Improved Automatic Drowning Detection Approach with YOLOv8

Nouf Alharbi
College of Computer Science and Engineering, Taibah University, 42353 Madinah, Saudi Arabia
nmoharbi@taibahu.edu.sa (corresponding author)

Layan Aljohani
College of Computer Science and Engineering, Taibah University, 42353 Madinah, Saudi Arabia
TU4150939@taibahu.edu.sa

Anhar Alqasir
College of Computer Science and Engineering, Taibah University, 42353 Madinah, Saudi Arabia
TU4150188@taibahu.edu.sa

Taghreed Alahmadi
College of Computer Science and Engineering, Taibah University, 42353 Madinah, Saudi Arabia
TU4052241@taibahu.edu.sa

Rehab Alhasiri
College of Computer Science and Engineering, Taibah University, 42353 Madinah, Saudi Arabia
TU4052120@taibahu.edu.sa

Dalia Aldajan
College of Computer Science and Engineering, Taibah University, 42353 Madinah, Saudi Arabia
TU4152887@taibahu.edu.sa

Received: 27 August 2024 | Revised: 23 September 2024 | Accepted: 27 September 2024

 DOI: https://doi.org/10.48084/etasr.8834

ABSTRACT

Although swimming is a popular activity that promotes relaxation and stress relief, drowning remains a serious global problem. According to the World Health Organization (WHO), drowning is the third most common cause of death. This study delves into implementing deep learning techniques for precise drowning detection. From this point of view, a drowning detection system was designed using the YOLOv8 model, which is a powerful tool for object detection and classification tasks. Using a large dataset, the YOLOv8 model was trained to recognize drowning patterns and movements and increase the likelihood of successful rescue operations by reducing response times and improving water safety. The proposed system uses deep learning techniques and YOLOv8 technology with data augmentation techniques to enhance the model's robustness to variations in lighting, pose, and background conditions. The system performance was evaluated using the Swimming and Drowning Detection dataset achieving 90.1% accuracy compared to 88.5% with YOLOv5.

Keywords-YOLOv8; deep learning; computer vision; object detection; drowning detection

I. INTRODUCTION

Swimming is considered one of the most important and best exercises that helps people relieve stress and relax. Swimming pools are widely available in resorts and tourist clubs. The biggest fear is that people who swim in pools can drown. According to a World Health Organization (WHO) report, drowning is the third leading cause of death worldwide, with an estimated 236,000 drowning deaths annually worldwide [1]. Studies have investigated the appropriate ways to reduce drowning incidents and save people. These methods include educating parents about the dangers of drowning, increasing the supervision while swimming by specialists, providing resources such as life jackets and other aids, or continuous monitoring of security cameras by staff to monitor any signs of distress or drowning. However, these solutions are insufficient and can be considered primitive because they rely on visual or auditory signals that can be inaccurate in noisy or crowded environments. Therefore, there is a need for technologies and methods to help in immediately detecting drowning cases, preventing deaths, and helping rescue workers and owners of places with swimming pools contain and reduce drowning cases [2].

YOLO is a single-stage object detector and uses a fully Convolutional Neural Network (CNN). It has a simple architecture and is specially designed to detect objects in one stage by considering all range proposals. It can detect multiple objects in a single image or video frame with high accuracy and speed using features from the entire image to predict each bounding box of the object. Combining a previously multi-step process, YOLO uses a single neural network to perform classification and prediction of the bounding boxes of detected objects. This significantly improves recognition performance and can run much faster than running two separate neural networks separately for object detection and classification. The YOLO architecture family is gaining importance due to its high compatibility with industrial requirements such as accuracy, lightweight, and generally being more efficient than others in terms of speed and accuracy [3]. YOLOv8 is the latest version of the YOLO family and is a powerful and flexible tool for object detection, providing faster and more accurate results than previous versions. It is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection, image segmentation, and image classification tasks [4]. This study used the YOLOv8 algorithm for training and deep learning.

Many artificial intelligence techniques have been studied in the field of drowning detection. In [5], a deep learning-based computer vision approach was proposed for drowning detection using and evaluating five pre-trained CNN models, namely SqueezeNet, GoogleNet, AlexNet, ShuffleNet, and ResNet50, with ResNet50 showing the best performance with a prediction accuracy of 100%. This study also discussed different approaches to automatic drowning detection, including sensor-based and vision-based approaches. However, a technique for rapid and automatic detection of drowning victims was proposed using pre-trained CNN models. The strength of the proposed technique relied on transfer learning: Pre-trained CNN models enable transfer learning, meaning that knowledge gained through training for one task (e.g. image recognition) is applied to a different but related task (e.g. drowning detection). However, the weaknesses of such pre-trained models lie in being trained on generic image datasets and may not be specifically designed for drowning detection. This can lead to suboptimal performance in the detection of drowning instances. In addition, the high accuracy of the models may be due to overfitting because of using small datasets for training.

In [6], the development of a real-time automatic child drowning detection method was discussed, using YOLOv5 and Faster R-CNN models based on video surveillance. This study collected a dataset of various live scene videos of infants swimming and drowning and trained the models using supervised learning experiments. The results showed that YOLOv5s outperformed the Faster R-CNN model in terms of speed. However, the mean Average Precision (mAP) of YOLOv5s was 89%, while the mAP of the Faster R-CNN model was 92.24%, indicating a slightly lower accuracy for YOLOv5s compared to Faster R-CNN. However, the YOLOv5s model achieved 75 FPS, while the Faster R-CNN model could only handle 6 FPS video.

Video surveillance allows continuous monitoring of the swimming pool area and enables real-time detection of possible drowning incidents. This is especially important for young children, who cannot send distress signals in emergencies. However, there is the possibility of false detection in cases where the targets are in a complicated environment with toy interference. As a result, there is an urgent need for careful placement and calibration of cameras to ensure optimal coverage of the swimming pool area and minimize blind spots. In [7], the use of Unmanned Aerial Vehicles (UAVs) was discussed in maritime Search And Rescue (SAR) operations. This study highlighted the limitations imposed, as it does not have sufficient ability to detect drowning people and small targets in images taken from high altitudes. A lightweight model was also proposed, which relied on YOLOv5s to address these limitations. The proposed model included improvements such as an enlarged layer for small object detection, the use of Ghost and C3Ghost modules to improve the lightweight network, and improved detection performance for rescue targets in marine casualties. The experimental results showed that the improved model outperformed the original YOLOv5s in terms of mAP for different detection thresholds. In addition, the improved model had reduced parameters, size, and weights compared to the compact model, making it suitable for UAV deployment. However, this study did not specify whether the improved YOLOv5s model was tested or optimized for diverse environmental conditions (e.g., weather, lighting, sea state), which are critical factors in maritime SAR operations.

In [8], an automatic video-based drowning detection system for swimming pools was introduced using active contours. The system was based on real-time video analysis from cameras installed around the swimming pool and used HSV color space analysis and contour detection to detect the area of interest in each frame of video sequences under real-world conditions. It could detect drowning people in indoor swimming pools and send an alarm to the lifeguard if the person is missing for a certain period. The system was tested on several video sequences recorded in swimming pools and demonstrated high accuracy and the ability to track people in real time. However, this study did not explain how the system works in outdoor swimming pools with different lighting conditions or in crowded pool environments with multiple swimmers. Furthermore, active contour-based drowning detection systems can produce false alarms, which means that they can incorrectly identify as drowning someone who is not.

In [9, 10], underwater computer vision-based drowning detection systems were proposed to address the challenges of drowning detection using a two-stage algorithm and embedded artificial intelligence hardware. The first stage used YOLOv5n to identify near-vertical human objects, which effectively excluded non-drowning people and established a reliable foundation for drowning detection. In the second stage, a lightweight Drowning Detection Network (DDN) was proposed based on a deep Gaussian model for fast detection. This approach enables the detection of anomalies in high-level semantic features, overcoming the scarcity of actual drowning videos by focusing on detecting anomalous behavior. The effectiveness of this approach was demonstrated by experimental results, highlighting its potential for real-world applications. The proposed drowning detection system utilized a deep Gaussian model for unsupervised anomaly detection in high-level semantic features and outperformed pixel-level-based methods in robustness and performance. The effectiveness of the algorithm may be affected by prolonged occlusion, size anomalies, or atypical behavior. Further advances should address these limitations to expand the applicability of the algorithm.

Through extensive research and knowledge of the strengths and weaknesses of each project, YOLOv8 is considered an improved version and one of the latest developments in the field of object detection due to its fast ability to analyze images and videos. By using this technology, drowning cases can be detected more efficiently and robustly than other technologies such as Faster R-CNN and CNN. To that extent, this paper proposes a deep learning model using the YOLOv8 algorithm to detect drowning cases effectively and overcome the limitations of previous approaches that have difficulty in dealing with overlapping objects and changes in lighting settings [11]. YOLO is characterized by high accuracy, speed, and excellent processing. This algorithm is used to effectively and quickly detect drowning cases by training the model on a large number of images showing drowning people. When the system is running, the images are passed to the YOLO model, which consequently identifies the people within the bounding box and displays their classification label. The main objectives of the proposed system are as follows:

- Construct a deep learning model for drowning detection and classification.
- Early detect and classify drowning cases effectively and accurately.
- Improve water safety.

II. EXPERIMENTAL METHODS

A. *Proposed Approach*

Drowning detection is critical to keeping people safe in the water. To improve the quality and effectiveness of drowning detection, the proposed model uses YOLOv8. Due to its design based on deep neural networks [11], it is an effective approach to solving this problem and helping to protect people's lives in water. This approach consists of two phases, as shown in Figure 1. First, data preprocessing is applied and second, a YOLO model is built. The first phase applies data-augmentation techniques and processes to ensure detection accuracy. The second phase includes building the YOLO model and training it in different drowning cases. This phase begins with feature extraction in the backbone and then enhances and fuses the features in the neck. In the final phase of condition detection and classification, the drowning person appears within the bounding box and is then classified according to its condition.

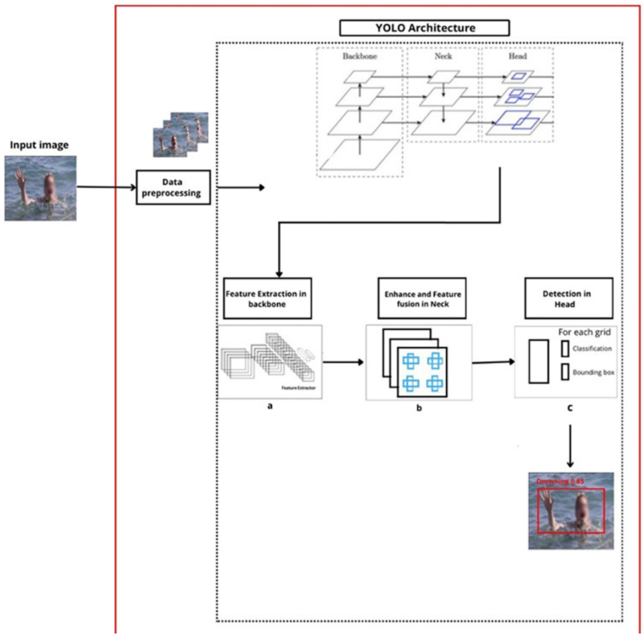

Fig. 1. The proposed approach for drowning detection framework using the YOLOv8 model.

1) *First Stage: Data Preprocessing*

Preprocessing is the process of preparing input data (images) and annotations to train the YOLO object detection model. Properly preprocessed data contributes to better model performance and generalization to real-world scenarios.

Dataset Preparation [12]: The swimming and drowning detection dataset [13] contains 12,365 outdoor images of a swimming pool and is divided into three sets. A training set of 10,140 images, a validation set of 1,478 images, and a test set of 747 images were used to classify three labels as follows: swimming, drowning, and person out of water. After the dataset was obtained and verified, it was uploaded to Google Drive and linked to Google Colab for use in building the model.

Data format: To prepare data for YOLO, it is necessary to ensure that the annotations are in the specific format that YOLO requires. YOLO expects annotation data in a specific format to train and test its models. Each annotation file corresponds to an image, and the annotation file should have the same name as the image but with a different extension. For example, having an image named "image001.jpg", the corresponding annotation file should be "image001.txt". Each line in the annotation file represents an object annotation and should contain the following information, separated by spaces:

<class_id> <center_x> <center_y> <width> <height>

where <class_id> is the integer ID of the class, and <x_center>, <y_center>, <width>, and <height> are the normalized center coordinates, width, and height of the bounding box, respectively.

Data augmentation is used to create new from existing data by applying various transformations, as shown in Figure 2, such as flipping, rotating, cropping, and changing brightness or contrast. Data augmentation is commonly used in computer vision tasks, as it helps increase the volume of data by applying some techniques that improve the performance of the model and offer more diversity in training data. This helps the model to generalize better and, by subjecting it to different transformations, become more robust and less sensitive. Augmentation was used in the drowning detection model to increase the volume and variety of data to facilitate the drowning detection task. In this phase, many techniques were applied to expand the data and improve the images in this project.

- Image cropping was performed so that the model could learn to detect drowning even when the image was partially occluded.
- The images were flipped horizontally and vertically to help the model learn to detect drowning from all sides.
- The images were rotated at specific angles to increase the size of the data and make the model more powerful in detecting drowning cases in more than one direction.
- The brightness and contrast of the images were changed, with the new images becoming darker or brighter, allowing the model to detect drowning at different lighting levels. The image contrast was also changed to be detectable at different lighting levels.

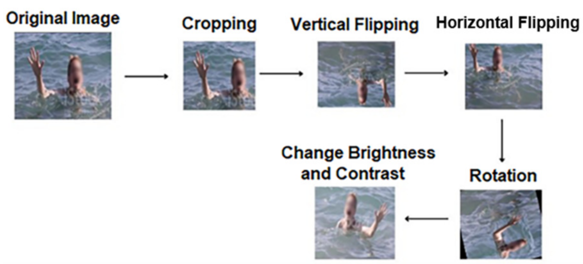

Fig. 2. Data augmentation techniques.

B. *Second Stage: Building the YOLO Model (YOLO Architecture)*

YOLO is a state-of-the-art object detection algorithm that is so fast that it has become one of the standard and most popular object detection methods in computer vision. The YOLO family belongs to single-stage object detection models that process the entire image in a single forward pass. In this project, YOLOv8 was chosen because of its high accuracy and detection speed. YOLO can detect drowning in one step, eliminating the need for two detection stages that slow down the detection process. After preprocessing the dataset, the YOLO model was created and the dataset was passed through it. The model generates predictions about the bounding-box coordinates and displays the classification. When the image passes through the YOLO model, the first stage occurs in the backbone, where features are extracted. Then it is moved to the neck, where the inherited features of the backbone are integrated and strengthened. Then, the head consists of task-specific layers designed to make a final prediction or inference based on information from the backbone and the neck.

1) *Feature Extraction in Backbone*

The backbone phase in YOLOv8 for drowning detection is the first of the three phases in the model architecture, and it is responsible for extracting features from the input images. These features are then used by the neck and head phases to make predictions about whether or not a person is drowning. YOLOv8 takes the preprocessed image as an input and divides it into a grid. The preprocessed input image goes through a series of convolutional layers in the backbone network. The layers in the backbone are designed to capture features of varying scale and complexity. For example, low-level features, such as edges, can be captured in the early layers, while high-level features and object semantics are captured in the deeper layers. The final output of the backbone is a series of feature maps with rich spatial and semantic information. Subsequent layers then use these feature maps to predict bounding boxes, class probabilities, and confidence values for object detection. The backbone phase is crucial for extracting meaningful features from the input image and enables YOLOv8 to achieve accurate and efficient object detection across different scales and complexities within the image.

2) *Enhance and Feature Fusion in Neck*

The neck stage in YOLOv8 drowning detection is the second of three stages in the model structure. It is responsible for processing the features extracted from the backbone and also integrating the features for drowning prediction. The neck phase consists of two main components, the Path Aggregation Network (PANet) and the Feature Pyramid Network (FPN) [14]. PANet is used to aggregate features from different levels of the backbone. This allows the model to capture information from different scales, which is important for object detection. FPN effectively fuses features of various scales to construct a more comprehensive representation. The neck phase output usually consists of refined feature maps that are most suitable for object detection, and each of these maps represents a different scale. The neck phase is an important part of the YOLOv8 architecture, as it helps improve the accuracy of model predictions by collecting information from different scales and focusing on the most relevant parts of the feature maps.

3) *Detection in Head*

After using the backbone as a feature extractor, allowing the representation of a feature map in the neck, this information extracted from the backbone and neck is passed to the head. The head consists of special layers designed to produce the final prediction. Here, the drowning condition is detected and classified within the bounding box.

III. RESULTS AND DISCUSSION

Model evaluation is a fundamental process to assess the quality and performance of the model. The purpose of the evaluation is to estimate the effectiveness of the model and determine its compliance with established requirements and objectives. The evaluation can also provide evidence-based recommendations for improving the process or making changes to the model. To evaluate the performance of the proposed model, YOLOv5 was also trained on the same dataset. This comparative approach was adopted to evaluate the precision and effectiveness of each algorithm in this specific dataset. Both YOLOv8 and YOLOv5 are powerful object detection models that differ in their underlying architecture, leading to unique strengths and weaknesses. The evaluation metrics that were used in this experiment are as follows.

A. Mean Average Precision (mAP)

mAP [15] is an important metric for evaluating object detection models and measuring accuracy. It is used to evaluate popular models such as Fast R-CNN, YOLO, and Mask R-CNN. mAP is the average precision across all detected classes. After training and evaluating the performance of the models, the obtained results were 0.901 and 0.885 for YOLOv5 and YOLOv8, respectively. Despite that YOLOv5 model accuracy is slightly higher, the detection speed of YOLOv8 was outstanding at 10.24 ms compared to 23.76 ms for YOLOv5, showing that it perfectly suits this specific detection task.

B. Precision

Precision [16] measures the percentage of true positives among all positive predictions and assesses how well the model prevents false positives.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \tag{1}$$

As shown in Figure 3, the precision results of the proposed model appeared at 100% or close to it. This means that all the positive predictions provided by the model were correct. In other words, all cases that the model classified as positive were actually positive. The precision of the proposed model for identifying drowning individuals was 100%, meaning that every individual that the model classified as drowning was actually drowning. The precision-confidence curve is a crucial visualization that showcases the model's precision at different confidence thresholds. For YOLOv8, a precision score of 1.00 signifies that the model made no false-positive predictions at the confidence threshold of 0.958. In other words, when the model identified an object with a confidence score exceeding 0.958, it was unequivocally accurate in its predictions. However, YOLOv5 achieved the same precision but with a lower confidence score of 0.933.

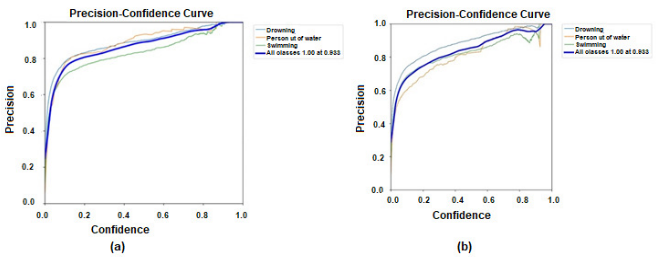

Fig. 3. Precision-confidence curve using: (a) YOLOv5, (b) YOLOv8.

C. Recall

Recall is the proportion of detected true positives among all true positives and measures the ability of the model to detect all instances of a class.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{2}$$

As shown in Figure 4, YOLOv8 outperformed YOLOv5 with a recall of 0.97, which means that the model correctly identified 97% of all positive cases. Recall, also known as sensitivity, measures the proportion of actual positive cases that are correctly identified by the model. There is a trade-off relationship between precision and recall, meaning that when recall is very high, precision can be low, and vice versa.

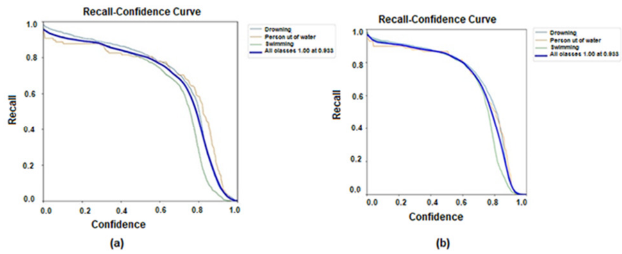

Fig. 4. Recall - confidence curve using: (a) YOLOv5, (b) YOLOv8.

D. F1-score

The F1 confidence curve [17] illustrates how sensitive the model is to the threshold used to classify a positive instance. It also helps identify the threshold that provides the best balance between precision and recall. It also shows how robust the model is across different levels of confidence at the same time. It is vital in applications where the cost of false positives and false negatives varies. The F1 confidence curve can provide insights beyond what a single F1 score might reveal, especially in cases where models have similar F1 scores but different precision-recall balances.

$$F1 = 2 * \frac{precision * recall}{-recision + recall} \tag{3}$$

As shown in Figure 5, the F1 scores for YOLOv5 and YOLOv8 were 0.84, and 0.87 respectively, which indicates the harmonic mean of precision and recall was higher for YOLOv8. The F1 score combines both precision and recall into one value, providing a balance between the two. An F1-score of 0.87 for YOLOv8 indicates a strong performance of the model in terms of both precision and recall.

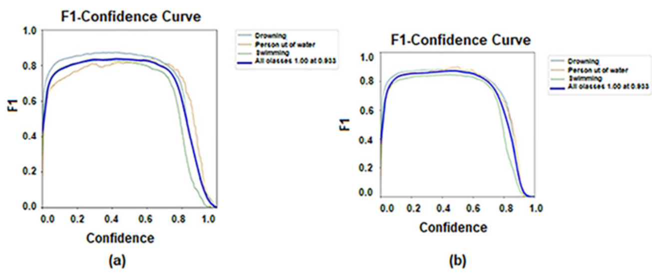

Fig. 5. F1 confidence curve using: (a) YOLOv5, (b) YOLOv8.

E. Confusion Matrix

The confusion matrix [18] is a visual tool that summarizes how well the model distinguishes between different classes. As shown in Figure 6, the matrices display the predictions of the actual classification and the expected classification for both models. These matrices help to examine the performance of the object detection models and reveal their strengths and weaknesses. The confusion matrix plays a crucial role in evaluating the model's ability to analyze background performance and, along with other metrics, helps identify biases or limitations in the model.

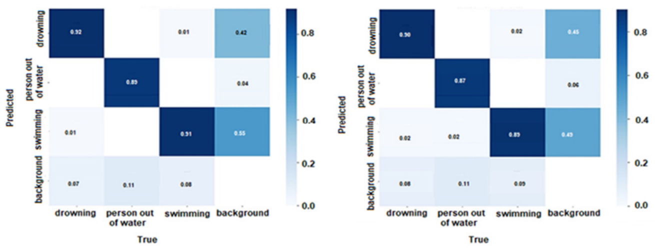

Fig. 6. Confusion matrix using: (a) YOLOv5, (b) using YOLOv8.

IV. CONCLUSION

This study proposed a deep learning-based drowning detection system using the YOLOv8 algorithm, demonstrating significant improvements in detection speed and accuracy over YOLOv5. By training the model on a large dataset, the YOLOv8 approach achieved an accuracy of 90.1%. This accuracy was compared with previous studies that employed YOLOv5 and Faster R-CNN, which, while effective, exhibited limitations in both speed and handling of complex environments [6, 7]. Several similar studies in the field have also used deep learning approaches to improve drowning detection. For instance, in [5], five pre-trained CNN models were used, including ResNet50 and SqueezeNet, to detect drowning incidents, achieving a prediction accuracy of 100% for ResNet50. However, this approach was limited by the small dataset of only 200 images, which may have caused overfitting. This study used a significantly larger dataset, leveraging YOLOv8's advanced object detection capabilities to provide a more robust and generalizable model for real-world applications.

Furthermore, the findings align with other studies that explored vision-based drowning detection using YOLO models. In [6], YOLOv5 and Faster R-CNN were employed for real-time infant drowning detection, with YOLOv5 demonstrating superior speed (75 FPS) but slightly lower accuracy compared to Faster R-CNN. Although YOLOv5 was proficient in handling real-time detection tasks, YOLOv8 outperformed it in terms of speed (10.24 ms) and was more adept at handling variations in lighting and environmental conditions, which makes it better suited for diverse aquatic settings. In [7], the use of UAVs with YOLOv5s was explored to detect drowning individuals, showing enhancements in lightweight detection but still facing challenges in high-altitude image analysis. The improvements introduced in the proposed model, particularly in small object detection and feature fusion, provide potential solutions for such challenges, highlighting the broader applicability of YOLOv8 in both surveillance and rescue operations.

In conclusion, this study demonstrated that YOLOv8 is not only an incremental improvement in terms of performance but also offers practical benefits for real-world applications. This model represents a significant step forward in the field of automatic drowning detection by addressing limitations such as overlapping objects and varying environmental conditions.

REFERENCES

[1] A. E. Peden, J. Passmore, A. C. Queiroga, R. Sweeney, and J. Jagnoor, "Closing the gap for drowning prevention across Europe," *The Lancet Public Health*, vol. 7, no. 9, pp. e728–e729, Sep. 2022, https://doi.org/10.1016/S2468-2667(22)00193-1.

[2] A. Rahman, A. E. Peden, L. Ashraf, D. Ryan, A.-A. Bhuiyan, and S. Beerman, "Drowning: Global Burden, Risk Factors, and Prevention Strategies," in *Oxford Research Encyclopedia of Global Public Health*, 2021.

[3] N. Alharbi, "Exploring Advance Approaches for Drowning Detection: A Review," *Engineering, Technology & Applied Science Research*, vol. 14, no. 4, pp. 16032–16039, Aug. 2024, https://doi.org/10.48084/etasr.7804.

[4] M. Sohan, T. Sai Ram, and Ch. V. Rami Reddy, "A Review on YOLOv8 and Its Advancements," in *Data Intelligence and Cognitive Informatics*, Tirunelveli, India, 2024, pp. 529–545, https://doi.org/10.1007/978-981-99-7962-2_39.

[5] M. Shatnawi, F. Albreiki, A. Alkhoori, and M. Alhebshi, "Deep Learning and Vision-Based Early Drowning Detection," *Information*, vol. 14, no. 1, Jan. 2023, Art. no. 52, https://doi.org/10.3390/info14010052.

[6] Q. He, Z. Mei, H. Zhang, and X. Xu, "Automatic Real-Time Detection of Infant Drowning Using YOLOv5 and Faster R-CNN Models Based on Video Surveillance," *Journal of Social Computing*, vol. 4, no. 1, pp. 62–73, Mar. 2023, https://doi.org/10.23919/JSC.2023.0006.

[7] J. Bai, J. Dai, Z. Wang, and S. Yang, "A detection method of the rescue targets in the marine casualty based on improved YOLOv5s," *Frontiers in Neurorobotics*, vol. 16, Nov. 2022, https://doi.org/10.3389/fnbot.2022.1053124.

[8] N. Salehi, M. Keyvanara, and S. A. Monadjemmi, "An Automatic Video-based Drowning Detection System for Swimming Pools Using Active Contours," *International Journal of Image, Graphics and Signal Processing*, vol. 8, no. 8, pp. 1–8, Aug. 2016, https://doi.org/10.5815/ijigsp.2016.08.01.

[9] T. Liu, X. He, L. He, and F. Yuan, "A video drowning detection device based on underwater computer vision," *IET Image Processing*, vol. 17, no. 6, pp. 1905–1918, 2023, https://doi.org/10.1049/ipr2.12765.

[10] V. Sathiyapriya, K. Kaviya, R. Kavitha, R. Ajitha, and S. Jothika, "Drowning Detection System," *International Journal for Research Trends and Innovation*, vol. 8, no. 5, pp. 747–751, 2023.

[11] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023, https://doi.org/10.3390/make5040083.

[12] W. Chen and A. Quan-Haase, "Big Data Ethics and Politics: Toward New Understandings," *Social Science Computer Review*, vol. 38, no. 1, pp. 3–9, Feb. 2020, https://doi.org/10.1177/0894439318810734.

[13] "Swimming and Drowning Detection Dataset," *Roboflow*. https://universe.roboflow.com/university-g3h71/swimming-and-drowning-detection.

[14] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, Jun. 2018, pp. 8759–8768, https://doi.org/10.1109/CVPR.2018.00913.

[15] Y. Cui, D. Guo, H. Yuan, H. Gu, and H. Tang, "Enhanced YOLO Network for Improving the Efficiency of Traffic Sign Detection," *Applied Sciences*, vol. 14, no. 2, Jan. 2024, Art. no. 555, https://doi.org/10.3390/app14020555.

[16] W. Cullerne Bown, "Sensitivity and Specificity versus Precision and Recall, and Related Dilemmas," *Journal of Classification*, vol. 41, no. 2, pp. 402–426, Jul. 2024, https://doi.org/10.1007/s00357-024-09478-y.

[17] I. P. Sary, S. Andromeda, and E. U. Armin, "Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection using Aerial Images," *Ultima Computing : Jurnal Sistem Komputer*, vol. 15, no. 1, pp. 8–13, Jun. 2023, https://doi.org/10.31937/sk.v15i1.3204.

[18] J. Liang, "Confusion Matrix: Machine Learning," *POGIL Activity Clearinghouse*, vol. 3, no. 4, Dec. 2022.