AN INTELLIGENT MOBILE APPLICATION TO MONITOR CHILDREN'S SAFETY USING DEEP LEARNING AND OBJECT DETECTION

Yutong Zhang[1], Ang Li[2]

[1]Sage Hill School, Newport Coast, CA 92657
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

In the realm of home safety, the heightened risk of injury among unsupervised children, particularly from window-related falls, represents a significant challenge [1]. This study introduces an innovative solution to mitigate such risks through a novel integration of technology and artificial intelligence. We propose a comprehensive system that harnesses the power of Deep Learning, specifically utilizing the YOLOv8n algorithm, in conjunction with a Raspberry Pi platform for real-time hazard detection [2]. To address the critical aspect of data transfer, our system employs Firebase for efficient and timely communication between components. Acknowledging the limitations posed by initial model inaccuracies, our approach involved augmenting our dataset to ensure diversity and mitigate the risk of overfitting, thereby enhancing the model's predictive accuracy. This paper details our experimentation with various configurations, including an attempt to utilize YOLOv8x, which was ultimately revised to YOLOv8n due to computational constraints of the Raspberry Pi [3]. The robustness of our system was rigorously tested across diverse scenarios involving windows and doors, establishing a comprehensive dataset that underscores the system's effectiveness. By integrating real-time detection with an intuitive user interface, our system offers a proactive tool for parents to enhance home safety for their children. This contribution not only addresses a pressing societal issue but also advances the application of Deep Learning and IoT technologies in the domain of domestic safety.

KEYWORDS

YoloV8n, Object detection and recognition, Raspberry PI, OpenCV, Picamera2

1. INTRODUCTION

The prevalent problem addressed in this research is the increased risk of injury to unsupervised children at home, particularly incidents involving falls from windows. This issue has long been a concern in child safety, with a history of increasing awareness over the past decades. Despite advancements in home safety measures, the risk persists, underscoring the importance of this problem.

The significance of this problem lies in its direct impact on child safety [4]. Unsupervised children, often due to parents' work commitments or other reasons, are vulnerable to various household hazards, with window falls being among the most serious. These incidents not only

David C. Wyld et al. (Eds): NATL, SEAPP, SIGEM, MLTEC, CSEA – 2024
pp. 37-54, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.14205

cause immediate harm but can also lead to long-term physical and psychological consequences for the child and the family.

Statistically, the magnitude of this problem is substantial. In the United States, over four million children suffer injuries annually while being home alone, a figure that starkly highlights the gravity of the situation. Such statistics not only reflect the physical harm to children but also suggest a broader societal impact, including healthcare costs and emotional trauma to families.

This problem affects not only the individual children and their families but also has wider implications for community health and safety. In the long run, failing to address this issue can lead to increased healthcare expenses, loss of productivity for the parents, and long-term psychological effects on the children involved. The development of a robust mobile application designed to alert parents of potential dangers faced by their children at home aims to mitigate these risks, offering a proactive solution to enhance child safety and parental peace of mind.

In summarizing the methodologies from Section 5, three innovative solutions employing computer vision and deep learning for critical societal issues are highlighted:

1. Pen-Holding Gesture Recognition: Qu et al.'s research focuses on recognizing correct pen-holding gestures in children, utilizing YOLOv8 for enhanced accuracy [5]. This method aims to assist in correcting pen-holding techniques, crucial for children's developmental stages. While achieving high accuracy, its limitation lies in its deployment constraints, primarily its incompatibility with mobile devices, limiting its applicability in dynamic educational settings. Our project seeks to extend this technology to mobile platforms, providing broader accessibility and incorporating real-time feedback mechanisms for immediate corrective actions.

2.Child Safety in Domestic Environments: Tan and Goh's system aims to prevent common household accidents involving infants and toddlers by detecting the proximity of children to dangerous objects using CCTV and YOLOv8 [6]. Despite its real-time monitoring capability, the system's reliance on 2D views could overlook certain risks. Our improvement efforts focus on broadening detection capabilities to encompass a wider range of hazardous scenarios, including climbing and potential airway obstructions, alongside enhancing the user interface for more intuitive caregiver interaction.

Elderly Fall Detection: Marshal et al.'s project leverages YOLOv5 to identify falls among the elderly through video analysis. It offers a significant upgrade over wearables by providing accurate, real-time detection. However, privacy concerns and environmental complexities pose challenges. Our project addresses these by incorporating additional sensing modalities and privacy-preserving technologies, aiming for a more adaptable and respectful solution integrated within existing healthcare frameworks.

The proposed solution to mitigate the risk of unsupervised children sustaining injuries at home involves the development of a specialized application, designed using Flutterflow and integrated with a Deep Learning model on a Raspberry PI 4G platform [7]. In essence, this application utilizes a picamera to detect the proximity of a child to potential danger zones, such as windows or doors, and subsequently alerts the user of the app when such conditions are detected.

The core of this system lies in the deployment of the advanced Yolov8 Deep Learning Model, renowned for its precision in object recognition and detection. This model has been meticulously trained using Robotflow to distinguish specific target items, achieving high accuracy in differentiating between a child and other objects within the home environment. Consequently, this system has demonstrated exceptional efficiency and accuracy in real-time object detection.

This innovative application provides parents with the capability to remotely monitor their children's safety, thereby bridging the gap in supervision. Additionally, the establishment of a data transfer mechanism between Raspberry PI and Flutterflow facilitates effective and instantaneous communication between these platforms.

Comparative analysis with previous iterations employing Yolov5n has revealed significant enhancements in performance with the adoption of Yolov8n, notably in terms of reduced response time and improved accuracy. This advancement underscores the superiority of the current method over previously explored alternatives, offering a more robust and reliable solution for ensuring the safety of children in unsupervised home settings.

In the experimental phase (Section 4), our project aimed to validate the effectiveness and reliability of our system across two distinct scenarios: window detection and door detection, each constituting a potential safety hazard for unsupervised children.

For window detection, we utilized a dataset of 9515 images, focusing on identifying various window structures. The experiment was structured to assess the model's accuracy in recognizing windows across different settings, employing a substantial portion of the dataset for training, validation, and testing. The significant finding was the model's high accuracy, indicative of its capability to discern and accurately identify window structures, thus demonstrating its potential to alert parents or guardians of a child's proximity to potential fall hazards.

The door detection scenario employed a dataset of 1000 images, aiming to recognize door structures accurately. Similar to the window detection experiment, this test also focused on the model's ability to detect and distinguish doors, crucial for preventing unsupervised exits or entries by children. The results showed commendable accuracy, although slightly lower than the window detection scenario, suggesting the model's effectiveness in door detection but also highlighting room for improvement in distinguishing between closely related objects.

The outcomes of both experiments underscored the effectiveness of our model in real-time hazard detection, significantly attributed to the robust dataset preparation and the utilization of YOLOv8 for object detection. However, the slight discrepancy in accuracy between window and door detection experiments suggests that while our system is highly capable, ongoing optimization and model refinement are necessary to enhance its precision and reliability in all potential hazard scenarios.

2. CHALLENGES

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

2.1. The Accuracy of the Detection

The detection is not very accurate from time to time from previous experimental projects, cause they employed less advanced detection model and resulted in this consequence. Thus, we change the model to a more powerful one: yolov8. Meanwhile, when we try to distinguish between doors and windows, the model was hard to tell the exact difference from the two similar objects. In further research studies, I focus on how to improve the quality of dataset and eventually figured out the method of combining the two datasets together. The raspberry PI was experiencing much more computational burden when I applied the heavier model on it, which resulted in lower performance. Thus, we deployed the nano version of YOLOv8 on it and significantly boost the chip’s performance.

2.2. Managing the Extensive Volume of Images

To address the challenge of managing the extensive volume of images and their associated URLs generated by our surveillance system, we considered implementing Binary Large Objects (BLOBs) within our database structure. This method would enable us to store image data directly in the database, offering a streamlined approach to data management by facilitating the direct insertion and retrieval of binary data.

After careful consideration, we decided to integrate this approach with Firebase, a cloud-based database platform known for its real-time data synchronization and efficient storage solutions. By uploading our images as BLOBs to Firebase, we successfully leveraged Firebase's scalable infrastructure to handle our project's data storage needs. This integration not only allowed us to maintain system performance despite the large data volume but also ensured quick and easy access to the stored images and their URLs.

Firebase's robust storage capabilities effectively managed the binary data of our images, while its database services provided a seamless mechanism for storing and retrieving the URLs associated with these images. This setup benefited our project by eliminating the complexities and overhead associated with external file storage systems, thus simplifying our data management process.

Furthermore, Firebase's security features and easy-to-use API facilitated a secure and straightforward implementation, ensuring that our images were stored safely and could be accessed efficiently whenever needed. The success of this approach validated our decision to use BLOBs for image storage in conjunction with Firebase, demonstrating an effective solution to the challenge of managing a large volume of surveillance images and their URLs within our project's database component.

2.3. The Model

Implementing the YOLOv8 Nano model on the Raspberry Pi presented several challenges, primarily due to the device's limited computational power and memory capacity. First, the Raspberry Pi struggled with the intensive demands of running deep learning models, which are resource-intensive and typically require powerful GPUs to operate efficiently. In addressing the Raspberry Pi's performance constraints, we rigorously tested various YOLOv8 models to identify one that balances accuracy with the required frame rate for real-time surveillance. The Raspberry Pi, with limited computational resources, necessitated a model that could operate efficiently without sacrificing performance. After evaluating the trade-offs between detection precision and processing speed, we opted for the YOLOv8 Nano model. This lightweight version is tailored for devices with restricted processing capabilities, offering an optimal mix of speed and accuracy. Implementing YOLOv8 Nano ensured our system could deliver real-time object detection while maintaining the Raspberry Pi's performance, striking a crucial balance essential for our project's success.

3. SOLUTION

There are 3 major components that connects the program together: 1. Target Detection and Data Capture System, 2. Data Storage and Transmission System and 3. User Interface and Interaction System. Their close cooperation aims at monitoring child safety near doors and windows. At its core, the Raspberry Pi and PiCamera capture images, utilizing YoloV8 and OpenCV for object detection to identify when a child approaches potential dangers [8]. This information is then transmitted to Firebase, acting as a central hub for data storage and management, enabling real-

time updates and secure data handling. Firebase's functionality extends to serving HTTP requests, facilitating seamless data flow and accessibility. The culmination of this process is realized in an app developed with FlutterFlow, which provides an intuitive interface for real-time monitoring and alerts. This design philosophy ensures that parents and caregivers can quickly familiarize themselves with the system, enabling them to effortlessly discern potential dangers and receive timely warnings if a child approaches a hazardous area. Emphasizing simplicity and ease of use, the app is engineered to facilitate immediate understanding and action, thereby enhancing the safety and well-being of children. To extend its accessibility and user-friendliness, the app also introduces a guest mode option. This feature is designed to serve customers who may need temporary access to the monitoring capabilities, ensuring that the system's benefits can be extended to a wider audience without compromising the security and privacy of the primary users. Leveraging Firebase for backend operations allows for efficient data synchronization and notification management, ensuring that users are promptly alerted to potential safety risks. This integrated approach combines sophisticated AI detection with robust data infrastructure and user-centric design to create a powerful tool for child safety [9].

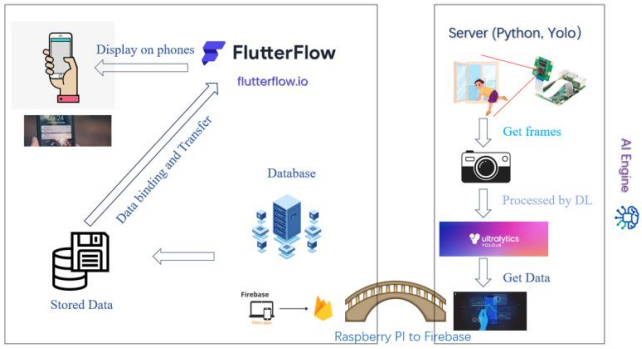

Figure 1. Overview of the solution

The Object Detection component utilizes Raspberry Pi and PiCamera, employing Neural Networks via YoloV8 for real-time image analysis to protect child from potential dangers. This AI-driven system processes visual data, identifying specific patterns indicative of children. It integrates with wider security monitoring systems and the results of the inspection can be seen in the application. After image processing we also use a simple algorithm: Bounding Box Collision Detection realizes whether a child is near the target object or not [10].

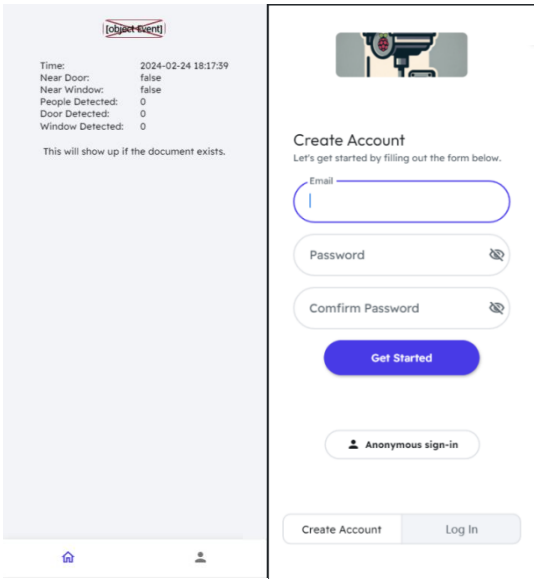

Figure 2. Screenshot of the app

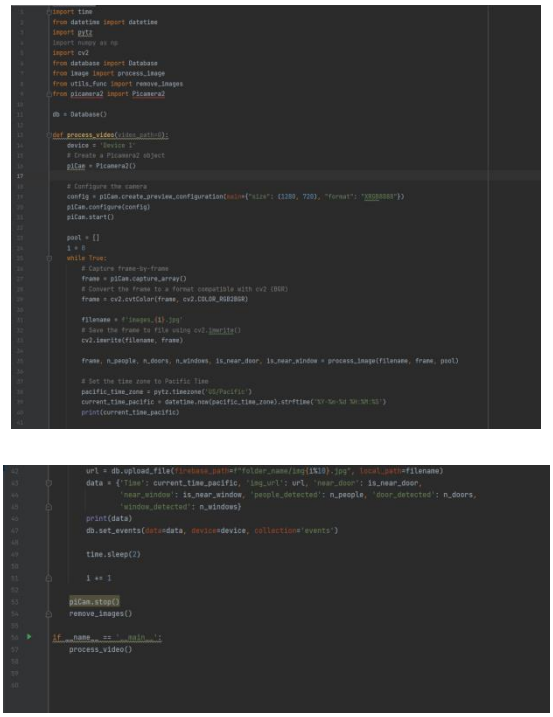

Figure 3. Screenshot of code 1

This Python script exemplifies a real-time surveillance system, operational via the Raspberry Pi and Picamera2, aimed at enhancing security and monitoring capabilities. Initiated typically at system startup or when activated, it continuously captures video frames, scrutinizing each for the presence of individuals, doors, and windows. Central to this are the next2door and next2window functions, assessing proximity of detected persons to critical entry points, while process_image meticulously consolidates these findings into a detailed analysis per frame. The process_video method embodies the system's essence, orchestrating the seamless capture, analysis, and data

logging flow, integrating sophisticated image processing with hardware operations and systematic data management.

Variables such as frame, capturing individual video snapshots, and filename, denoting each frame's storage name, alongside a data dictionary compiling analysis outcomes and metadata, facilitate the subsequent storage and retrieval processes. This setup presumes a server backend, likely Firebase, to harbor the processed data, managing uploads and event logging with precision, encapsulating detection insights, timestamps, and corresponding image references.

The operational sequence begins with the Picamera2's configuration, transitioning into a relentless capture loop. Post capture, frames undergo conversion to a compatible format, followed by an exhaustive object detection phase. Successive steps involve logging these detections, punctuated by image uploads to Firebase, ensuring data is methodically cataloged. Intermittent pauses allow for system recuperation and readiness for subsequent captures, culminating in a cleanup routine post-termination.

This system does not rely on advanced concepts like NLP or Neural Networks but is built upon the concept of real-time data synchronization and authentication. Real-time data synchronization ensures that the data displayed to the users is always current, without needing to manually refresh or query the database. Authentication is crucial for protecting privacy and ensuring that data access is controlled and secure.

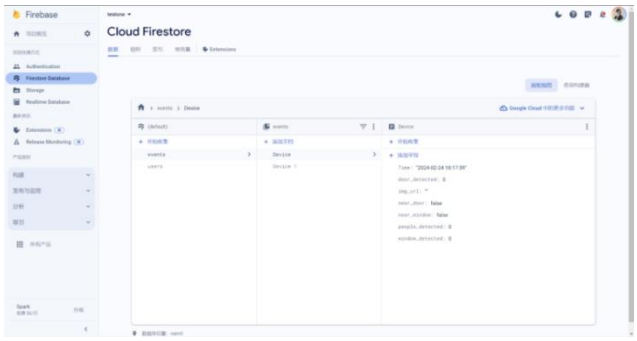

Figure 4. Screenshot of the firestore

```
import firebase_admin
from firebase_admin import credentials, storage, firestore

class Database:
    def __init__(self):
        # Replace the projectID eg.
        self.bucket_name = 'testone-b55de.appspot.com'
        # self.bucket_name = '<projectID>.appspot.com'

        # You need to download the serviceaccount.json
        self.fb_cred = 'service_account.json'
        cred = credentials.Certificate(self.fb_cred)
        firebase_admin.initialize_app(cred,
                                      {'storageBucket': self.bucket_name})
        self.db = firestore.client()  # this connects to our Firestore database

    def set_events(self, collection: str, device: str, data: dict):
        collection = self.db.collection(collection)  # opens collection
        doc = collection.document(device)  # specifies the  document
        doc.set(data, merge=True)

    def exists_on_cloud(self, filename):
        bucket = storage.bucket()
        blob = bucket.blob(filename)
        if blob.exists():
            return blob.public_url
        else:
            return Nothing

    def upload_file(self, firebase_path, local_path):
        bucket = storage.bucket()
        blob = bucket.blob(firebase_path)
        blob.upload_from_filename(local_path)
        print('This file is uploaded to cloud.')
        blob.make_public()
        url = blob.public_url
        return url
```

Figure 5. Screenshot of code 2

In this project, User Interface and Interaction System component doesn't rely on complex concepts like NLP or Neural Networks but emphasizes user experience (UX) design principles to ensure ease of use and efficiency [14]. Through FlutterFlow, a visually intuitive and interactive app interface is created, enabling users to view alerts, monitor situations, and interact with the system seamlessly. This component is critical for facilitating effective communication between the system and its users, ensuring that safety alerts are promptly delivered and understood.

4. EXPERIMENT

4.1. Experiment 1

The primary objective of this project is to enhance child safety by monitoring their movements when they are home alone. Consequently, the accuracy of the AI model is paramount. It must effectively detect, identify, and assess whether children are moving towards doors or windows.

The Window Detection dataset, accessible through Roboflow Universe, comprises a substantial collection of 9,515 images, specifically gathered for advancing window recognition technology in diverse building types. The images in this dataset are uniformly standardized to a resolution of 640x640 pixels, ensuring high-quality inputs for machine learning algorithms. Covering a broad array of window structures, the dataset includes various annotations that detail the presence of windows across multiple environments and building styles. Each image in the dataset has been enriched with annotations that detail the window structure, capturing the nuances of different designs and conditions such as lighting, shadows, and reflections. In our rigorous data preparation process, 8,337 images have been allocated to the training set, constituting 88% of the total dataset, to provide a robust training foundation. Additionally, 788 images (8% of the dataset) have been reserved for the validation set to tune model parameters, and 390 images (4% of the dataset) have been designated for the testing set to objectively assess model performance. This deliberate distribution ensures a comprehensive dataset that supports the development of highly accurate and generalizable window detection models.

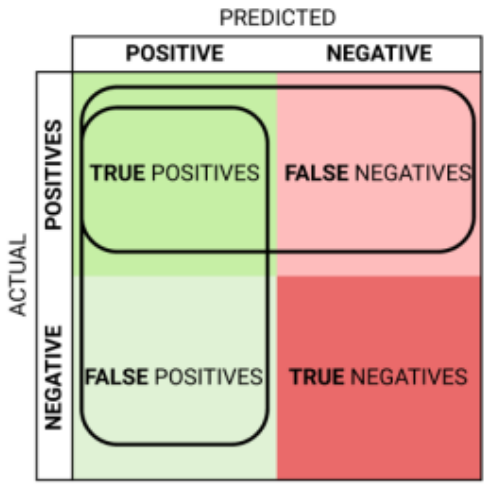

Figure 6. Predicted and actual

1) Precision

Precision refers to the probability of correctly predicting positive samples among the predicted positive samples. It is calculated using the following formula:

$$P = \frac{TP}{TP + FP} \tag{17}$$

View Source

The variable TP represents the number of true positive samples predicted as positive, and FP represents the number of false positive samples predicted as positive.

2) Recall

Recall represents the probability of correctly predicting positive samples among all the predicted samples. Its calculation is as follows:

$$R = \frac{TP}{TP + FN} \tag{18}$$

View Source

The variables FN represent the number of true positive samples predicted as negative.

3) mAP

mAP (mean Average Precision) is the mean value of precision for all detection categories. It is calculated using the following formula:

$$mAP = \frac{\sum_0^N AP_n}{N} \tag{19}$$

Figure 7. Functions

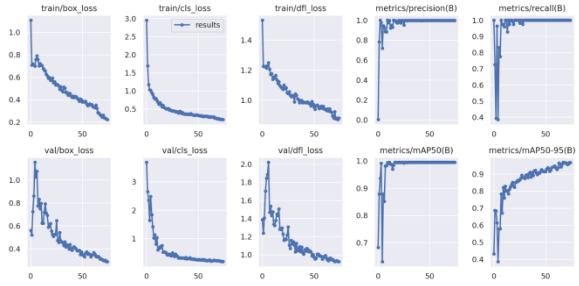

Figure 8. The trend

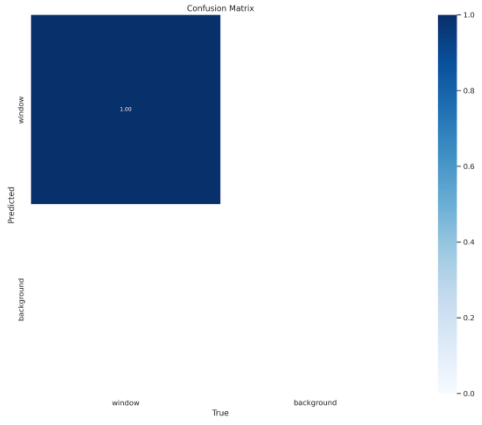

Figure 9. Confusion matrix 1

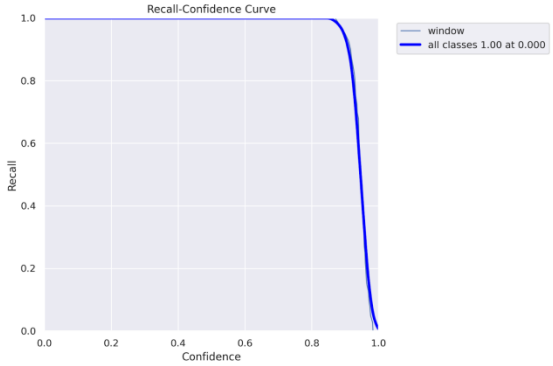

Figure 10. Recall-confidence curve

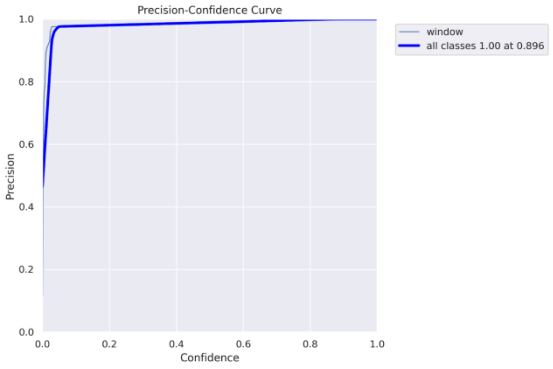

Figure 11. Precision-confidence curve

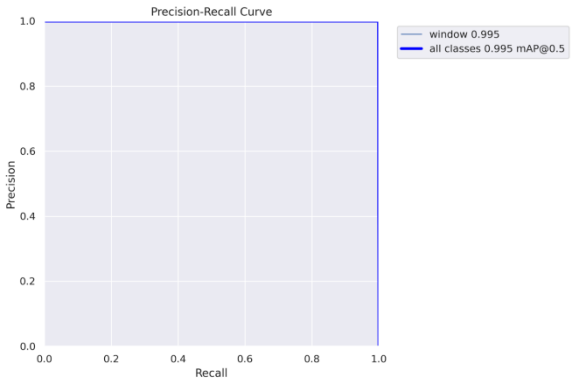

Figure 12. Precision-recall curve

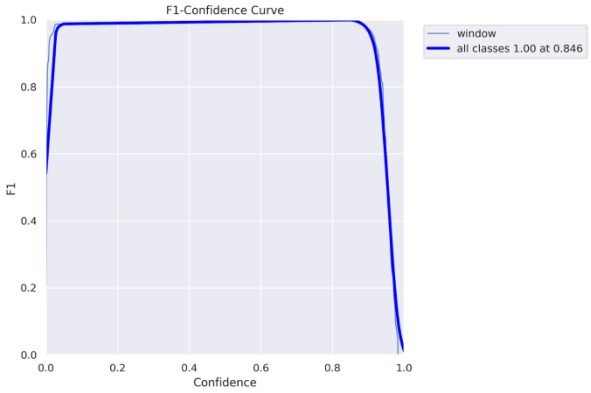

Figure 13. F1-confidence curve

The final results are shown as below:

Upon examination of the provided training and validation loss graphs, a discernible downward trend is observed, indicative of the model's learning efficacy. The training losses for box, class, and dfI demonstrate considerable decreases, suggesting a successful learning phase with the initial box loss decreasing from approximately 1.2 to below 0.2. Conversely, the validation losses, although improving, plateau at higher levels than their training counterparts, suggesting potential overfitting, a common challenge in machine learning models where a model performs well on training data but poorly on unseen data.

The precision and recall metrics for the training data predominantly exceed 0.6, indicating that the model maintains a commendable balance between identifying true positives (precision) and minimizing false negatives (recall). This balance is crucial for models aimed at object detection tasks, such as window detection in this context.

The mean Average Precision (mAP) scores, particularly mAP at Intersection over Union (IoU) of 0.5 (mAP50) and mAP at IoU ranging from 0.5 to 0.95 (mAP50-95), show substantial improvement throughout the training process. These metrics, essential for evaluating object detection models, reveal that the model's detection accuracy, particularly the overlap between the predicted and actual bounding boxes, has significantly improved over time.

The confusion matrix presents a flawless identification rate for the 'background' category, while the absence of data for the 'window' category in the matrix could indicate an imbalance in the validation dataset or an error in data presentation or model prediction capabilities.

The Recall-Confidence and Precision-Confidence Curves, ideally high across all confidence levels, depict notable declines at higher confidence thresholds. This drop may suggest overconfidence in the model's predictions, potentially leading to inaccuracies when the model is highly certain of its output.

The Precision-Recall Curve, exhibiting near-perfect precision across varying levels of recall, demonstrates the model's exceptional performance in maintaining high precision, a testament to its effective classification capability.

Lastly, the F1-Confidence Curve, achieving a peak F1 score around 0.846, indicates an optimal balance between precision and recall, underscoring the model's overall performance efficacy.

In conclusion, while the analysis reveals a model that is learning effectively and demonstrating commendable predictive performance, the signs of potential overfitting, as highlighted by the validation loss trends, necessitate further investigation. Addressing this could involve incorporating a more diverse validation dataset, applying regularization techniques, or adjusting the model's complexity. Additionally, the high performance at lower confidence levels juxtaposed with the declines at higher confidence thresholds suggests a need for recalibrating the model's confidence calibration mechanisms.

4.2. Experiment 2

The Door Detection dataset, hosted on Roboflow Universe, comprises 1,000 images showcasing a variety of door structures set against the backdrop of both urban and rural environments. These images are captured in high definition, each with a resolution of 480x640 pixels, to ensure the clarity and detail necessary for effective machine learning model training. The dataset is rich in diversity, featuring doors of different designs, sizes, and styles, which are common in residential buildings, office blocks, and public institutions. On average, each image is annotated with multiple instances of doors, highlighting the dataset's focus on detailed object detection within architectural contexts. The annotations detail the doors' bounding boxes, capturing variations in orientation, size, and appearance due to differing lighting conditions, perspectives, and environmental settings.

For practical application in deep learning projects, the dataset has been divided into training, validation, and testing sets. A total of 700 images have been allocated for the training set to provide a comprehensive learning foundation. The validation set consists of 200 images, designed to fine-tune model parameters and prevent overfitting. The remaining 100 images serve as the testing set, offering an unbiased evaluation of the model's performance in recognizing and classifying door structures under varied conditions. This structured dataset distribution supports a rigorous machine learning workflow, from model training through to evaluation, ensuring the development of robust door detection algorithms.

Loss Graphs (Training and Validation)

Box Loss, Class Loss, and dfI Loss: For both training and validation, the loss graphs show a decreasing trend. The training box loss starts near 1.2 and ends below 0.2, indicating significant learning. However, the validation box loss starts around 3.5 and plateaus around 1.0, which might suggest the model is not generalizing as well as we'd like. Similar trends are observed in class and dfI loss graphs, though the starting and ending values differ. This disparity between training and validation loss might indicate overfitting.

Precision and Recall (Training)

Precision and Recall: The precision graph is fluctuating but remains mostly above 0.6, which is relatively high. The recall graph shows a similar pattern, with most values above 0.6. This suggests the model is capable of identifying relevant instances (true positives) while keeping false positives low.

mAP Graphs (Training)

mAP at IoU=0.5 (mAP50): The graph shows improvement over time, starting from around 0.2 and reaching upwards of 0.8. This indicates the model is getting better at detecting objects with a good overlap (50% Intersection over Union).

mAP at IoU=0.5 to 0.95 (mAP50-95): This mAP graph shows a steady increase, from near 0.1 to about 0.5. Since this metric is more stringent, the lower score compared to mAP50 is expected. The consistent upward trend is a positive sign.

Confusion Matrix

The confusion matrix shows a high true positive rate for 'door' (0.86) and a perfect true negative rate for 'background' (1.00). However, there's a notable false negative rate for 'door' (0.14), which means the model sometimes misses detecting doors.

Recall-Confidence and Precision-Confidence Curves

These curves are ideal when they start high and remain high as confidence increases, which is mostly what we see. However, the sharp drop at high confidence levels suggests the model has few predictions with very high confidence, and when it does, it is not always correct.

Precision-Recall Curve

The Precision-Recall curve shows a precision level above 0.8 for most recall levels, which is excellent. However, the steps in the graph might indicate a small dataset or that the model has issues with certain instances.

F1-Confidence Curve

The F1 score combines precision and recall into a single metric. The curve shows a peak F1 score of around 0.78 at a confidence threshold of approximately 0.83, which is quite good. The F1 score is a more comprehensive metric than precision or recall alone and suggests the model is performing well on balancing the two.

The graphs show a model that learns and improves over time. The precision and recall are high, indicating the model is both relevant and precise in its predictions. The mAP scores show capability in detecting objects with various degrees of overlap. However, the validation loss plateau and the perfect segments in some curves could indicate potential overfitting or evaluation issues, which may attribute to the repetitive and similar pictures.

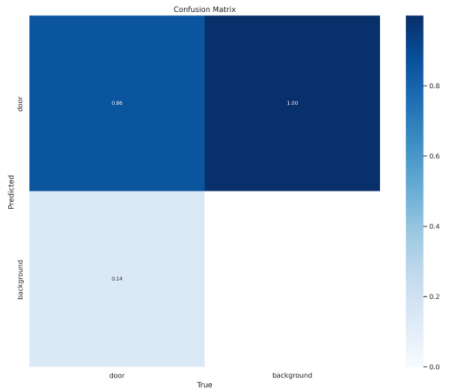

Figure 14. Figure of experiment 2

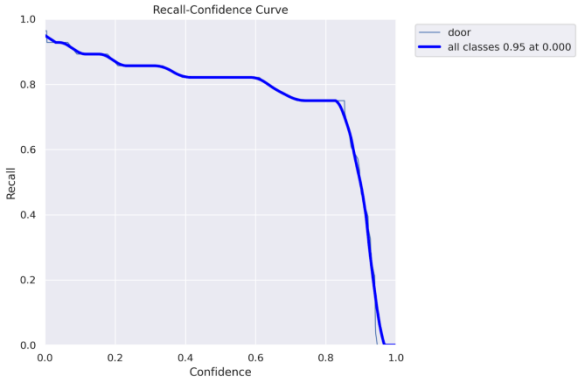

Figure 15. Recall-confidence curve

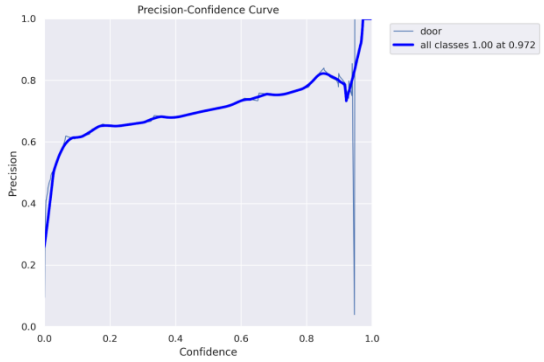

Figure 16. Precision-confidence curve

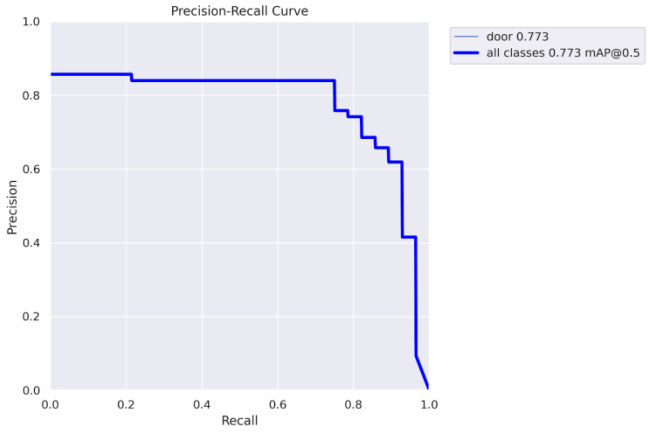

Figure 17. Precision-recall curve

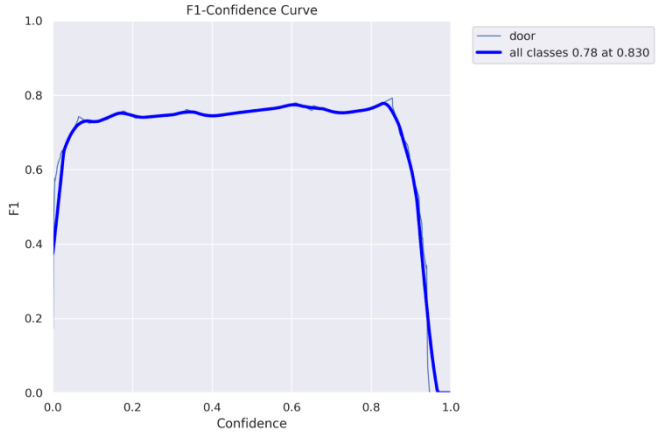

Figure 18. F1-confidence curve

The final results are obtained from testset:

Model	Window.pt	Door.pt
Accuracy	94.57%	86.48%

Figure 19. Model and accuracy

5. RELATED WORK

The scholarly source titled "Development of a real-time pen-holding gesture recognition system based on improved YOLOv8" by Wenjiao Qu et al. addresses the important issue of identifying correct pen-holding gestures among school-aged children [11]. This research is significant due to the correlation between pen-holding gestures and the visual acuity and skeletal development of children, a topic often overlooked by educators and parents. Utilizing computer vision (CV) technology, specifically the YOLOv8 model, the study introduces a novel approach by incorporating a pose estimation method with enhanced granularity and adding a status branch to each key point detection branch for improved semantic information utilization.

The solution proposed achieves an accuracy of 99.50%, demonstrating its effectiveness over advanced models like VIT and DETR by margins of 5.16% and 1.20%, respectively. This high level of accuracy is indicative of the model's ability to discern fine-grained differences in pen-holding gestures, which is critical given the subtlety of these gestures. The study also contributes a corresponding real-time recognition system designed to facilitate handwriting practice, highlighting the practical application of the research in educational settings.

However, the method has its limitations, notably its current inability to be deployed on mobile terminals, which restricts its utility in more dynamic classroom settings. The research suggests future directions focusing on model deployment to mobile devices to enhance accessibility and applicability.

Comparatively, the project I am working on aims to build upon this foundation by addressing some of the identified limitations. While Qu et al.'s method shows remarkable precision in recognizing pen-holding gestures, our project seeks to extend the functionality of such systems to more portable and accessible platforms, like smartphones and tablets, thereby increasing the

system's versatility and utility in real-world educational environments. Moreover, our approach may explore integrating feedback mechanisms within the app, offering users (teachers, parents, and students) immediate and actionable guidance on correcting pen-holding gestures, thus not only identifying but also rectifying improper techniques in real-time. This aspect of interactivity and user engagement could significantly enhance the learning experience and outcome, ensuring more widespread and effective use of the technology in improving pen-holding gestures among school-aged children.

The scholarly source titled "Enhancing Child Safety: Computer Vision-Based Accident Detection for Infants and Toddlers" by Jia He Tan and Ching Pang Goh presents a novel system aimed at monitoring infants and toddlers to prevent common household accidents [12]. Utilizing closed-circuit television (CCTV) cameras and machine learning algorithms, specifically the YOLOv8 model, the system is designed to detect the presence of dangerous objects such as scissors and knives and calculate the distance between these objects and children. The main objective is to alert caregivers when children come into proximity with potential hazards, thereby preventing accidents like cuts and burns. The system demonstrated satisfactory detection accuracies of 87% for scissors, 84% for knives, and a slightly reduced accuracy of 75% when both items were present simultaneously.

The effectiveness of this solution is grounded in its ability to provide real-time monitoring of children, reducing the likelihood of accidents and offering peace of mind to parents. However, the system has notable limitations, including its reliance on a 2D view for detection, which may miss instances where children are obscured by objects, and its inability to detect risks associated with climbing or covering the face with soft objects. These limitations suggest areas for potential improvement in future iterations of the system.

In comparison to this approach, our project also utilizes computer vision and deep learning technologies but seeks to extend the capabilities of the system to address some of the limitations identified. Specifically, our project aims to enhance the system's ability to detect a broader range of dangerous scenarios, including climbing and situations where a child's airway may be obstructed. To achieve this, our project aims to improve the system's user interface to make it more intuitive for parents and caregivers, providing detailed alerts and actionable advice in real-time. This could include suggestions for mitigating identified risks, such as moving dangerous objects out of reach or securing furniture to prevent climbing accidents.

By addressing these areas, our project aims to build upon the foundation laid by Tan and Goh's research, offering a more comprehensive and user-friendly solution for enhancing child safety in the home. Through continued development and refinement, we hope to contribute to the reduction of accidents among infants and toddlers, ensuring a safer environment for their exploration and growth.

The scholarly source titled "An Image-based Fall Detection System for the Elderly using YOLOv5" by S. Marshal et al. introduces an innovative solution aimed at enhancing the safety of the elderly by detecting falls through image processing [13]. Leveraging the YOLOv5 deep learning model, the system analyzes video feeds in real-time to identify instances where an elderly person may have fallen. This approach signifies a considerable advancement over traditional methods, which often rely on wearable devices or sensors and may suffer from issues such as false alarms, poor accuracy, and limited adaptability to various environments.

The effectiveness of this solution is underscored by its capacity to provide real-time, accurate fall detection, thereby enabling swift intervention and potentially reducing the severity of fall-related injuries. Nevertheless, the system does encounter challenges, including dependency on

continuous video surveillance, which raises privacy concerns, and the potential for decreased accuracy in complex or cluttered environments.

Comparatively, our project utilizes similar computer vision and deep learning technologies but aims to enhance the capabilities of the fall detection system further. Specifically, we intend to address the limitations identified in Marshal et al.'s approach by incorporating additional sensors and refining the deep learning model to improve detection accuracy in a wider range of settings, including those with poor lighting or occlusions. Additionally, we plan to develop a privacy-preserving mechanism that can anonymize individuals in video feeds, thus mitigating privacy concerns while maintaining the system's efficacy.

Moreover, our project seeks to improve the user interface and integrate the system more seamlessly into the existing healthcare infrastructure. This includes providing caregivers and medical personnel with actionable insights and alerts through an intuitive interface, enabling more effective monitoring and response strategies.

In essence, by building upon the foundation established by Marshal et al., our project endeavors to create a more comprehensive, accurate, and user-friendly fall detection system. This system aims not only to mitigate the immediate risks associated with falls among the elderly but also to integrate more closely with broader healthcare and support services, ultimately contributing to improved outcomes and enhanced quality of life for the elderly population.

6. CONCLUSIONS

Our project, while demonstrating high accuracy in detecting potential hazards to children, encounters limitations such as suboptimal frames per second (FPS) rates and occasional misdetections [15]. These issues are crucial for real-time monitoring and require refinement. To address these, optimizing the inner layers of the YOLOv8 model could enhance computational efficiency and maintain high accuracy, potentially through model pruning or quantization techniques that lighten the model without significant loss of performance. Additionally, transitioning from FlutterFlow to native Flutter development could offer more advanced functionalities and customization options, allowing for a more robust and versatile application. Given additional time, these improvements would be prioritized to refine the system’s responsiveness and accuracy, ensuring a more reliable and efficient monitoring solution.

This project represents a significant step forward in leveraging advanced deep learning technologies to enhance child safety within the home. By addressing the limitations and implementing suggested improvements, the potential of this system to prevent accidents and provide peace of mind to parents and guardians is immense. With continued development, this solution has the promise to set a new standard in home safety monitoring.

REFERENCES

[1] Kendrick, Denise, et al. "Home safety education and provision of safety equipment for injury prevention." Evidence-based child health: a Cochrane review journal 8.3 (2013): 761-939.

[2] Serre, Thomas. "Deep learning: the good, the bad, and the ugly." Annual review of vision science 5.1 (2019): 399-426.

[3] Xie, Wu, Feihong Feng, and Huimin Zhang. "A Detection Algorithm for Citrus Huanglongbing Disease Based on an Improved YOLOv8n." Sensors 24.14 (2024): 4448.

[4] Mello, Débora Falleiros de, et al. "Child safety from the perspective of essential needs." Revista latino-americana de enfermagem 22 (2014): 604-610.

[5] Zhang, Jian, et al. "SmartWriting: Pen-holding gesture recognition with smartwatch." ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, 2019.

[6] Ghailan, Khalid, et al. "Children domestic accidents profile in Jazan Region, a call for new policies to improve safety of home environment." Saudi Journal of Biological Sciences 28.2 (2021): 1380-1382.

[7] Garcia-Barrientos, Abel, et al. "Design and implementation of a car's black box system using a raspberry pi and a 4g module." Applied Sciences 12.11 (2022): 5730.

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

[9] Shi, Wenzhong, et al. "Change detection based on artificial intelligence: State-of-the-art and challenges." Remote Sensing 12.10 (2020): 1688.

[10] Chao, Wang, et al. "Improved hybrid bounding box collision detection algorithm." Journal of System Simulation 30.11 (2019): 4236-4243.

[11] Qu, Wenjiao, et al. "Development of a real-time pen-holding gesture recognition system based on improved YOLOv8." 2023 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML). IEEE, 2023.

[12] Tan, Jia He, and Ching Pang Goh. "Enhancing Child Safety: Computer Vision-Based Accident Detection for Infants and Toddlers." 2024 3rd International Conference on Digital Transformation and Applications (ICDXA). IEEE, 2024.

[13] Silva, Shehan, et al. "Effectiveness of Deep Learning Technologies to Assist Educational Inclusion for Children with Down Syndrome." 2023 5th International Conference on Advancements in Computing (ICAC). IEEE, 2023.

[14] Alshemali, Basemah, and Jugal Kalita. "Improving the reliability of deep neural networks in NLP: A review." Knowledge-Based Systems 191 (2020): 105210.

[15] Faccio, Daniele, and Andreas Velten. "A trillion frames per second: the techniques and applications of light-in-flight photography." Reports on Progress in Physics 81.10 (2018): 105901.

©2024 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.