A Comprehensive Drowning Detection System Employing Advanced Computer Vision for Enhanced Water Safety

Tan Zu Seng
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp065401@mail.apu.edu.my

Voon Jin Hui
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp064878@mail.apu.edu.my

Dr.Adeline Sneha J
Senior Lecturer / School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
adeline.john@apu.edu.my

Dr. Kamalanathan Shanmugam
Senior Lecturer / School of Technology
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
kamalanathan@apu.edu.my

Juhairi Aris Muhamad Shuhili
Lecturer / School of Computing
Asia Pacific University of Technology and Innovation (APU))
Kuala Lumpur, Malaysia
juhairi.shuhili@apu.edu.my

***Abstract*— The increasing number of drowning accidents, particularly in swimming pools, highlighted the urgent need to improve water safety. However, conventional surveillance methods have shown limited abilities in preventing drowning issues. This research examined previous research in the fields of underwater object detection and drowning behavior as well as explored similar projects and methodologies. A system architecture was proposed to address this issue which employed computer vision and deep learning technologies and integrated YOLOv3, a powerful object detection algorithm, with real-time video analysis and motion analysis. The impact of parameter modifications was investigated, specifically the Confidence score and Non-Maximum Suppression (NMS) threshold, on the effectiveness and accuracy of drowning detection. The study concluded that a compromise between a Confidence value of 0.5 and an NMS threshold of 0.5 produced optimal outcomes in terms of processing speed and accuracy through iterative experiments. Hence, the proposed drowning detection system has demonstrated improvement in enhancing water safety by reducing response times during drowning incidents.**

Keywords—drowning, computer vision, recognition, swimming pool, algorithms, deep learning, underwater

I. INTRODUCTION

Drowning remains a haunting and major reason for unintentional death, raising doubt about the safety of people, especially children, in swimming pools. Despite our efforts to assure water safety through traditional surveillance methods, the number of drowning accidents remains high, creating a continuing issue, particularly in occupied or remote places. These old approaches rely heavily on human supervision, which, at times, may be insufficient or ineffectual in averting such terrible incidents.

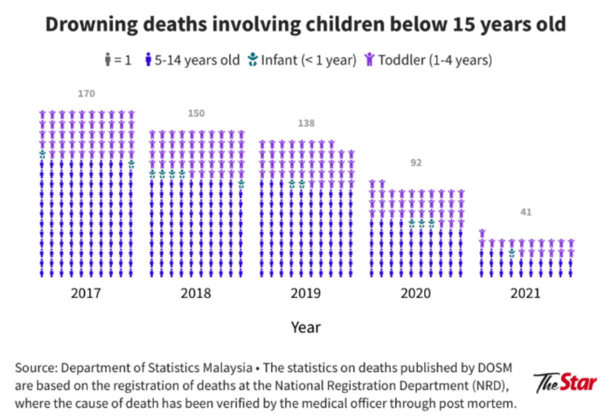

Fig. 1. Drowning Cases in Malaysia Statistic *(Pfordten, 2023)*

In Malaysia, where the appealing nature of swimming pools coexists with the dangers they pose, the Statistics Department has shed light on the gravity of the situation. Between 2017 and 2021, a shocking 591 drowning fatalities occurred among youngsters aged 0 to 14. (Pfordten, 2023) This worrisome number reveals a stark truth, emphasizing the importance of developing new and technologically sophisticated solutions to this important issue. The fact that 74% of these terrible instances affected boys is deeply concerning, highlighting the importance of focused initiatives and increased vigilance to protect our youth's lives. Furthermore, Malaysia had a series of terrible drowning accidents just six months ago, in 2023, showing the problem's persistence and urgency. On February 27, 2023, two siblings sadly drowned while swimming in an Ipoh pool (Amanda, 2023). Subsequent occurrences, such as a child discovered drowned in a kiddie pool in Tambun, highlighted the ongoing susceptibility of young lives to drowning threats (Loh, 2023).

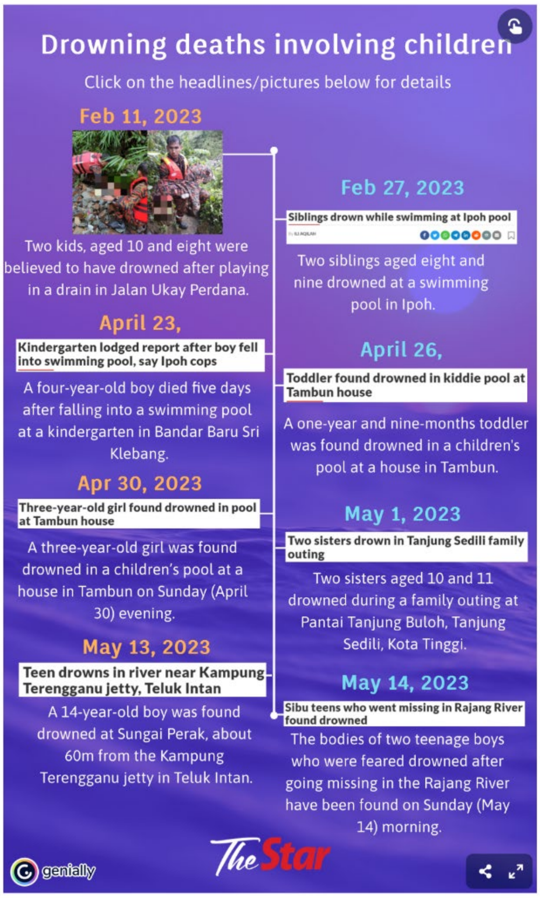

Fig. 2. Drowning Real Cases in Malaysia 2023 *(Pfordten, 2023)*

Given this context, the need to investigate cutting-edge technology that might improve drowning prevention efforts becomes critical. This journal article aims to offer a ground-breaking drowning detection system using artificial intelligence, computer vision, and machine learning. Our technological advances, designed to overcome the limits of traditional monitoring, employ complex algorithms to analyze video feeds from several cameras strategically positioned around swimming pools. As a result, it hopes to provide a proactive and effective approach to preventing drowning events, with the ultimate objective of saving lives and minimizing the devastating effects of this terrible but preventable cause of unintentional death.

A. Literature Review

1) Similar Projects

In the domain of underwater object detection, several projects have addressed challenges specific to marine environments. One notable project, "Underwater Object Detection Method Based on Improved Faster RCNN," focuses on enhancing the Faster RCNN algorithm for detecting marine organisms (Wang & Xiao, 2023). It introduces improvements to the backbone network, utilizes online hard example mining (OHEM), and employs techniques like Generalized Intersection Over Union (GIOU) and Soft Non-Maximum Suppression. Another project, "Detection and Tracking of Humans in an Underwater Environment Using Deep Learning Algorithms," explores the application of Faster RCNN in detecting and tracking humans underwater. It achieves a maximum confidence level of 99% in various angles, postures, and lighting conditions (Zheng & Zhan, 2020).

Another relevant project, "Automatic Real-Time Detection of Infant Drowning Using YOLOv5 and Faster R-CNN Models Based on Video Surveillance," specifically targets the critical issue of infant drowning in swimming pools (Dulhare & Ali, 2023). It employs both YOLOv5 and Faster RCNN models, demonstrating that YOLOv5 outperforms in terms of speed and precision. Lastly, the project "Two-Stage Underwater Object Detection Network Using Swin Transformer" introduces a two-stage algorithm for underwater object detection, utilizing the Swin Transformer as the backbone network (Liu et al., 2022). It addresses challenges such as color offset, low contrast, and target blur in underwater image data, achieving substantial improvements on the URPC2018 dataset.

The objective shared by the other four literature studies was to detect drowning behavior in different real-time scenarios using advanced technologies such as deep learning and computer vision. Regardless of whether the focus was emphasized on the underwater operation or the swimming pool, the researchers utilized innovative methodologies to enhance real-time monitoring to address the need for timely and accurate drowning detection. In the realm of swimming pool safety, the fourth project on “Drowning behavior detection in swimming pool based on deep learning” pointed out the limitations of lifeguard-based supervision and presented a comprehensive methodology for drowning detection (Lei et al., 2022). Additionally, the fifth project “Pose Estimation of Swimmers from Digital Images Using Deep Learning” highlighted the importance of pose estimation as a fundamental input for various tasks, including drowning detection, and explored the impact of dataset scale on model performance (Cao, 2021). Furthermore, the sixth project “Automatic Real-Time Detection of Infant Drowning Using YOLOv5” emphasized the feasibility and application value of YOLOv5s in reducing infant drowning incidents, considering model performance metrics such as Mean Average Precision (mAP) and Frames Per Second (FPS) (He et al., 2023). The last project entitled “A Machine Vision Approach for Underwater Remote Operated Vehicle to Detect Drowning Humans” involved a machine vision implemented on underwater remote-operated vehicles (ROVs) that compared the effectiveness of Faster RCNN and YOLOv3 for underwater human detection, focusing on enhancing accuracy and efficiency in drowning rescue operations (K* et al., 2020).

2) Approach

The methodologies employed across these projects vary, showcasing the versatility of approaches in the field. The first project improves Faster RCNN by enhancing the backbone network, incorporating OHEM, and optimizing bounding box regression mechanisms (Wang & Xiao, 2023). The second project leverages Faster RCNN for human detection underwater, demonstrating its capabilities in different scenarios (Zheng & Zhan, 2020).

The third project introduces YOLOv5 and Faster RCNN for real-time infant drowning detection, emphasizing the use of video surveillance and data augmentation techniques (Dulhare & Ali, 2023). The fourth project proposes a two-stage algorithm using Swin Transformer as the backbone, addressing challenges in underwater image data through path aggregation, online hard example mining, and improved ROI processing (Liu et al., 2022).

The fourth project utilized eight underwater cameras to create a coordinate system and employed the BR-YOLOv4 algorithm for swimmer behavior analysis, involving frame analysis, spatial relationship assessment, and discriminant threshold application (Lei et al., 2022). The fifth project combines HRNet with YOLOv5 for multi-swimmer pose estimation using the top-down method and uses YOLOv5s model for swimmer detection, demonstrating improved precision and mAP@0.5 (Cao, 2021). The sixth project collects data from swimming pools through cameras and employs YOLOv5 and Faster R-CNN models for real-time infant drowning detection and utilizes data augmentation techniques to enhance model generalization (He et al., 2023). The last project employed underwater ROVs equipped with cameras and GPS tagging for drowning detection and uses Faster RCNN and YOLOv3 algorithms for underwater human detection (K* et al., 2020).

3) *Recommendation*

Across the studies, the conclusions emphasize the effectiveness of the proposed methodologies. The first project concludes that the improved Faster RCNN model demonstrates increased accuracy in underwater object detection, specifically achieving a 3.3% improvement in mean Average Precision (mAP) (Wang & Xiao, 2023). The second project concludes that Faster RCNN is capable of detecting and tracking humans underwater with a high confidence level of 99%, even in challenging conditions (Zheng & Zhan, 2020). The third project recommends YOLOv5 as an optimal model for infant drowning detection, achieving high precision and faster processing speed (Dulhare & Ali, 2023). The fourth project concludes that the improved Faster RCNN model, utilizing Swin Transformer, outperforms other algorithms in complex underwater environments, significantly improving mean Average Precision (mAP) (Liu et al., 2022).

In summary, these projects collectively showcase the advancements in underwater object detection methodologies, ranging from improvements to existing algorithms, the application of deep learning techniques, to the introduction of novel models like Swin Transformer. The findings contribute valuable insights for our group's task of conducting a comparative analysis of object detection algorithms for drowning detection, guiding our understanding of the applicability of these algorithms in water safety applications.

BR-YOLOv4 algorithm was recommended to be adopted in the fourth project for drowning detection due to its superior results of accuracy, speed, and precision compared to YOLOv3 and YOLOv5. The authors acknowledged the impact of environmental factors on swimmer behaviour recognition and emphasized the need for addressing challenges like object overlapping in future research (Lei et al., 2022). In the fifth project, the YOLOv5s algorithm was recommended for multi-swimmer pose estimation, emphasizing the significance of dataset size concerning model complexity. The constraints of limited available datasets were acknowledged by the authors and the need for continuous optimization to enhance detection accuracy has been highlighted (Cao, 2021). The application of YOLOv5s was suggested for the sixth project which is real-time detection of infant drowning incidents in swimming pools. The authors acknowledged the challenges such as distributed targets or complex environments with toy disturbances and committed to continuous optimization of the YOLOv5s model for accurate detection (He et al., 2023). Last but not least, due to the limitation of using YOLOv3, the Faster RCNN was recommended for underwater drowning detection, particularly in scenarios involving overlapping frames (K* et al., 2020).

II. Materials And Methods

A. *Software Requirement*

The drowning detection system is programmed in Python 3.7 or above, with key modules and packages used to ensure reliable functionality. These include Albumentations for picture enhancement, Certifi for SSL certificate checking, and OpenCV-Python for computer vision applications. Matplotlib and NumPy are used for data visualization and manipulation, respectively, while Scikit-Learn enables machine learning capabilities. PyTorch (Torch) is the deep learning framework, and other utilities like Playsound for audio playback and Requests for HTTP requests extend to the system's flexibility. The presence of particular libraries, like as Qudida, demonstrates that the elements are optimized for drowning detection. This complete suite of software tools, compatible with Python 3.7 and above, serves as the system's basis, allowing for efficient algorithm development, training, and deployment. (Kariyawasam, 2023)

TABLE I: Required Python Modules with Version

Python Modules	Version
Albumentations	1.3.1
Certifi	2023.11.17
OpenCV-Python	4.8.1.78
Matplotlib	3.8.2
Numpy	1.21.6
PyTorch	1.13.1
Playsound	1.3.0
Qudida	0.0.4
Scikit-learn	1.0.2

B. Hardware Requirement

The drowning detection system should run optimally on a computer platform with a quad-core CPU, at least 8 GB of RAM, and a GPU with CUDA capabilities for accelerated processing. A minimum of 100 GB of disc space is recommended for storing datasets, training models, and interim results. The system can function well with a regular internet connection to retrieve external data sources or updates. In the absence of a GPU, the technique may still be executed on a CPU, but at reduced processing rates. Real-time video processing from cameras is possible and the system works with typical camera resolutions and frame rates.

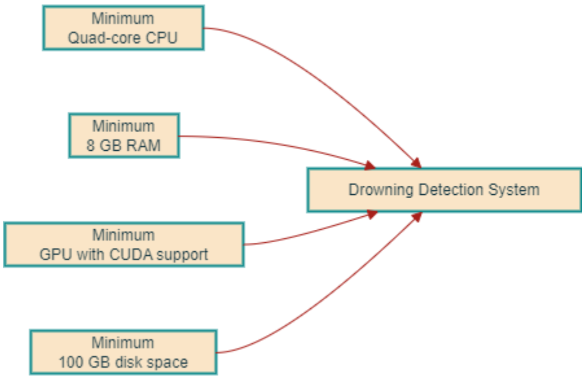

Fig. 3. Hardware Requirement for Drowning Detection System

C. Methodology

1) Dataset

As there is no dataset available for drowning and swimming, therefore a varied and representative dataset is obtained from GitHub from the “randhana” repository, YouTube, and Google using specific keywords such as “swimmers”, “swimming”, “drowning”, and “drowning in the swimming pool”. The dataset collected consists of 20 video clips and then is utilized in the development and evaluation of the drowning detection system. The dataset was rigorously chosen to include a variety of situations, environmental variables, and possible drowning incidences. It includes the following important components:

a) Face Detection Dataset

The algorithm's face detection component was based on a specific collection of video frames from a variety of drowning circumstances. This dataset contains:

- **Positive Samples:** Video frames of people in water-related situations, with an emphasis on instances where facial recognition is critical for detecting probable drowning victims.
- **Negative Samples:** Video frames that have no water-related context, assuring the model's specificity and generalizability.

The face recognition dataset is labeled to identify the presence and position of faces, making it easier to train and evaluate the SSD-based face identification model.

b) General Object Detection Dataset

The algorithm's general object identification component relies on a large dataset to train and evaluate the YOLOv3 model. This dataset includes:

- **Drowning-related Object:** Video frames of objects connected with probable drownings
- **Background Scenes:** Video frames that do not contain a drowning person or a person in a swimming state, which help the YOLOv3 model recognize relevant things from the surroundings.

Each image in the dataset has bounding boxes around things of interest, allowing the model to learn and generalize from these instances.

To ensure the algorithm's resilience, the dataset is separated into distinct subsets, 70% for training and 30% for testing, during the dataset splitting step. The training set was critical in training both the face detection and YOLOv3 models, and the testing set was then used to assess the algorithm's overall performance.

2) Architecture

The drowning detection system has a well-organized architecture, with each Python module playing an important part in ensuring a smooth and successful detection procedure. The cvlib module, which uses the power of YOLOv3 for cutting-edge object identification, serves as the system's core (Sharma, 2021). This module accurately recognizes humans in video streams, marking the initial stage in the drowning detection process. The Motion Analysis Module then takes over, following the motions of detected persons and using advanced algorithms to determine their safety (Gowtham, 2021). The dynamic interplay between the item detection and motion analysis modules serves as the system's backbone, ensuring a thorough grasp of the aquatic environment.

In addition to these fundamental components, the system includes a specific Data Handling Module, which streamlines interactions with the training dataset to optimize model performance (Hemanshi, 2021). The Alerting Module (Playsound Module) serves as the system's voice, rapidly informing important stakeholders when possible drowning occurrences are detected. Furthermore, the inclusion of YOLOv3 and additional utility modules for image processing, network connectivity, and device compatibility improves the system's flexibility in a variety of circumstances (Chung & Kamsin, 2023). Collectively, these components form a coherent and efficient design, utilizing modern computer vision and machine learning technologies to produce a robust drowning detection system that prioritizes individual safety in and around water bodies.

Apart from that, the drowning detection system has a well-structured flowchart shown in Figure 3.0, beginning with the receiving of video feeds from several cameras installed around a pool or water body. These video inputs are then routed via a Convolutional Neural Network (CNN), a sophisticated deep learning model adept at image processing tasks. The CNN, which was trained on a large collection of photos and videos of people in aquatic situations, can properly recognize human bodies in a variety of poses and lighting circumstances. The CNN output is then processed using the Yolov3 object detection method. Yolov3 specializes in quickly and precisely recognizing objects in photos, and in this case, it determines if a human is present in the water (Alamgeer et al., 2018). The system then determines whether the detected target is in a drowning scenario. If the analysis determines that a drowning scenario occurs, the system activates an alarm mechanism, which is represented by a buzzer or beep sound. If no drowning danger is found, the system cycles back to the first input step while continuously monitoring the video streams for prospective events. In essence, this flowchart illustrates a robust drowning detection system that uses powerful computer vision and machine learning technologies to quickly and reliably identify probable drowning accidents, improving overall water safety. (Bapardekar Yukta, 2022)

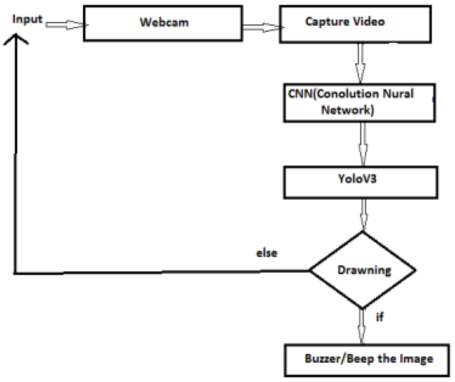

Fig. 4. Flowchart of Drowning Detection System *(Bapardekar Yukta, 2022)*

The block diagram of the drowning detection system shown in Figure 4.0 also demonstrates an extensive approach for analyzing indoor swimming pool captures with the ultimate purpose of detecting probable drownings and sounding a rescue alarm. Beginning with a picture recorded from an indoor pool, the system enters a pre-processing step. This step includes the translation of the image into the HSV colour space, which improves the system's capacity to distinguish objects based on colour. Binary thresholding is then used to transform the image into binary form to highlight certain characteristics or objects. The tracking phase follows, which includes contour detection to highlight object boundaries, calculating the area contained by each contour, and selecting the biggest contour. The biggest contour is believed to be the most visually important presence in the pool, maybe representing a human. When the analysis reveals that the greatest contour relates to a probable drowning incidence, the system generates an output signal, specifically a rescue alarm. This systematic technique combines colour analysis, contour recognition, and size evaluation to identify and respond to possible crises in indoor swimming pool environments, offering a quick and effective alarm system for the protection of those in trouble. (Nasrin Salehi, 2016)

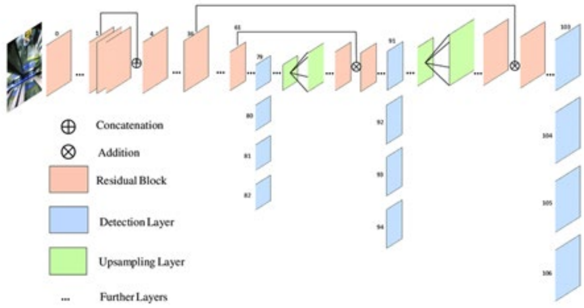

Fig. 5 Architecture of YOLOv3 *(Dai et al., 2020)*

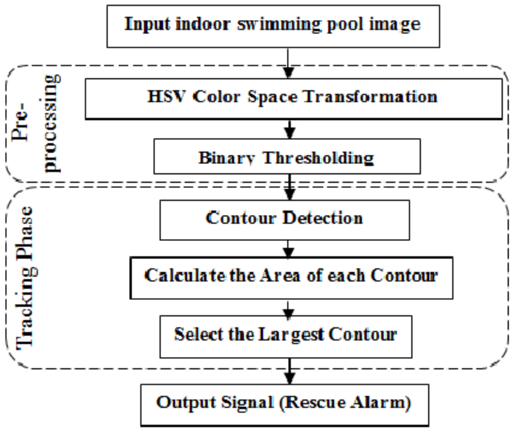

Fig. 6. Block Diagram of Drowning Detection System *(Nasrin Salehi, 2016)*

III. ALGORITHM IMPLEMENTATION

A. Purpose

The cutting-edge drowning detection system, which combines computer vision, deep learning, and the YOLOv3 object identification algorithm, represents a transformational goal centered on early detection and speedy alerting, as well as novel water safety solutions. Using YOLOv3's sophisticated object detection capabilities, the system rapidly locates persons who may be in difficulty in aquatic situations. This excellent early detection is supplemented by real-time video analysis using computer vision, providing a broad surveillance reach. When combined, this dynamic integration enables the system to proactively identify and swiftly inform caregivers, lifeguards, or authorities of probable drowning occurrences, dramatically lowering response time and improving the possibility of favorable outcomes. (Kariyawasam, 2023)

Beyond object recognition, their drowning detection system uses advanced algorithms for motion analysis, taking safety monitoring to a new level. This thorough method allows the system to distinguish between normal swimming activities and possible drownings by capturing individuals' movements in the water. The unique mix of early detection, quick warning, and motion analysis distinguishes their technology as a pioneering force, establishing new standards for water safety solutions. The commitment to harnessing cutting-edge technology demonstrates their determination to revolutionize processes, guaranteeing that their sophisticated system serves as a beacon in protecting lives in aquatic settings.

B. Parameters and Functions Modified

The object recognition in computer vision handles the task of identifying and localizing objects within an image or video. The key parameters used in object detection are the confidence level and Non-Maximum Suppression (NMS) thresholds. These two parameters play a crucial role in the NMS technique in order to select one entity from a large number of overlapping entities. (Prakash, 2021) Confidence level, also known as confidence score, refers to the model's certainty which is related to an object's presence within a bounding box. Each detected object is assigned a confidence score, hence, the higher this score, the greater the level of confidence in the object's presence. NMS thresholds serve as controllers for the overlapping between bounding boxes during the suppression process. It eliminates redundant and overlapping bounding, ensuring that closely grouped objects are retained. A lower NMS threshold would lead to greater suppression and remove entities with lower confidence scores but higher overlap. In reverse, higher NMS thresholds have less suppression, allowing the retention of more overlapping boxes. The optimum confidence score coupled with the appropriate threshold can detect objects accurately and filter out the most reliable objects. (Munawar, 2023)

The researchers have made changes to adjust these parameters after a few experiments. The iterative process aims to find an optimal combination of confidence level and NMS threshold for better drowning detection and reliable results. As drowning is considered as a safety-critical scenario, the researcher decided to decrease the confidence score and increase the NMS threshold to minimize the number of false positives in object detection. As a result, drowning detection became more accurate and precise.

IV. Results And Discussion

Following parameter modifications, notably targeting the Confidence value and Non-Maximum Suppression (NMS) thresholds, a study was performed to determine their impact on processing time per frame in the context of drowning detection. The findings include average, maximum, and minimum processing times per frame for various parameter settings, which were carefully documented to allow for a more comprehensive comparison.

TABLE II: Results of Confidence of 0.3 & Different NMS Thresholds

Confidence	NMS_Thresh	Average (sec)	Max (sec)	Min (sec)
0.3	0.1	0.2711	0.2930	0.2410
0.3	0.3	0.2698	0.3070	0.2430
0.3	0.5	0.2720	0.2980	0.2430

TABLE II shows the results of using a constant confidence value of 0.3 combined with several NMS threshold values ranging from 0.1, 0.3, and 0.5. Notably, when looking at the average processing time per frame, it is clear that a confidence value of 0.3 paired with an NMS threshold of 0.3 results in a faster processing speed, demonstrating its effectiveness in handling video frames quickly. Further study of the maximum and minimum processing times per frame indicates that the configuration with a confidence value of 0.3 and an NMS threshold of 0.1 leads the others in terms of processing time minimization. However, the trade-offs associated with decreasing the confidence threshold should be carefully considered, since they may increase the incidence of false positives or inaccurate detections (Tulyakov & Govindaraju, 2013). Furthermore, a lower NMS threshold increases the likelihood of keeping more duplicated bounding boxes, demanding a balanced approach to parameter optimization for reliable drowning detection.

TABLE III: Results of Confidence of 0.5 & Different NMS Thresholds

Confidence	NMS_Thresh	Average (sec)	Max (sec)	Min (sec)
0.5	0.1	0.2671	0.2990	0.2303
0.5	0.3	0.2683	0.2930	0.2325
0.5	0.5	0.2637	0.2890	0.2460

TABLE III highlights the results obtained from various combinations of Confidence values (0.5) and Non-Maximum Suppression (NMS) thresholds (0.1, 0.3, and 0.5). The average processing times vary just a little between the three adjustments. However, it is worth mentioning that the adjustment with a Confidence value of 0.5 and an NMS threshold of 0.5 has the lowest average processing time, at 0.2637 seconds. Additionally, the setting with a Confidence value of 0.5 and an NMS threshold of 0.1 records the lowest minimum processing time (0.2303 seconds), while the configuration with a Confidence value of 0.5 and an NMS threshold of 0.5 has the lowest maximum processing time (0.2890 seconds). The choice of NMS threshold appears to have an obvious impact on both average and maximal processing times. A higher NMS threshold (0.5) appears to contribute to a little quicker average processing time but at the expense of a slightly longer minimum processing time.

TABLE IV: Results of Confidence of 0.7 & Different NMS Thresholds

Confidence	NMS_Thresh	Average (sec)	Max (sec)	Min (sec)
0.7	0.1	0.2642	0.2970	0.2412
0.7	0.3	0.2659	0.3010	0.2400
0.7	0.5	0.2731	0.3540	0.2357

TABLE IV shows the results of using a constant confidence value of 0.7 combined with several NMS threshold values ranging from 0.1, 0.3, and 0.5. Based on the results, increasing the NMS threshold from 0.1 to 0.5 results in an increase in average processing time from 0.2642 to 0.2731 seconds. As the NMS threshold increases, it correspondingly increases the maximum processing time. The setup with a confidence level of 0.7 and an NMS threshold of 0.5 has the highest maximum processing time (0.3540 seconds). Conversely, the minimum processing time shifts with the lowest value recorded for the configuration with a Confidence of 0.7 and an NMS threshold of 0.5 (0.2357 seconds). When comparing settings with the same confidence level (0.7), it is clear that a higher NMS threshold results in slower processing time. This is notably obvious in an increase in both average and maximum processing times. The choice of NMS threshold appears to have a significant impact on both average and maximum processing times. A higher NMS threshold results in slower but potentially more accurate detection since it allows for stronger suppression of duplicated bounding boxes.

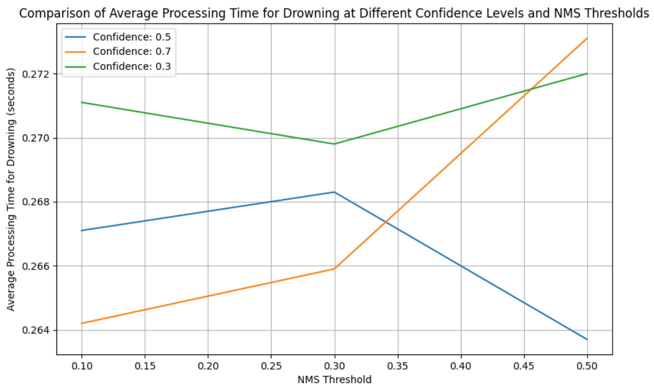

Fig. 7. Comparison Average Processing Time per Frame

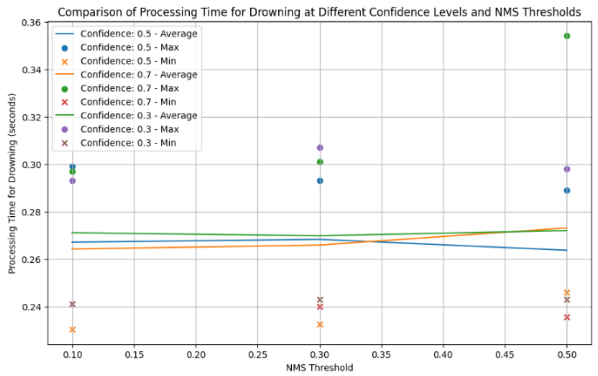

Fig. 8. Comparison Average, Maximum & Minimum Processing Time per Frame

After obtaining the results of three tables with different Confidence values and NMS thresholds, the results have been converted into a graph using Python matplotlib.pyplot. The average processing time per frame is plotted as a line graph while the maximum and minimum processing time are plotted as a scatter plot graph. According to Fig.1, it consistently shows that a Confidence value of 0.5 paired with an NMS threshold of 0.5 leads to a shorter processing time per frame. When compared to alternative adjustments, this configuration routinely shows competitive or even better average, maximum, and minimum processing speeds. Lowering the NMS threshold, particularly to 0.5, may lead to more accurate findings in terms of bounding box suppression, thereby lowering false positives (DS, 2023). However, it is critical to recognize the tradeoff between processing speed and accuracy. A higher NMS threshold typically leads to a slower processing time, as seen in the tables. The choice of Confidence and NMS threshold settings is critical for optimizing the drowning detection system. A setup with 0.5 confidence and 0.5 NMS threshold appears to establish a compromise between processing efficiency and detection accuracy. Lowering the confidence and NMS levels may improve processing speed, but careful evaluation is required. It may result in more false positives (incorrect detections) and the retention of redundant bounding boxes, lowering the system's accuracy. In a nutshell, a Confidence value of 0.5 coupled with an NMS threshold of 0.5 appears to be a good setup, providing competitive processing performance without considerably reducing drowning detection accuracy.

V. Conclusion

In conclusion, this research has delved into the drowning issue, especially in swimming pools, and proposed a comprehensive drowning detection system that employs advanced computer vision and deep learning and the YOLOv3 object recognition algorithm to detect drowning incidents accurately. This algorithm is instrumental in identifying and localizing objects, crucially recognizing human presence in aquatic environments. The system's architecture integrates real-time video analysis and motion analysis to enhance drowning detection capabilities. The primary parameters under consideration are the Confidence score and the Non-Maximum Suppression (NMS) threshold. The researcher learned that a high NMS threshold contributes to increased accuracy but at the expense of higher processing times. On the other hand, a low Confidence value can lead to an elevation in false positives. Striking a balance is crucial, as the Confidence value cannot be too low, and the NMS threshold cannot be too high. after a few evaluations of different parameter configurations, the recommended configuration of a Confidence value of 0.5 and an NMS threshold of 0.5 provides an optimal compromise, ensuring both efficiency and accuracy in drowning detection. It also showcases greater performance in terms of average, maximum, and minimum processing times per frame. This balanced approach enhances the overall performance of the system.

References

Alamgeer, Z., Selvaperumal, S. K., Prum, S., & Fei, T. K. (2018). Review of Car Make & Model Recognition Systems. *Journal of Applied Technology and Innovation*, 9-16.

Amanda. (2023, February 28). *9yo older sister attempted to save 8yo brother from drowning in Ipoh pool, both tragically pass away*. Retrieved from Wau Post: https://waupost.com/9yo-older-sister-attempted-to-save-8yo-brother-from-drowning-in-ipoh-pool-both-tragically-pass-away/#:~:text=As%20reported%20by%20local%20newspaper,brother%20drowning%20in%20the%20pool.

Bapardekar Yukta, B. R. (2022). DROWNING DETECTION SYSTEM USING CNN. *International Journal of Creative Research Thoughts (IJCRT)*, 2320-2882. Retrieved from https://ijcrt.org/papers/IJCRTO020016.pdf

Cao, X. (2021). Pose Estimation of Swimmers from Digital Images Using Deep Learning. *Auckland University of Technology*, 1-119.

Chung, C. V., & Kamsin, D. I. (2023). Informative and Deep Learning System to Help. *Journal of Applied Technology and Innovation*, 7-10.

Dai, Y., Liu, W., Li, H., & Liu, L. (2020). Efficient Foreign Object Detection Between PSDs and Metro Doors via Deep Neural Networks. *IEEE Access*, 1-12. doi:10.1109/ACCESS.2020.2978912

DS, K. (2023, March 14). *Non-Max Suppression*. Retrieved from Medium: https://korlakuntasaikamal10.medium.com/non-max-suppression-b0569e89e9e7

Dulhare, U. N., & Ali, M. H. (2023). Underwater human detection using faster R-CNN with data augmentation. *Materials Today: Proceedings*, 1940-1945.

Gowtham. (2021, September 23). *Motion Detection and Tracking using OpenCV Python*. Retrieved from Medium: https://medium.com/@gowtham180502/motion-detection-and-tracking-using-opencv-python-ea75ef927e72

He, Q., Mei, Z., Zhang, H., & Xu, X. (2023). Automatic Real-Time Detection of Infant Drowning Using YOLOv5 and Faster R-CNN Models Based on Video Surveillance. *Journal of Social Computing*, 62-73. doi:10.23919/JSC.2023.0006

Hemanshi. (2021, June 26). *data handling in python*. Retrieved from Medium: https://lightupthesky1111.medium.com/data-handling-in-python-385107e6c9a1

K*, Y. S., M, K., & S, V. P. (2020). A Machine Vision Approach for Underwater Remote Operated Vehicle to Detect Drowning Humans. *Advance in Science, Technology and Engineering System Journal*, 1734-1740. doi:10.25046/aj0506207

Kariyawasam, P. (2023, May 9). *Drowning Detection*. Retrieved from Github: https://github.com/randhana/Drowning-Detection-

Lei, F., Zhu, H., Tang, F., & Wang, X. (2022). Drowning behavior detection in swimming pool based on deeplearning. *Springer Link*, 1683-1690. doi:https://doi.org/10.1007/s11760-021-02124-9

Liu, J., Liu, S., Xu, S., & Zhou, C. (2022). Two-Stage Underwater Object Detection Network. *IEEE Access*, 117235-117247. doi:10.1109/access.2022.3219592

Loh, I. (2023, April 26). *Toddler found drowned in kiddie pool at Tambun house*. Retrieved from The Star: https://www.thestar.com.my/news/nation/2023/04/26/toddler-found-drowned-in-kiddie-pool-at-tambun-house

Munawar, M. R. (2023, July 15). *Relationship Between Confidence and NMS Thresholds in Object Detection*. Retrieved from VisoByte: https://www.visobyte.com/2023/07/Relationship-Between-Confidence-and-NMS-Thresholds-in-Object-Detection.html

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

Pfordten, D. (2023, May 25). *INTERACTIVE: Why so many of our kids are drowning and what can be done*. Retrieved from The Star: https://www.thestar.com.my/news/nation/2023/05/25/interactive-drowning-among-leading-causes-of-death-for-kids

Prakash, J. (2021, June 2). *Non Maximum Suppression: Theory and Implementation in PyTorch*. Retrieved from LearnOpenCV: https://learnopencv.com/non-maximum-suppression-theory-and-implementation-in-pytorch/

Sharma, G. (2021, July 16). *cvlib : Yolo Object Detection in Seconds!* Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/06/yolo-object-detection-in-seconds-cvlib/

Tulyakov, S. (., & Govindaraju, V. (. (2013). Matching score fusion methods. *Artificial Intelligence for Future Generation Robotics*, 151-175.

Wang, H., & Xiao, N. (2023). Underwater Object Detection Method Based on Improved Faster RCNN. *Applied Sciences*, 1-15. doi:https://doi.org/10.3390/app13042746

Zheng, M., & Zhan, W. (2020). From Body Parts Responses to Underwater Human Detection: A Deep Learning Approach. *Master of Science in Computer Science*, 1-43.