

Article

An Intelligent Baby Monitor with Automatic Sleeping Posture Detection and Notification

Tareq Khan 

School of Engineering, Eastern Michigan University, Ypsilanti, MI 48197, USA; tareq.khan@emich.edu

Abstract: Artificial intelligence (AI) has brought lots of excitement to our day-to-day lives. Some examples are spam email detection, language translation, etc. Baby monitoring devices are being used to send video data of the baby to the caregiver's smartphone. However, the automatic understanding of the data was not implemented in most of these devices. In this research, AI and image processing techniques were developed to automatically recognize unwanted situations that the baby was in. The monitoring device automatically detected: (a) whether the baby's face was covered due to sleeping on the stomach; (b) whether the baby threw off the blanket from the body; (c) whether the baby was moving frequently; (d) whether the baby's eyes were opened due to awakening. The device sent notifications and generated alerts to the caregiver's smartphone whenever one or more of these situations occurred. Thus, the caregivers were not required to monitor the baby at regular intervals. They were notified when their attention was required. The device was developed using NVIDIA's Jetson Nano microcontroller. A night vision camera and Wi-Fi connectivity were interfaced. Deep learning models for pose detection, face and landmark detection were implemented in the microcontroller. A prototype of the monitoring device and the smartphone app were developed and tested successfully for different scenarios. Compared with general baby monitors, the proposed device gives more peace of mind to the caregivers by automatically detecting un-wanted situations.

Keywords: human pose detection; face detection; deep learning; cloud messaging; hypertext transfer protocol (HTTP) server; jetson nano



Citation: Khan, T. An Intelligent Baby Monitor with Automatic Sleeping Posture Detection and Notification. *AI* **2021**, *2*, 290–306. <https://doi.org/10.3390/ai2020018>

Academic Editor: Giovanni Diraco

Received: 27 April 2021

Accepted: 15 June 2021

Published: 18 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Smart baby monitoring devices are being used to obtain and send video and audio data of the baby to the caregiver's smartphone, but most of these devices are unable to recognize or understand the data. In this project, a novel baby monitoring device is developed which automatically recognizes the undesired and harmful postures of the baby by image processing and sends an alert to the caregiver's smartphone—even if the phone is in sleep mode. Deep learning-based object detection algorithms are implemented in the hardware, and a smartphone app is developed. The overall system is shown in Figure 1.

The research problem addressed in this paper is to develop methods to automatically detect (a) whether the baby's face is covered due to sleeping on the stomach; (b) whether the baby has thrown off the blanket from the body; (c) whether the baby is moving frequently; (d) whether the baby's eyes are opened due to awakening. One of the challenges of deep learning models is running them in embedded systems where resources such as memory and speed are limited. The methods must work in an embedded system with low latency. The system should also work in both day and night conditions. The detection methods should not be biased and be inclusive to all races of babies.

The objective of this study is to develop a baby monitoring device that will automatically recognize the harmful postures of the baby by image processing and send an alert to the caregiver's smartphone. The work will be considered successful when the proposed baby monitor can automatically detect the targeted harmful postures and send a notification to the smartphone. Experiments with different postures will be conducted and the latency of the detection algorithms will be measured.

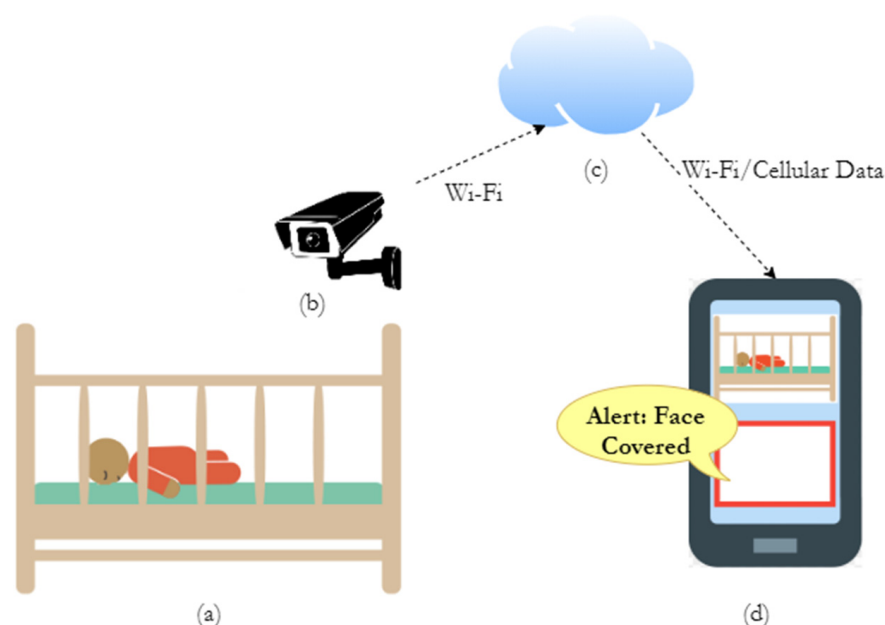


Figure 1. The overall smart baby monitoring system: (a) baby sleeping; (b) smart baby monitoring device automatically detects harmful postures such as face covered, thrown off the blanket, frequently moving or awake; (c) image data are sent to the smartphone through the Internet with the message of any harmful situation; (d) a real-time video of the baby is shown in the smartphone, and notifications and alerts are generated whenever the caregiver receives a message of the harmful situation of the baby.

The needs and significances of the proposed system are mentioned below:

- About 1300 babies died due to sudden infant death syndrome (SIDS), about 1300 deaths were due to unknown causes, and about 800 deaths were caused by accidental suffocation and strangulation in bed in 2018 in the USA [1]. Babies are at higher risk for SIDS if they sleep on their stomachs as it causes them to breathe less air. The best and only position for a baby to sleep is on the back—which the American Academy of Pediatrics recommends through the baby’s first year [2]. Sleeping on the back improves airflow. To reduce the risk of SIDS, the baby’s face should be uncovered, and body temperature should be appropriate [3]. The proposed baby monitor will automatically detect these harmful postures of the baby and notify the caregiver. This will help to reduce SIDS.
- Babies—especially four months or older—move frequently during sleep and can throw off the blanket from their body [4]. The proposed system will alert when the baby is moving frequently and also whether the blanket is removed. Thus, it helps to keep the baby warm.
- Babies may wake up in the middle of the night due to hunger, pain, or just to play with the parent. There is an increasing call in the medical community to pay attention to parents when they say their babies do not sleep [5]. The smart baby monitor detects whether the baby’s eyes are open and sends an alert. Thus, it helps the parents know when the baby is awake even if he/she is not crying.
- When a baby sleeps in a different room, the caregivers need to check the sleeping condition of the baby after a regular interval. Parents lose an average of six months’ sleep during the first 24 months of their child’s life. Approximately 10% of parents manage to get only 2.5 h of continuous sleep each night. Over 60% of parents with babies aged less than 24 months get no more than 3.25 h of sleep each night. A lack of sleep can affect the quality of work and driving; create mental health problems, such as anxiety disorders and depression; and cause physical health problems, such as obesity, high blood pressure, diabetes, and heart disease [6]. The proposed smart device will automatically detect the situations when the caregiver’s attention is required and

generate alerts. Thus, it will reduce the stress of checking the baby at regular intervals and help the caregiver to have better sleep.

- The proposed baby monitor can send video and alerts using the Internet even when the parent/caregiver is out of the home Wi-Fi network. Thus, the parent/caregiver can monitor the baby with the smartphone while at work, grocery, park, etc.
- Do smart devices make us lazy? It surely depends on the ethical and responsible use of technology. Smart devices allow humans to have more time for creative work by automating routine tasks [7,8].

There are commercial baby monitoring devices such as the MBP36XL baby monitor by Motorola [9] and the DXR-8 video baby monitor by Infant Optics [10] available in the market that can only send video and audio data but are unable to automatically recognize harmful postures of the baby. The Nanit Pro smart baby monitor [11] can monitor the breathing of the baby; however, the baby must wear a special breathing dress—which is an overhead. A recent Lollipop baby monitor [12] can automatically detect crying sounds and a baby crossing a certain boundary from the crib. The Cubo Ai smart baby monitor [13] can detect faces covered due to sleeping on the back; however, it cannot detect a blanket removed, frequent moving, and awake or sleep state from the eyes. The detailed algorithms used in these commercial baby monitors are not publicly available. The proposed work embraces an open science approach, and the methods are described in detail for the researchers to repeat the experiments. The rest of the paper is organized as follows. In Section 2, materials and methods are discussed with the proposed detection algorithms and the prototype development. Results are discussed in Section 3. In Section 4, the discussion and the limitations of the study are presented. Finally, Section 5 presents the conclusion.

2. Materials and Methods

The steps taken to develop the detection algorithms of harmful and undesired sleeping postures from image data and prototype development of the smart baby monitor are briefly shown in Figure 2. They are described below.

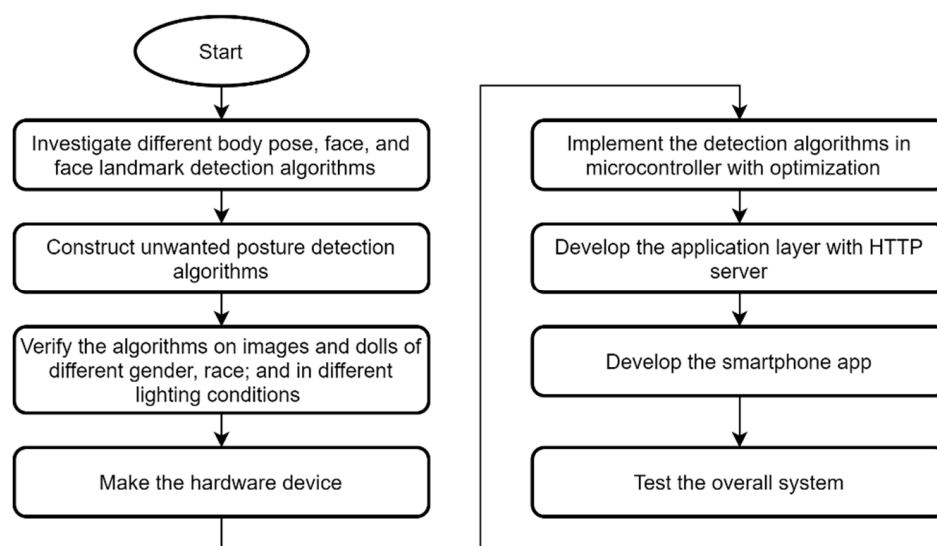


Figure 2. Flowchart showing the steps for developing the detection algorithms and the prototype.

2.1. Detection Algorithms

The experimental setup as shown in Figure 3 is used to develop the alerting situation detection algorithms. A night-vision camera [14] is interfaced with an NVIDIA Jetson Nano microcontroller [15]. Realistic baby dolls [16–20] of both genders and different races—Asian, Black, Caucasian, Hispanic—were put under the camera during experiments. Both a daylight condition and night vision condition—where the doll is illuminated by

infrared light—were taken into consideration. A brief description of the detection of the four alerting situations—face covered, blanket not covering the body, frequently moving and awake—is described below.

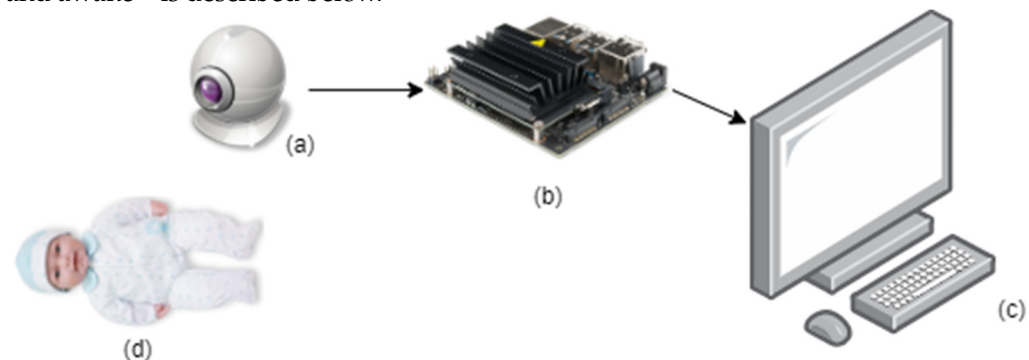


Figure 3. Experimental setup: a night-vision camera (a) is interfaced with an NVIDIA Jetson Nano microcontroller (b). Monitor, wireless keyboard, and wireless mouse (c) were connected with the Jensen Nano. Realistic baby doll (d) of both genders and different races were put under the camera during experiments.

2.1.1. Detection of Face Covered and Blanket Removed

The nose of the baby is detected from the image to decide whether the face is covered due to sleeping on the stomach or for other reasons. To detect a blanket removed, the visibility of the lower body parts such as the hip, knee, and ankle are detected. The pseudocode for a face covered and blanket removed detection is shown in Figure 4.

```
TotalHuman, BP := DetectBodyParts (Image)
If TotalHuman == 1:
    isFaceCovered := NOT BP[0].isNoseVisible
    isBlanketRemoved := BP[0].isLeftHipVisible OR
                        BP[0].isRightHipVisible OR
                        BP[0].isLeftKneeVisible OR
                        BP[0].isRightKneeVisible OR
                        BP[0].isLeftAnkleVisible OR
                        BP[0].isRightAnkleVisible
```

Figure 4. Pseudocode for a face covered and blanket removed detection.

Pose detection techniques [21–23] are used to detect the body parts. Densenet 121 [24] is used as the backbone network for feature extraction. The features are then fed into two-branch multi-stage transposed convolution networks. These branch networks simultaneously predict the heatmap and Part Affinity Field (PAF) matrices. The model was trained for 160 epochs with the COCO [25] dataset. The COCO dataset is a large dataset containing 1.5 million object instances and 80 categories of objects. It has images of 250,000 people; of them, 56,165 people have labeled key points such as nose, eye, ear, etc. [26].

The heatmap is a matrix that stores the confidence of a certain pixel containing a certain part. There are 18 heatmaps associated with each one of the body parts. PAFs are matrices that give information about the position and orientation of pairs. A pair is a connection between parts. They come in couples: for each part, there is a PAF in the ‘x’ direction and a PAF in the ‘y’ direction. Once the candidates for each one of the

body parts are found, they are then connected to form pairs guided by the PAFs. The line integral along the segment connecting each couple of part candidates are computed over the corresponding PAFs (x and y) for that pair. A line integral measures the effect of a PAF among the possible connections between part candidates. It gives each connection a score, that is saved in a weighted bipartite graph. The weighted bipartite graph shows all possible connections between candidates of two parts and holds a score for every connection. Then, the connections are searched that maximize the total score; that is, solving the assignment problem using a greedy algorithm. The last step is to transform these detected connections into the skeletons of a person. Finally, a collection of human sets is found, where each human is a set of parts, and where each part contains its relative coordinates.

2.1.2. Frequent Moving Detection

The motion of the baby is detected by image processing [27,28]. To detect motion, the captured image is first converted to grayscale, as color information is not required for motion detection. Then, Gaussian Blur [29] is applied to smooth the image. In the Gaussian Blur operation, the image is convolved with a Gaussian filter. The Gaussian filter is a low-pass filter that removes the high-frequency camera noises. Then, an absolute difference image is calculated by subtracting the image from a previously captured image. The previously captured image is captured, gray-scaled, blurred and saved one second before. The difference image contains larger values where motion is detected, and smaller values where no or insignificant motion is detected. The image is then threshold [30] to make it a binary-black and white-image. It converts the motion regions to white and non-motion background regions to black. Then, the white region is enlarged by dilation [31] and contours [32] are drawn around them. Then, the area of each contour is calculated. If the area of any contour is larger than a threshold area, then it indicates a transient motion. This thresholding avoids small movements to be considered as a transient motion.

The term *frequent moving* is defined if there is at least one transient motion in every three consecutive blocks of time. A block of time is declared to be 10 s. Whenever a transient motion is detected, a *block movement flag* is set as true. A first-in-first-out (FIFO) of size three is used to store the last three block movement flags. After every 10 s, an item from the FIFO is removed and the *block movement flag* is put in the FIFO. The *block movement flag* is then set as false. If all the entries of the FIFO are true, then a *frequent moving flag* is set to true.

2.1.3. Awake Detection

To detect whether the baby is awake or asleep, the eye landmarks from the image are processed. The flowchart for awake detection is shown in Figure 5.

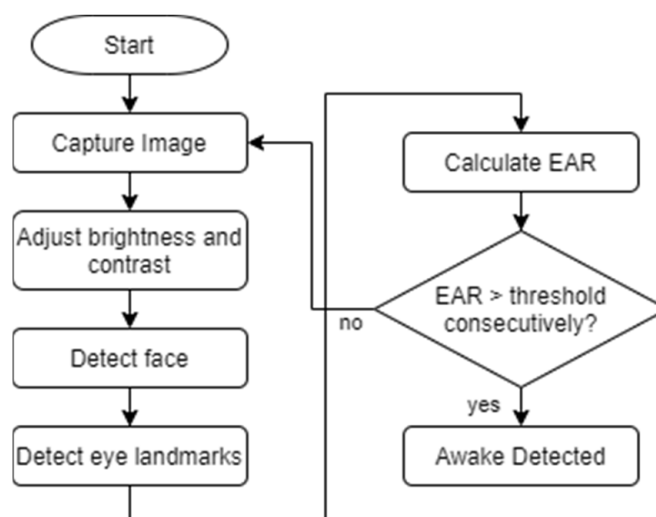


Figure 5. Flowchart for awake detection (EAR = eye aspect ratio).

The face of the baby is detected using the Multi-Task Cascaded Convolutional Neural Network (MTCNN) [33]. It is a deep learning-based method. It can detect not only faces but can also detect landmarks such as the location of the two eyes, nose, and mouth. The model has a cascade structure with three networks. First, the image is rescaled to a range of different sizes. Then, the first model (Proposal Network or P-Net) proposes candidate facial regions. Addition processing such as non-maximum suppression (NMS) is also used to filter the candidate bounding boxes. Then, the second model (Refine Network or R-Net) filters the bounding boxes, and the third model (Output Network or O-Net) proposes facial landmarks. These three models are trained for face classification, bounding box regression, and facial landmark localization namely. It was found that reducing the brightness and increasing the contrast of the image gives better face detection for both day and night light conditions. Therefore, the brightness and the contrast are adjusted of the image before passing it through the MTCNN. The eye landmarks detected by the MTCNN only provide the location of the eyes and cannot be used to detect whether the eyes are open or closed.

Once the face bounding box is detected, the region of interest (ROI) is then passed to a facial landmark detector [34–36]. In this method, regression trees are trained using a gradient boosting algorithm with labeled datasets [37] to detect the x and y locations of 68 points on the face such as on the mouth, eyebrows, eyes, nose, and jaw. On each eye, the six locations are detected, as shown in Figure 6. The eye aspect ratio (EAR') is then calculated using Equation (1).

$$EAR' = (BF + CE)/2AD \quad (1)$$

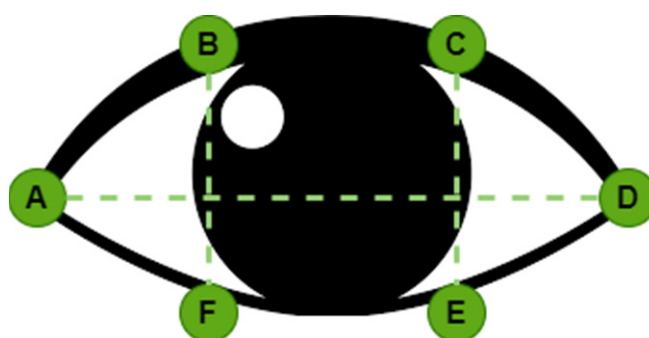


Figure 6. Eye landmarks. Two points (B and C) on the upper eyelid, two points (F and E) on the lower eyelid, and two points (A and D) on the two corners of the eye.

When an eye is open, the EAR' will be larger; when an eye is closed, the EAR' will be smaller. The average of the left and right eye EAR s are used to find the final EAR . If the EAR is larger than a threshold, then eye open is detected. The threshold is set to 0.25.

While awake, the baby may blink, and the eye gets closed for a short amount of time. It is not desirable to change the status to sleeping for blinking, as this is misleading. It is implemented that if the eye is closed consecutively for a defined number of loop cycles, then sleep is detected. In the same way, if the eye is opened consecutively for a defined number of loop cycles, then an awake state is detected.

2.2. Prototype Development

The proposed system consists of the smart baby monitor device and the smartphone app. They are briefly described below.

2.2.1. Smart Baby Monitor Device

The smart baby monitor device is placed above the baby's crib. It takes images of the baby, detects harmful or undesired situations, and sends a notification to the caregiver's smartphone. The hardware and the software parts of this device are briefly described below.

Hardware: The single-board computer—NVIDIA® Jetson Nano™ [15]—is used as the main processing unit. It is a small size and low-power embedded platform where neural network models can run efficiently for applications such as image classification, object detection, segmentation, etc. It contains a Quad-core ARM A57 microprocessor running at 1.43 GHz, 4 GB of RAM, a 128-core Maxwell graphics processing unit (GPU), a micro SD card slot, USB ports, and other built-in hardware peripherals. A night-vision camera [14] is interfaced with the Jetson Nano using a USB. When the surrounding light is enough, such as in the daytime, it captures color images. This camera has a built-in light sensor and infrared (IR) LEDs. When the surrounding light is low, the IR LEDs automatically turn on and it captures grayscale images. To connect with the Internet wirelessly, a Wi-Fi adaptor [38] is connected to the USB port of the Jetson Nano. A 110V AC to 5V 4A DC adapter is used as the power supply. A cooling fan with pulse width modulation (PWM)-based speed control is placed above the microprocessor. The hardware block diagram is shown in Figure 7.

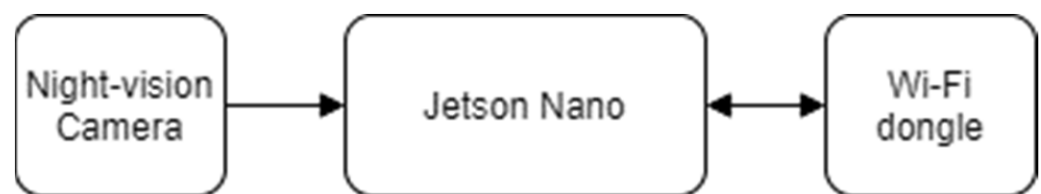


Figure 7. Block diagram of the baby monitor hardware.

Software: Linux4Tegra (L4T)—a version of Ubuntu operating system (OS)—is installed on a 32 GB SD card of the Jetson Nano board. The application software is developed in Python language and the necessary packages are installed.

The device connects to the home router using Wi-Fi to access the Internet. To stream real-time video to the smartphone and to receive commands from the smartphone that is outside of the home Wi-Fi network, the device must be accessed from outside of the home network. A Hypertext Transfer Protocol (HTTP) server, known as Flask [39], is implemented in the device. The device's private IP is made static and port forwarding [40] is configured. Thus, the server can be accessed from outside of the home network using its public IP and the port number.

The smartphone app sends commands using an HTTP GET request to the baby monitor device to start, stop, and configure settings such as enabling or disabling one or more detections. Depending upon the words contained in the Uniform Resource Locator (URL) of the GET requests, call back functions are executed to start, to stop, and to configure settings of the device. A separate thread is used that captures VGA (640 × 480) images and stores them in a global object with thread locking [41]. This thread continues to capture images after it receives the start command. To stream video, a separate thread reads the captured image from the global object considering thread locking, compresses the image to JPEG, adds headers to the encoded stream as an image object, and then sends it to the HTTP client, i.e., to the smartphone. Making a separate thread for streaming video solves the latency problem that will be caused by the detection algorithms in a single loop-based program.

After the device receives the start command, a separate thread reads the captured image considering thread locking—to detect the harmful and undesired situations. Depending upon the detection alerts requested from the user, it continuously executes the requested detection algorithms—such as face covered, blanket removed, moving, or awake—as discussed in Section 2.1. To reduce the inference time of MTCNN for face detection and Densenet121-based body part detection on the Jetson Nano, NVIDIA-TensorRT [42,43] is used. TensorRT includes a deep learning inference optimizer and runtime that delivers low latency and high throughput for deep learning inference applications. TensorRT provides INT8 and FP16 optimizations and the reduced precision significantly reduces the inference latency.

If a change occurs in any of the requested detection results, a message containing the results of the detections is sent to the user's smartphone using Firebase Cloud Messaging (FCM) [44]. FCM can send a message of a maximum of 4 KB at no cost using the Internet to a client app. Using FCM, a smartphone app can be immediately notified whenever new data are available to sync. The message is sent using a Server key—that is generated from the cloud server where the smartphone app is registered.

2.2.2. Smartphone App

The Android platform was used to develop the smartphone app. The first screen of the app contains a WebView object [45] to show the real-time video of the baby; a toggle button to start and to stop the baby monitor device; labels for displaying alerts, last detection status update time, and connection status with the baby monitor device using the Internet. It also contains a button to clear the alerts manually by the user and the button is visible only when there is at least one alert present.

The app contains a settings menu for configuration. It contains check boxes for enabling or disabling real-time video and the four detection alerts such as face covered, blanket removed, moving or awake; textboxes for the baby monitor device's public IP and port number; and checkboxes for enabling or disabling voice and vibration alert. To make an HTTP request to the baby monitor device from the smartphone app, the public IP of the device is required. If the smartphone is connected with the same Wi-Fi network of the device, then the public IP of the Wi-Fi network can be auto-filled by pressing a button—that sends an HTTP request to <https://ipecho.net/plain> (accessed on 17 June 2021).and it responds to the IP from where the request was made. Once the user exits from the settings menu by pressing the back button, the app saves the data in the *settings.dat* file and sends an HTTP GET request to the device using the *settings* word and a binary string indicating the requested detections in the URL. If the baby monitor device is online, then it responds with a success message, and the connection status in the app with the device is shown as connected.

When the user presses the start/stop toggle button, the app sends HTTP GET requests to the device containing the word start/stop, namely in the URL. After starting, the URL for the WebView is set as the public IP of the device with port number and thus it shows the real-time video of the baby in the app. Whenever the detection status is changed in the device, a new FCM message containing the current status arrives in the smartphone app. The flowchart in Figure 8 shows the actions taken by the smartphone whenever the FCM message is received. In this way, the data are synchronized between the device in the home and the user's smartphone—at whatever place the user may be in the world. When it receives an FCM message, a call back function is called and the app saves the current detection status—such as *is face covered*, *is blanket removed*, *is moving*, or *is awake*—with the current date/time information in the *status.dat* file. The app then generates smartphone notifications, voice alerts, and vibration alerts—depending on the last detection status and alert settings. The voice alert is generated using text-to-speech and it speaks out loud the harmful situation such as “Alert: Face Covered”, “Alert: Blanket Removed”, etc. The voice alerts and the phone vibration alerts are repeated using a timer. If the FCM message contains no alerts, then the timer is disabled, speech is turned off, and the notifications are cleared. The first screen label of the app is then updated to show the current alert status of the baby. Once the user is aware of the alerts, the user may click a clear alert button to manually stop the alerts.

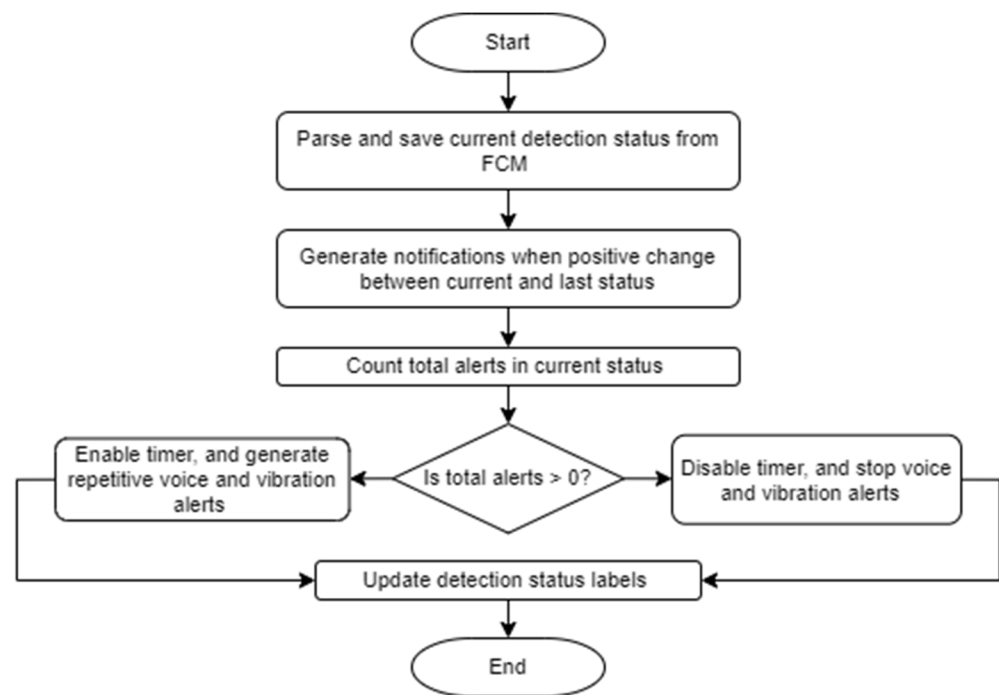


Figure 8. Alert generation after FCM message arrives.

3. Results

The detection algorithm results and the prototype results are discussed below.

3.1. Detection Algorithm Results

The face covered and blanket removed detection results are shown in Figures 9 and 10, respectively. They show the locations of the detected body parts and the skeleton in green color on different realistic baby dolls in different light conditions. In Figure 11, the different processing steps are shown for moving detection. Figure 12 shows the face and eye landmark detection for awake detection in green color for eye open and eye close situations in two different light conditions. Here, the EAR is 0.43 in Figure 12a when the eye is open, and the EAR is 0.17 in Figure 12b when the eye is closed.

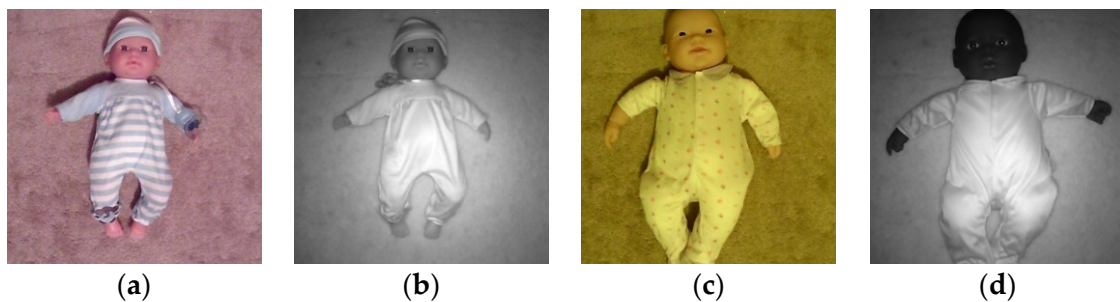


Figure 9. Cont.

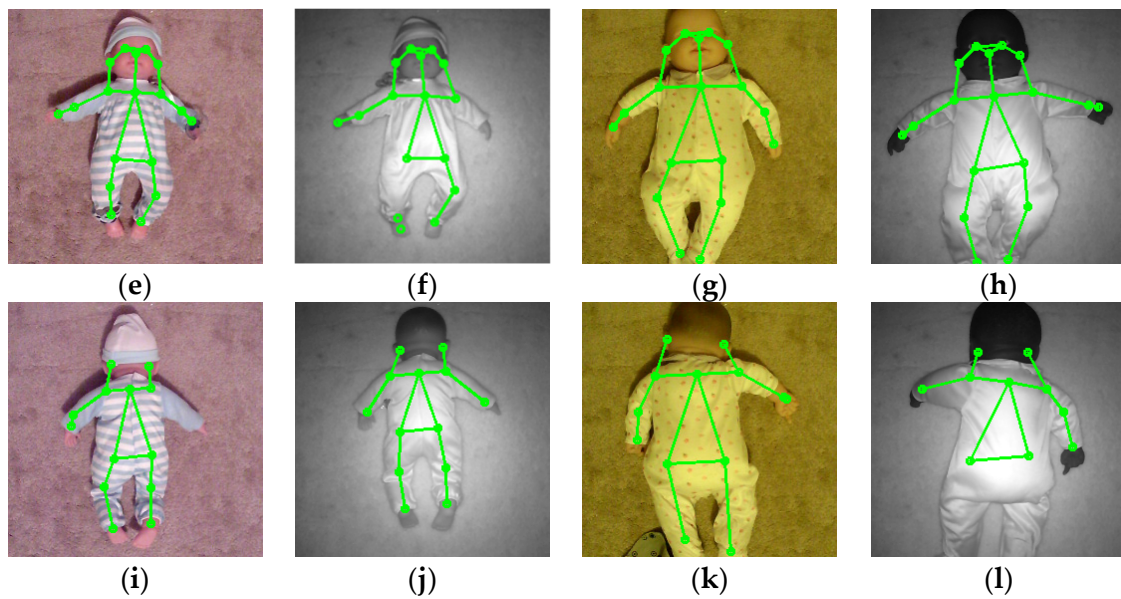


Figure 9. (a–d): captured images of dolls in the day and night conditions; (e–h) nose detected (with other body parts), indicating face is not covered; (i–l) nose undetected, indicating face is covered.

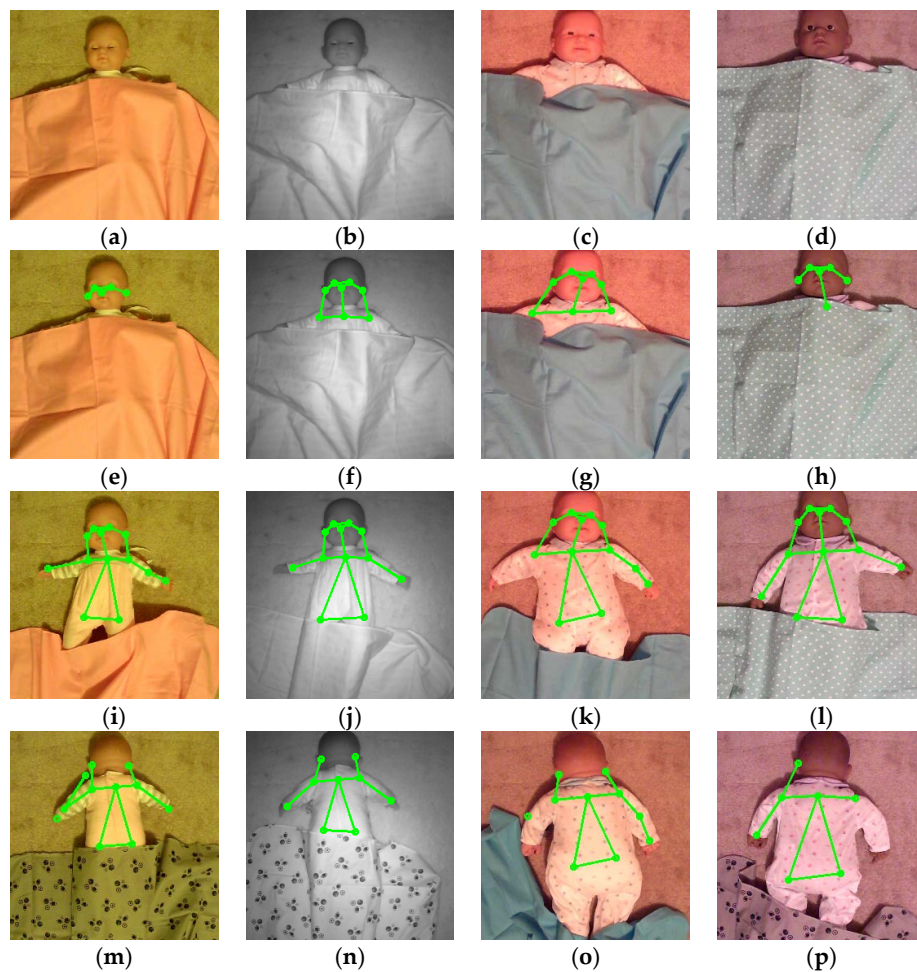


Figure 10. (a–d): captured images of dolls with a blanket in the day and night conditions; (e–h) lower body parts not detected, indicating blanket is not removed; (i–l) lower body parts (such as hips) detected, indicating blanket is removed; (m–p) nose undetected and lower body parts detected, indicating face is covered and blanket is removed.

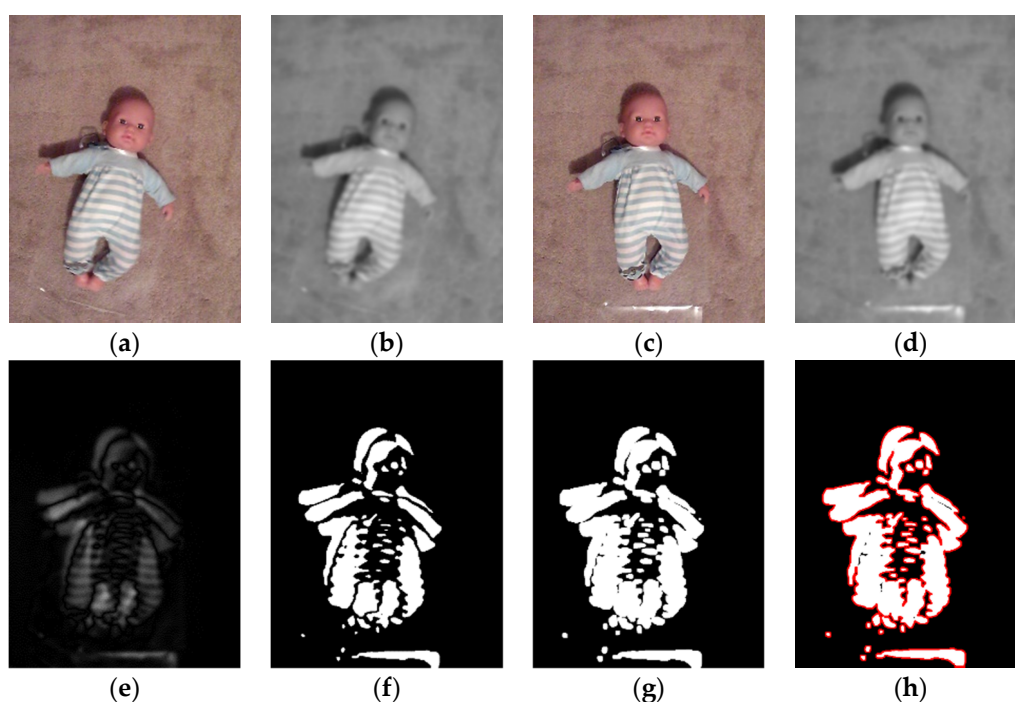


Figure 11. (a) the previous frame; (b) previous frame grayscaled and blurred; (c) the current frame; (d) current frame grayscaled and blurred; (e) absolute difference image between (b,d); (f) black and white binary image using thresholding; (g) image after dilation; (h) contours shown in red.

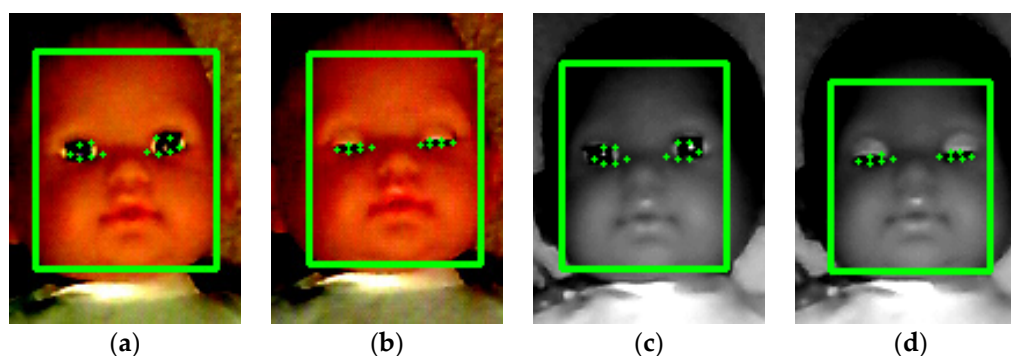


Figure 12. Face and eye landmark detection: (a) eye opened in day (EAR = 0.43); (b) eye closed in day (EAR = 0.17); (c) eye opened in night (EAR = 0.42); (d) eye closed in night (EAR = 0.19).

3.2. Prototype Results

A prototype of the proposed smart baby monitor device and smartphone app has been developed and tested successfully. A photograph of the prototype device is shown in Figure 13. The physical dimension of the Jetson Nano board is 69 mm × 45 mm and the total power consumption of the proposed device is measured to be around 24.5 watts [46]. Different harmful and unwanted situations such as face covered due to sleeping on the stomach, blanket removed, frequently moving, and awake were created in baby dolls in both daylight and night environments, and the proposed baby monitor was able to detect, send, and generate alerts in the smartphone. Some screenshots of the smartphone app are shown in Figure 14. The smartphone was taken out of the range of home Wi-Fi and it was connected with the Internet using the cellular network. In this scenario, the video stream was received, and the alerts were also generated successfully in the smartphone.

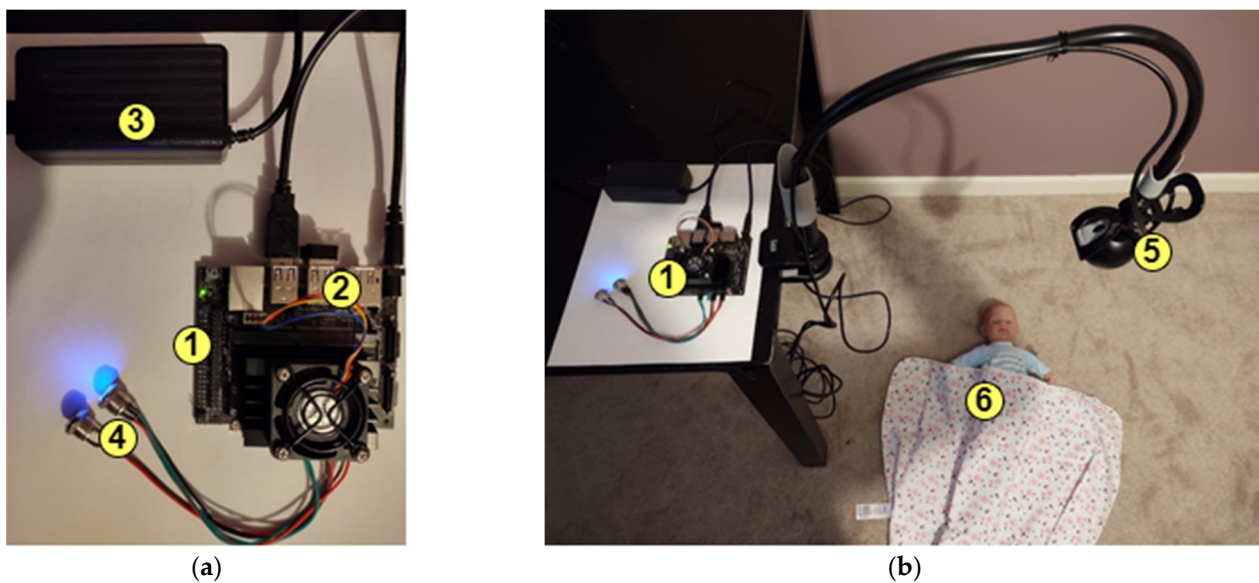


Figure 13. Photograph of the smart baby monitor device: (a) closeup view of the device—(1) Jetson Nano microcontroller board, (2) USB Wi-Fi dongle, (3) DC power adaptor, (4) power and reset switches; (b) birds eye view of the setup—(5) camera with night vision capability, (6) doll with blanket.

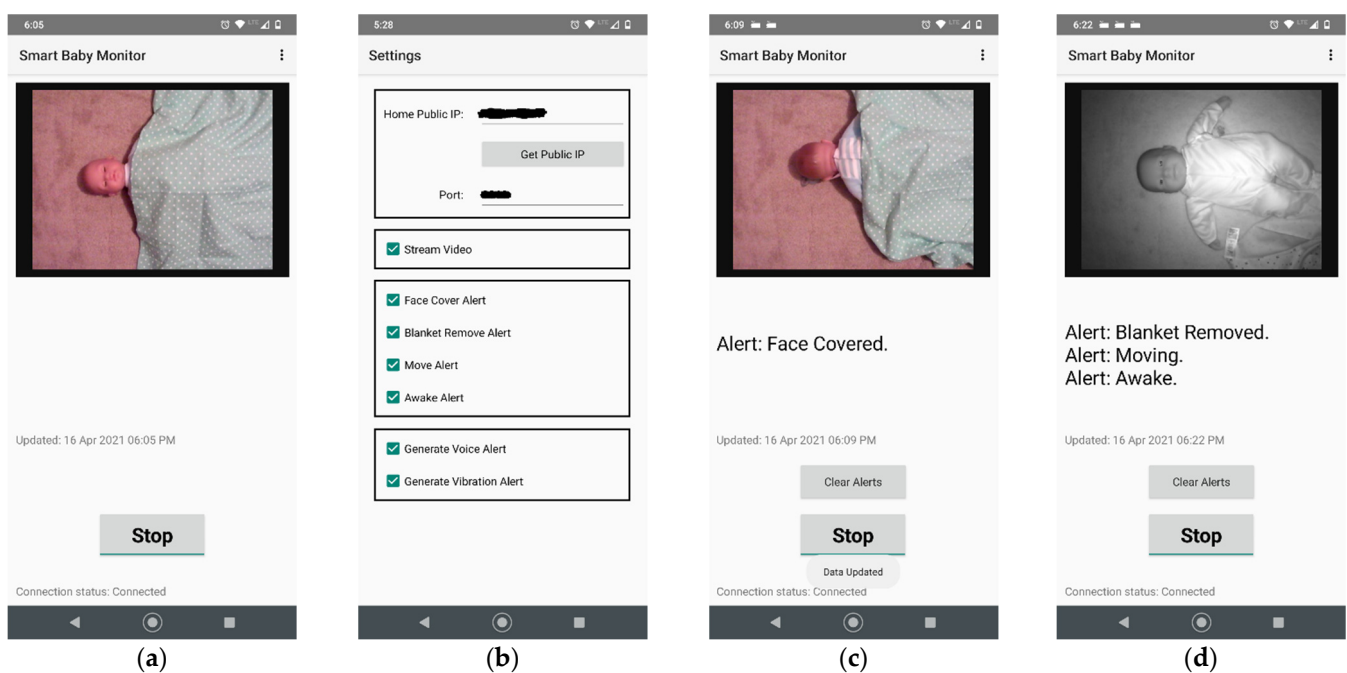


Figure 14. Screenshots of the smartphone app: (a) The first screen of the app showing live video stream, last status update date and time, start/stop toggle button, and connection status; (b) settings window for configuring the device's public IP and port, video stream, and detection alerts; (c) voice and vibration alert generated when the baby's face is covered; (d) alert generated in night light condition when the baby threw off the blanket, frequently moving and eye opened due to awake.

Table 1 shows the latency for detecting body parts for the face covered and blanket removed detection, moving detection, awake detection, and when all of these detections are enabled—for video streaming enabled and disabled cases. Here, it is seen that the detection times are fast (much less than a second)—due to the implementation of NVIDIA-TensorRT [42,43]. Though video streaming is performed using a separate thread, it is seen that the detection latencies are slightly less when video streaming is turned off. After the

detection, the proposed device sends a notification using FCM to the smartphone. The notification generally reaches within a second.

Table 1. Latency of detection algorithms on the Jetson Nano device.

Video Streaming	Detection	Latency (Second)
Yes	Body parts (for face covered and blanket removed)	0.1096
	Moving	0.0001
	Awake	0.0699
	All (Body parts + Moving + Awake)	0.1821
No	Body parts (for face covered and blanket removed)	0.1091
	Moving	0.0001
	Awake	0.6820
	All (Body parts + Moving + Awake)	0.1807

Along with baby dolls, the proposed detection methods were also applied on some baby images available online, as shown in Figure 15. Here, it is seen that the proposed methods can successfully detect the no alert situation as in Figure 15a, the face covered and blanket removed condition as in Figure 15b, the only blanket removed condition as in Figure 15c,d, the eye closed condition as in Figure 15e, and the awake condition as in Figure 15f. Frequent moving detection does not need human pose or face detection; thus, its results do not depend on whether doll or real baby is used.

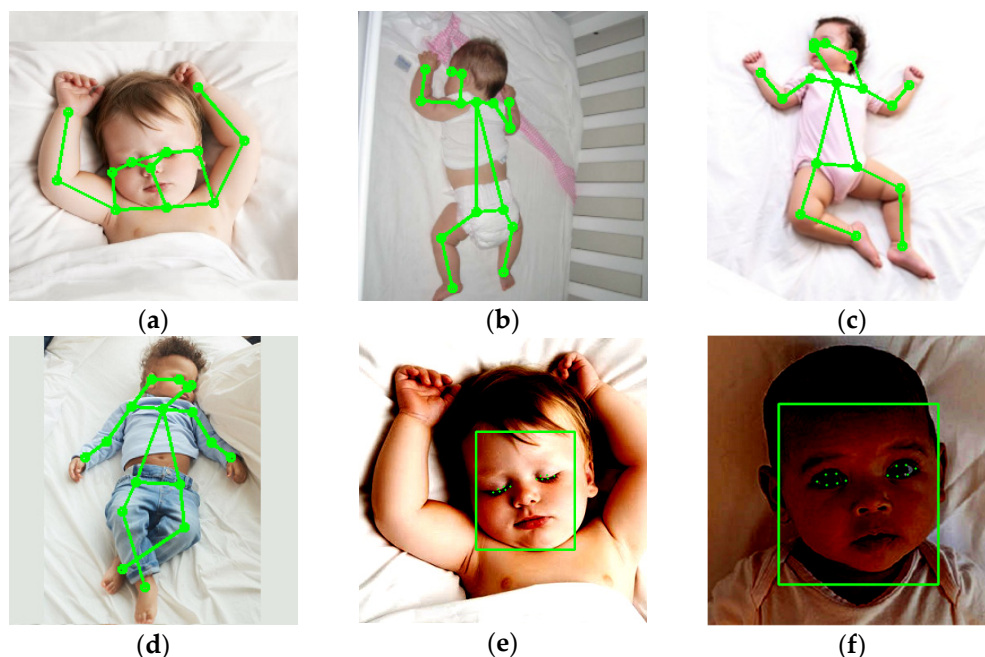


Figure 15. Detection methods applied on real baby images: (a) baby sleeping on the back with the blanket on—no alert; (b) baby sleeping on stomach causing nose undetected—face covered alert. Hip, knee, and ankle are visible—blanket removed alert; (c,d) baby sleeping on the back and nose is visible. Hip, knee, and ankle are visible—blanket removed alert; (e) baby sleeping, showing eye landmarks (EAR = 0.19); (f) baby awake, showing eye landmarks (EAR = 0.35)—baby awake alert.

4. Discussion

Table 2 shows the comparison of the proposed work with other works. To date, there is no baby monitor work found in the literature that can detect a thrown off blanket, or being awake due to eyes being opened. This research focused on detecting alarmable

situations from image data only. Implementing detections from audio data and other wearable sensors is planned for the future.

To detect the nose for face covered detection, the body parts detection method as described in [22,23] are used. Other possible nose detection methods could be the MTCNN [33] and the facial landmark detector in [34]. One problem with [34] is that if any of the facial landmarks are covered, then all the other facial landmarks cannot be detected. As the body parts detection method can be used for both face covered and blanket removed detection, this method is preferred over other options.

The logic definition of blanket removed detection, *isBlanketRemoved*, as shown in Figure 3, can be made configurable by the user using the smartphone. Users may configure logical OR/AND among the visibleness of hip, knee, and ankle according to their needs as simply OR-ing these body parts may not be suitable for everyone's needs. During the experiment, it was found that sometimes, the detections of some body parts are missed in a few frames—especially during night light conditions. This detection fluctuation is common in most object detection algorithms. Training the model with more grayscale baby images might improve the detections in night conditions.

Frequent moving is defined as if there is at least one transient motion in three consecutive blocks of time. The duration of a single block of time and the number of consecutive blocks can be made configurable by the smartphone app; thus, the user can choose these values according to their needs.

For awake detection, the MTCNN [33] face detector is used. One limitation of the available face detectors is that sometimes, it misses the face if it is not aligned vertically. Another option is to use the Haar cascade [47–49] face detector, which runs fast. However, it was found during the experiment that it misses the face often, especially if the image contains a side face and in low light conditions. It is also possible to combine frequent moving and awake detection using logical AND—so the user will be notified when the baby is frequently moving when awake only.

Table 2. Comparison with other works.

Work	Motorola [9]	Infant Optics [10]	Nanit [11]	Lollipop [12]	Cubo Ai [49]	Proposed
Live Video	Yes	Yes	Yes	Yes	Yes	Yes
Boundary	No	No	No	Yes	Yes	No
Cross Detection	No	No	No	Yes	Yes	No
Cry detection	No	No	No	Yes	Yes	No
Breathing Monitoring	No	No	Yes	No	No	No
Face Covered	No	No	No	No	Yes	Yes
Detection	No	No	No	No	No	Yes
Blanket Removed Detection	No	No	Yes	No	No	Yes
Frequent Moving Detection	No	No	No	No	No	Yes
Awake Detection from Eye	No	No	No	No	No	Yes

In this work, the focus is given to detections from image data only. It is planned to process audio data and detect cry sounds in the future. Another future work is to use a thermal camera to read the baby's body temperature and generate alerts based on those data. Privacy and security are a concern for this type of device. To make the system more secure, it is planned to utilize POST requests instead of GET requests and also to encrypt the data.

The models such as Densenet121 for pose detection, the MTCNN for face detection, and regression trees for face landmark detections are trained with human images from datasets such as COCO [25], 300-W [37], etc. As these datasets contain human images of various postures having complex backgrounds, it can be expected that the models trained with these images will work with real humans and infants. The experiments carried out with the online baby images in Figure 14 indicate its applicability with real subjects.

However, new challenges might be faced when it is tested on real subjects such as baby sucking thumbnail while sleeping, only side face visible, alignment issues, etc. To solve these problems, a new dataset of baby images having complex sleeping postures may need to be developed, the data labelled, and then the models retrained using transfer learning [50] with the new dataset. It is planned to test the system with real baby subjects in the future after institutional review board (IRB) approval.

5. Conclusions

In this paper, an intelligent baby monitor device is developed that automatically detects face covered, blanket removed, frequent moving, and awake using deep learning and image processing algorithms. The algorithms are implemented in a microcontroller-based system interfaced with a camera and results show that they run successfully in real-time with low latency. The device implements an HTTP server and sends alerts to the caregiver's smartphone using cloud messaging whenever one or more of these unwanted situations occur. Though some of the recently available baby monitors implement some automatic detection features, the proposed work contributes a new study for detecting the blanket removed from the baby's body and awake detection by analyzing the eye images. Applying this new and useful knowledge in baby monitors can imply more peace of mind for the caregivers. The prototype of the baby monitor device and the smartphone app has been tested successfully with images and dolls of different races in both day and night light conditions.

Author Contributions: Conceptualization, methodology, software, validation, analysis, investigation, resources, writing—original draft preparation, writing—review and editing, visualization, supervision, project administration, funding acquisition, by T.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Faculty Research Fellowship (FRF) award of Eastern Michigan University.

Acknowledgments: The author would like to thank Aditya Annavarapu for collecting sleeping baby images.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. U.S. Department of Health & Human Services. Sudden Unexpected Infant Death and Sudden Infant Death Syndrome. Available online: <https://www.cdc.gov/sids/data.htm/> (accessed on 22 February 2021).
2. The Bump. Is It Okay for Babies to Sleep on Their Stomach? Available online: <https://www.thebump.com/a/baby-sleeps-on-stomach/> (accessed on 22 February 2021).
3. Illinois Department of Public Health. SIDS Fact Sheet. Available online: http://www.idph.state.il.us/sids/sids_factsheet.htm (accessed on 22 February 2021).
4. How Can I Keep My Baby Warm at Night? Available online: <https://www.babycenter.in/x542042/how-can-i-keep-my-baby-warm-at-night> (accessed on 4 March 2021).
5. 5 Reasons Why Your Newborn Isn't Sleeping at Night. Available online: <https://www.healthline.com/health/parenting/newborn-not-sleeping> (accessed on 4 March 2021).
6. Nordqvist, C. New Parents Have 6 Months Sleep Deficit During First 24 Months of Baby's Life. *Medical News Today*, 25 July 2010. Available online: <https://www.medicalnewstoday.com/articles/195821.php/> (accessed on 22 February 2021).
7. Belyh, A. The Future of Human Work is Imagination, Creativity, and Strategy. *CLEVERISM*, 25 September 2019. Available online: <https://www.cleverism.com/future-of-human-work-is-imagination-creativity-and-strategy/> (accessed on 22 February 2021).
8. Janssen, C.P.; Donker, S.F.; Brumby, D.P.; Kun, A.L. History and future of human-automation interaction. *Int. J. Hum. Comput. Stud.* **2019**, *131*, 99–107. [CrossRef]
9. MOTOROLA. MBP36XL Baby Monitor. Available online: <https://www.motorola.com/us/motorola-mbp36xl-2-5-portable-video-baby-monitor-with-2-cameras/p> (accessed on 3 June 2021).
10. Infant Optics. DXR-8 Video Baby Monitor. Available online: <https://www.infantoptics.com/dxr-8/> (accessed on 3 June 2021).
11. Nanit Pro Smart Baby Monitor. Available online: <https://www.nanit.com/products/nanit-pro-complete-monitoring-system?mount=wall-mount> (accessed on 3 June 2021).
12. Lollipop Baby Monitor with True Crying Detection. Available online: <https://www.lollipop.camera/> (accessed on 22 February 2021).

13. Cubo Ai Smart Baby Monitor. Available online: <https://us.getcubo.com> (accessed on 1 June 2021).
14. Day and Night Vision USB Camera. Available online: <https://www.amazon.com/gp/product/B00VFLWOC0> (accessed on 3 March 2021).
15. Jetson Nano Developer Kit. Available online: <https://developer.nvidia.com/embedded/jetson-nano-developer-kit> (accessed on 3 March 2021).
16. 15" Realistic Soft Body Baby Doll with Open/Close Eyes. Available online: <https://www.amazon.com/gp/product/B00OMVPX0K> (accessed on 3 March 2021).
17. Asian 20-Inch Large Soft Body Baby Doll. Available online: <https://www.amazon.com/JC-Toys-Asian-Baby-20-inch/dp/B074N42T3S> (accessed on 3 March 2021).
18. African American 20-Inch Large Soft Body Baby Doll. Available online: <https://www.amazon.com/gp/product/B01MS9SY16> (accessed on 3 March 2021).
19. Caucasian 20-Inch Large Soft Body Baby Doll. Available online: <https://www.amazon.com/JC-Toys-Baby-20-inch-Soft/dp/B074JL7MYM> (accessed on 3 March 2021).
20. Hispanic 20-Inch Large Soft Body Baby Doll. Available online: <https://www.amazon.com/JC-Toys-Hispanic-20-inch-Purple/dp/B074N4C6J7> (accessed on 3 March 2021).
21. NVIDIA AI IOT TRT Pose. Available online: https://github.com/NVIDIA-AI-IOT/trt_pose (accessed on 8 March 2021).
22. Xiao, B.; Wu, H.; Wei, Y. Simple baselines for human pose estimation and tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
23. Cao, Z.; Simon, T.; Wei, S.; Sheikh, Y. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1302–1310.
24. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
25. COCO Dataset. Available online: <https://cocodataset.org> (accessed on 8 March 2021).
26. Faber, M. How to Analyze the COCO Dataset for Pose Estimation. Available online: <https://towardsdatascience.com/how-to-analyze-the-coco-dataset-for-pose-estimation-7296e2ffb12e> (accessed on 8 March 2021).
27. Rosebrock. Available online: <https://www.pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/> (accessed on 18 March 2021).
28. Opencv-Motion-Detector. Available online: <https://github.com/methylDragon/opencv-motion-detector> (accessed on 18 March 2021).
29. Gedraite, E.S.; Hadad, M. Investigation on the effect of a Gaussian Blur in image filtering and segmentation. In Proceedings of the ELMAR-2011, Zadar, Croatia, 14–16 September 2011; pp. 393–396.
30. Image Thresholding. Available online: https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html (accessed on 18 March 2021).
31. Dilation. Available online: https://docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html (accessed on 18 March 2021).
32. Contours. Available online: https://docs.opencv.org/master/d4/d73/tutorial_py_contours_begin.html (accessed on 18 March 2021).
33. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [CrossRef]
34. Kazemi, V.; Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1867–1874.
35. Dlib Shape Predictor. Available online: http://dlib.net/imaging.html#shape_predictor (accessed on 1 April 2021).
36. Rosebrock. Facial Landmarks with Dlib, OpenCV, and Python. Available online: <https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python> (accessed on 1 April 2021).
37. Facial Point Annotations. Available online: <https://ibug.doc.ic.ac.uk/resources/facial-point-annotations> (accessed on 1 April 2021).
38. Edimax 2-in-1 WiFi and Bluetooth 4.0 Adapter. Available online: <https://www.sparkfun.com/products/15449> (accessed on 7 April 2021).
39. Flask Server. Available online: <https://flask.palletsprojects.com/en/1.1.x> (accessed on 7 April 2021).
40. Yatritrivedi; Fitzpatrick, J. How to Forward Ports on Your Router. Available online: <https://www.howtogeek.com/66214/how-to-forward-ports-on-your-router/> (accessed on 7 April 2021).
41. Thread-Based Parallelism. Available online: <https://docs.python.org/3/library/threading.html> (accessed on 7 April 2021).
42. NVIDIA-TensorRT. Available online: <https://developer.nvidia.com/tensorrt> (accessed on 1 March 2021).
43. TensorRT Demos. Available online: https://github.com/jkjung-avt/tensorrt_demos (accessed on 1 March 2021).
44. Firebase Cloud Messaging. Available online: <https://firebase.google.com/docs/cloud-messaging> (accessed on 7 April 2021).
45. WebView. Available online: <https://developer.android.com/guide/webapps/webview> (accessed on 13 April 2021).
46. Jetson's Tegrastats Utility. Available online: <https://docs.nvidia.com/jetson/14t/index.html#page/Tegra%20Linux%20Driver%20Package%20Development%20Guide/AppendixTegraStats.html> (accessed on 21 April 2021).
47. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA, 8–14 December 2001.

-
48. Viola, P.; Jones, M. Robust real-time face detection. *Int. J. Comput. Vis.* **2004**, *57*, 137–154. [[CrossRef](#)]
 49. Lienhart, R.; Maydt, J. An extended set of Haar-like features for rapid object detection. In Proceedings of the International Conference on Image Processing, Rochester, NY, USA, 22–25 September 2002.
 50. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A Survey on Deep Transfer Learning. In *Artificial Neural Networks and Machine Learning—ICANN 2018*; Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I., Eds.; ICANN 2018; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; Volume 11141. [[CrossRef](#)]