sustainability

MDPI

Article

Implementation of Automated Baby Monitoring: CCBeBe

Soohyun Choi [1], **Songho Yun** [2] and **Byeongtae Ahn** [3,*]

[1] School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, Korea; ff152@gist.ac.kr

[2] School of Electronics and Communication Engineering, Chonnam National University, Gwangju 61186, Korea; 134330@jnu.ac.kr

[3] Liberal & Arts College, Anyang University, Anyang 14028, Korea

* Correspondence: ahnbt@anyang.ac.kr

Received: 26 February 2020; Accepted: 20 March 2020; Published: 23 March 2020

Abstract: An automated baby monitoring service CCBeBe (CCtv Bebe) monitors infants' lying posture and crying based on AI and provides parents-to-baby video streaming and voice transmission. Besides, parents can get a three-minute daily video diary made by detecting the baby's emotion such as happiness. These main features are based on OpenPose, EfficientNet, WebRTC, and Facial-Expression-Recognition.Pytorch. The service is integrated into an Android application and works on two paired smartphones, with lowered hardware dependence.

Keywords: baby monitoring; automated monitoring; emergency alert; big data

1. Introduction

It is important to watch babies, but keeping eyes on them all the time is hard for parents. Some parents use general CCTV to monitor babies, but it cannot notify parents of emergencies. Also, even if some wearable devices are made to send alert messages to parents, many parents worry about electromagnetic waves from the devices. Moreover, with these systems, users need to buy special expensive hardware. Inspired by these problems, we designed an automated baby monitoring service that alerts emergencies such as crying and rolling over, with lowered dependency on hardware. The main features of the service include: (1) monitoring dangerous lying posture based on OpenPose; (2) detecting infant cry using EfficientNet; (3) parents-to-baby video streaming and voice transmission; (4) recognizing the baby's emotion; and (5) notifying of detected events.

2. Related Work

2.1. OpenPose

OpenPose is an open-source real-time system for detecting the 2D pose of humans by finding body, foot, hand, and facial keypoints [1]. This library provides high-quality estimation, which is enough to perform in real-time. To detect the baby's body parts accurately and to infer immediately whether the posture of the baby is dangerous, we chose OpenPose and transformed it to run on TensorFlow. Figure 1 shows the detected skeletons of the baby sleeping in safe and dangerous positions.

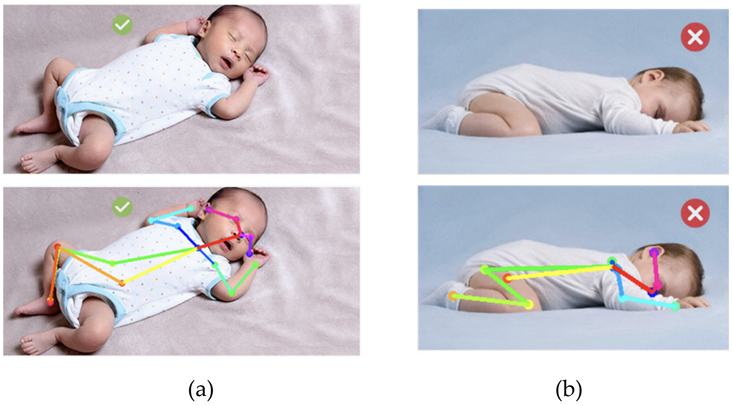
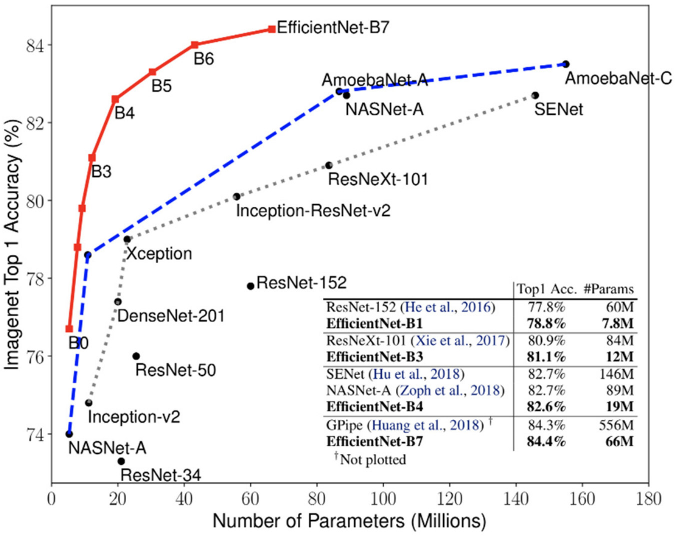


Figure 1. Body, foot, hand, and facial keypoints of a baby sleeping in safe (**a**) and dangerous (**b**) positions detected by OpenPose.

2.2. *EfficientNet*

EfficientNet is a highly effective ConvNet in terms of accuracy and efficiency [2]. By a compound scaling method that balanced depth, width, and resolution of the network, its performance surpassed previous ConvNets (Figure 2). To identify infant crying, we applied EffientNet via transfer learning with spectrogram images of crying sounds and non-crying sounds on various noises.



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**78.8%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.1%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.6%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

†Not plotted

Figure 2. High performance of EfficientNet compared to previous ConvNets [2].

2.3. *Firebase Cloud Messaging*

Firebase Cloud Messaging (FCM) is a cross-platform messaging solution that enables messages to be sent stably and for free [3]. If a notification message is sent to the client application such as iOS, Android, or Web, the user can notice that the data is available to sync. FCM was applied in this design to send alerts and sync data to the client application when a possible hazard to the baby is detected or a daily video diary is newly created.

2.4. *WebRTC*

WebRTC provides Real-Time Communications (RTC) of audios and videos between browsers and mobile applications without extra software [4]. WebRTC APIs are simple, yet they are more effective than Real-Time Messaging Protocol (RTMP) or Real-Time Streaming Protocol (RTSP) in the aspects of Quality of Experience (QoE) and Quality of Service (QoS) for Android applications [5]. Also, RTSP has network security issues related to the UDP firewall and does not support outdated legacy mobile devices [6]. Regarding all these aspects, we adapted WebRTC so that our service can be run effectively and securely on unused smartphones in a caregivers' house. We utilized WebRTC to develop real-time video monitoring and voice transmission and to figure out dangerous postures and baby cries from the media.

2.5. *Face_recognition*

Face_recognition is the simplest library used to recognize faces in pictures. It is based on the face recognition tool of dlib C++ library, developed by deep learning, and has a recorded accuracy of more than 99% on the Labeled Faces in the Wild benchmark [7]. We used this library to extract the face of the baby and then classify its emotion.

2.6. *Facial-Expression-Recognition.Pytorch*

The Facial-Expression-Recognition.Pytorch model recognizes facial expressions on images based on CNN and runs on Pytorch, trained by FER2013 and CK+ dataset. It classifies emotions from the faces into "angry," "disgusted," "fear," "happy," "sad," "surprised," and "neutral" [8]. We applied this model on the images of the face extracted by Face_recognition and then captured the baby's emotion, described in Figure 3. This emotional information is also used to create a daily video diary.

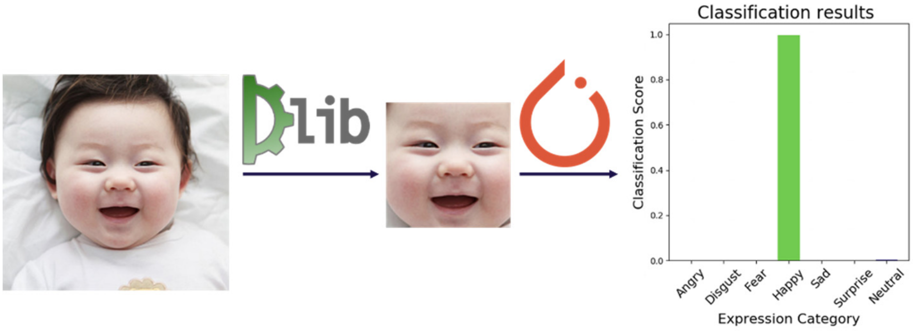


Figure 3. Classification results of input demo image by Face_recognition and Facial-Expression-Recognition.Pytorch.

3. Case Studies

Before designing the automated baby monitoring service, we researched domestic and international baby monitoring services similar to ours.

For domestic services, BeneCam delivers live video of a newborn baby to parents through IP cameras in postpartum care centers (Figure 4). Parents or families can see their babies live and take pictures or videos on the mobile application [9]. However, it cannot recognize dangerous events and automatically send alerts. Baby Crying Notifier monitors baby crying and sends a call to another registered phone [10] (Figure 5). It helps parents to respond immediately, but it cannot monitor the posture of the baby and works only if the phones are registered by phone plans.

배내캠은 IP카메라를 통해 언제 어디서든
신생아와 가족들을 연결해주는 서비스입니다.
아기와 나의
24시간
소중한 우리아기 배내캠과 함께하세요.

Figure 4. Benecam [9].

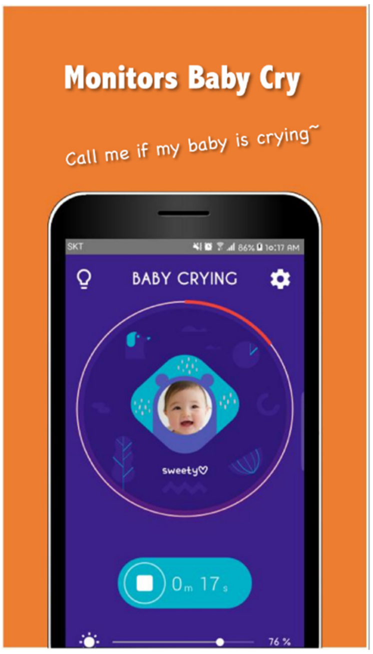


Figure 5. Baby Crying Notifier [10].

In the United States, Cocoon Cam catches the breathing rate and the movement of the baby during sleep with a camera provided and without wearable devices [11] (Figure 6). Yet, it cannot recognize the emotion of the baby and is highly expensive due to the camera. As shown in Figure 7, Owlet Cam also provides a camera to monitor the baby with high-resolution, but Smart Sock should be connected to keep track of heart rate and oxygen saturation of the baby, as well as sleep condition [12]. Even though the wearable device can provide accurate biological information, it can disturb the baby's quality of sleep.

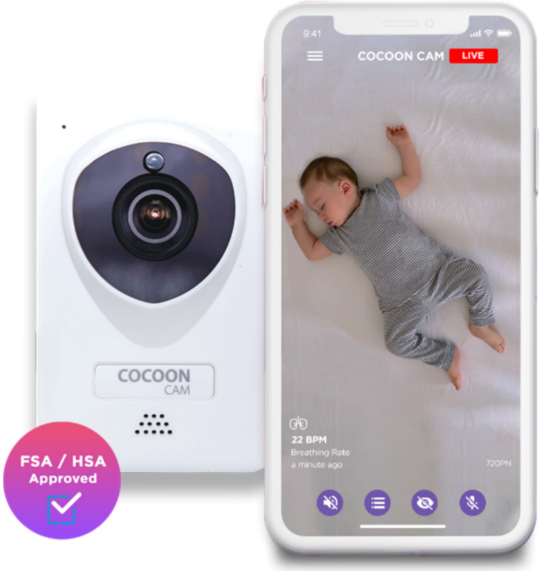


Figure 6. Cocoon Cam [11].

Figure 7. Owlet Cam and Smart Sock [12].

4. Design of CCBeBe

The automated baby monitoring service CCBeBe is focused on software. It runs on Android OS as a form of a mobile application. The key functions of the service are monitoring and detecting dangerous lying posture and crying, video streaming from parents to a baby with voice transmission and creating a three-minute daily video diary by recognizing emotions from the faces of the baby. These features work on two Android smartphones which are paired by tokens. A camera on a baby-side smartphone monitors the baby's status and detects possible hazards such as "no face detected" or "crying". If found, an alert message with the baby's image is automatically sent to a parent-side smartphone by Firebase Cloud Messaging (FCM). Figure 8 is a system diagram of CCBeBe.

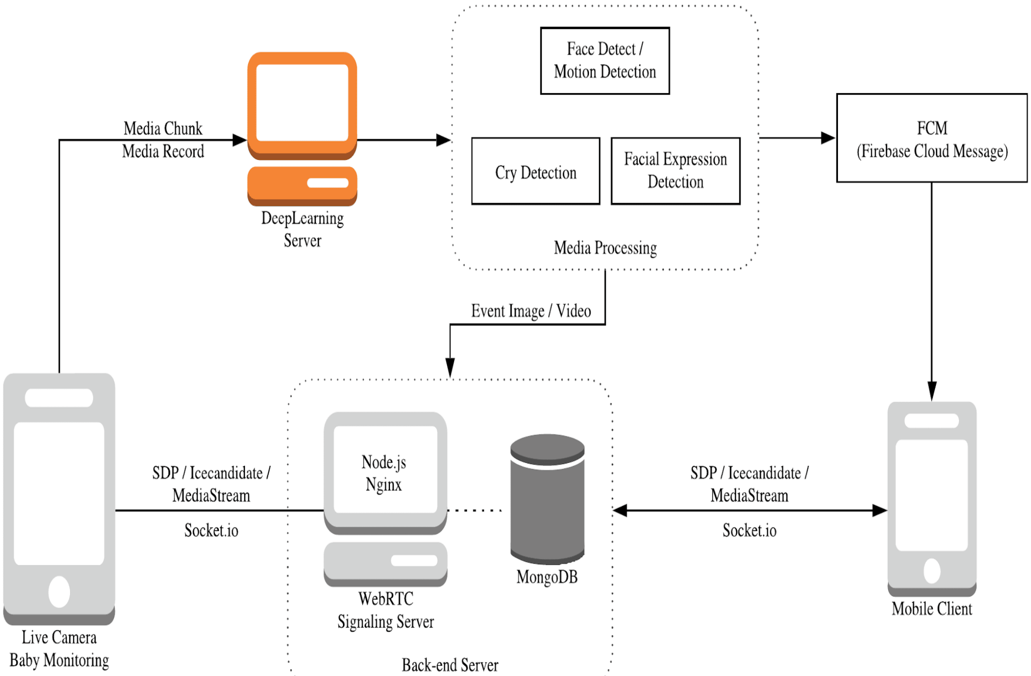


Figure 8. Automated baby monitoring design.

The service mainly monitors whether the face of the baby is detected or not. Powered by OpenPose, the pose detector model runs on the Chrome web browser on Node.js backend server with user data received. OpenPose model is applied to the live media stream by loading Tensorflow.js and ML5.js.

Then, the skeleton of the baby's posture detected is rendered on the browser by P5.js. If there is no face detected, the parent-side smartphone receives an alert message, including the image. Sending the alert, six-second-long videos recorded three seconds before and after the occurrence of the event, respectively, are also saved on the server, so that parents can check what occurred during the event on the timeline of the application.

The infant cry detector also warns if the baby is in an emergency. Input audio from the baby-side smartphone is processed in the deep learning server. The audio is periodically saved as a seven-second Waveform Audio File Format (WAV) and then converted into a mel-spectrogram, which contains characteristic information of the audio signal. The mel-spectrogram is entered into EfficientNet which is pre-trained by baby crying and environmental sounds and classified into "crying" and "not crying". If a cry is recognized, it pushes notification messages to parents who use the baby-side smartphone and the six-second video is saved on the backend server as well.

Besides the emergency warning features, the function of live video streaming and voice transmission allows parents to watch their baby and send their voice to the baby live, making sure that the baby is safe. It is implemented by Socket.IO and WebRTC API on the backend signaling server. The server accesses to the live camera, which monitors the baby, and captures the audio and video so that share the information with the mobile client.

In addition to the features above, a 3-minute-long video diary of the baby is created and pushed to users daily by FCM. The video diary is created by detecting facial expressions of the baby by Face-recognition library and Facial-Expression-Recognition.Pytorch model. If the baby is recognized "angry," "disgust," "fear," "happy," "sad," "surprise" or "neutral" for 2–3 seconds in the media processing unit in the deep learning server, the images and videos are saved and then concatenated into a video by FFmpeg on the backend server.

5. Implementation of CCBeBe

To connect a parent-side mobile client with a baby-side mobile client, a WebRTC signaling server was built on Amazon Elastic Compute Cloud (EC2) with Ubuntu OS, where Socket.IO, Node.js, Nginx, and MongoDB were used. Implementing WebSocket server on this Node.js server, Session Description Protocol (SDP) and candidate information of Internet Connectivity Establishment configuration (ICE candidate, IceCandidate in WebRTC API) are exchanged. A WebSocket client was built on Web client to send SDP and IceCandidate information to the signaling server. Then, to capture and exchange audio and video streams of the baby with the parent-side mobile client, the getUserMedia() method and RTCPeerConnection interface were used. The image data of the baby is transferred by RTCDataChannel so that the data is secured with Datagram Transport Layer Security (DTLS).

The deep learning server was built on a desktop computer with three NVIDIA GeForce GTX 1080Ti GPUs, running on Ubuntu OS. The baby's posture focused on the face, crying, and emotion are detected on each GPU of the deep learning server. For posture detection, if four or fewer key points are accurately detected by the server (left eye, right eye, nose, left ear, and right ear), it is considered as "no face detected" and the variable noFaceCount increases by 1. If noFaceCount exceeds the threshold (3), the server sends an alert message including the image by FCM to the parent-side mobile client. Figure 9 describes the process on the deep learning server where the emotion of the baby is estimated from capturing the face of the baby and the skeleton of the baby is tracked from the media stream received from the parent-side mobile client. The video clip used for this demonstration is from YouTube.

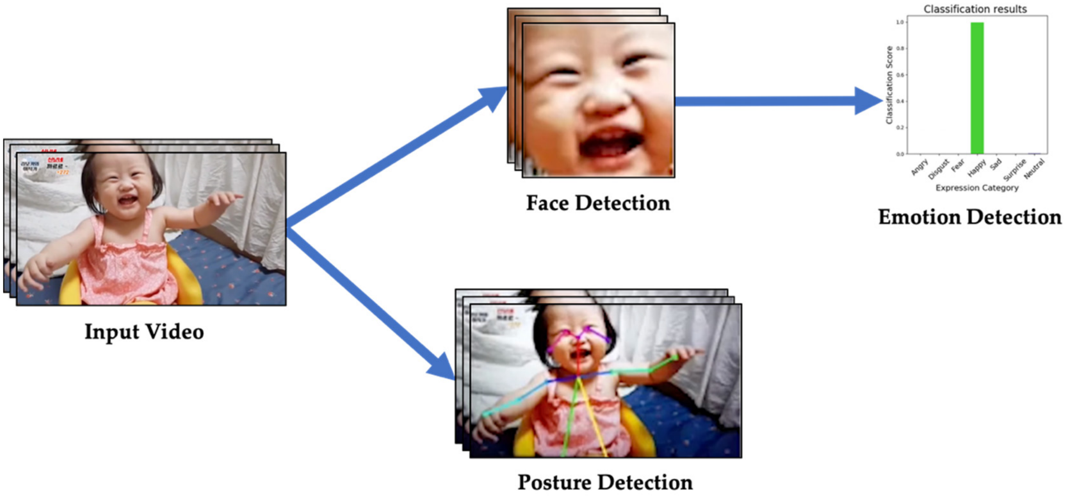


Figure 9. Emotion and skeletons of a baby processed on the server.

Especially for crying detection, we used transfer learning to train the deep neural networks of EfficientNet by mel-spectrograms of 3599 crying sounds of babies("crying") and 3607 environmental sounds("not crying"). The sample sounds are obtained from "donateacry-corpus" and "ESC-10" dataset from "ESC-50: Dataset for Environmental Sound Classification" from GitHub repositories and by trimming and extracting YouTube video clips from AudioSet, then the spectrograms are extracted by LibROSA [13,14]. Mel-spectrogram samples of each class are depicted in Figures 10 and 11. 80% of the datasets were used to train the model and 20% were used to test the model. We trained each EfficientNet architecture with version B0 to B7 built for Keras and the performance of EfficientNet-B3 was the best for our datasets so we adapted it. We added a GlobalMaxPooling2D and a Dense layer to EfficientNet-B3 by using Keras Sequential model. To classify inputs into two classes "crying(1)" and "not crying(0)," we applied the sigmoid activation function on the Dense layer and binary_crossentropy of Keras for loss function. Also, RMSprop optimizer was used to optimize the loss with the learning rate of 0.00002 and batch size was set to 512. The learning process was finished at epoch 219 where the accuracy was stopped to grow. The classification report data was created by classification_report of scikit-learn. For crying detection, the precision of our model was recorded as 0.96. F1 score of crying detection is not much higher, but when we sliced and recognized the spectrogram at intervals of a second, the accuracy was relatively high for some specific patterns of crying. As suggested in Conclusions and Work, we are continuously researching ways to improve accuracy. Details of the classification report and confusion matrix are depicted in Figures 12 and 13, respectively.

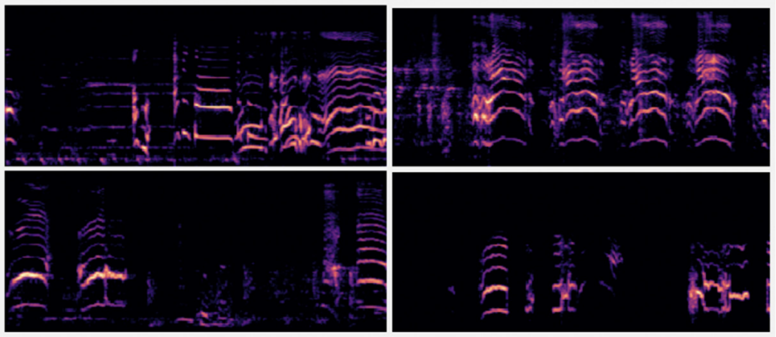


Figure 10. Mel-spectrogram samples of baby crying sounds (x-axis: time, y-axis: frequency).

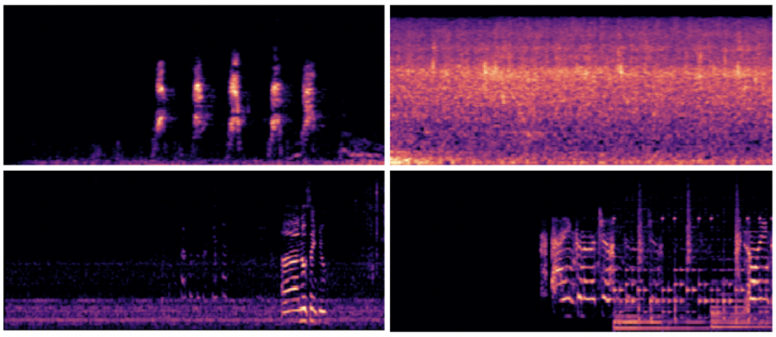


Figure 11. Mel-spectrogram samples of non-crying environmental sounds (dog, rain, music, talk in clockwise).

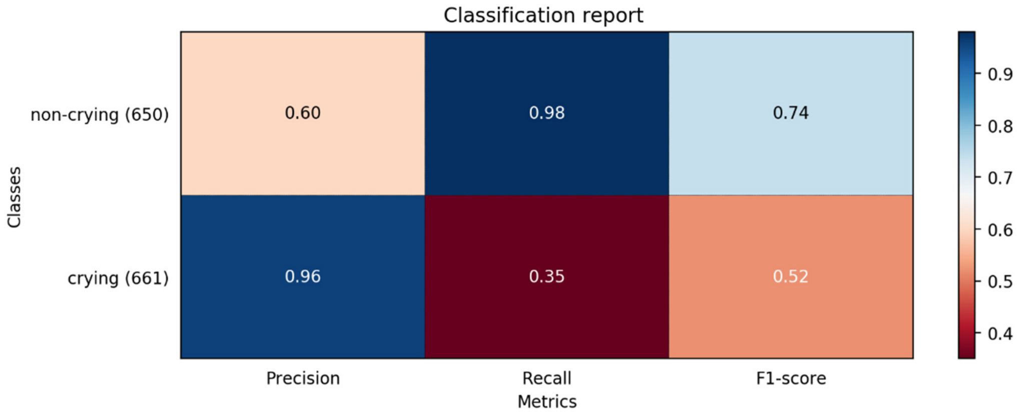


Figure 12. Classification report of the baby crying sound detector.

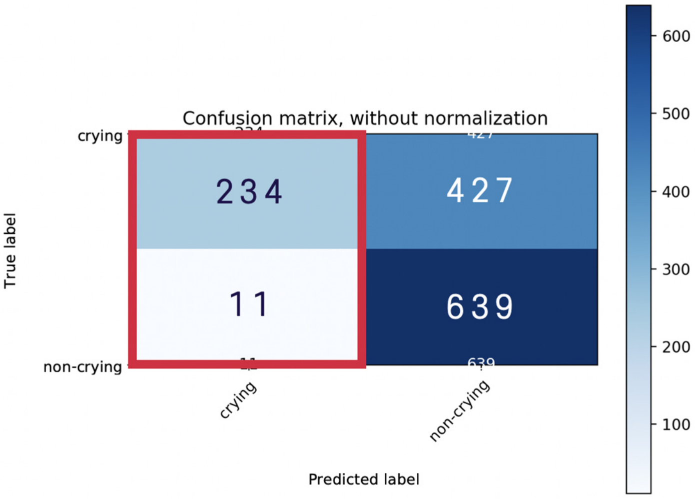
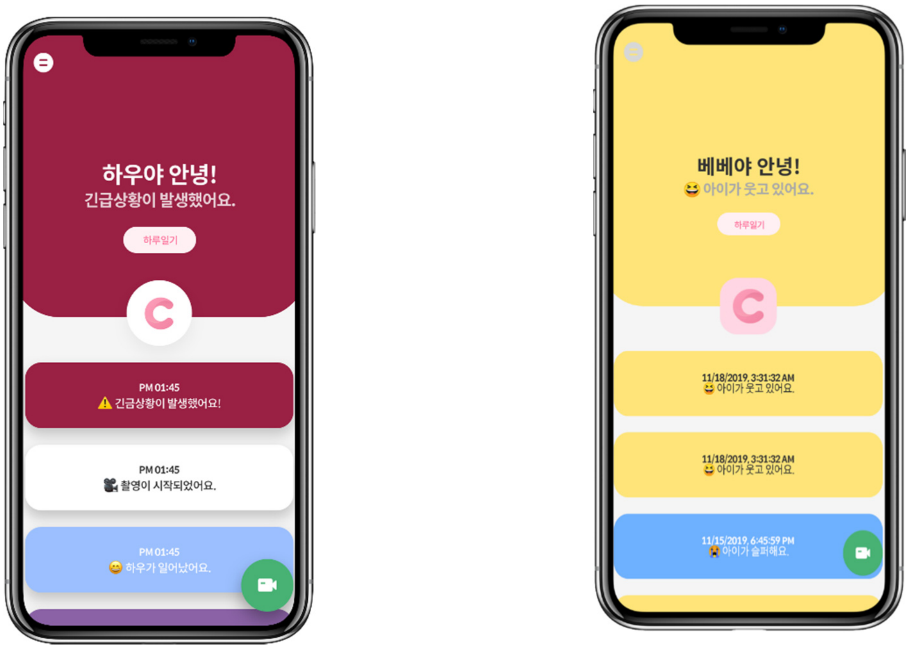


Figure 13. Confusion matrix of the baby crying sound classification with test data.

Our system design was developed as an Android application named CCBeBe. User Interface (UI) was designed for parents to identify events immediately on the main page, using colors and emojis. If there is an emergency, the overall color of the UI turns red and a message with warning sign emoji is notified on the top of the timeline, as described in Figure 14a. Also, if a smiling face of the baby is detected, the upper part of the main page turns yellow, and a message is sent with grinning squinting face emoji, as depicted in Figure 14b.



하우야 안녕!
긴급상황이 발생했어요.

PM 01:45
⚠ 긴급상황이 발생했어요!

PM 01:45
촬영이 시작되었어요.

PM 01:45
😊 하우가 일어났어요.

베베야 안녕!
😆 아이가 웃고 있어요.

11/18/2019, 3:31:32 AM
😆 아이가 웃고 있어요.

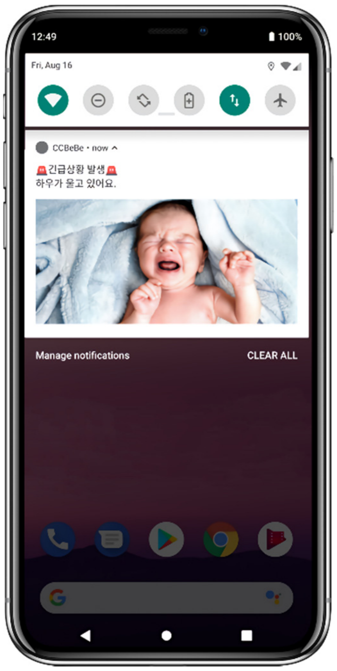
11/18/2019, 3:31:32 AM
😆 아이가 웃고 있어요.

11/15/2019, 6:45:59 PM
😭 아이가 슬퍼해요.

(a) (b)

Figure 14. (**a**). Timeline with latest emergency alert. (**b**). Timeline with latest smiling alert.

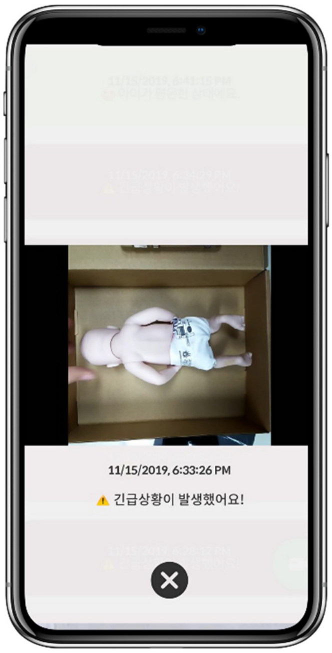
When users receive notification messages, a snapshot of the event is also pushed so that users can figure out the situation and deal with the event immediately (Figure 15). If users click the push message, it turns to the main page of the application. As users click the event bar on the timeline, they can see a six-second-long video, which is a combination of a three-second-long video taken before and after the event respectively (Figure 16). To watch the baby in real-time, clicking the green button below on the parent-side smartphone makes parents see live video of the baby (Figure 17b) streamed from the baby-side smartphone (Figure 17a).



CCBeBe • now

🚨긴급상황 발생🚨
하우가 울고 있어요.

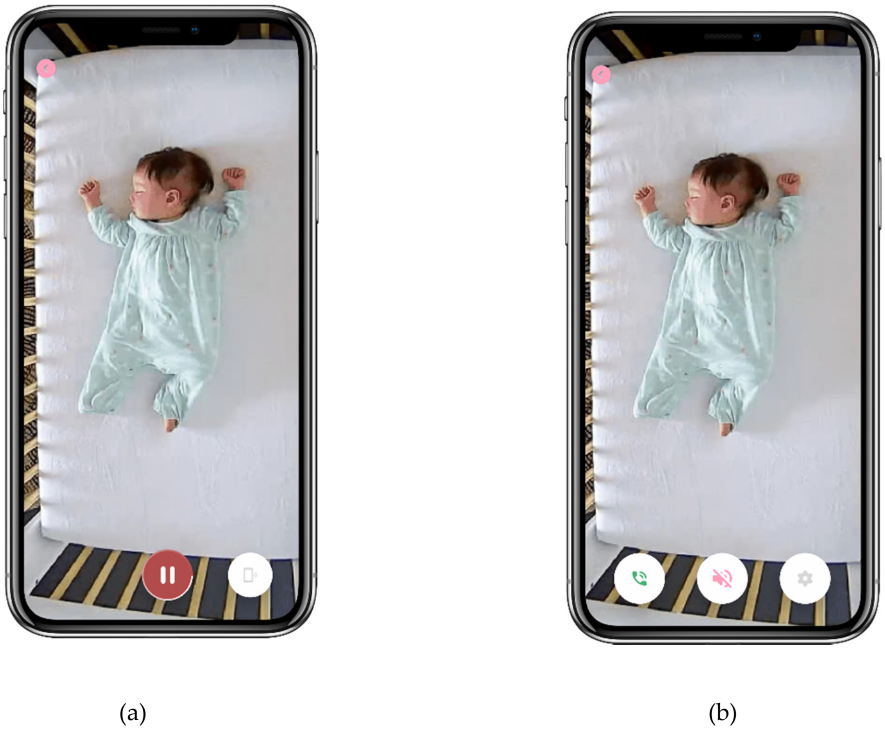
Figure 15. Example of an alert message pushed to a parent-side mobile client by Firebase Cloud Messaging (FCM).



11/15/2019, 6:33:26 PM

⚠ 긴급상황이 발생했어요!

Figure 16. Example of a short video recorded during the event of rolling over.



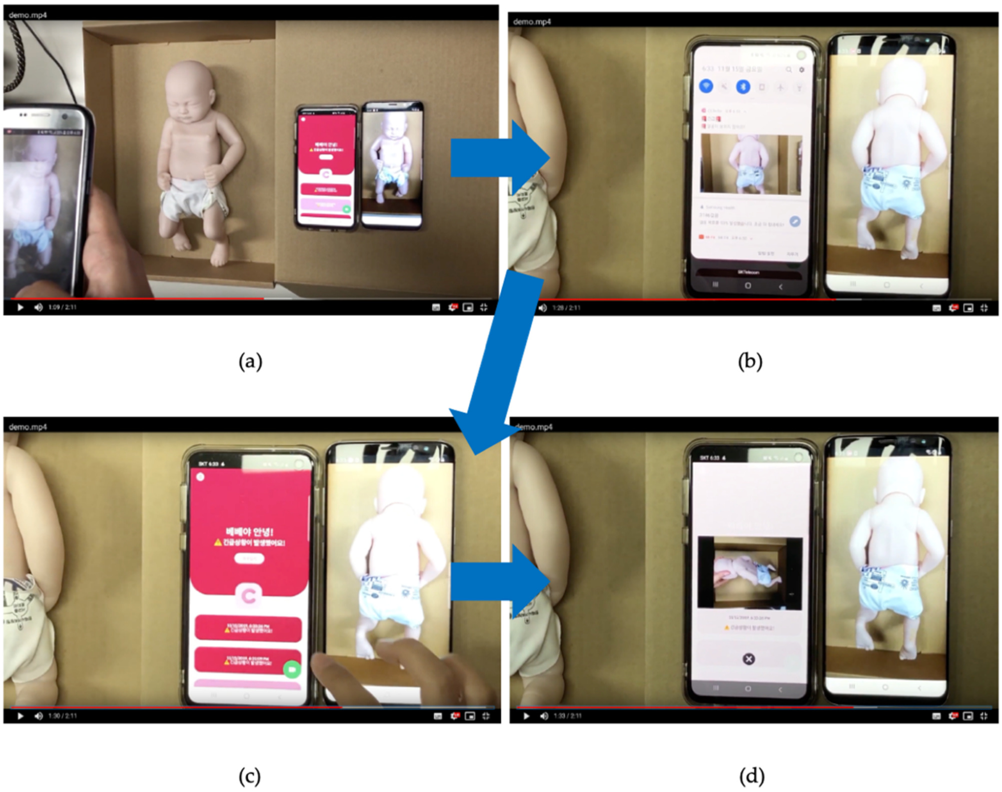
(a)

(b)

Figure 17. (**a**). Baby-side smartphone sending the video of the baby. (**b**). Parent-side smartphone receiving the media.

We tested the whole system using Samsung Galaxy S7, S8, and S10e smartphones. S7 was used as a baby-side smartphone, which films the baby, and S10e was used as a parent-side smartphone, which gets a notification. S8 was additionally used to mirror the screen of the parent-side smartphone (S7) to demonstrate clearly. All these smartphones were logged into our CCBeBe application with the same account. Four scenarios with a baby doll were tested to demonstrate the system of CCBeBe and the function of real-time video streaming and voice transmission was tested before these scenarios. The scenarios below are arranged in order of the functions introduced earlier.

First, we examined the main function, which detects dangerous posture by monitoring the existence of the face of the baby. The baby-side smartphone recorded the baby doll lying straight in the beginning (Figure 18a). Then, we flipped the doll to simulate the rolling over. Since there is no face detected by the system, an alert message "(police car light emoji) Baby's face is invisible!" was pushed to the parent-side smartphone with an image of the baby with flipped posture, and the timeline was updated with the new event (Figure 18b,c). By clicking the event bar, we can see that the video where the posture of the baby doll was changed from the lying straight to flipped back (Figure 18d).



(a)

(b)

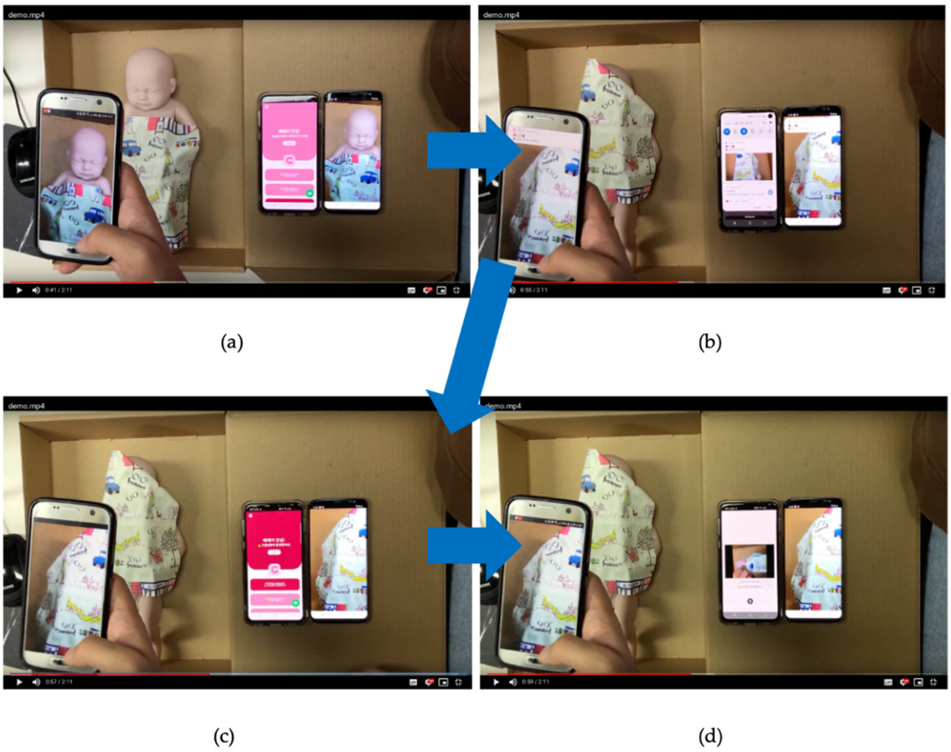
(c)

(d)

Figure 18. (**a**) Recording the baby doll lying straight before rolling over. (**b**) Push message received by parent-side smartphone after the rolling over. (**c**) Timeline updated with a red event bar. (**d**). Playing the recorded video by clicking the event bar.

The second scenario is also for verifying the hazardous position detector by monitoring the baby's face but with a face covered by a blanket. In Figure 19a, the body of the baby doll was covered by the blanket, except the face, then the face was covered a few seconds later. The alert message pushing functioned similarly to the first scenario and the event video was also saved during the covering the baby's face (Figure 19b–d).

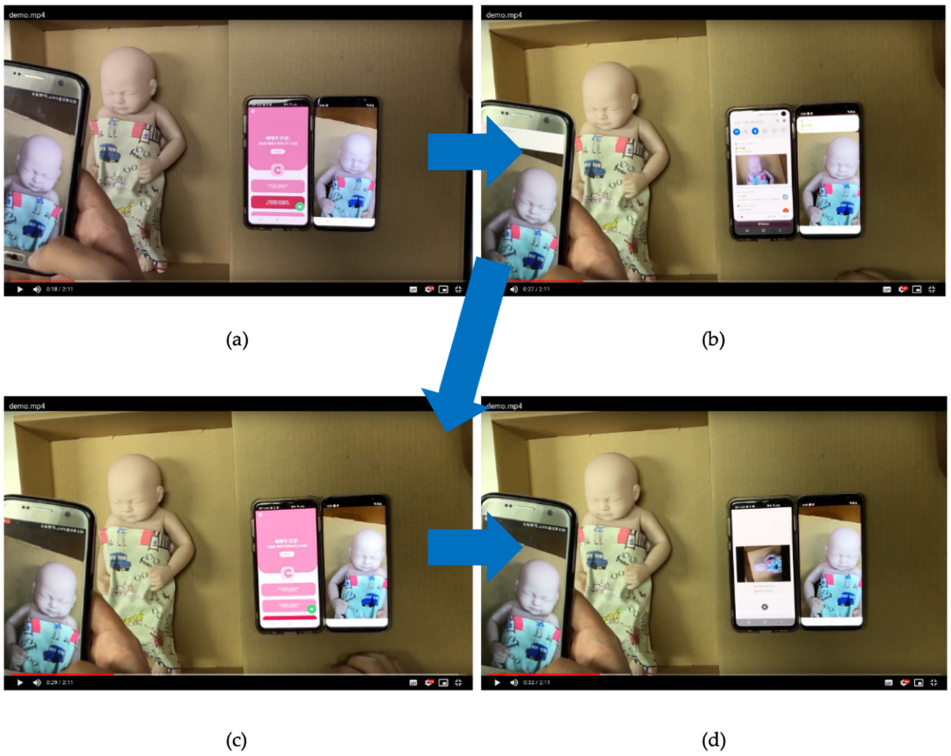
Then, we tested the function of detecting and sending alerts of a baby crying. In Figure 20a, the baby-side smartphone started recording the baby doll and we played a baby crying sound from Youtube video [15,16]. Soon, the parent-side smartphone received a push message "(crying face emoji) Baby has cried." with a picture taken at the event (Figure 20b). Touching the message, we could see the crying event bar created on the top of the main page of our application (Figure 20c). Also, we identified the event by touching the bar and playing a short video, showing the baby doll before and after the crying (Figure 20d).



(a) (b)

(c) (d)

Figure 19. (**a**) Recording the baby doll covered by the blanket except for the face. (**b**) Push message received by parent-side smartphone after the face covered. (**c**) Timeline updated with a red event bar. (**d**) Playing the recorded video by clicking the event bar.



(a) (b)

(c) (d)

Figure 20. (**a**) Recording the baby doll with a baby crying sound played. (**b**) Push message received by parent-side smartphone while crying. (**c**) Timeline updated with a pink event bar. (**d**) Playing the recorded video by clicking the event bar.

Finally, we played a YouTube video of a laughing baby and recorded the screen with the baby-side smartphone (Figure 21a) [17]. The smiling face was captured by the system, as shown in Figure 9, and it was notified as "(grinning squinting face emoji) Baby is smiling." on the parent-side smartphone (Figure 21b,c). The event video was also created so that it was able to check it on the timeline (Figure 21d). Meanwhile, the function of auto-creating a daily video diary could not be implemented yet due to the limited time.

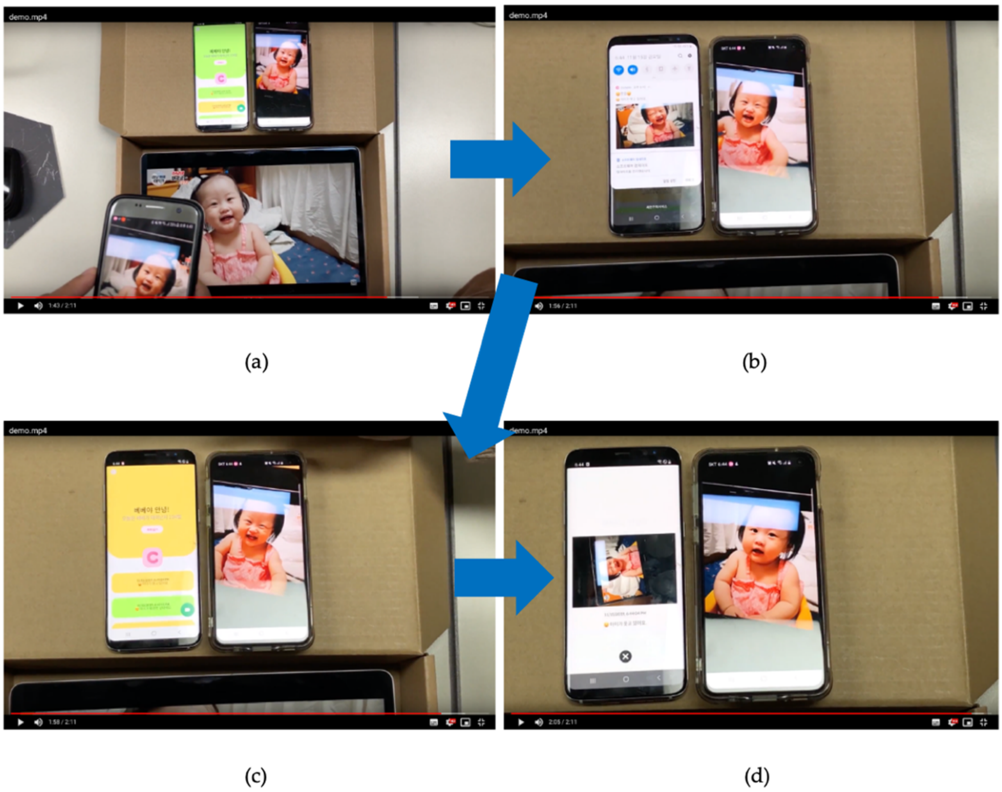


Figure 21. (**a**) Recording the screen of a laptop playing YouTube video of a smiling baby. (**b**) Push message received by parent-side smartphone while smiling. (**c**) Timeline updated with a yellow event bar. (**d**) Playing the recorded video by clicking the event bar.

6. Comparison

In this section, our CCBeBe is compared to other commercialized services and it is summarized in Table 1. We compared the key properties of the services by six entries: (1) real-time video/audio streaming, (2) low hardware dependency, (3) posture detection, (4) crying detection, (5) emotion detection and video diary (automatically creating), and (6) real-time breath monitoring.

Table 1. Comparison of CCBeBe(CCtv Bebe) and other services.

	BeneCam	**Cocoon Cam**	**CCBeBe**
Real-time video/audio streaming	O	O	O
Low hardware dependency	X	X	O
Posture detection	X	O	O
Crying detection	X	O	O
Emotion detection and video diary	X	X	O
Real-time breath monitoring	X	O	X

Basically, BeneCam, Cocoon Cam, and CCBeBe all provide real-time video/audio streaming. However, BeneCam and Cocoon cam have a higher dependency on hardware because the application of these two services runs only with the provided camera. Compare to them, CCBeBe can run on various models of Android smartphones if WebRTC protocols can be used. Regarding safety, our service and Cocoon Cam both monitor the baby's posture and detect crying, but the function of emotion detection and creating daily video diary only exists in our service. Nonetheless, since we excluded wearable devices from our design, there is no function of real-time respiration rate monitoring in our service.

7. Conclusions and Work

Parenting is not easy, especially for parents who raise newborn babies. It is important to monitor them, but in modern society, it is hard to watch babies all the time. Also, existing commercialized services provide real-time streaming, but there are some limitations of hardware dependency by camera or wearable devices. The automated baby monitoring service CCBeBe is designed to alleviate the burden on parents by instantly detecting posture and crying and sending alert messages. Installed on two Android smartphones paired by tokens, each smartphone works as a baby-side monitoring camera and a parent-side monitor that receives push messages. The AI-based service monitors the baby's lying posture and crying by utilizing OpenPose and EfficientNet, which helps parents to cope with the situation on time. By sending an alert by FCM, it saves a short video to identify the exact event detected. Additionally, by capturing the emotions of the baby, a three-minute daily video diary is provided to parents by the service. Since this service is designed to be implemented as a mobile application, it is easy to install and apply to the real world. Plus, eliminating a software-specific camera or a wearable device, it is economic and does not disturb infants during the sleep.

In the future, the accuracy of the posture detector could be increased by considering other key points of the body together with facial points. For crying, long short-term memory (LSTM) could be added for utilizing the time-series feature of the crying datasets and increasing the accuracy. Also, it can be considered to train the emotion detector using the faces of babies instead of adults. Considering real-time features, the service could be light and embedded to run on the smartphone itself and baby cries should be detected on audio streams rather than using the audio file saved on the server. Besides, since a camera on a smartphone is used, it can be hard to cover a wide range of angles depending on the smartphone model.

Author Contributions: Formal analysis and manuscript writing, S.C.; Design and implementation of the research, S.C., S.Y.; Project administration, B.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The researcher claims no conflicts of interest.

References

1. Cao, Z.; Hidalgo Martinez, G.; Simon, T.; Wei SSheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**. [CrossRef] [PubMed]
2. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2019**, arXiv:1905.11946 [cs.LG].
3. Firebase. Available online: https://firebase.google.com/ (accessed on 28 December 2019).
4. WebRTC. WebRTC Home. Available online: https://webrtc.org/ (accessed on 28 December 2019).
5. Santos-González, I.; Rivero-García, A.; Molina-Gil, J.; Caballero-Gil, P. Implementation and Analysis of Real-Time Streaming Protocols. *Sensors* **2017**, *17*, 846. [CrossRef] [PubMed]
6. Ma, K.J.; Bartos, R.; Xu, J.; Nair, R.; Hickey, R.; Lin, I. Method and System for Secure and Reliable Video Streaming with Rate Adaptation. U.S. Patent Application No. 13/483, 18 December 2012.
7. Geitgey, A. GitHub Repository. Face-Recognition. Available online: https://github.com/ageitgey/face_recognition (accessed on 28 December 2019).
8. Wu, J.; GitHub Repository. Facial-Expression-Recognition.Pytorch. 2019. Available online: https://github.com/WuJie1010/Facial-Expression-Recognition.Pytorch (accessed on 28 December 2019).

9. BeneCam. Single Broadcast of My Baby Can't Take My Eyes off from. Available online: http://www.benecam.co.kr/ (accessed on 28 December 2019).
10. YuRimpapa. Baby Crying Notifier (Baby Sleep Monitor & Alarm & Lullaby) Trial. Available online: https://play.google.com/store/apps/details?id=com.jb.babycrying_trial&hl=ko (accessed on 28 December 2019).
11. Cocoon Cam. Cocoon Cam Baby Monitor -HD Video and Breathing Monitoring. Available online: https://cocooncam.com/ (accessed on 28 December 2019).
12. Owlet Cam. Owlet Cam -Owlet Baby Care US. Available online: https://owletcare.com/pages/owlet-cam (accessed on 28 December 2019).
13. Veres, G.; GitHub Repository. Donateacry-Corpus. Available online: https://github.com/gveres/donateacry-corpus (accessed on 28 December 2019).
14. Xu, Z. The Analytics and Applications on Supporting Big Data Framework in Wireless Surveillance Networks. *Int. J. Soc. Humanist. Comput.* **2017**, *2*, 141–149. [CrossRef]
15. wi0915. Baby Crying. Available online: https://youtu.be/qS7nqwGt4-I (accessed on 28 December 2019).
16. You-Shyang, C.; Chien-Ku, L.; Chyuan-Yuh, L.; Huan-Ming, C.; Li-Chuan, W. Electronic Commerce Marketing-Based Social Networks in Evaluating Competitive Advantages Using SORM. *Int. J. Soc. Humanist. Comput.* **2017**, *2*, 261–277.
17. Mocha, K.; Jakka, L. Excited!! Laughing Baby #2. Daily Life of Leokhee (A Smiling Baby). Available online: https://youtu.be/niVVdeB7-Yk (accessed on 28 December 2019).