Implementation of deep-learning-based edge computing for preventing drowning

Yi-Tung Chan*, Tai-Wei Hou, Yu-Lun Huang, Wen-Hsin Lan, Pin-Chia Wang, Cih-Ting Lai

Department of Electrical Engineering,
R.O.C. Naval Academy, Taiwan

*Corresponding Author: eadown92@gmail.com

Abstract

With the increasing popularity of computer vision and ocean engineering, maritime visual surveillance has emerged as an important area of study. Despite the development of various detection or surveillance techniques for maritime locations, maritime visual surveillance remains a challenge owing to the complex, unconstrained, and diverse nature of such places. In addition, only a few studies have investigated edge computing in actively preventing people from drowning. Considering several people lose their lives due to drowning, the flourishing technologies of artificial intelligence (AI) and deep learning have been used for our study. In this paper, an implementation of deep-learning-based edge computing to prevent drowning with the use of NVIDIA Jetson Nano is proposed.

Keywords: Artificial Intelligence, Deep Learning, Edge Computing, NVIDIA Jetson Nano, Drowning.

1. Introduction

According to the global report of World Health Organization (WHO)(1), drowning is the 3rd leading cause for unintentional deaths worldwide, accounting for 7 % of all injury-related deaths. More than 360,000 people drown worldwide each year. Thus, drowning is a major public health concern1(2). Notably, over half of the world's drowning cases occur in the WHO’s Western Pacific Region and WHO’s South-East Asia Region.

Despite the development of various surveillance for object detection or recognition in maritime locations, maritime visual surveillance remains a challenge owing to the complex, unconstrained, and diverse nature of such environments. Electro-optical (EO) sensors are typically used for security-related visual surveillance purposes, such as ship tracking, counting, and classification(3,4,5,6). However, as 24-hour manual surveillance is impractical, achieving automatic maritime surveillance using object and foreground detection has become an important goal. In addition, the development of video surveillance applications, for utilization in maritime environments, has been hindered by the noise and complexity of sea scenes that arise from factors such as water motion, dynamic backgrounds, waves, sea foam, water spray, reflections, wave ripples, and wake(5,7). Furthermore, because harbors comprise a vast area and are not patrolled or monitored all the time even if it is a naval port, the crucial period to rescue a drowning person is overlooked. Although
numerous tools have been designed to supply lifeguards to save people from drowning, such as the remote-controlled lifebuoy, these tools are restricted in action; this implies that these can be used only when the lifeguards notice that someone is drowning. However, in reality, drowning accidents occur in a fraction of time and are difficult to notice. Specifically, it is nearly impossible for security guards to always keep a close watch on all the monitors simultaneously, which renders them unable to immediately notice someone drowning. Besides, conventional warning systems cannot rescue people immediately using edge computing.

To overcome these problems, this paper proposes an active protection edge computing device that can ensure safety through the use of a simple visual camera, which provides early warnings when a potential drowning victim is noticed and alarms the security guards. In this paper, the implementation of deep-learning-based edge computing to prevent drowning with the use of NVIDIA Jetson Nano is proposed.

2. Methods

The flowchart of the proposed method is shown in Fig. 1. Firstly, the drowning dataset is collected for training and testing the deep learning model for classifying drowning and not-drowning classes in this study. Next, the operation graphical user interface (GUI) and deep learning model use a program implemented in PyTorch library. Finally, the drowning prevention device is implemented on NVIDIA Jetson Nano, the smallest, yet the most powerful, GPU-based edge computing device, for validating the proposed method.

Fig. 1: Flowchart of the proposed drowning-prevention device using deep-learning-based edge computing.

2.1 Dataset Collection and Pre-processing

For collecting the required drowning image dataset, 30 volunteers, unrelated to the study, were asked to enter the water, and they were made to wear three kinds of costumes, which were the navy uniform, navy fatigue dress, and swimsuit, as shown in Fig. 2. These volunteers were asked to make different poses, as shown in Fig. 3. The camera chosen to shoot the photos included Canon 800d, Sony RX100M3, and Canon55D. The different cameras were used for simultaneous shooting, so that diversified images could be collected from different sensor sources for a robust training model. The drowning poses were classified into two groups: arms above the water and in the water, as illustrated by Figs. 3(a) and 3(b). The not-drowning poses were classified into six groups, including four styles of swimming: backstroke, breaststroke, butterfly stroke, and freestyle stroke, separately, as illustrated by Figs. 3(c), 3(d), 3(e), and 3(f). Further, the other not-drowning poses were that of playing and wearing a lifebuoy or kickboard, as illustrated by Figs. 3(g) and 3(h).
Each pose was separated into four different shot angles as shown in Fig. 4: front view, back view, right view, and left view. Further, the images were divided into two categories of drowning and not-drowning datasets. Finally, the collected dataset had 1,557 drowning images and 3,110 not-drowning images. During the pre-processing step, each image was normalized to 224×224 resolution.

2.2 Training and Testing Deep Learning Model

Convolutional neural networks (CNN) are considered the state-of-the-art model in image recognition tasks. The well-known AlexNet architecture, a CNN implementation, is applied to train the deep learning model for image classification in our study. AlexNet contains eight layers with weights; the first five are convolutional and the other three are fully connected, as depicted in Fig. 5. AlexNet had participated in the ImageNet Contest and had shown good results.

The training phase flowchart is illustrated in Fig. 6. Firstly, the collected drowning dataset is randomly...
partitioned into three-fourth for the training model, as illustrated in Fig. 6(a) and one-fourth for testing model. The training dataset contains 1,168 drowning images and 2,333 not-drowning images. The testing dataset contains 389 drowning images and 777 not-drowning images.

Firstly, the samples are sent to a computer, as illustrated in Fig. 6(a). Secondly, the code is programmed in the Jupyter Notebook, as illustrated in Fig. 6(b), to create a model for identifying drowning and not-drowning images. Thirdly, as illustrated in Fig. 6(c), a model is deployed to Jetson Nano, which can be combined with the camera.

![Flowchart](image)

**Fig. 6:** Overall implementation flowchart.

Further, we designed an operation GUI, as illustrated in Fig. 6(b), by using Ipywidgets of Jupyter for the training procedure, as shown in Fig. 7. In Fig. 7(a), ‘dataset’ and ‘category’ sheet can import the training images from the chosen category. Further, the computer shows the number of selected training images in the ‘count’ text field. The ‘train’ button, as shown in Fig. 7(b), trains our model with the selected training images, and the field of ‘progress’ shows the progression made by the computer. Finally, we can save or load the training model by using the buttons shown in Fig. 7(c).

In the training phase, as shown in Fig. 8, the training accuracy increases linearly over time, until it reaches approximately 100 %, whereas the training loss keeps decreasing linearly until it reaches approximately 0. The training loss reaches its minimum after 13 epochs and then stalls. Specifically, after 13 epochs, the proposed model’s prediction accuracy reaches approximately 0.99 for the training dataset. Thus, we stop the training phase and use the ‘evaluate’ button as shown in Fig. 7(b) for testing the training model. Finally, the model’s accuracy is determined to be 0.85 for the testing dataset.

![Accuracy and Loss](image)

**Fig. 7:** GUI employed in this study.

**Fig. 8:** Training accuracy and training loss.
2.3 Implementing Drowning Prevention Device

The hardware used in the proposed drowning prevention device is NVIDIA Jetson Nano, which consists of 4x USB 3.0, USB 2.0 Micro-B, MIPI CSI-2 DPHY lanes, and HDMI 2.0 and eDP 1.4 display unit. It is one of the most powerful edge devices available in the market with 128 CUDA core and 472 GFLOPS-capable GPU. It enables the development of small, low-power artificial intelligence (AI) systems.

In the drowning prevention device, we mounted the Raspberry Pi Camera, whose model product was Pi NoIR Camera V2, on NVIDIA Jetson Nano. Further, the training deep learning model was deployed in Jetson Nano and evaluated in a real environment.

3. Experimental Analysis

The implemented drowning detection device, as shown in Fig. 9(a), which has deployed the training model on Jetson Nano, conducted realistic experimental application in a swinging pool. The volunteers acted out drowning poses in the pool for evaluating the proposed device. The drowning detection device and the laptop were mounted for realistic experimental application purposes as shown in Fig. 9(b). The laptop could remotely control the Jetson Nano by using the router and Jupyter software, and we can see the prediction result from the model in Fig. 9(c).

Examples of the obtained results are shown in Fig. 10. We can see that our drowning detection device can correctly classify the drowning persons as shown in Fig. 10(a). The results demonstrated that our implementation of the deep-learning-based edge computing can classify drowning persons through the use of NVIDIA Jetson Nano.

Notably, the examples of Type I error (false positives) are shown in Fig. 10(b). We note that the prediction results were not as stable as expected. Specifically, when someone was drowning, the device might alter the prediction leading to inaccurate determination of whether the person was drowning or not.

(a) Drowning (b) Not-drowning

Fig. 9: Realistic illustration of our experimental application.

Fig. 10: Real-time prediction results when someone is drowning; these are obtained by using our training model with NVIDIA Jetson Nano.
We analyzed the results of false positives and arrived at the following three conclusions:

1) Different cameras: Although different cameras were used to collect the images for the drowning dataset, for the realistic experimental applications, our team used the Raspberry Pi Camera to shoot. There is a deviation in the pictures that were shot using the single-lens reflex camera and the ones shot using the Raspberry Pi Camera.

2) Different illumination conditions: The environment for the realistic experimental applications is different for the drowning dataset, which creates different illumination conditions for the pictures in the two periods.

3) Lack of spatiotemporal clues: In the future, we will add new images in the drowning dataset using the Raspberry Pi Camera. In addition, data augmentation will also be applied to create a robust training model that can adapt to different sensor sources, shot angles, and illumination conditions. Furthermore, we will improve our device by using object detection in video sequences.

4. Conclusions

In this study, we implemented a deep-learning-based edge computing device for prevention of drowning. In addition, it was shown that edge computing with AI technology can be realistically applied in our lives. Apart from mounting our drowning prevention device on the shore, it also can be deployed in an autonomous lifebuoy, which has propeller blades and can move automatically, as an active-preventing drowning system. This would reduce the complexity involved in saving people from drowning and greatly increase people’s safety both in the harbor and shore areas.

Acknowledgement

This research was supported by the Ministry of Science and Technology of Taiwan [grant number MOST 107-2218-E-012-001-MY2 and MOST 108-2221-E-012-005]. A patent application for this invention is currently in process.

References

(1) W. H. Organization and others: “Global report on drowning: preventing a leading killer”, World Health Organization, 2014

(2) W. H. Organization and others: “Preventing drowning: an implementation guide”, World Health Organization, 2017

(3) W. C. Hu, C. Y. Yang, and D. Y. Huang: “Robust real-time ship detection and tracking for visual surveillance of cage aquaculture,” Journal of Visual Communication and Image Representation, Vol. 22, No. 6, pp. 543–556, 2011

(4) D. K. Prasad, C. K. Prasath, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek: “Challenges in Video Based Object Detection in Maritime Scenario Using Computer Vision”, International Journal of Computer and Information Engineering, Vol. 11, No. 1, pp. 31–36, 2017

(5) D. D. Bloisi, F. Previtali, A. Pennisi, D. Nardi, and M. Fiorini: “Enhancing Automatic Maritime Surveillance Systems with Visual Information”, IEEE Transactions on Intelligent Transportation Systems, Vol. 18, No. 4, pp. 824–833, 2017

(6) D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek: “Video Processing From Electro-Optical Sensors for Object Detection and Tracking in a Maritime Environment: A Survey”, IEEE Transactions on Intelligent Transportation Systems, Vol. 18, No. 8, pp. 1993–2016, 2017

(7) A. Sobral, T. Bouwmans, and E. H. Zahzah: “Double-constrained RPCA based on saliency maps for foreground detection in automated maritime
surveillance”, AVSS 2015 - 12th IEEE International Conference on Advanced Video and Signal Based Surveillance, 2015

(8) Fujii, Hiroaki Izuma, and Shigeru Nonami, "Drowning accident monitoring device", Japan Patent JP2001184574A, July 06, 2001.

(9) Hajime Fujii, Hiroaki Izuma, and Shigeru Nonami, "Drowning accident monitoring device", Japan Patent JP2001184574A, July 06, 2001.

(10) Hung Chou, Yi Ju Chen, Ssu Ssu Yang, Tien Cheng, and Yu Tsai, "Near-drowning behavior detection method based on support vector machine", China Patent CN103413114A, May 17, 2013.

(11) Eyal Golan and Tamar Avraham, "Methods and systems for drowning detection", United States Patent US20190034712A1, January 19, 2017.

(12) Robert Francis Bodi, "System, Device, and Method of Detecting Dangerous Situations", United States Patent US20180040223A1, October 17, 2017.

(13) Tsun Mu Wang, "Wisdom deep learning water rescue system", Taiwan 107118800A, May 31, 2018.

(14) A. Krizhevsky, I. Sutskever, and G. E. Hinton: “ImageNet classification with deep convolutional neural networks”, Communications of the ACM, Vol. 60, No. 6, pp. 84–90, 2017

(15) https://zhuanlan.zhihu.com/p/77480151. [Accessed: 08-Jan-2020]