Ensuring Child Safety: An IoT-Based Surveillance System for Remote Monitoring and Detection of Anomalous Behavior

Khawaja Rabeet Tanveer*, Muhammad Shaheer Luqman†, and Anam Qureshi ‡
*Department of Computer Science, NUCES FAST Karachi
Email: k213374@nu.edu.pk
†Department of Computer Science, NUCES FAST Karachi
Email: k214655@nu.edu.pk
‡Department of Computer Science, NUCES FAST Karachi
Email: anam.qureshi@nu.edu.pk

***Abstract*—Amidst the economic pressures compelling many families to juggle multiple jobs, the ability to maintain consistent parental care for children diminishes. Yet, it remains essential that families feel assured their children are safe in their absence. Addressing these concerns, our research proposes an Internet of Things (IoT) surveillance system with real-time video and audio monitoring capabilities. This system detects anomalies such as crying, movement beyond camera range, and potential signs of child abuse, notifying connected devices with alerts tailored to the severity of the situation. A robust Artificial Intelligence model has been configured to map out the poses of the caretaker and child to detect any signs of abuse and stress. Parents and other concerned adults can monitor their children remotely using our system.**

***Index Terms*—Internet of Things (IoT) Computer vision (CV) Artificial intelligence (AI) Image processing Object detection Pose recognition Edge computing**

I. INTRODUCTION

In today's economic landscape, where the necessity for dual-income households often displaces the traditional caregiving roles within families, ensuring the safety and well-being of children becomes a paramount concern. Research indicates a significant global issue, with approximately 900,000 child deaths annually attributed to negligence, underscoring the critical need for effective monitoring solutions (WHO) [1]. This backdrop emphasizes the urgency for robust childcare surveillance systems that can reliably monitor and respond to potential risks in real time.

Current surveillance systems, although widely used, often lack the advanced features needed to fully address concerns about childcare safety. This gap in research highlights the need for more effective monitoring solutions [2]. Our proposed solution aims to fill this gap by using Internet of Things (IoT) technology and artificial intelligence (AI) to create an advanced childcare surveillance system. This system will have real-time video and audio monitoring, allowing parents to monitor their children remotely from anywhere. Additionally, it will use advanced AI algorithms [3] to detect unusual actions like crying, leaving the camera's view, or signs of potential abuse or distress. By using state-of-the-art AI models, the system can accurately identify and respond to critical incidents, giving parents timely alerts and notifications based on the seriousness of the situation.

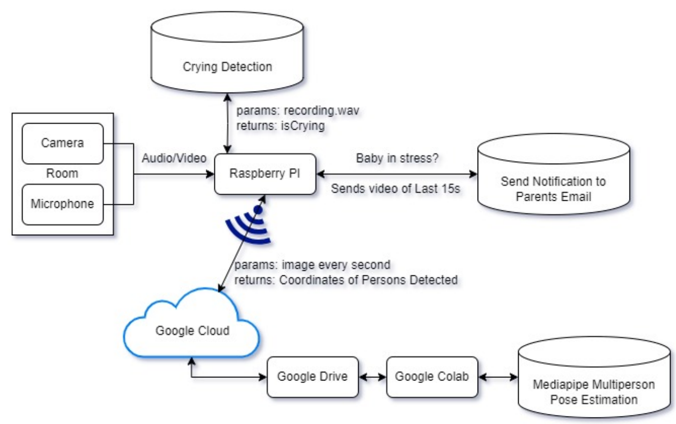

Fig. 1. Proposed architecture of child safety surveillance system

The scope of our project encompasses the design, development, and implementation of the IoT based childcare surveillance system described above. This includes the selection and integration of hardware components such as cameras, microphones, and a Raspberry Pi 4.0 device, as well as the development of software components for real-time monitoring, a pose detection system based on Mediapipe Pose Detection Model [16] to detect physical abuse, and a Tensorlite crying detection model[13] integrated with a notification/alert systems. The detailed design of the proposed solution is presented in figure 1. While our focus is primarily on childcare monitoring, the underlying technology and methodologies employed in our system may have broader applications in other surveillance and monitoring domains.

II. FUNCTIONAL AND NON FUNCTIONAL REQUIREMENTS

A. Functional Requirements

- Real-Time Surveillance: The IOT-Based Child Surveillance System (ICSS) should provide real-time video

and audio surveillance capabilities to enable parents and caregivers to monitor their children remotely.
- Detection of Anomalies: The system should be capable of detecting anomalies such as shouting and crying, indicating potential distress or emergencies.
- Detection of Physical Abuse: The ICSS should employ Pose Detection to detect signs of physical abuse, including aggressive behavior between the caregiver and the child.
- Line of Sight: The system should be able to detect if the caretaker or child leaves the field of view of the cameras, providing alerts to the parent or caregiver.

B. Non-Functional Requirements

- Security Confirmations: The ICSS should implement robust security measures to ensure the confidentiality, integrity, and availability of data. This includes user authentication, encryption of data transmissions, and secure storage of sensitive information from bad actors.
- Performance Requirements: The system should demonstrate high performance in terms of real-time monitoring and detection capabilities. It should be able to process video and audio streams efficiently, with minimal latency, to provide timely alerts to users.

III. Frameworks and Technologies Used

A. TensorFlow

It is an open-source machine learning framework developed by Google. It provides a comprehensive ecosystem of tools, libraries, and resources for building and deploying machine learning models. With its flexible architecture and extensive documentation, TensorFlow is widely used for various applications, including computer vision, natural language processing, and reinforcement learning. By leveraging TensorFlow's high-level APIs and pre-trained models, developers can accelerate the development process and achieve state-of-the-art performance in their machine-learning applications. Furthermore, TensorFlow's compatibility with a wide range of hardware platforms, including CPUs, GPUs, and TPUs, makes it suitable for both research and production environments [4].

B. MediaPipe

MediaPipe is an open-source framework developed by Google for building cross-platform real-time multimedia processing pipelines. It offers a range of pre-built components and tools for tasks such as object detection, pose estimation, and hand tracking. MediaPipe's modular design and efficient implementation make it suitable for a wide range of multimedia applications, including augmented reality, gesture recognition, and video analytics. With MediaPipe, developers can easily integrate advanced computer vision and machine learning algorithms into their applications, enabling innovative user experiences and interactions [5].

C. Cloud-based Storage and Computing

Cloud computing means shared computing resources (networks, servers, applications) are delivered as a service [8] [9] [10] over the Internet from cloud to customer. According to [12], Cloud is an interconnected network of servers providing services for people or businesses. Cloud supports real-time operation, processing, analyzing, connecting, managing, and securing IoT devices as well as applications [11]. Platforms such as Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure offer a range of services, including object storage, databases, compute instances, and machine learning services. By leveraging cloud-based resources, organizations can reduce infrastructure costs, improve scalability, and accelerate time-to-market for their applications. Additionally, cloud-based services offer advanced features such as automatic scaling, data replication, and built-in security controls, enabling developers to focus on building and deploying their applications without worrying about infrastructure management.

D. Raspberry Pi 4B

Raspberry Pi 4B is a low-cost, credit-card-sized single-board computer developed by the Raspberry Pi Foundation. It is the latest model in the Raspberry Pi series, offering improved performance, enhanced connectivity, and expanded capabilities compared to previous models. With its quad-core ARM Cortex-A72 processor, up to 8GB of RAM, and support for dual 4K displays, Raspberry Pi 4B is well-suited for a wide range of applications, including IoT projects, home media centers, and desktop computing. Additionally, Raspberry Pi 4B features a variety of ports, including USB 3.0, Gigabit Ethernet, HDMI, and GPIO pins, making it versatile and easy to integrate with external devices and peripherals. Whether used for educational purposes, hobbyist projects, or commercial applications, Raspberry Pi 4B offers a cost-effective and flexible solution for building innovative and interactive systems [6].

IV. Implementation of Modules

A. Crying Detection

To mitigate false alarms caused by loud background noise, microphone white noise, and other environmental anomalies, our crying detection model operates on a looping mechanism set to every n seconds, where n is presently configured to 5. The system verifies the presence of crying over two complete loops before triggering a notification alert to inform parents that stress has been detected by the monitor. This module is processed at the edge devices.

B. Abuse Detection

Our approach to implementing physical abuse detection relies on pose detection facilitated by the MediaPipe library. Each segment of the video feed is divided into 30 frames. Within each frame, the hand coordinates of the caretaker the larger of the two detected poses—are scrutinized to determine if they breach the bounding box around the child, represented by the smaller pose. In such instances, the system calculates the distance between the two poses to estimate the speed of the adult hand's movement. Additionally, the system checks if the child has started crying using the crying detection model after these conditions are met. Physical abuse is reported only when all three criteria—breaching the bounding box, movement speed of the adult hand, and crying detection—are simultaneously flagged. This method depends on adapting the MediaPipe Pose Detection system to support Multi Pose Detection capabilities

C. Line of Sight

Using line-of-sight detection, our system generates an image every second from real-time video frames. This image undergoes processing by the Pose Detection Model to identify two distinct poses within it. These pose parameters are defined relative to height, distinguishing between an adult's larger pose and a child's smaller pose, based on a specified threshold value, denoted as n. If either pose is not detected, the system triggers an alert to notify parents that either the child or caretaker is not visible.

D. Communication Protocols and Raspberry Pi 4b

In our project, the Raspberry Pi 4B utilizes IPV4 (Internet Protocol Version 4) for data transmission. IPV4 is chosen for its seamless compatibility with existing network infrastructures and services, simplifying deployment and management. Its straightforward implementation supports real-time data transmission and low latency communication, critical for timely surveillance operations.

V. Experimental Results

This research project employs two deep learning models: one designed for the detection of a child's cry and another tailored for multi-person pose estimation

A. Crying Detection

The deep learning model used in this study was selected from Hai Trieu's Public GitHub Repository [13]. To rigorously assess its performance and establish measures, we employed the Infant Cry Sounds Dataset sourced from Kaggle [14]. This dataset comprises a diverse array of audio samples, encompassing crying, laughter, silence, and ambient noise, reflecting real world conditions.

Following extensive evaluation, the model achieved the following performance metrics:

- Accuracy: 0.94
- Precision: 1.00
- Recall: 0.74
- F1 Score: 0.85

Predicted Outcomes

Actual Outcome	T	F	
	216	0	F′
	18	50	T′

B. Media-Pipe Pose Detection

For pose estimation, we utilized Google's Mediapipe Pose Estimation Solution [15]. In particular, we used the Lite Model which is capable of running on a Raspberry Pi. The landmark model in MediaPipe Pose predicts the location of 33 pose landmarks as shown in figure 2.

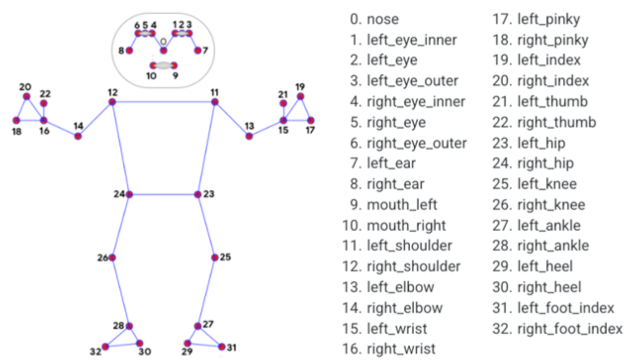

Fig. 2. Pose landmarks as predicted by Mediapipe Pose

The BlazePose GHUM Lite model's quality is evaluated using three validation datasets—Yoga, Dance, and high intensity interval training(HIIT)—each containing images of a single person distance at 2-4 meters from the camera. For consistency with other solutions, the evaluation criteria focuses on 17 keypoints from the common objects in context(COCO) topology. This approach ensures that the model's performance is comparable to other well-performing publicly available solutions across different verticals [16].

Percentage of Correct Key-points (PCK)@0.2 is a metric used to evaluate pose estimation models. It considers a detected joint correct if the distance between the predicted and true joint positions is less than 20% of the torso diameter. This metric is applied to keypoints such as the head, shoulders, elbows, wrists, hips, knees, and ankles. PCK@0.2 is used for both 2D and 3D evaluations, with higher values indicating better model accuracy in predicting joint positions as shown in figure 3 [17].

VI. Challenges

A. Lack of Processing Power

A significant hurdle in our project was synchronizing video after introducing Multi-Pose Detection in MediaPipe. Medi-

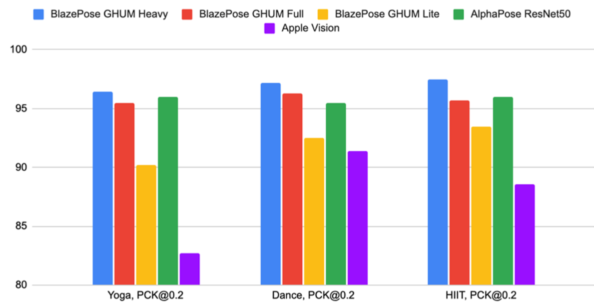

Fig. 3. Accuracy of models predicting joint positions [16]

aPipe's single-pose detection runs smoothly on a CPU in real-time with few issues. However, the multi-pose detection isn't as efficient. It struggled to synchronize the video properly on CPUs, which was our first major challenge. MediaPipe's GPU setup also posed challenges. Even on devices like the Raspberry Pi, the GPU wasn't powerful enough to handle the workload effectively. We found that MediaPipe's GPU optimization wasn't up to par based on our research and community feedback. Switching to cloud computing wasn't feasible due to limitations on processing power in free-tier services. As a last resort, we implemented a solution using a Google Cloud API to send data to Google Drive, where it could be accessed and processed using Google Colab. While this solved the synchronization issue, the overall processing time remained longer than desired for efficient project execution.

B. Video-on-Demand (VOD) Review

In response to challenges associated with physical abuse detection, the VOD review mechanism offers an alternative approach for incident analysis and review. When the system detects signs of distress, it automatically captures and sends a video snippet of the preceding moments to the parent's email address. This enables parents to retrospectively review the footage and investigate the circumstances leading to the child's distress, providing valuable insights into potential incidents of concern. This was expected to work however limitations of resources have caused us to get stuck at a point where we can either record the screen for a VOD review or provide a live feed.

C. Reason Behind Action

With the pose detection for physical abuse being a dead end, we settled for a system of VOD(video on demand) Review. Whenever the baby is detected as having been crying, the last 15 or 20 seconds, depending on the value of a set variable 'n', are emailed to the parent along with the notification that is sent to them alerting them of the child's stress. By doing so, the parents themselves can review the footage and decipher the events that took place that led to the child's crying. This solution bypasses the need for high processing and is a lower resource-demanding method at the cost of the parents reviewing the footage manually. From a purely rational standpoint, the parents want to know what caused their child to cry in general, so it works on that end as well.

VII. Conclusion and Future Work

In concluding this study, we have successfully developed and implemented a crying detection model that minimizes false alarms through a robust looping technique. Our system includes functionalities for Line-of-Sight Detection, real-time feed transmission, and automated notification and video generation. While these achievements represent significant progress in infant cry and stress surveillance (ICSS), the challenge of real-time physical abuse detection persists.

Moving forward, there are several avenues for future exploration. One critical area for improvement is the implementation of real-time physical abuse detection using pose detection libraries such as OpenPose or MoveNet, which offer potential optimizations to streamline data processing without introducing delays [7]. Additionally, enhancing user engagement and response times could be achieved through the development of a mobile application that delivers notifications directly to users' smartphones, thus improving the accessibility and usability of the surveillance system.

References

[1] World Health Organization, "Child safety and unintentional injuries," 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/child-safety-and-unintentional-injuries. [Accessed: Jun. 19, 2024].

[2] C. J. Moerman and A. Lier, "Smart Surveillance Systems for Child Safety: Current Limitations and Future Directions," *Int. J. Child-Comput. Interact.*, vol. 26, pp. 100232, 2020. doi: 10.1016/j.ijcci.2020.100232.

[3] R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 1-36, 2019. doi: 10.1145/3281487.

[4] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, Savannah, GA, USA, 2016, pp. 265-283. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf.

[5] A. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," in *arXiv preprint arXiv:1906.08172*, 2019. Available: https://arxiv.org/abs/1906.08172.

[6] E. Upton, "Raspberry Pi 4 Model B Datasheet," Raspberry Pi Trading Ltd., Cambridge, UK, 2019. Available: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf.

[7] Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., Sheikh, Y. (2019). OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(1), 172-186. Available: https://ieeexplore.ieee.org/document/8897194.

[8] Mohsin Nazir. Cloud Computing: Overview & Current Research Challenges. IOSR Journal of Computer Engineering. 8(1). 2012. pp.14-22. Available: https://www.iosrjournals.org/iosr-jce/papers/Vol8-issue1/I0811422.pdf.

[9] B D, Parameshachari Rachana, C. (2017). CLOUD COMPUTING: A RESEARCH PERSPECTIVE ON THE SECURITY ISSUES. Available: https://www.semanticscholar.org/paper/Cloud-Computing%3A-A-Research-Perspective-on-the-B-D-Rachana/8eb8e79b553ee1e5eebaf2e39d9396c3f13a0e5f.

[10] Rachana, C.R., Banu, R., Ahammed, G.A. and Parameshachari, B.D., 2017, August. Cloud Computing–A Unified Approach for Surveillance Issues. In IOP Conference Series: Materials Science and Engineering (Vol. 225, No. 1, p. 012073).IOP Publishing. Available: https://iopscience.iop.org/article/10.1088/1757-899X/225/1/012073.

[11] Arun K Mani1, M.Gokilavani, Shreevani D, Samra Said and Unnikrishnan K N. A Review: IoT And Cloud Computing For Future Internet. International Research Journal of Engineering and Technology (IRJET). 6(5). 2019. pp.1098-1102. Available: https://irjet.net/archives/V6/i5/IRJET-V6I5241.pdf.

[12] Chamandeep Kaur. The Cloud Computing and Internet of Things (IoT). International Journal of Scientific Research in Science, Engineering and Technology. 7(1). 2020. pp.19-22. DOI: https://doi.org/10.32628/IJSRSET196657.

[13] H. Trieu, "PBL5_BabyCryingDetection_raspberry," GitHub. Available: https://github.com/haitrieu123580/PBL5_BabyCryingDetection_raspberry.

[14] R. Jahangir, "Infant Cry Sounds," Kaggle. Available: https://www.kaggle.com/datasets/raiyanjahangir1939/infant-cry-sounds.

[15] Google AI, "MediaPipe Pose Landmarker," Google AI Dev. Available: https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker.

[16] Google AI, "MediaPipe Pose Landmarker Documentation," GitHub. Available: https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/pose.md.

[17] C. Sudux, "Human Pose Estimation 101 - Percentage of Correct Key Points (PCK)," GitHub. Available: https://github.com/cbsudux/Human-Pose-Estimation-101?tab=readme-ov-file#percentage-of-correct-key-points---pck.