International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

Design and Implementation of Smart Baby Feeding System: A Fusion of Image Processing and IoT Technologies.

Enjamuri Baktha Singh[1], Dr. G. Thirupathi *[2], Pothe Deepak [3], Tankari Jyothi [4]

Submitted:12/03/2024 **Revised**: 27/04/2024 **Accepted**: 04/05/2024

Abstract: A cradle is an environment that gives comfort, protection to the babies. When parents are busy working with the household things, they put their babies in the cradle. When babies are in the cradle, the parents should always check them time to time and feed them when they are hungry. There are existing systems that gives the baby good feeling by swinging the cradle, and some plays music to make the baby happy, and some are equipped with lots of sensor which are transmitted to a mobile app where parents can monitor them in live. In this research, we intend to design and develop an Internet of Things (IoT) and machine learning based system integrated with cradle which feeds the baby as a response to the baby's cry using a robotic arm. It uses Raspberry pi combined with Arduino microcontroller to perform the operation. This system continuously watches the baby and saves his mouth location. When the baby cries, the system sends the direction signals to control the robotic arm, which is holding sipper filled with milk, to put sipper in the baby's mouth. To send direction signals to the arm, we developed an algorithm which works based on the locations of the mouth and the sipper. The system discards the locations other than mouth, which can be danger to the baby, ensuring safety and certainty. In optimal conditions (such as light, sound), our system successfully detected the mouth and gave directions to the arm with 98% accuracy to move the sipper to the mouth. In future, this system can be modified in such a way that the arm can be controlled remotely with mobile devices to feed the baby.

Keywords: *Arduino microcontroller, Artificial Intelligence, Deep-Learning, Image Processing, Internet of Things, Smart Cradle.*

1. Introduction

When a baby is in a cradle, monitoring him is an important task for a parent. The parent should check his status time to time. When baby cries, the parent may be sometimes busy working household things and cannot control the circumstances of the baby remotely. Hunger for food is one of the major reasons why a baby cry. Sometimes the parent is busy working on household things by keeping him in a cradle. When baby cries for hunger the parent must stop their work and feed him.

In the existing literature, the baby is continuously monitored while in the cradle [1]. Whenever there is an occurrence, such as urination or the baby awakening from sleep, an alert is immediately sent to the parent's device. It also creates rocking motion to the cradle automatically via geared motor. Rocking motion of the cradle as a response to the crying of the baby [2]. The sensors are activated when baby cries and signal is sent to controller and a rocking motion is created via motor. Low-cost IoT-based system for monitoring in real time is created [3]. An affordable IoT-based system for real-time monitoring has been developed. The team created a novel algorithm that is crucial for enhancing baby care when parents are not present. This system uses sensors to track important parameters like optimum temperature, moisture, and also crying. Additionally, parents can monitor their babies though external web camera remotely. The IoT-based cradle system enhances safety, comfort, and also convenience for both infants and caregivers. This advanced cradle is outfitted with various sensors, including those for temperature, humidity, motion, and sound, which continuously track the baby's surroundings. These sensors gather data in real-time and transmit it to a cloud server for processing and analysis. Caregivers can access this information via a mobile app or web interface, enabling them to monitor the baby's condition remotely and receive alerts if any irregularities are detected. Additionally, the Smart Cradle features automated rocking, adjustable incline, and soothing lullabies, all of which can be controlled remotely through the mobile app. In all the existing systems mentioned before, there is a lack of feature i.e., feeding the baby without human interaction.

Our proposed system is completely a new feature and focuses on feeding the baby without human interaction. The system is triggered by the baby's cry sound and immediately acts to feed the baby using a robotic arm equipped with a milk sipper. The system spots the baby's mouth and moves the sipper into the mouth. If the baby still cries it alerts the parent through a sound. This system makes use of Computer Vision using the existing system python frameworks i.e., MediaPipe [15]. This framework uses Convolutional Neural Networks (CNN) [17] to implement face detection. The system uses Raspberry Pi and Arduino as the main controllers for computer vision and arm control. The Raspberry pi gets the location of the mouth using the web

[1] SNIST, Dept. of CSE, Hyderabad, Telangana-501301, INDIA.
[2] SNIST, Dept. of CSE Hyderabad, Telangana-501301, INDIA.
ORCID ID: 0000-0002-9081-2816
[3] SNIST, Dept. of CSE Hyderabad, Telangana-501301, INDIA.
[4]SNIST, Dept. of CSE Hyderabad, Telangana-501301, INDIA.
** Corresponding Author Email: thirupathi.g@email.com*

cameras associated with the cradle and motor signals are sent to the Arduino through serial communication. The Arduino in turn controls the arm to feed the baby. The system feeds the baby up to a variable time, and if baby still cries it alerts the parent through sound and if the baby is satisfied after feeding then system will get back to the normal position. The overall cost of making the arm physically is around 30,000/-

In future, the system can be integrated with all existing cradle systems to create a hybrid smart cradle. The parent can have full control over the smart cradle from the smallest of monitoring the cradle to the largest of controlling the circumstances remotely even on the robotic arm so parent can choose whether to the control the arm manually or automatically to feed the baby. For the integration of this device with the existing system doesn't cost anything, but it is matter of work. But for the features which are going to be added in future the estimated cost is around 50,000 to 70,000/-.

2. Literature survey

The system setup allows the cradle to seamlessly integrate with a smartphone, typically an Android device. An Arduino microcontroller is utilized to assemble all the required sensors and hardware components. Continuous monitoring of the baby inside the cradle will be conducted [5]. If any activity such as urination or the waking up from sleep happens, a notification through an SMS will be sent to the device of the parents. The Smart cradle have additional features such as automatically rocking the baby using the motor mechanism.

The system helps to supervise and soothe the infant. This is an automatic cradle which operates based on sensors. When the baby cries, the sensors will be triggered. The primary purpose of this cradle is to comfort a crying baby and lull it to sleep without requiring the presence of parents [9]. This mechanism soothes the baby's cries and induces sleep. It is structured in a manner that it functions alongside a mobile application, enabling parents to activate and deactivate the cradle and also override its automated functions. Additionally, this system empowers parents to keep track of their infants remotely.

A cost-effective and streamlined IoT-based system for real-time monitoring. They additionally devised a novel algorithm for the system, pivotal in delivering superior baby care during the absence of parents [1, 3]. They also created a new algorithm for the system that plays a key role in providing better baby care while parents are away. In this system, Node Micro-Controller Unit (NodeMCU) Controller Board is exploited to gather the data read by the sensors and uploaded via Wi-Fi to the AdaFruit MQTT Server. The setup utilizes sensors to track the infant's crucial parameters, including ambient temperature, moisture, and crying. The system's structure comprises a baby cradle that automatically swings using a motor in response to the baby's cries [10]. Additionally, parents can oversee their infants' well-being through an external webcam and remotely activate the lullaby toy positioned on the baby cradle via the MQTT server to soothe the baby.

An IoT-based offers security, ease, and practicality for both infants and caregivers. This setup is furnished with a range of sensors, including temperature, humidity, motion, and sound sensors, which consistently monitor the baby's surroundings. These sensors gather data in real-time and transmit it to a cloud-based server for processing and analysis [11]. Caregivers can retrieve this data through a mobile application or a web interface, enabling them to remotely monitor the baby's status and environment and receive alerts in the event of any issues [10]. Additionally, this system features automated rocking motion [13][14], adjustable incline, and soothing lullabies, all of which can be controlled remotely [16][17]. Moreover, the system can generate insights and recommendations. Caregivers have the flexibility to customize the settings according to their needs and preferences.

3. Proposed Work

The proposed work is completely new introduction and an additional feature that can be added to all existing smart cradles. Whatever the machine which is going to be built is completely starts with its core idea. In that meanwhile we selected the components by researching and analyzing the existing software systems and cradle designs that could help our work. We found various components and software programs used in the different smart cradle systems that range from the smallest feature of swinging to the largest of live video streaming of the baby. In that process the hardware and software we used for our work are given in the appendix section.

We divided the explanation of the proposed work into four sections. In the first section we will explain how the system architecture is, and how it is constructed. In second section, control flow of the system is explained, which shows the flow of the system execution. In the third section, the algorithm, which is developed to direct the arm to the mouth location and it's time complexity is explained. Then in fourth section, we have added few research questions that our work is going to address is mentioned.

3.1. System Architecture

The architecture of this system consists of various devices which work independently of each other yet work towards single goal. The Fig.1. Shows the system architecture of the smart baby feeding. The camera and Microphone, which are the one of the major components, provides inputs i.e., video stream and audio stream to the system to work. The Raspberry Pi, which is the heart of the system, processes the audio and video and sends the direction signals to Arduino microcontroller to move the arm. Initially, the video is processed using Convolutional Neural Networks (CNN) and the locations of the mouth and sipper is retrieved with the help of OpenCV. An algorithm developed, which is explained in the third section, which takes the locations of the mouth and the sipper as input from the OpenCV, to send the direction signals to arm. But the arm cannot take these signals directly, so Arduino is introduced here as a mediator between arm and signals sent by raspberry pi. The arm is equipped with Ramps 1.4 which understands the Arduino instructions and converts them into motion of the motors.

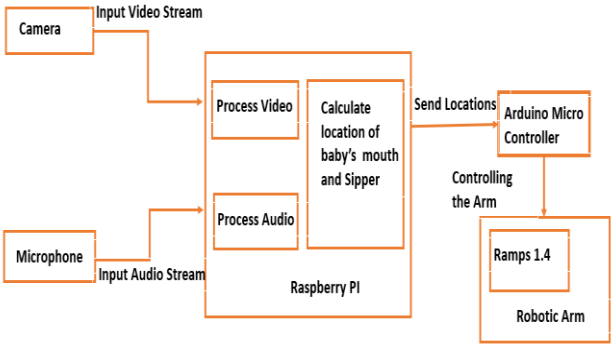


Fig. 1. System Architecture of Smart Baby Feeding

The mechanical arm is the major component which feeds the baby. We created a simple 3 axis robotic arm where end of the arm holds the sipper. The stepper motors are used at these 3 joints to create the motion and the movements of these motors receive signals though their drivers. The connections of these motors with their drivers requires lots of wiring which leads to complexity. So, in this area, Ramps 1.4 board is introduced. This is a 3d printer controller board allows to connect stepper motors and drivers without or nearly less wiring. Every of these 3 joints of the arm has separate motor and has independent motion. So, to make the arm work as we desired, we need to coordinate the motors to move the sipper to the right place. How we are going to do that? This is the area where the Arduino mega comes in.

The Arduino is programmable microcontroller board that can be integrated into a variety of electronic projects. It is programmed in a way that it coordinates the motors to reach the baby's mouth. Since stepper motors doesn't know where they are, the area where they move are defined through simple signal. The Arduino sends the signals to the Ramps which in turn sends motion signals to the motors. There is no need of any extra knowledge to connect the Ramps 1.4 to the Arduino mega, we simply need to mount the male headers of the Ramps board with the female headers of the Arduino.

Now the major part, we need to make sure that the arm is moving in such a way that it reaching the baby's mouth and feeding him correctly. For that need we need to recognize the baby's mouth and the sipper tip and keep the sipper tip in the mouth, and also it should not move the sipper to deep that might choke him and leads to death and sipper should not move into the nose that might block his breath. So how we are going to handle it?

Computer vision [16], a field of artificial intelligence (AI) that allows computers to understand and identify objects and people in videos and images. Building a program that can recognize the baby's mouth is an expensive task and most advanced one. But a framework is already existed for facial recognition i.e., MediaPipe [15], is an open-source framework for creating pipelines to conduct computer vision inference on diverse sensory data like video or audio.. We used two webcams as X and Y axis to the sides of the cradle and with the help of this framework the baby's mouth and nose are detected. And a TensorFlow Lite model of the sipper is developed to recognize the sipper tip by gathering various images of the sipper. We keep track of lips outline and form a coordinate that is the centroid of the lips and send the signals to the arm to move the sipper to the centroid of the mouth. And by chance of any baby's movement if it going in wrong direction other than lips, then it immediately waits a moment and again moves the arm in the right direction.

This overall recognition process is done in a mini machine, the Raspberry Pi, which comprises a series of small single-board computers (SBCs) that are the size of a credit card. We run the MediaPipe frame work in this computer using the python language.

The Arm is controlled by the Arduino, but the recognition and tracking of the mouth is done in the Raspberry Pi. How we are going to control the arm to move it to the baby's mouth? How we are going to tell the Arduino to move the arm to the exact location?

Serial Communication between Raspberry Pi and the Arduino is the bridge connecting the location of the baby's mouth and the Arm motion. The stream commands which has what motor should act at what time are sent from the Raspberry Pi to the Arduino. Then the Arduino receives the command and moves the respective motor in the commanded direction.

The activation of the arm to feed the baby can be done through various factors. We can activate the arm, when baby is sad (by emotion detection) or when baby is crying (sound detection). Here we are going to introduce sound detection. Mediapipe also has the facility to classify sound. It has a pre-existing model that can classify various sounds. It includes baby's cry as one of the categories. So, the arm is going to respond to when the baby is crying. So, when the baby cries, the arm immediately responds and feeds the baby.

The system observes the continuous stream of sound coming from the external environment. If it finds the baby's cry sound, then it immediately responds and moves the sipper to baby's mouth. It feeds the baby up to a variable time. When this variable time is reached, the arm reset to the initial position. If the baby is cried during this feeding time or if the baby is not stopping his cry, then it alerts the parent through sound. In the next section, we are going to explain how the control flow is for this system.

3.2. Control Flow of the System

The Fig.2. Shows the control flow of the proposed work. The system initially starts by checking whether all functions working or not. If everything works fine, then it continues its execution, if not it alerts the user about the issue. By ensuring that everything is fine, it sets the arm to initial position and then starts capturing video and audio simultaneously. The system continuously watches the baby and tracks the mouth location. When baby cries, the understands that it is the time to feed the baby. But cry can happen in two scenarios, first the baby is crying in the cradle, second, the baby is crying outside of the cradle. So, the system only continues if the baby is inside the cradle. If the baby is crying and not in the cradle, then it alerts the user that baby is crying somewhere. If the baby is in the cradle, it continues the execution to feed the baby. The baby is fed only up to a variable time set by the user, and after the feeding time it alerts the user that baby's feeding is completed. During the feeding time, if baby still cries, we can't say why he is crying, so it alerts the user. After all operations has been done, the arm moves back to the initial position. The system runs in an infinite loop until the user shuts it down. Now let us talk about the important thing in the next section, i.e., Algorithm.

3.3. Working of the Algorithm

The distance is nothing but how far two objects are, and the two points to consider in our system are baby's mouth and the sipper tip. The arm has 3 joints since it is a 3-axis robotic arm. The algorithm aims to align the x, y, z axes of the sipper tip with the baby's mouth, especially when the sipper is positioned too far above the baby's mouth, the signal is sent to arm to move it in a way that the sipper should come near to the mouth. When the distance between the mouth and sipper becomes zero i.e., into the baby's mouth, then this phase is called as feeding. After a variable time-interval the arm comes back to the initial position.

Algorithm: Step by step instructions:

Step 1. Initialization:

a. Initialize the robotic arm with three motors: Motor1 for z- axis rotation, Motor2 for y-axis rotation of the main base, and Motor3 for y-axis rotation of the elbow joint.

b. Set the initial position of the ArmTip, which represents the tip of the robotic arm.

Step 2. Target Setup: Define the TargetPoint as the position of an object on the ground that the robotic arm needs to approach.

Step 3. Movement Calculation:

a. Continuously monitor the positions of ArmTip and TargetPoint.

b. Calculate the differences in position along each axis: dx (horizontal), dy (vertical), and dz (depth).

Step 4. Movement Decision:

a. Determine which axis has the largest difference:

b. If dz (depth) is the largest difference: Send a signal to move Motor2 to adjust the z-axis rotation, aiming to minimize the distance in the depth direction.

c. Else, if dx (horizontal) is larger than dy (vertical): Send a signal to move Motor1 to adjust the y-axis rotation of the main base, aiming to minimize the distance in the horizontal direction.

d. Otherwise: Send a signal to move Motor3 to adjust the y-axis rotation of the elbow joint, aiming to minimize the distance in the vertical direction.

Step 5. Iteration: Repeat steps 3-4 until the ArmTip is within a predefined threshold distance from the TargetPoint or until a predetermined maximum number of iterations is achieved

Step 6. Completion: Once the ArmTip is sufficiently close to the TargetPoint or once the maximum number of iterations is reached, the algorithm will be terminated.

The above step by step instructions of an algorithm is converted into pseudo code.

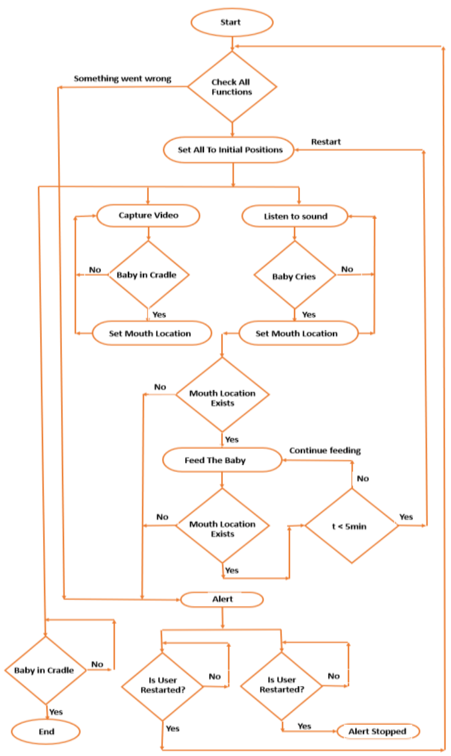


Fig.2. Control flow of the proposed work.

Pseudo Code:

1. **Initialize** the robotic arm with three motors: Motor1 for z-axis rotation, Motor2 for y-axis rotation of the main base, and Motor3 for y-axis rotation of the elbow joint.

2. **Set** the ArmTip position as the starting point of the robotic arm.

3. **Define** the TargetPoint as an object's position on the ground.

4. **Loop:**

a. Measure the current position of ArmTip and TargetPoint.

b. Calculate the differences in position along each axis

(dx, dy, dz).

c. Determine which axis has the largest difference:

if dz > dx and dz > dy:

then Send a signal to move Motor2 to adjust the z-axis rotation.

else if dx > dy:

then Send a signal to move Motor1 to adjust the y-axis rotation of the main base.

else:

Send a signal to move Motor3 to adjust the y-axis rotation of the elbow joint.

d. **Repeat** the loop **until** the ArmTip approaches the TargetPoint within a specified threshold distance.

The above pseudo-code outlines the steps of the algorithm in a more abstract, language-agnostic way. It describes the loop that continuously measures the position of the ArmTip and TargetPoint, calculates the differences along each axis, and adjusts the robotic arm's motors accordingly until the ArmTip reaches the TargetPoint within a specified threshold distance.

Time Complexity Analysis:

The time complexity of the provided pseudo-code depends on several factors:

1. The number of iterations of the loop: The loop continues until the ArmTip approaches the TargetPoint within a specified threshold distance. The number of iterations depends on how quickly the arm converges to the target point.

2. The complexity of measuring the position of ArmTip and TargetPoint: This involves some computational overhead, but it's typically constant or linear in relation to the number of dimensions being measured.

3. The complexity of calculating differences along each axis (dx, dy, dz): This step involves simple arithmetic operations and is constant time.

4. The complexity of moving the motors: This depends on the efficiency of the motor control algorithm and the hardware being used. In practice, it's typically constant time or linear in relation to the number of motors.

Overall, the time complexity of the provided pseudo-code is dominated by the number of iterations of the loop, as it determines how many times the position measurements and motor adjustments are performed. Therefore, we can denote the time complexity as O(k), where k is the number of iterations required for the arm to converge to the target point. However, the exact value of k depends on factors such as the initial arm position, the target point, and the threshold distance.

3.4. Research Questions:

Below are some of the research questions that we will address in this work.

RQ1: What if the arm moves the sipper into the nose or ears?

RQ2: How accurately the arm can move towards mouth location?

4. Results

Fig.3. Shows the 3-axis arm is built to feeding the baby. The arm is designed manually and used 3d printing for physical transition. Stepper motors and timing belts are attached for motion. Ramps 1.4 is the main commutator between the motors and the Arduino. The maximum speed of each stepper motor is 3000 rpm. We set the speed of each stepper motor to 600 rpm for smooth operation.

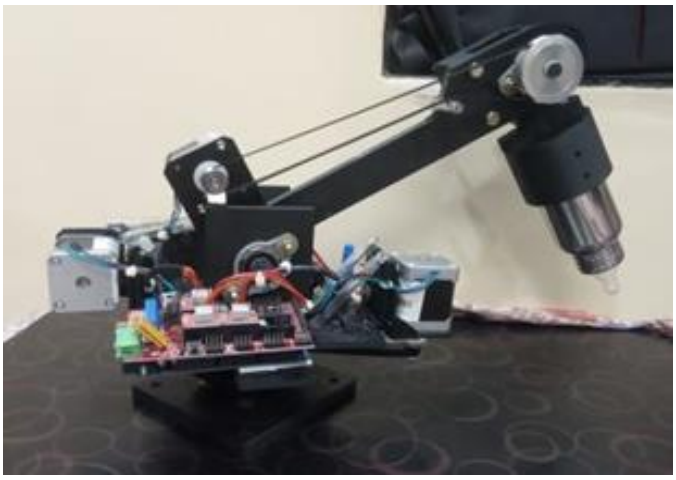


Fig.3. 3-Axis robotic arm to feeding the baby

Our proposed system initializes smoothly, seamlessly transitioning into operation with its initial settings configured that shown in Fig.4. This marks the commencement of the system's functionality, poised to execute tasks according to its predetermined parameters and specifications.

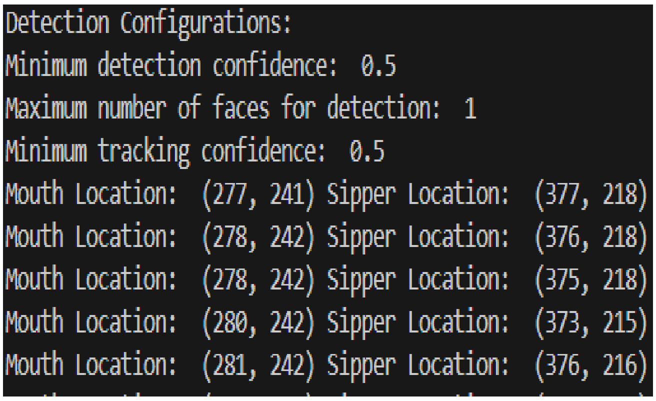


Fig.4. System Initialization

The system adeptly identifies both the mouth and danger locations. "DL," designated as Danger Location, encompasses vital areas such as the eyes and nose, crucial for assessing potential risks or hazards. Conversely, "SL" denotes the Sipping Location, pinpointing where feeding occurs.

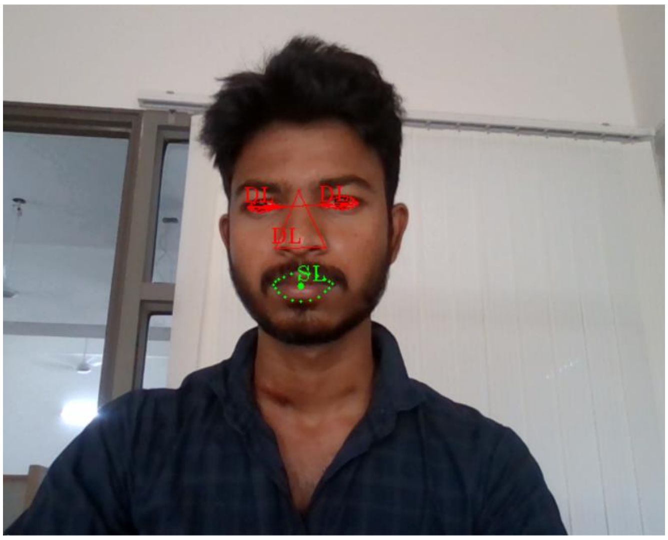


Fig.5. System Identifies Different Areas Recognition

RQ1. Answer: The system successfully identified the danger locations, which are represented as DL in the Fig.5. These are the places which the system will not direct the arm. With the results, our proposed system able to provide safety-ness of the baby.

When system is started to move the sipper to the mouth location, it started tracking the mouth and the sipper coordinates which is shown in Fig.6. Green line is an indicator of coordinator and index finger considered as a sipper. Based on the x and y axis coordinates and the distance between the mouth and the sipper, the system sent signals for the motion of the sipper. When we placed the finger in front of the camera, our fingertip is the sipper, and the system tends to give directions towards the mouth location. The distance between the finger and the mouth is represented by a line in the mentioned images.

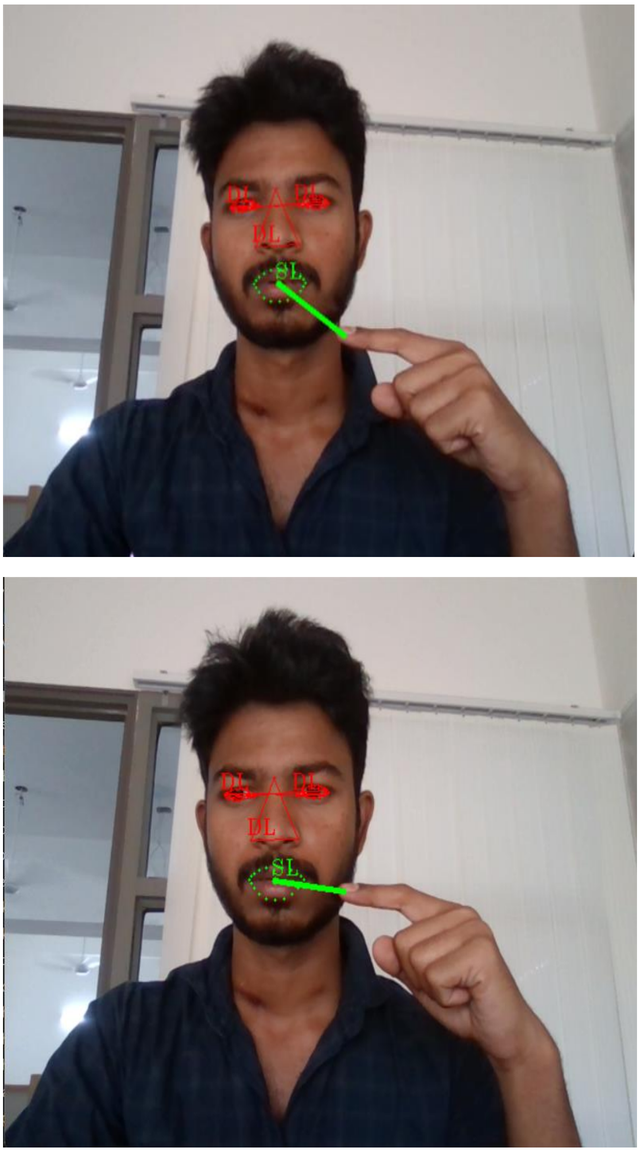
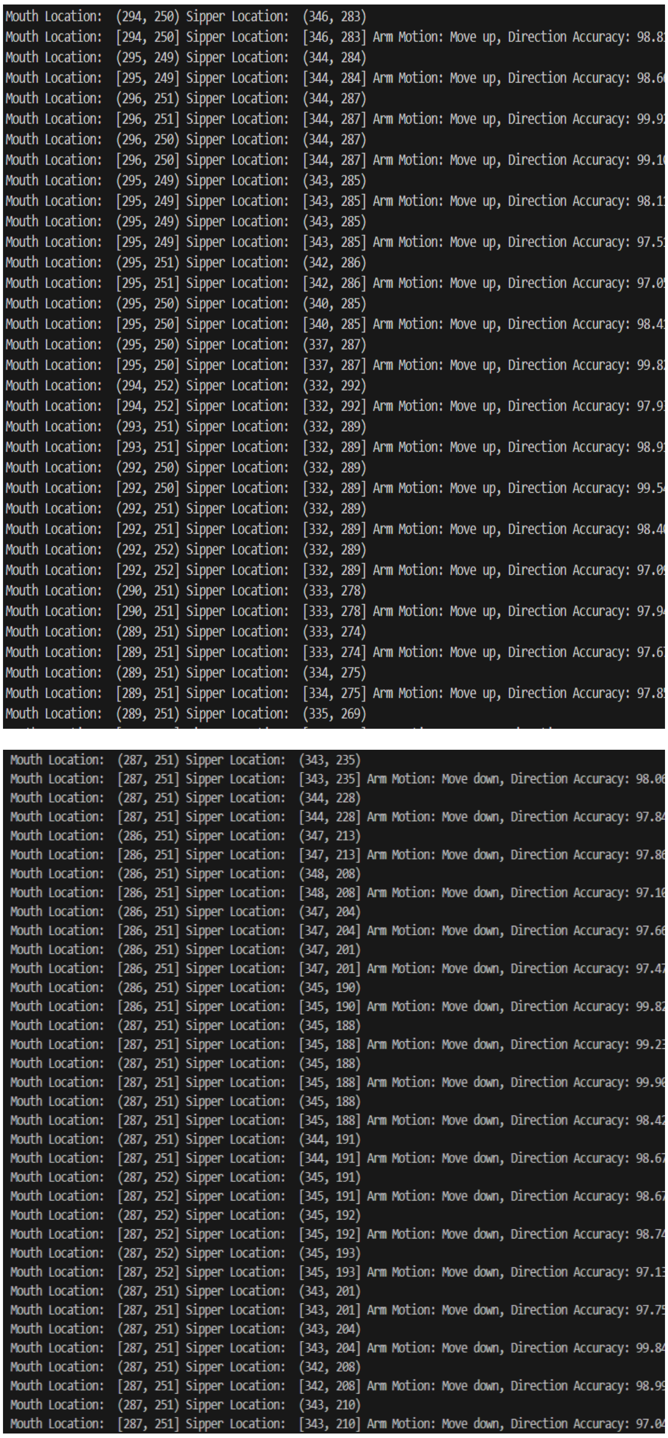


Fig. 6. Showing coordinates with Distance between Sipping and Mouth.

The Fig. 7. shows the directions to which the fingertip should move in order to reach the mouth point. It shows the mouth location and sipper location coordinates with respect the screen and along with the direction to which the finger should move to reach the arm. It also shows how accurately the arm is moving towards the mouth.



```
Mouth Location:  (294, 250) Sipper Location:  (346, 283)
Mouth Location:  [294, 250] Sipper Location:  [346, 283] Arm Motion: Move up, Direction Accuracy: 98.8
Mouth Location:  (295, 249) Sipper Location:  (344, 284)
Mouth Location:  [295, 249] Sipper Location:  [344, 284] Arm Motion: Move up, Direction Accuracy: 98.6
Mouth Location:  (296, 251) Sipper Location:  (344, 287)
Mouth Location:  [296, 251] Sipper Location:  [344, 287] Arm Motion: Move up, Direction Accuracy: 99.9
Mouth Location:  (296, 250) Sipper Location:  (344, 287)
Mouth Location:  [296, 250] Sipper Location:  [344, 287] Arm Motion: Move up, Direction Accuracy: 99.1
Mouth Location:  (295, 249) Sipper Location:  (343, 285)
Mouth Location:  [295, 249] Sipper Location:  [343, 285] Arm Motion: Move up, Direction Accuracy: 98.1
Mouth Location:  (295, 249) Sipper Location:  (343, 285)
Mouth Location:  [295, 249] Sipper Location:  [343, 285] Arm Motion: Move up, Direction Accuracy: 97.5
Mouth Location:  (295, 251) Sipper Location:  (342, 286)
Mouth Location:  [295, 251] Sipper Location:  [342, 286] Arm Motion: Move up, Direction Accuracy: 97.0
Mouth Location:  (295, 250) Sipper Location:  (340, 285)
Mouth Location:  [295, 250] Sipper Location:  [340, 285] Arm Motion: Move up, Direction Accuracy: 98.4
Mouth Location:  (295, 250) Sipper Location:  (337, 287)
Mouth Location:  [295, 250] Sipper Location:  [337, 287] Arm Motion: Move up, Direction Accuracy: 99.8
Mouth Location:  (294, 252) Sipper Location:  (332, 292)
Mouth Location:  [294, 252] Sipper Location:  [332, 292] Arm Motion: Move up, Direction Accuracy: 97.9
Mouth Location:  (293, 251) Sipper Location:  (332, 289)
Mouth Location:  [293, 251] Sipper Location:  [332, 289] Arm Motion: Move up, Direction Accuracy: 98.9
Mouth Location:  (292, 250) Sipper Location:  (332, 289)
Mouth Location:  [292, 250] Sipper Location:  [332, 289] Arm Motion: Move up, Direction Accuracy: 99.5
Mouth Location:  (292, 251) Sipper Location:  (332, 289)
Mouth Location:  [292, 251] Sipper Location:  [332, 289] Arm Motion: Move up, Direction Accuracy: 98.4
Mouth Location:  (292, 252) Sipper Location:  (332, 289)
Mouth Location:  [292, 252] Sipper Location:  [332, 289] Arm Motion: Move up, Direction Accuracy: 97.0
Mouth Location:  (290, 251) Sipper Location:  (333, 278)
Mouth Location:  [290, 251] Sipper Location:  [333, 278] Arm Motion: Move up, Direction Accuracy: 97.9
Mouth Location:  (289, 251) Sipper Location:  (333, 274)
Mouth Location:  [289, 251] Sipper Location:  [333, 274] Arm Motion: Move up, Direction Accuracy: 97.6
Mouth Location:  (289, 251) Sipper Location:  (334, 275)
Mouth Location:  [289, 251] Sipper Location:  [334, 275] Arm Motion: Move up, Direction Accuracy: 97.8
Mouth Location:  (289, 251) Sipper Location:  (335, 269)
```

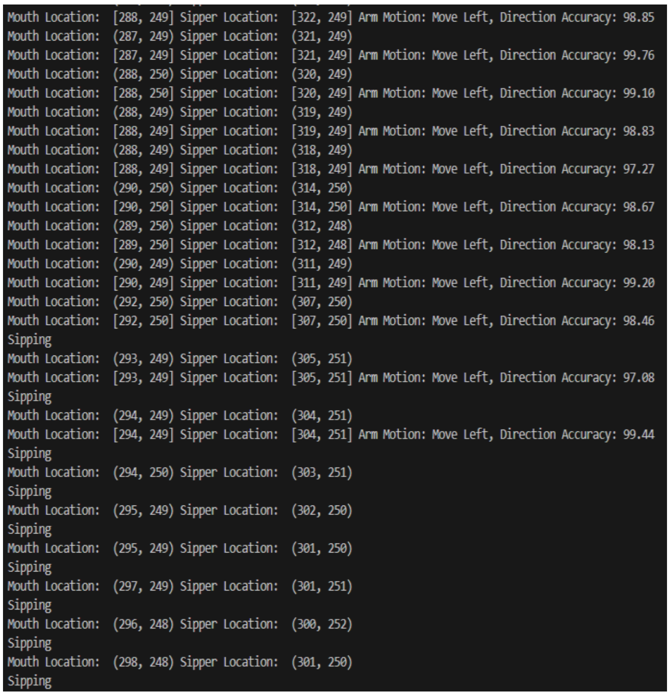
```
Mouth Location:  (287, 251) Sipper Location:  (343, 235)
Mouth Location:  [287, 251] Sipper Location:  [343, 235] Arm Motion: Move down, Direction Accuracy: 98.0
Mouth Location:  (287, 251) Sipper Location:  (344, 228)
Mouth Location:  [287, 251] Sipper Location:  [344, 228] Arm Motion: Move down, Direction Accuracy: 97.8
Mouth Location:  (286, 251) Sipper Location:  (347, 213)
Mouth Location:  [286, 251] Sipper Location:  [347, 213] Arm Motion: Move down, Direction Accuracy: 97.8
Mouth Location:  (286, 251) Sipper Location:  (348, 208)
Mouth Location:  [286, 251] Sipper Location:  [348, 208] Arm Motion: Move down, Direction Accuracy: 97.1
Mouth Location:  (286, 251) Sipper Location:  (347, 204)
Mouth Location:  [286, 251] Sipper Location:  [347, 204] Arm Motion: Move down, Direction Accuracy: 97.6
Mouth Location:  (286, 251) Sipper Location:  (347, 201)
Mouth Location:  [286, 251] Sipper Location:  [347, 201] Arm Motion: Move down, Direction Accuracy: 97.4
Mouth Location:  (286, 251) Sipper Location:  (345, 190)
Mouth Location:  [286, 251] Sipper Location:  [345, 190] Arm Motion: Move down, Direction Accuracy: 99.8
Mouth Location:  (287, 251) Sipper Location:  (345, 188)
Mouth Location:  [287, 251] Sipper Location:  [345, 188] Arm Motion: Move down, Direction Accuracy: 99.2
Mouth Location:  (287, 251) Sipper Location:  (345, 188)
Mouth Location:  [287, 251] Sipper Location:  [345, 188] Arm Motion: Move down, Direction Accuracy: 99.9
Mouth Location:  (287, 251) Sipper Location:  (345, 188)
Mouth Location:  [287, 251] Sipper Location:  [345, 188] Arm Motion: Move down, Direction Accuracy: 98.4
Mouth Location:  (287, 251) Sipper Location:  (344, 191)
Mouth Location:  [287, 251] Sipper Location:  [344, 191] Arm Motion: Move down, Direction Accuracy: 98.6
Mouth Location:  (287, 252) Sipper Location:  (345, 191)
Mouth Location:  [287, 252] Sipper Location:  [345, 191] Arm Motion: Move down, Direction Accuracy: 98.6
Mouth Location:  (287, 252) Sipper Location:  (345, 192)
Mouth Location:  [287, 252] Sipper Location:  [345, 192] Arm Motion: Move down, Direction Accuracy: 98.7
Mouth Location:  (287, 252) Sipper Location:  (345, 193)
Mouth Location:  [287, 252] Sipper Location:  [345, 193] Arm Motion: Move down, Direction Accuracy: 97.1
Mouth Location:  (287, 251) Sipper Location:  (343, 201)
Mouth Location:  [287, 251] Sipper Location:  [343, 201] Arm Motion: Move down, Direction Accuracy: 97.7
Mouth Location:  (287, 251) Sipper Location:  (343, 204)
Mouth Location:  [287, 251] Sipper Location:  [343, 204] Arm Motion: Move down, Direction Accuracy: 99.8
Mouth Location:  (287, 251) Sipper Location:  (342, 208)
Mouth Location:  [287, 251] Sipper Location:  [342, 208] Arm Motion: Move down, Direction Accuracy: 98.9
Mouth Location:  (287, 251) Sipper Location:  (343, 210)
Mouth Location:  [287, 251] Sipper Location:  [343, 210] Arm Motion: Move down, Direction Accuracy: 97.0
```

Fig. 7. Arm directions towards mouth and with the accuracy

RQ2. Answer: The above results showing that the direction accuracy is about 98% on average.

When the sipper moved to the mouth location, this is the time where baby starts sucking the sipper. As you can see in the Fig. 8., the system shows sipping when the fingertip reached to the mouth location. When the sipper reaches the mouth, the system stops sending direction signals to the arm, which holds the arm in stationary position. The Fig. 9. shows that the system successfully directed the arm to move the sipper to the baby's mouth with around 98% of accuracy, resulting the sipper at the baby's mouth.

Fig. 8. System Indicating the Sipping



```
Mouth Location:  [288, 249] Sipper Location:  [322, 249] Arm Motion: Move Left, Direction Accuracy: 98.85
Mouth Location:  (287, 249) Sipper Location:  (321, 249)
Mouth Location:  [287, 249] Sipper Location:  [321, 249] Arm Motion: Move Left, Direction Accuracy: 99.76
Mouth Location:  (288, 250) Sipper Location:  (320, 249)
Mouth Location:  [288, 250] Sipper Location:  [320, 249] Arm Motion: Move Left, Direction Accuracy: 99.10
Mouth Location:  (288, 249) Sipper Location:  (319, 249)
Mouth Location:  [288, 249] Sipper Location:  [319, 249] Arm Motion: Move Left, Direction Accuracy: 98.83
Mouth Location:  (288, 249) Sipper Location:  (318, 249)
Mouth Location:  [288, 249] Sipper Location:  [318, 249] Arm Motion: Move Left, Direction Accuracy: 97.27
Mouth Location:  (290, 250) Sipper Location:  (314, 250)
Mouth Location:  [290, 250] Sipper Location:  [314, 250] Arm Motion: Move Left, Direction Accuracy: 98.67
Mouth Location:  (289, 250) Sipper Location:  (312, 248)
Mouth Location:  [289, 250] Sipper Location:  [312, 248] Arm Motion: Move Left, Direction Accuracy: 98.13
Mouth Location:  (290, 249) Sipper Location:  (311, 249)
Mouth Location:  [290, 249] Sipper Location:  [311, 249] Arm Motion: Move Left, Direction Accuracy: 99.20
Mouth Location:  (292, 250) Sipper Location:  (307, 250)
Mouth Location:  [292, 250] Sipper Location:  [307, 250] Arm Motion: Move Left, Direction Accuracy: 98.46
Sipping
Mouth Location:  (293, 249) Sipper Location:  (305, 251)
Mouth Location:  [293, 249] Sipper Location:  [305, 251] Arm Motion: Move Left, Direction Accuracy: 97.08
Sipping
Mouth Location:  (294, 249) Sipper Location:  (304, 251)
Mouth Location:  [294, 249] Sipper Location:  [304, 251] Arm Motion: Move Left, Direction Accuracy: 99.44
Sipping
Mouth Location:  (294, 250) Sipper Location:  (303, 251)
Sipping
Mouth Location:  (295, 249) Sipper Location:  (302, 250)
Sipping
Mouth Location:  (295, 249) Sipper Location:  (301, 250)
Sipping
Mouth Location:  (297, 249) Sipper Location:  (301, 251)
Sipping
Mouth Location:  (296, 248) Sipper Location:  (300, 252)
Sipping
Mouth Location:  (298, 248) Sipper Location:  (301, 250)
Sipping
```

Fig.9. Sipping in Process.

5. Conclusion and Future Work

The advancement of technology has rapidly accelerated. Given its significant development, technology can contribute to society in various ways. This endeavor represents another contribution to society, particularly for working parents, need not to feed the baby when they are hungry where automation does for them. The health and the wellness of the baby is so important for their parents and they have to fill their needs time to time. This work fulfils one of such need and creates another level of baby caring.

The proposed work is only the core idea of feeding the baby but there's a lot more things that can be integrated from the existing systems. We can create complete monitoring system, that can collect the data of the circumstance of the baby from the smallest of his temperature to the largest of his emotions and send them to a mobile application, so the parents can watch the children remotely and can control the circumstances, even feeding the baby. The ideas are more, maybe we need to accelerate.

Acknowledgments

We would like to express our sincere gratitude to Sreenidhi Group of Institutions for providing funds to publish the paper. We are also thankful to Prof. Aruna Varanasi and Prof. Sreedhar Bhukya for helping us to provide the valuable feedback and suggestions to get the quality of the paper. We would like to thank all the participants of this study for their cooperation and participation. Finally, we are indebted to our families and friends for their unwavering support and encouragement throughout this research endeavour.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Patil, Aniruddha & Patil, Nitesh & Mishra, Deepak & Mane, Yogita. (2018). Smart Baby cradle. 1-5. 10.1109/ICSCET.2018.8537356.

[2] S, Kavitha & R, Neela & M, Sowndarya & Madhuchandra, & Kamal, Harshitha. (2019). Analysis on IoT Based Smart Cradle System with an Android Application for Baby Monitoring. 136-139. 10.1109/ICATIECE45860.2019.9063773.

[3] W. A. Jabbar, H. K. Shang, S. N. I. S. Hamid, A. A. Almohammedi, R. M. Ramli and M. A. H. Ali, "IoT-BBMS: Internet of Things-Based Baby Monitoring System for Smart Cradle," in IEEE Access, vol. 7, pp. 93791-93805, 2019, doi: 10.1109/ACCESS.2019.2928481

[4] Thopate, Kaushalya & Gawade, Mayuri & Savale, Vaishali & Musale, Prajakta. (2023). Smart Cradle: A Technology-Enabled Solution for Safer and Better Infant Sleep. International Journal on Recent and Innovation Trends in Computing and Communication. 11. 223-228. 10.17762/ijritcc.v11i7.7849.

[5] Ingale, Aniket & Kshirsagar, Somnath & Nikam, Sonal & Nagalkar, Vinay. (2021). BABY MONITORING SYSTEM. 7.

[6] S. Durga, S. Itnal, K. Soujanya, C. Z. Basha and C. Saxena, "Advanced and effective baby care monitoring Smart cradle system using Internet of Things," 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2021, pp. 35-42, doi: 10.1109/ICOSEC51865.2021.9591955.

[7] Patil, Aniruddha & Patil, Nitesh & Mishra, Deepak & Mane, Yogita. (2018). Smart Baby cradle. 1-5. 10.1109/ICSCET.2018.8537356.

[8] Durga, S. kanaka et al. "Advanced and effective baby care monitoring Smart cradle system using Internet of Things." 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC) (2021): 35-42.

[9] M. H. McGillion, E. Duceppe, K. Allan et al., "Postoperative remote automated monitoring: need for and state of the science," Canadian Journal of Cardiology, vol. 34, no. 7, pp. 850–862, 2018.

[10] Shreelatha, Shreya Pai, Sonal Cynthia Pereira,Tanya Nicole, Ms.Ushadevi A, "Advanced Baby Monitor", International Journal of Internet of Things 2017, 6(2): 51-55 DOI: 10.5923/j.ijit.20170602.09.

[11] Telepatil, A. R., Patil, P. P., Yajare, S. S., & Jadhav, S. R. (2019). Intelligent baby monitoring system. *International Journal of Research in Advent Technology*, *7*(6), 191-194.

[12] Aslam Forhad Symon, Nazia Hassan, Humayun Rashid,Iftekhar Uddin Ahmed, S M Taslim Reza, "Design and Development of a Smart Baby Monitoring System based on Raspberry Pi and Pi Camera", Proceedings of the 2017 4th International Conference on Advances in Electrical Engineering, 30 September, 2017, Dhaka, Bangladesh DOI:10.1109/ICAEE.2017.8255338

[13] M. Goyal and D. Kumar, "Automatic e-baby cradle swing based on baby cry," International Journal of Computers and Applications, vol. 975, p. 8887, 2013.

[14] F. Symon, N. Hassan, H. Rashid, I. U. Ahmed, and S. M. Taslim Reza, "Design and development of a smart baby monitoring system based on Raspberry Pi and Pi camera," in 2017 4th International Conference on Advances in Electrical Engineering (ICAEE), pp. 117–122, Dhaka, Bangladesh, 2017.

[15] D. M. Ibrahim, M. A. Hammoudeh, S. Ambreen, and S. Mohammadi, "Raspberry Pi-based smart infant monitoring system," International Journal of Engineering Research and Technology, vol. 12, no. 10, pp. 1723–1729, 2019.

[16] A. Rudyansyah, H. L. Warnars, F. L. Gaol, and T. Matsuo, "A prototype of baby monitoring use raspberry pi," in 2020 International Conference on ICT for Smart Society (ICISS), pp. 1–4, Bandung, Indonesia, 2020.

[17] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C., Yong, M.G., Lee, J., Chang, W., Hua, W., Georg, M., &

Grundmann, M. (2019). MediaPipe: A Framework for Building Perception Pipelines. ArXiv, abs/1906.08172.

[18] I. Culjak, D. Abram, T. Pribanic, H. Dzapo and M. Cifrek, "A brief introduction to OpenCV," 2012 Proceedings of the 35th International Convention MIPRO, Opatija, Croatia, 2012, pp. 1725-1730. keywords:

[19] R. Chauhan, K. K. Ghanshala and R. C. Joshi, "Convolutional Neural Network (CNN) for Image Detection and Recognition," 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, 2018, pp. 278-282, doi: 10.1109/ICSCCC.2018.8703316.