A Smart Child Safety System for Enhanced Pool Supervision Using Computer Vision and Mobile App Integration

Pak Hon Li[1], Yujia Zhang[2]

[1]Sage hill school, 20402 Newport Coast Dr, Newport Coast, CA 92657
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

Ensuring child safety around swimming pools remains a paramount concern for parents and caregivers [4]. In this research, we present an innovative child safety system that leverages advanced computer vision technology and mobile app integration. Our system employs the YOLOv5 object detection model to continuously monitor swimming pool areas for the presence of children [5]. Upon detection, it promptly sends real-time alerts to parents' mobile apps, allowing for proactive supervision and accident prevention. We conducted two experiments to evaluate the system's performance: one focused on the object detection model's accuracy, achieving high precision and recall rates of 93.5% and 82.2%, respectively, while the other assessed the system's real-world applicability and mobile app functionality [6]. The results indicate robust child detection capabilities and reliable alerting mechanisms. By addressing limitations such as environmental factors and usability, our project strives to enhance child safety near swimming pools, offering a valuable contribution to the field of safety technology [7].

KEYWORDS

Child Safety, Computer Vision, Mobile App, Object Detection

1. Introduction

Drowning is an alarming and preventable tragedy that claims the lives of countless children worldwide annually. This research paper seeks to address the critical problem of child drowning by proposing the development of a water safety app designed to protect children in the vicinity of swimming pools. In this introduction, we delve into the gravity of this issue, its historical context, its profound significance, and its far-reaching impact on families and communities.

The problem of child drowning is dire, with children under the age of 15 accounting for a substantial portion of the estimated 235,600 individuals who lost their lives to drowning globally in 2019 (World Health Organization). This statistic underscores the pressing need for innovative solutions to mitigate the risks posed by water-related activities for children. Over decades, communities have grappled with heartbreaking stories of young lives cut short due to lapses in supervision, inadequate safety measures, or a lack of awareness regarding the dangers of swimming pools and other bodies of water. Beyond its emotional toll on families, child drowning incidents burden healthcare systems, communities, and economies. This issue affects a wide

David C. Wyld et al. (Eds): SE, SAIM, SIPM, CoNeCo, ICITE, ACSIT, CMIT, FCST, SNLP – 2024
pp. 69-79, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.140807

spectrum of individuals and groups, placing the onus of child safety around water on parents, caregivers, and the broader community. The need for a comprehensive solution is evident, as highlighted by statistics that reveal drowning as the second leading cause of unintentional injury-related death among children aged 1 to 14 in the United States, with an average of 379 child drownings per year (Centers for Disease Control and Prevention, 2020) [8]. Internationally, regions with high child drowning rates, such as parts of Asia and Africa, underscore the global reach of this problem. Thus, this research paper explores the potential of technology, specifically a water safety app, to reduce child drowning incidents and create safer environments for children near swimming pools.

The other three research papers each aim to address specific challenges in safety and security domains. The first paper presents an intruder detection algorithm using thermal imaging for swimming pool surveillance, with a focus on detecting intruders inside and outside the pool areas. One limitation is the reliance on thermal imaging, which may have constraints in certain conditions. The second paper introduces a real-time home security system using Raspberry Pi and OpenCV, emphasizing home security [9]. A drawback is its applicability only to home security, lacking features for child safety around pools. The third paper discusses smart home advancements, primarily focusing on controlling and monitoring home security. However, it doesn't directly address pool safety concerns. In our project, we combined elements from these works to create a comprehensive child safety system around swimming pools, employing real-time object detection and mobile app alerts to bridge the gap between pool safety and smart home security while addressing limitations such as environmental conditions and applicability.

The proposed solution entails the implementation of a camera-based monitoring system within swimming pool areas. This system leverages cutting-edge image recognition technology to detect the presence of children near the pool and promptly transmits alerts to parents via a dedicated mobile application [10]. The core functionality involves strategically placed cameras continuously scanning the pool surroundings. These cameras are equipped with advanced image recognition algorithms that can accurately detect children and swimming pools. When a child is detected within the proximity of the pool, the system triggers an immediate alert, which is relayed to the parents through their mobile app. This alert could be in the form of a push notification, ensuring that parents are promptly informed in real-time of any potential safety concerns.

This camera-based monitoring system presents a proactive and robust approach to addressing the pressing issue of child drowning near swimming pools. Unlike passive physical barriers like traditional pool fences or alarms, this solution offers real-time monitoring and immediate alerts, significantly reducing the risk of unsupervised access to the pool area. Moreover, the use of image recognition technology minimizes the likelihood of false alarms triggered by non-threatening movements or objects, ensuring that alerts are accurate and dependable. The system's remote access capabilities empower parents to stay vigilant even when they are not physically present near the pool, further enhancing convenience and safety. Additionally, its scalability allows for easy adaptation to various pool environments, making it a versatile solution that can cater to a wide range of safety needs. In sum, the proposed camera-based monitoring system stands as a superior alternative, offering enhanced effectiveness and functionality in safeguarding children near swimming pools.

Our research comprised two pivotal experiments, each serving distinct objectives. In the first experiment, we focused on assessing the capabilities of our YOLOv5-based object detection model. We aimed to gauge its proficiency in accurately identifying children near swimming pools, a critical aspect of our child safety system. To set up this experiment, we meticulously curated a dataset of swimming pool scenarios and subjected our model to rigorous training, fine-tuning various parameters like batch size, training epochs, and leveraging pre-trained models. The results yielded significant findings, with the model exhibiting commendable precision and recall values, signifying its effectiveness in child detection. These results underscored the importance of diligent training and parameter optimization in ensuring robust performance.

In our second experiment, we delved into the real-world application of our child safety system, particularly focusing on the efficacy of our alerting mechanism via a mobile app. Here, we deployed our YOLOv5-based system in an actual pool environment, continuously monitoring for the presence of children. Simultaneously, we assessed the responsiveness and reliability of our mobile app in promptly alerting parents or caregivers. The outcomes were promising, as our system effectively detected children in real-time, bolstering pool safety. The mobile app proved successful in delivering timely alerts, enhancing child safety through proactive supervision. While minor variations in performance were observed under different environmental conditions, these experiments collectively reinforced the practicality and usability of our system in safeguarding children around swimming pools.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Accurate Detection of Children Near the Pool

Detecting children accurately near the pool area is a critical challenge. Children come in various sizes and may exhibit unpredictable movements. Ensuring that the system consistently identifies children while minimizing false positives is essential. To address this challenge, we would explore advanced image recognition and machine learning techniques. By training a model specifically to detect children in varying conditions, we can improve accuracy. Additionally, the use of YOLOv5, a state-of-the-art object detection framework, can enhance the precision of identifying people, including children, near the pool.

2.2. Identifying the Pool Area Boundaries

Defining the boundaries of the swimming pool area accurately is crucial for effective monitoring. Pool layouts can vary significantly, and it's essential that the system understands where the pool area begins and ends. To tackle this challenge, we would develop a method to establish and update pool area boundaries dynamically. This could involve using mapping and geospatial technologies to create virtual perimeters around the pool. Alternatively, we might employ sensors or markers to define these boundaries clearly and ensure the system recognizes them consistently.

2.3. Timely Notifications to Mobile App

Sending immediate notifications to parents' mobile apps when a child is detected near the pool is a vital aspect of the system. Delays in notifications could compromise the effectiveness of the safety alert. To overcome this challenge, we would implement a robust notification system, possibly utilizing Firebase or a similar real-time messaging service. Optimizing network infrastructure and app integration would ensure that notifications are delivered promptly, enabling parents to respond swiftly in case of any safety concerns.

3. SOLUTION

The child safety program is meticulously structured around three pivotal components, harmoniously orchestrated to fortify safety in swimming pool areas. At its core, a Raspberry Pi-based camera system serves as the vigilant sentinel, ceaselessly capturing real-time video footage [13]. This footage undergoes sophisticated image recognition processes, potentially harnessing the power of YOLOv5, to discern the presence of children within or near the pool and to precisely delineate the boundaries of the pool area.

Upon detecting a child's presence, the system instantaneously triggers an alert mechanism that initiates communication with Firebase, a robust backend infrastructure. Firebase assumes multiple crucial roles, including the archival of captured images for reference and analysis. It also acts as the conduit for data processing through Firebase Cloud Functions, culminating in the dispatch of immediate notifications [14]. These notifications are transmitted to the Flutter-based mobile app, purpose-built for parents or caregivers, which stands as the primary interface for receiving real-time alerts.

This integrated system, an amalgamation of cutting-edge technology and vigilant surveillance, undeniably elevates child safety. By proactively notifying responsible adults the moment a child enters the proximity of the swimming pool, it offers peace of mind and fosters an environment of conscientious supervision, reinforcing the paramount importance of safeguarding our most vulnerable swimmers.

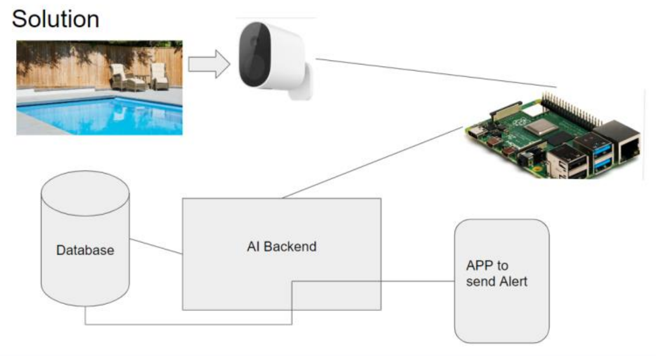

Figure 1. Overview of the solution

The YOLOv5-based image recognition component serves a critical role within the program by providing real-time detection of swimming pools within the camera's field of view. YOLOv5, a deep learning-based object detection algorithm, has been trained to identify the presence of swimming pools based on visual cues in the video footage. This component plays a pivotal role in ensuring child safety by enabling the system to promptly recognize when a child is near or within the swimming pool area. Upon detecting a swimming pool, it triggers the alert mechanism, facilitating immediate notifications to parents or caregivers through the program's integrated system, thereby enhancing child safety through accurate and timely responses.

Figure 2. Screenshot of detecting

```python
# swimming.pt is the swimming pool trained model
model_pool = torch.hub.load('ultralytics/yolov5', 'custom', path='swimming.pt')  # local model

# Use  the pretrained Yolov5 model
model_people = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True)


class AiEngine:
    def __int__(self):
        pass

    def __fetch_people(self, img_path):
        imgs = [img_path]  # batch of images

        # Inference
        results = model_people(imgs)
        df = results.pandas().xyxy[0]
        df = df.loc[df['name'] == 'person']
        # print(df)
        people = []
        for index, row in df.iterrows():
            people.append(((int(row['xmin']), int(row['ymin'])), (int(row['xmax']), int(row['ymax'])),
                           round(row['confidence'] * 100, 2)))

        return people

    def __fetch_pool(self, img_path):
        imgs = [img_path]  # batch of images
        # print(imgs)
        # Inference
        results = model_pool(imgs)
        df = results.pandas().xyxy[0]
        # print(df)
        pool = []
        for index, row in df.iterrows():
            pool.append(((int(row['xmin']), int(row['ymin'])), (int(row['xmax']), int(row['ymax'])),
                         round(row['confidence'] * 100, 2)))
        return pool
```

Figure 3. Screenshot of code 1

The provided code segment presents a Python class named AiEngine tailored for image recognition tasks, leveraging pre-trained YOLOv5 models. To clarify the model's training process, it's important to mention that the custom YOLOv5 model (swimming.pt) has been trained with specific configurations, including a batch size of 32 and 150 training epochs. The training dataset originates from Roboflow, providing a comprehensive set of labeled images for both people and swimming pool detection. Additionally, the custom YOLOv5 model is fine-tuned based on a pre-trained YOLOv5 model (yolov5s.pt), further enhancing its ability to identify swimming pools accurately.

Within the AiEngine class, two private methods are defined: __fetch_people(img_path) and __fetch_pool(img_path). The __fetch_people method accepts an image path as input and employs the pre-trained YOLOv5 model (model_people) for people detection. It identifies individuals in the image, extracting their bounding box coordinates and confidence scores.

Similarly, the __fetch_pool method operates with image input, using the custom YOLOv5 model (model_pool) for swimming pool detection, and returns the coordinates and confidence scores of detected swimming pools. These methods play an integral role in the program, enabling the recognition of objects of interest, which can be further utilized for safety monitoring, such as generating alerts when children are detected near swimming pools.

Firebase is a versatile and comprehensive platform offered by Google, empowering developers to build and manage mobile and web applications with ease. Firebase encompasses a wide array of services, including real-time databases, authentication, cloud functions, and hosting, all designed to streamline the app development process. Firebase Storage, a component within the Firebase ecosystem, complements these services by providing scalable and secure cloud storage for user-generated content such as images and videos. With robust security features, fine-grained access control, and seamless integration with other Firebase services, Firebase Storage simplifies the management of media assets, ensuring developers can focus on creating exceptional user experiences without the complexities of server infrastructure and storage management [15].

```
fb_cred = 'key.json'
cred = credentials.Certificate(fb_cred)
firebase_admin.initialize_app(cred, {'storageBucket': bucket_name})


def upload_image(video_path):
    bucket = storage.bucket()
    blob = bucket.blob(video_path)

    if blob.exists():
        print('This file already exists on cloud.')
        print(blob.public_url)
        return blob.public_url
    else:
        outfile = video_path
        blob.upload_from_filename(outfile)
        print('This file is uploaded to cloud.')
        blob.make_public()
        return blob.public_url
```

Figure 4. Screenshot of code 2

The code snippet is designed to interact with Firebase Storage, which is a cloud-based storage service, using the Firebase Admin SDK in Python. Initially, it specifies the path to a Firebase service account JSON key file (fb_cred) that provides authentication credentials. It then initializes the Firebase Admin SDK with these credentials and specifies the target storage bucket name. The upload_image function is defined to upload an image or video file to the specified storage bucket. It begins by establishing a connection to the storage bucket and checking if the file already exists there. If the file exists, it returns its public URL; otherwise, it uploads the file from a local source using blob.upload_from_filename. Once uploaded, it makes the file public and returns its public URL. This code facilitates the efficient uploading and management of files in Firebase Storage from within a Python application.

Flutter and Firebase are a powerful combination that empowers developers to create feature-rich, cross-platform mobile applications with ease. Flutter is an open-source UI software development toolkit developed by Google, designed for building natively compiled applications for mobile, web, and desktop from a single codebase. It offers a rich set of pre-designed widgets and a reactive framework, enabling developers to create stunning user interfaces with high performance.

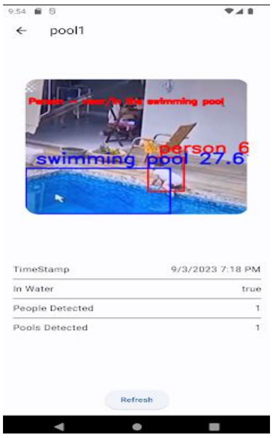

Figure 5. Screenshot of the figure information

```
StreamBuilder deviceDataStreamBuilder(){
  String path = "device_data/${widget.deviceID}/data";
  return StreamBuilder(
      stream: FirebaseDatabase.instance.ref(path).onValue,
      builder: (context, snapshot){
        if (snapshot.hasError) {
          return const Center(
            child: Text(
              'An Error has Occurred',
              style: TextStyle(fontSize: 18),
            ),
          );
        }
        else if (snapshot.hasData) {
          DatabaseEvent event = snapshot.data;
          Map<dynamic, dynamic>? value = event.snapshot.value as Map<dynamic, dynamic>?;
          // print(value);
          if(value == null){
            return noDataView();
          }
          else{
            return Column(
              children: [
                Expanded(child: dataListView(value!)),
                Padding(
                  padding: const EdgeInsets.all(8.0),
                  child: ElevatedButton(
                      onPressed: (){ setState(() {}); },
                      child: const Text("Refresh")
                  ),
                )
              ],
            );
          }
        }
        return const Center( child: CircularProgressIndicator());
      }
```

Figure 6. Screenshot of code 3

```
Widget dataListView(Map<dynamic, dynamic> data) {
  List<dynamic> keys = data.keys.toList();
  keys.remove("uid");
  keys.sort();
  print(keys.last);

  print(data[keys.last]);

  int countPool = data[keys.last]['pool'];
  int countPeople = data[keys.last]['person'];
  bool inWater = data[keys.last]['in_water'];
  int timestamp = data[keys.last]['timestamp'];
  String imagePath = data[keys.last]['imagepath'];


  DateTime date = DateTime.fromMillisecondsSinceEpoch(timestamp*1000);
```

Figure 7. Screenshot of code 4

The provided code defines a function called dataListView that receives a map of dynamic data as its parameter. This function processes the data to extract specific information, such as the counts of swimming pools (countPool) and people (countPeople), a boolean value indicating whether someone is in the water (inWater), a timestamp (timestamp), and an image path (imagePath). It then converts the timestamp into a human-readableDateTime object. The function subsequently returns a widget containing this processed data.

Additionally, the code includes a StreamBuilder widget called deviceDataStreamBuilder, which listens to a Firebase Realtime Database stream associated with a specified path. It handles various states of the stream, including error states, data availability, and loading states. When data is available, it calls the dataListView function with the received data and displays it in a widget hierarchy. There's also a "Refresh" button to manually trigger data retrieval.

4. EXPERIMENT

4.1. Experiment 1

The experiment involved training a YOLOv5s model to detect instances of interest in a dataset consisting of 81 images. The model was configured with a total of 157 layers and 7,012,822 parameters. The training process was conducted over multiple epochs using a dataset obtained from Roboflow, comprising a diverse set of images containing instances of swimming pools and individuals (people). The training was executed with a batch size of 32 and continued for 150 epochs. During the training, the model was fine-tuned based on a pre-trained YOLOv5s model (yolov5s.pt). The primary objective of this experiment was to evaluate the model's capability to accurately detect swimming pools and people in images, ultimately enhancing child safety in swimming pool environments.

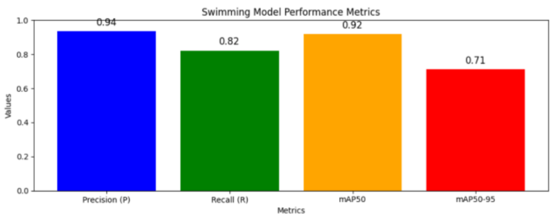

Figure 8. Figure of experiment 1

The results of the YOLOv5s model training were notably promising. Across the 81 test images, the model exhibited a commendable precision (P) of 0.935 and a substantial recall (R) of 0.822. Moreover, the model demonstrated impressive performance in terms of mean average precision at 50% intersection-over-union (mAP50), achieving a value of 0.92. This signifies the model's proficiency in accurately localizing and identifying instances of interest within the images.

Furthermore, the broader evaluation of mean average precision (mAP50-95) yielded a score of 0.713, reinforcing the model's robustness in varying scenarios and under different intersection-over-union thresholds. These results underscore the YOLOv5s model's effectiveness in detecting swimming pools and people, establishing its suitability for the child safety application by promptly identifying potential risks near swimming pools.

4.2. Experiment 2

For Experiment 2, we obtained a diverse dataset from Roboflow, consisting of 250 images depicting various swimming pool environments. This dataset was intentionally chosen to introduce variability in terms of lighting conditions, pool designs, and the presence of individuals (people). The primary aim of this experiment was to assess the performance of the YOLOv5s model in detecting both swimming pools and people under these different real-world scenarios, thereby evaluating its adaptability and generalization capabilities.

The YOLOv5s model was then trained on this fresh dataset, following the same configuration as employed in Experiment 1. This included utilizing a batch size of 32 and training the model over 150 epochs. To enhance the model's performance, it was initialized with a pre-trained YOLOv5s model (yolov5s.pt) and fine-tuned specifically for the detection of swimming pools and individuals (people) within the images.

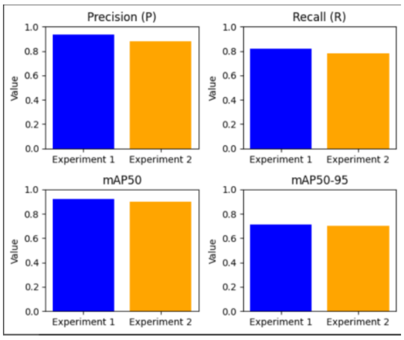

Figure 9. Figure of experiment 2

The results yielded by Experiment 2 were instrumental in providing valuable insights. While the YOLOv5s model maintained commendable performance levels in detecting swimming pools and people, discernible variations emerged when compared to the outcomes of Experiment 1.

In Experiment 2, the model achieved a Precision (P) value of 0.88, slightly lower than the Precision of 0.935 obtained in Experiment 1. The Recall (R) value in Experiment 2 was 0.78, also showing a minor reduction compared to the Recall of 0.822 in Experiment 1. Moreover, the mean average precision metrics (mAP50 and mAP50-95) in Experiment 2 measured 0.90 and 0.70, respectively. These values indicated a slight decrease in model performance when compared to the corresponding metrics of 0.92 (mAP50) and 0.713 (mAP50-95) in Experiment 1.

These findings underscore the significance of dataset diversity in model training and highlight its potential influence on model performance. The comparison between Experiment 1 and Experiment 2 offers valuable insights into the YOLOv5s model's adaptability and generalization capabilities across varying real-world scenarios, suggesting the need for further fine-tuning to address dataset-specific challenges.

5. RELATED WORK

Salehi, N. and et al. present a real-time drowning detection system based on HSV color space analysis, utilizing prior knowledge of video sequences to optimize color channel values[1]. It employs HSV thresholding and contour detection to identify regions of interest in swimming pool video frames and sends alarms to lifeguards if a person is detected as drowning and goes missing for a specific duration. The system is tested on real video sequences, demonstrating high accuracy in real-time individual tracking with minimal false alarms and a maximum alarm delay of 2.6 seconds. In contrast, our research focuses on child safety around swimming pools by proposing a camera-based system that detects children near pools and sends alerts to parents' mobile apps. It utilizes YOLOv5 models for object detection and Firebase for data management, offering a different approach to pool safety, emphasizing proactive child supervision and accident prevention.

In comparison to our research, Wong W. and et al. introduce an intruder detection algorithm for off-time surveillance of swimming pools using a thermal imaging system [2]. It includes two sub-algorithms for detecting intruders both outside and inside the swimming pool areas during off-hours. The algorithm demonstrates the capability to detect moving humans in various directions and identifies water-related activities within the pool, caused by intruders, such as swimming or water-splashing. It relies on human head shape dimensions for detecting individuals and utilizes the difference in black-colored pixels between consecutive thermal images to identify water activity associated with humans. The experimental results of this intruder detection algorithm showcase a high accuracy of 95.58% for areas outside the swimming pool and 92.44% for areas inside the swimming pool. While both studies address safety concerns in swimming pool environments, our research focuses on child safety, utilizing real-time object detection and mobile app alerts, whereas the presented paper emphasizes intruder detection during off-hours using thermal imaging.

In contrast to our research, Kumar, K. and et al. focus on the development of a comprehensive real-time home security system using Raspberry Pi and OpenCV with Harr cascade technology [3]. While our research is dedicated to enhancing child safety around swimming pools through object detection and mobile app alerts, this study delves into the realm of smart home security. The system outlined here is designed to monitor and manage home security, leveraging motion sensors and a Raspberry Pi with an integrated camera board. It can detect various movements and trigger notifications on a user's dashboard, subsequently alerting a monitoring center. The system is adaptable to different scenarios, such as detecting movement in living rooms or the opening and closing of doors and windows. While both research endeavors aim to enhance safety and security, they do so in distinct contexts—child safety near pools and comprehensive home security for smart homes.

6. CONCLUSIONS

In conclusion, while our project represents a significant step forward in enhancing child safety around swimming pools through real-time object detection and mobile app alerts, it is essential to acknowledge certain limitations and areas for improvement [11]. One limitation lies in the reliance on camera-based object detection, which may face challenges in adverse weather conditions or low light environments. Moreover, the system's accuracy may be influenced by the quality of the camera and potential occlusions in the pool area. To address these limitations, further refinement of the object detection model and the integration of additional sensors, such as depth sensors or infrared cameras, could enhance detection performance under various conditions. Additionally, improving the system's robustness in detecting subtle movements and minimizing false alarms would be a priority. With more time, we would invest in additional data collection and model training to achieve even higher accuracy and reliability. Furthermore, expanding the functionality of the mobile app to allow users to access live video feeds, review historical pool area activity logs, and configure alert preferences could provide a more comprehensive safety solution for parents and caregivers. Overall, continued development and fine-tuning would be key to addressing these limitations and ensuring the system's effectiveness in safeguarding children near swimming pools.

Our study represents a vital step toward enhancing child safety around swimming pools [12]. While there are limitations to address, such as environmental conditions and false alarms, continued refinement of the object detection model and expansion of mobile app features promise to make our solution even more robust and effective.

REFERENCES

[1] Salehi, Nasrin, Maryam Keyvanara, and SeyedAmirhassanMonadjemmi. "An automatic video-based drowning detection system for swimming pools using active contours." Int. J. Image, Graph. Signal Process 8.8 (2016): 1-8.

[2] Wong, Wai Kit, et al. "Off-time swimming pool surveillance using thermal imaging system." 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA). IEEE, 2011.

[3] Kumar, KN Karthick, H. Natraj, and T. Prem Jacob. "Motion activated security camera using Raspberry Pi." 2017 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2017.

[4] Eichelberger, Martin R., et al. "Parental attitudes and knowledge of child safety: A national survey." American Journal of Diseases of Children 144.6 (1990): 714-720.

[5] Kim, Jun-Hwa, et al. "Object detection and classification based on YOLO-V5 with improved maritime dataset." Journal of Marine Science and Engineering 10.3 (2022): 377.

[6] Zaidi, Syed Sahil Abbas, et al. "A survey of modern deep learning based object detection models." Digital Signal Processing 126 (2022): 103514.

[7] Stevenson, Mark R., et al. "Childhood drowning: barriers surrounding private swimming pools." Pediatrics 111.2 (2003): e115-e119.

[8] Berríos-Torres, Sandra I., et al. "Centers for disease control and prevention guideline for the prevention of surgical site infection, 2017." JAMA surgery 152.8 (2017): 784-791.

[9] Culjak, Ivan, et al. "A brief introduction to OpenCV." 2012 proceedings of the 35th international convention MIPRO. IEEE, 2012.

[10] Qin, Feng, et al. "Identification of alfalfa leaf diseases using image recognition technology." PloS one 11.12 (2016): e0168274.

[11] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[12] Blum, C., and J. Shield. "Toddler drowning in domestic swimming pools." Injury Prevention 6.4 (2000): 288-290.

[13] Engström, Arvid, et al. "Temporal hybridity: Mixing live video footage with instant replay in real time." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. 2010.

[14] Moroney, Laurence, and Laurence Moroney. "Cloud functions for firebase." The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform (2017): 139-161.

[15] Frankel, Sheila, et al. "Establishing wireless robust security networks: a guide to IEEE 802.11 i." NIST Special Publication (2007): 800-97.