BIOMETRIC SMART SECURITY SYSTEM WITH CHILD CARE FOR A SMART SOCIETY

Shashika Lokuliyana[1], I.U.Mundigala[2],G.H.Ashan Sanjeewa[3]
Department of Computer Systems and Networking,Department of Information Systems Engineering
Sri Lanka Institute of Information Technology
Malabe,Sri Lanka
[1]shashika.l@sliit.lk, [2]iyanthiupeksha@gmail.com,[3]ashansanjee@gmail.com,

***Abstract*— This research is mainly focused on Infant movement detection and alerting, in order to enhance their security within the home premises. As the first move, the research focuses on the identification of the human and classifying whether an adult or a baby. Then a model was built up in three classifications to identify static and dynamic positions of the infant, through Image Processing and analysis. In order to enhance the accuracy of the custom classifiers an already trained model using 1 million image set was retrained by customized image sets. To present this research as a smart home solution modern technology were used in implementing the close connection between the infant and the parent.**

Keywords*— *Custom classifiers; Frame analysis; Image Processing; Object detection

I. INTRODUCTION

Ensuring the security of children is a major problem parent's face in now days. It has become a great challenge with the busy schedules they have in their day-to-day routine. But oin the other hand physical development is a foundation of learning. Motor skills perform an important aspect in infants and toddler's brain development. So that parents should allow their infants and toddlers to move freely. But to let them move freely around the premises parents should first make sure that their own premises is safe enough. However, most parents tend to perform day to day work such as cleaning, washing clothes, and sweeping while they look after the child. But there might be instances where their concentration about the child goes off due to the other work they perform. In instances like this the infant might move to a dangerous place like balcony, veranda or else might touch sockets or run towards the kitchen. Hence there is a possibility of an accident within seconds in loss of concentration about the child. In order to overcome these risks ,CCTV systems, Child care systems and Baby care products were introduced to the current market. But many drawbacks such as inaccuracy, latency, and high cost can be found in them. The aim of this research is to Fig. out such drawbacks and overcome them in an effective manner.

II. LITERATURE REVIEW

There are many wearable baby movement detectors in the market. Among them below mentioned are the prominent [1]

D. Sproutling

This is a unique baby monitor which monitors data like heart rate, skin temperature, and the baby's sleeping position. Also, this will detect if the baby moves or repositions. This is placed around the baby's ankle and it pings all information to an application.

E. Mimo Onesie

This device monitor's information like breathing patterns, temperature, body position while sleeping. A special feature in this device is it is made of a washable and stretchy type of cotton.

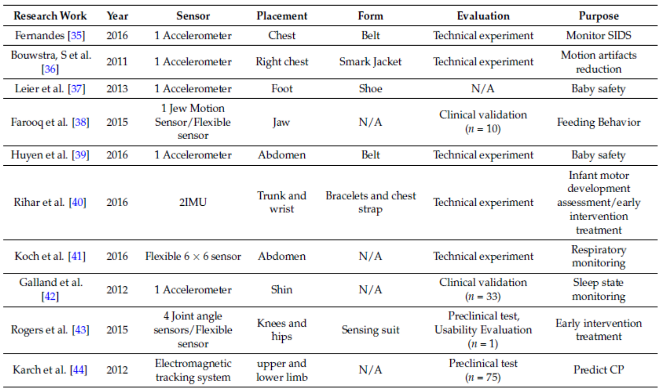
F. Mon Baby

Mon Baby is a wearable device which can be clipped into baby's clothing. This device monitors motions like rolling, fall or leaving the proximity of the tracker. All the information generated from the device is synced with a mobile device through a companion application.

G. Owlet Sock

This device comes in the shape of a sock and has the capability of monitoring the heart rate, skin temperature and oxygen level of the infant.

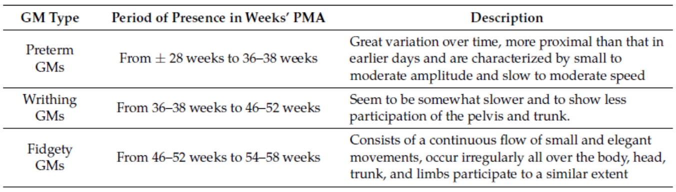
But a common drawback which can be seen in all these detectors are they are either wearable devices or devices which can be fixed to the baby's clothing. Hence the baby might become uncomfortable as it limits the motion of the baby. Also, the weight of the device might affect the baby in an unfavorable way. At the same time, the wearable devices might cause problems in the infant skin like rashes due to wearing them for a long time. At the same time, the greatest lacuna which can be identified in theseis devices are they mainly focus on about the infant's health and do not focusing onabout the security of the infant. Hence the primary aim of this particular research was to developing a way to monitor the infant's movement externally and alerting the parents about the infant's movement. Several Techniques have been used to detect the movement of infants and alerting. Following are such researches which have been conducted by different researchers [2].



Research Work	Year	Sensor	Placement	Form	Evaluation	Purpose
Fernandes [35]	2016	1 Accelerometer	Chest	Belt	Technical experiment	Monitor SIDS
Bouwstra, S et al. [36]	2011	1 Accelerometer	Right chest	Smark Jacket	Technical experiment	Motion artifacts reduction
Leier et al. [37]	2013	1 Accelerometer	Foot	Shoe	N/A	Baby safety
Farooq et al. [38]	2015	1 Jew Motion Sensor/Flexible sensor	Jaw	N/A	Clinical validation ($n = 10$)	Feeding Behavior
Huyen et al. [39]	2016	1 Accelerometer	Abdomen	Belt	Technical experiment	Baby safety
Rihar et al. [40]	2016	2IMU	Trunk and wrist	Bracelets and chest strap	Technical experiment	Infant motor development assessment/early intervention treatment
Koch et al. [41]	2016	Flexible 6 × 6 sensor	Abdomen	N/A	Technical experiment	Respiratory monitoring
Galland et al. [42]	2012	1 Accelerometer	Shin	N/A	Clinical validation ($n = 33$)	Sleep state monitoring
Rogers et al. [43]	2015	4 Joint angle sensors/Flexible sensor	Knees and hips	Sensing suit	Preclinical test, Usability Evaluation ($n = 1$)	Early intervention treatment
Karch et al. [44]	2012	Electromagnetic tracking system	upper and lower limb	N/A	Preclinical test ($n = 75$)	Predict CP

Fig.1 Researches done from 2012-2016 about wearable baby monitors.

Overview of the wearable sensor system to monitor infant movements.
The below table depicts the General Movement Assessment with the age of the infant.



GM Type	Period of Presence in Weeks' PMA	Description
Preterm GMs	From ± 28 weeks to 36–38 weeks	Great variation over time, more proximal than that in earlier days and are characterized by small to moderate amplitude and slow to moderate speed
Writhing GMs	From 36–38 weeks to 46–52 weeks	Seem to be somewhat slower and to show less participation of the pelvis and trunk.
Fidgety GMs	From 46–52 weeks to 54–58 weeks	Consists of a continuous flow of small and elegant movements, occur irregularly all over the body, head, trunk, and limbs participate to a similar extent

Fig.2 General movement assessment with the age of infants

Hence it is clear that the wearable motion detector should not restrict the movements of the infant as motor skills development is a mandatory area for the physical and mental growth and development of the infant.

III. Methodology

H. Human Body detection and classifying it whether an adult or a child

For the child detection process, open cv libraries and deep neural network (DNN) [3] libraries with the caffe [4] train model have been used. With the using the caffe train model and the DNN libraries we were able to get the higher processing frame rate and the efficient resources utilization. The main source code of the object detection is built on open cv [5] libraries in python language. With the information, we collect from previous researches we used python language instead of using C++. Because the open cv Python is more familiar with many software platforms and it supports many other libraries that we expect to use in this project, like Imutils [6], DNN etc.. As the hardware requirement this system needs Intel i3 (6.th gen) / 2 GB Ram / GPU with 100CUDA cores as minimum system requirements. And the camera should have the minimum 3.2 megapixels and should have the capability to stream at least 720p video feed. As the first step, we have developed the image classifier training background. authors have tried out in different platforms and we realized that the Linux environment is the best platform to train the model. Because it provides the capability to directly work with the libraries. In windows environment, we have tried with the Open VINO™ [7] Toolkit also. But it didn't perform as the manual training methods in Linux. Previously we tried the tenser flow training methods. But its required more resources to train the classifier. Then we switched to the caffe train model that required fewerless resources compared to the tenser flow training platform. When executing the pre trained classifier we noticed that caffe model perform faster than tenser flow. Another thing is able to recognize any part of the human body. For the image training step, we have used 500 positive images and 300 negative images. With the caffe train model, we resize all of them in to 100px X 100px size. Because the training process gets more time to learn when the image size is large. For this project, we used pictures of the child in different sides. Then for the code step, we have used the open cv libraries and develop the source code in python language. for the source code development, Authors have used the PyCharm python IDE[8]. The pycharm IDE had many tools to make easier to develop python code. It already designs for python development projects and contains all the libraries in buildingt that authors want. With the caffe model, we use the deep neural network and OpenCV. And also imutils and utils[9] libraries to differentiate the object form the background. First, we resize the video feed and get frame sequence (With a delay) and compare with the pre trained model. When it detects a child in the image it started to track the object and run the alarm triggering a function when the object reaches to the pre-defined area. In our proposed method, its abele to detect multiple child objects in the video feed. And it can detect the child in any direction.

When the process in running stage the resource utilization at athe higher level. We tested the software that has Intel 3rd gen i5 processer, GTX 630m GPU (137 CUDA Cores) and 8GB ram. These are the resource utilization records we collected in the developing stage. with this resource utilization, it gives 15-24fps processing rate. With the new generation, i3 processers are able to process more than this fps with the integrated iris graphic processes.

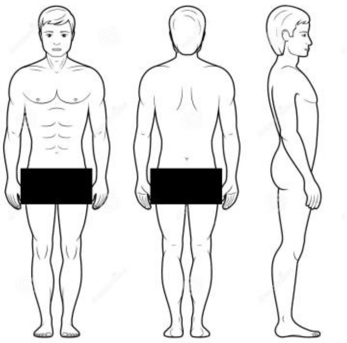
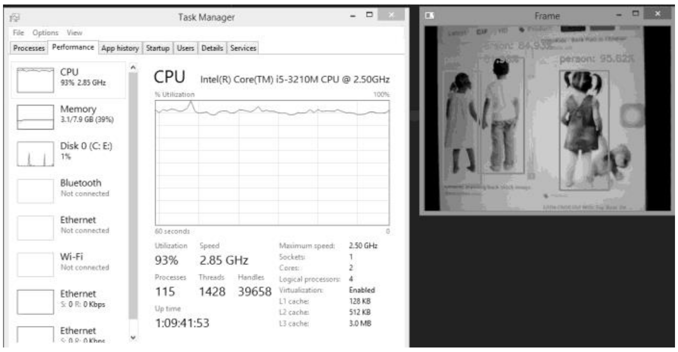


Fig.3.Angles which the camera detect the human object


Task Manager
File Options View
Processes Performance App history Startup Users Details Services
Frame

CPU 93% 2.85 GHz
Memory 3.1/7.9 GB (39%)
Disk 0 (C: E:) 1%
Bluetooth Not connected
Ethernet Not connected
Wi-Fi Not connected
Ethernet S: 0 R: 0 Kbps
Ethernet S: 0 R: 0 Kbps

CPU Intel(R) Core(TM) i5-3210M CPU @ 2.50GHz
% Utilization 100%
60 seconds 0

Utilization 93%
Speed 2.85 GHz
Processes 115
Threads 1428
Handles 39658
Up time 1:09:41:53

Maximum speed: 2.50 GHz
Sockets: 1
Cores: 2
Logical processors: 4
Virtualization: Enabled
L1 cache: 128 KB
L2 cache: 512 KB
L3 cache: 3.0 MB

Fig 4. Performance of the machine while the detection process takes place.

I. *Child Movement Detection Analysis, and Alarm Triggering*

This component is mainly developed using Python as it is a high-level language and the execution speed is considerable higher when compared to the other languages. TensorFlow Inception Model Version 3 was used to retrain the new model with customized classifications, image manipulation was handled by Open CV , NVIDIA CUDA and CUDNN libraries were also used in image processing and analysis using the maximum performance of the external NVIDIA graphic card of the machine. The development process was done within the PyCharm Integrated Development Environment This component mainly identifies the movement of the infant In order to develop this part the custom model were built upon the already built TensorFlow Inception Model Version 3[10]TensorFlow Inception Model Version 3 is a trained model using large set of images which are in ImageNet[11] since 2012. Initially, three classifications were built as "crawling baby", "standing baby" and "sitting baby" using about 350 images per classification. Classifications "crawling baby" and "standing baby" acts as positive cases as they are warning states as the baby is in a dynamic position [12]. "Sitting baby" acts as a negative case due to the position is static and it is not a warning state. The video stream is split in to frames for the purpose of analyzing so.the Open CV libraries were in the manipulation of images. Using Tensorflow GPU libraries the frame set is matched with the retrained graph frame by frame. To make this process efficient and accurate NVIDIA CUDA and CUDNN libraries were used getting the optimum performance of the GPU of the machine. In matching, after the most likely prediction state classification is determined then the confidence level of it is also determined after tallying with the model. To sort out the most likely prediction state and the confidence level Numpy Python library hais collaborated in numerical calculations. If the most likely prediction is in a positive case / warning state and also its confidence level is greater than the desired (85%) level alerting should be triggered.

1) *Alerting*

Fig.5, Illustrates the high-level architecture of the alerting system which is explained as follows.

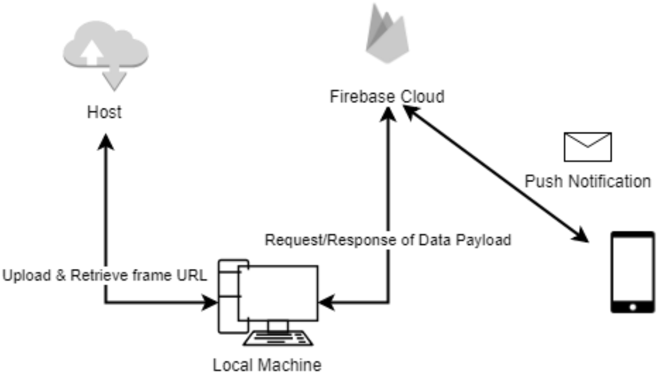


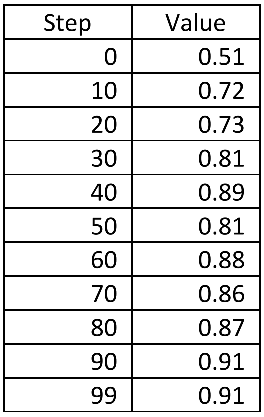
Fig.5.High level diagram of the alerting system

As this research presents a smart solution alerting is done using push notifications to notify the user about the predicted warning states. In order to accomplish that task Firebase Cloud Messaging was used. As Python was used as the foundation of the research Python Cloud Messaging library named PyFCM[13] used to maintain the connectivity and perform the functions of sending push notifications.. Push notifications were customized with extra parameters than the default such as message title, message body, and image URL. The title and the message body contains the warning message and the time stamp of the incident and the Image URL contains the detected frame with the highest confidence level. Since we were manipulating an Open CV image the image was saved in the local machine and then using FTP (File Transfer Protocol) the image was uploaded to a separate host. Then the uploaded image URL was retrieved by the host. That particular URL was sent as the parameter of the data pay load Image URL. Those three parameters were sent to the Firebase Cloud as a data payload. We can access the data pay load through the Android Mobile Application we have developed and connected with the same application in the Firebase Cloud console.

IV. Results and Discussion

The accuracy calculation in our research, we have done some validations and evaluation tests to the trained data set. This has been automatically done by a tool named Tensor board [14]. During our training process, we were feeding data to this tensor board. To do this overall task we need to pre allocate our training image set, defined in 3 classes and also random images which are not included in previously defined three classes. So, for the evaluation purpose, we have defined the learning rate variable as 0.005 [15]. So that the training process will be a bit slow, but accuracy will be high. To train our data set initially we have used 100 steps and 350 images per each category. As auto generated accuracy graphs in our tensor board it seems that this research is with an average accuracy of 80.90%[16]. Refer to TABLE 1.

TABLE I. VALIDATION ACCURACY COUNT OVER 100 STEPS



Step	Value
0	0.51
10	0.72
20	0.73
30	0.81
40	0.89
50	0.81
60	0.88
70	0.86
80	0.87
90	0.91
99	0.91

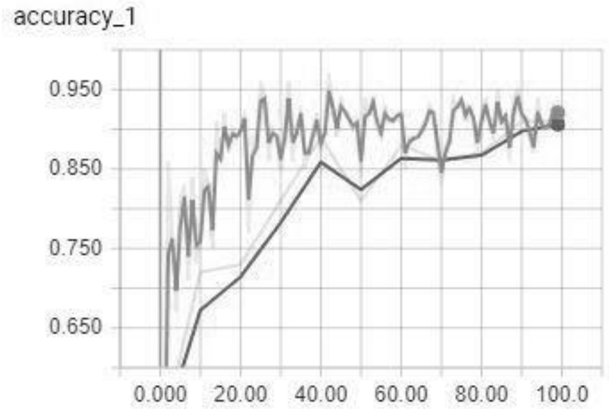


Fig.6 Accuracy graph over validation and training data

V. CONCLUSION AND FUTURE WORK

According to our accuracy of the previous evaluation, we can see that the accuracy level of this model can be improved using several factors such as , Increasing the data set of each class, Increasing the training steps according to the data set ,lowering the learning rate than 0.005[17]. To increase the efficiency of the training process and the detection task we can use , NVIDIA GEFORCE[18] higher range of graphic card with a good capacity, use "mobilenet" architecture instead of "inception_v3" (Use of inception_v3 causes higher accuracy but slows down the process. For higher efficiency mobilenet can be used but at the same time it is lower in accuracy than inception_v3), Keep learning rate in 0.01. Hence it is clear that the efficiency and the accuracy of the process can be enhanced by following the above procedures.

REFERENCES

[1] Kenney, B. (2018). Top 10 Wearables for Your Baby for Peace of Mind. [online] SmartWatches.org. Available at: https://smartwatches.org/fitness/top-10-wearables-for-your-baby/ [Accessed 8 Aug. 2018].

[2] Chen, H., Xue, M., Mei, Z., Bambang Oetomo, S. and Chen, W. (2016). A Review of Wearable Sensor Systems for Monitoring Body Movements of Neonates. Sensors, 16(12), p.2134.

[3] NVIDIA Developer. (2018). NVIDIA cuDNN. [online] Available at: https://developer.nvidia.com/cudnn [Accessed 9 Aug. 2018].

[4] Caffe.berkeleyvision.org. (2018). Caffe | ImageNet tutorial. [online] Available at: http://caffe.berkeleyvision.org/gathered/examples/imagenet.html [Accessed 9 Aug. 2018].

[5] "OpenCV library," About - OpenCV library. [Online]. Available: https://opencv.org/. [Accessed: 09-Aug-2018].

[6] Rosebrock, A. (2018). My imutils package: A series of OpenCV convenience functions - PyImageSearch. [online] PyImageSearch. Available at: https://www.pyimagesearch.com/2015/02/02/just-open-sourced-personal-imutils-package-series-opencv-convenience-functions/ [Accessed 9 Aug. 2018].

[7] Software.intel.com. (2018). OpenVINO™ Toolkit | Intel® Software. [online] Available at: https://software.intel.com/en-us/openvino-toolkit [Accessed 9 Aug. 2018].

[8] JetBrains. (2018). PyCharm: Python IDE for Professional Developers by JetBrains. [online] Available at: https://www.jetbrains.com/pycharm/ [Accessed 9 Aug. 2018].

[9] "python-utils," PyPI. [Online]. Available: https://pypi.org/project/python-utils/. [Accessed: 09-Aug-2018].

[10] IBecoming Human: Artificial Intelligence Magazine. (2018). Transfer Learning: retraining Inception V3 for custom image classification. [online] Available at: https://becominghuman.ai/transfer-learning-retraining-inception-v3-for-custom-image-classification-2820f653c557 [Accessed 8 Aug. 2018].

[11] Towards Data Science. (2018). Train Inception with Custom Images on CPU – Towards Data Science. [online] Available at: https://towardsdatascience.com/training-inception-with-tensorflow-on-custom-images-using-cpu-8ecd91595f26 [Accessed 8 Aug. 2018].

[12] TensorFlow. (2018). How to Retrain an Image Classifier for New Categories | TensorFlow Hub | TensorFlow. [online] Available at: https://www.tensorflow.org/hub/tutorials/image_retraining [Accessed 8 Aug. 2018].

[13] Olucurious, "olucurious/PyFCM," GitHub, 23-Feb-2018. [Online]. Available: https://github.com/olucurious/PyFCM. [Accessed: 09-Aug-2018].

[14] TensorFlow. (2018). TensorBoard: Visualizing Learning | TensorFlow. [online] Available at: https://www.tensorflow.org/guide/summaries_and_tensorboard [Accessed 8 Aug. 2018].

[15] TensorFlow. (2018). TensorBoard: Graph Visualization | TensorFlow. [online] Available at: https://www.tensorflow.org/guide/graph_viz [Accessed 8 Aug. 2018].

[16] Codelabs.developers.google.com. (2018). TensorFlow For Poets. [online] Available at: https://codelabs.developers.google.com/codelabs/tensorflow-for-poets/ [Accessed 8 Aug. 2018].

[17] Codelabs.developers.google.com. (2018). Image Classification Transfer Learning with Inception v3. [online] Available at:

[18] "Drivers," Compare and Buy GPUs | GeForce. [Online]. Available: https://www.geforce.com/drivers. [Accessed: 09-Aug-2018].