IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 11, November 2017

Detection of Human Face in a Meeting Using Audio Sensitive Camera

[1] Giridhar N R, [2] Aniketh M, [3] Ashish Kumar Vishwakarma, [4] Ganesh Prateek L, [5] Dr. Jharna Majumdar
[1,2] Student, Dept. of Computer Science & Engg.
[3,4] Student, Dept. of Electronics & Communication Engg.
[5] DEAN R&D Prof. and Head Dept. of MTech Computer Science & Engg., Head Center for Robotics Research
Nitte Meenakshi Institute of Technology, Bangalore – 560 064, India

Abstract: - **A meeting outcome is a desired result or product that will be achieved by the end of the meeting. It is also necessary to take note of key points made in the meeting. In this paper, we achieve this goal using an autonomous system. The entire system is built on both hardware and software components. The system is developed on an embedded board containing a video recording camera, audio sensors and a Dynamixel motor. The camera rotates in the direction the sound originates and starts to record. The recorded video is fed to the software part where face detection is done. Face detection is a computer technology that is used to identify human faces in digital images. We make use of three colour models like RGB, HSV and YCbCr to perform skin segmentation to obtain probable regions where human face is present. We determine the best colour model which is suitable for skin segmentation. From the segmented image obtained from the previous step, Lucas-Kanade Optical Flow Method [6] is used in determining the location of the face in the frame.**

Keywords: - **Embedded board, Audio sensors, Camera, Shell Scripting, Color models, Skin Segmentation, Optical Flow method.**

I. INTRODUCTION

Decision made in a meeting may involve a lot of time, with important things at stake in meetings, minutes meetings are considered of great value. They serve as record of what was discussed and decided in a meeting. To do this particular task we have developed an autonomous system that is equipped with a video recording camera [3], to record the happenings of a meeting scenario, audio sensors [2] to identify from which direction the sound originates in order to rotate the camera to that direction, a Dynamixel [5] motor that is used to move the camera platform, an embedded board [1] which controls the overall functioning of the system and acts as the media through which the content that is recorded is sent to the software component, which detects human face. L.S. Balasuriya, et al [7] in their work have developed a system that will automatically detect human face using a deformable template algorithm based on image invariants. The deformable template was run down to know the exact location of face in a scene. Avenir K. Troitsky [8] in their article proposed a fast two-level multiple face detection algorithm. For location of face, detected facial elements points are used. These points are used for preliminary face location. Padmapriya et al [9], have done a face detection design using Adaboost algorithm. The simplest feature, pixel intensity set estimations is used. This is where the sum of the luminance of the pixels in the white region of the feature is subtracted from the sum of the luminance in the remaining gray section. A structure of gradually more complex classifiers called a cascade is built to improve computational competence.

II. AUDIO SENSITIVE CAMERA

A. Objective

The main objective of this work is to make a setup for meeting with an audio sensitive camera. The camera will have motor control to take the video at whatever location the guest is speaking. Video generated at the end of the meeting would be recorded thus containing both video and audio signals. The captured video and audio is stored where the algorithm developed to detect human face is applied to summarize a meeting. The following section explains the hardware that is built. Later on Face detection algorithm is explained.

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 11, November 2017

B. Flow chart

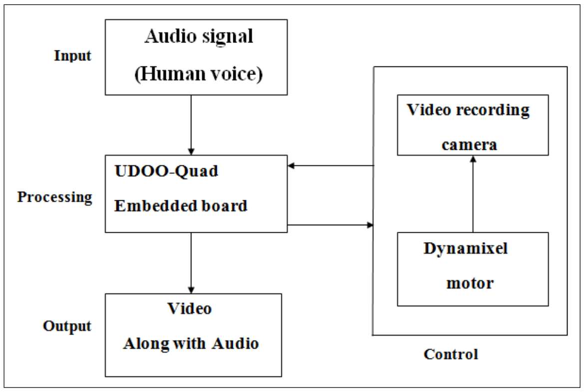


Fig.1. Flow of work

The input comprises of human voice which is detected by the sound sensors. These voice signals are sent to the embedded board to determine what direction the sound appears from. Embedded board, is the UDOO-Quad embedded board, which houses the operating system and controls the overall functioning of the system. Also, the video content recorded is stored on the embedded board. Control comprises of the Dynamixel motor and the camera platform. The Dynamixel motor is programmed through the embedded board and controls the rotation of the camera. The camera is guided towards the speaker by the motor upon receiving appropriate signals from the embedded board. The output consists of the recorded video along with corresponding audio.

C. Components utilized

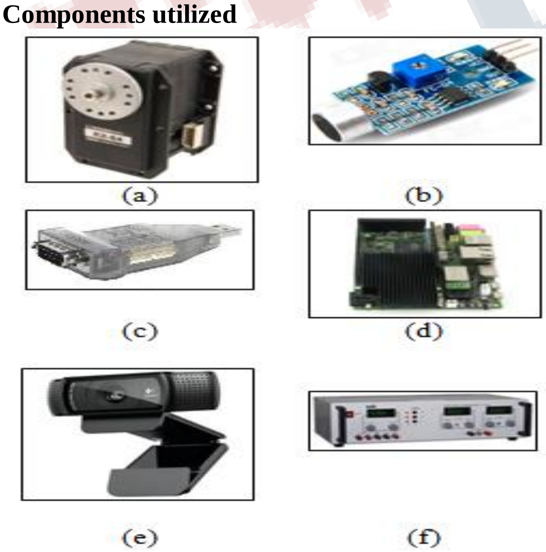


Fig. 2. (a) Dynaxmixel motor (b) Audio sensor (c)USB2Dynamixel (d) UDOO Quad Embedded Board (e) Recording camera (f) Regulated Power Supply

D. Hardware setup

i. A Dynamixel motor is used to control the platform over which the camera is mounted.
ii. RPS (Regulated power supply) is used to power up the Dynamixel.
iii. USB2Dynamixel is used to communicate between the UDOO board and Dynamixel motor. (To send and receive position feedback).
iv. The whole structure is built such that it supports the alignment of Dynamixel and camera platform and also is flexible enough to bring in adjustments.
v. C920 Logitech HD Pro Camera is mounted on the camera platform. This is used to record video and audio.
vi. Recording the video, storing it, control of Dynamixel and sound sensors is managed through the UDOO embedded board.
vii. The GPIOs of UDOO board are used to control sound sensors.

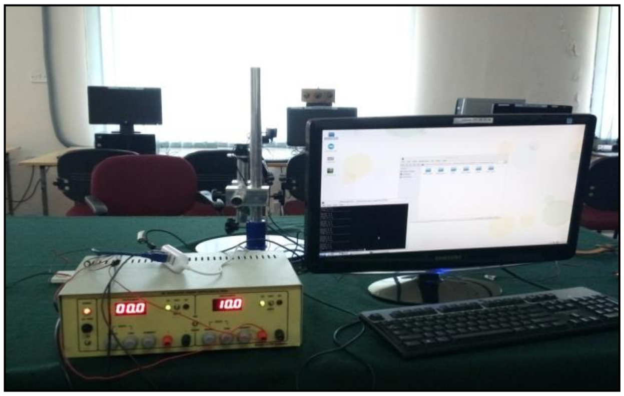
This gives the hardware setup of the components that comprise the audio sensitive camera.

E. Software setup

GCC version 5.4 and above is required to run the commands.

i. GCC version 5 is installed and built on the UDOO board Linux environment.
ii. The necessary Dynamixel SDK is built along with this.
iii. The control part of the program is shell scripted.
iv. Python program is used to record from the camera.
v. This forms the necessary software setup required.

(a)



(b)

Fig. 3. (a) Structure of the Audio Sensitive Camera, (b) Basic layout of the audio sensitive camera setup.

F. Working of the setup

i. When the guest is speaking in the meeting, the sound sensors pick up audio signals and determine the direction of sound.
ii. Depending on this, position values are written to the Dynamixel and it rotates the camera platform, thus pointing the camera towards the guest who is speaking.
iii. The above process repeats whenever direction of sound changes.
iv. The video is recorded along with audio and is stored for further use.

The overall working of the system is governed by shell scripting along with the programs to control the Dynamixel motor.

G. Linux Operating System

To manage the overall functioning of the system and control the inputs from various sensors attached, a Linux operating system is made use of on the UDOO-Quad embedded board. The one used here is UDOO Ubuntu 2.1.4 and as the on-board memory is quite less compared to computers, a lightweight minimal desktop environment is utilized along with this Linux operating system.

H. Programming and control

The control of camera in recording the video and simultaneous storage of the recorded content on the UDOO board is implemented through Python programming (Python 2.7.13).

i. Embedded Python can do much more than C in less program memory.
ii. Program has to take very less processor resource. Using C would burden the processor.

Embedded Python has JIT (Just in time) compilers which increase runtime speed.

III. FACE DETECTION USING OPTICAL FLOW

The importance that is given to detect human faces in a scene is increasing exponentially. Many methods have been proposed to detect human face. Paul Viola and Michael Jones in their Robust Real Time Object Detection [10] came up with a method which made use of Integral Image, AdaBoost learning algorithm and a set of classifiers. The performance of this system is comparatively the best. This is also the widely used algorithm. The output generated from the previous stage, i.e. video recorded in a meeting is fed into this software part to detect human faces. The algorithm that we have proposed consists of four key stages – Skin segmentation, Block based grouping, Elimination of false objects and Applying Optical Flow.

A. Skin Segmentation

Skin segmentation is applied on each frame (24 bit) to obtain the regions where skin is present. We make use of three color models – RGB, HSV and YCbCr. The best color model is decided and the same is chosen for further stages of face detection. Khamar Basha Shaik, et al [11] in their work have made a comparative study of skin segmentation using HSV and YCbCr color space and have shown that YCbCr color space is efficient for separating skin pixels from non-skin pixels. In their experiment, threshold is applied for chrominance components, i.e. Cr and Cb. In this paper we have defined threshold for RGB color space to detect skin pixels. In this paper a study with respect to efficiency is done between RGB and YCbCr color space. The output obtained after this phase is an 8 bit thresholded image.

In our experiment the threshold for R, G and B components are

$$R > 50 \;\&\&\; G > 30 \;\&\&\; B > 20 \;\&\&$$
$$R > G \;\&\&\; B < R \;\&\&\; |R - G| > 15$$

In the work of Khamar Basha Shaik, et al [11] threshold for chrominance components, i.e. Cr and Cb are

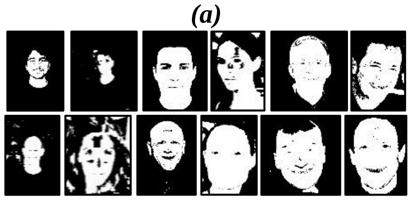
$$Cr > 150 \;\&\&\; Cr < 200 \;\&\&$$
$$Cb > 100 \;\&\&\; Cb < 150$$

For experimental purpose, we have made use of images where people belong to different skin complexions.

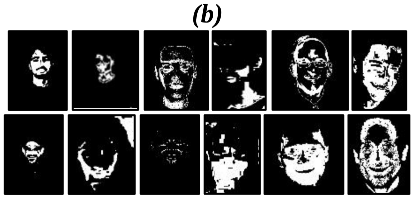
ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 11, November 2017

(a)

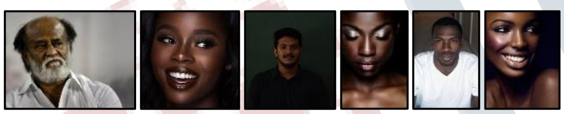


(b)

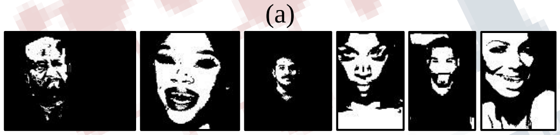


(c)

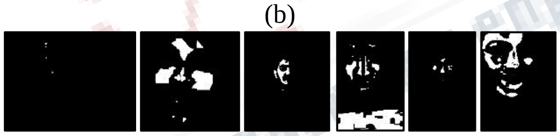
Fig. 4. (a) Images containing human skin having light color complexion, (b) Images Obtained after applying the specified R, G, B threshold values, (c) Images Obtained after applying the values specified in [11] for Cb, Cr components



(a)



(b)

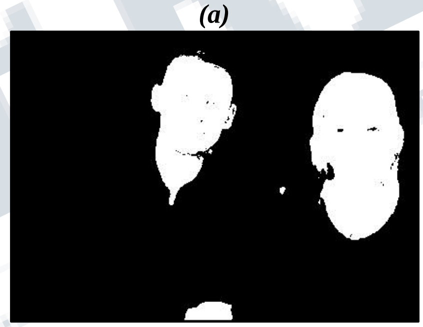


(c)

Fig. 5. (a) Images containing human skin having dark color complexion, (b) Images Obtained after applying the specified R, G, B threshold values, (c) Images Obtained after applying the values specified in [11] for Cb, Cr component

The segmentation of skin pixels using RGB and YCbCr color space is shown in Fig. 4 and Fig. 5 respectively. Our main objective in this part is to extract as much skin pixels as possible. In the case of YCbCr color space, separation of skin color is done, but it fails when the brightness is too less or too high. If the skin complexion is dark as in Fig. 5, chrominance values do not fall into the specified range in such cases YCbCr color model fails to extract as much as skin pixels from the image. In case of RGB color space, the specified range is suitable for all kind situations, with respective to brightness and skin complexions, the same is shown in Fig. 4. So RGB color space can be applied for the complex color images with uneven illumination. We make use of RGB color space to do the skin segmentation in the stages to come.

(a)

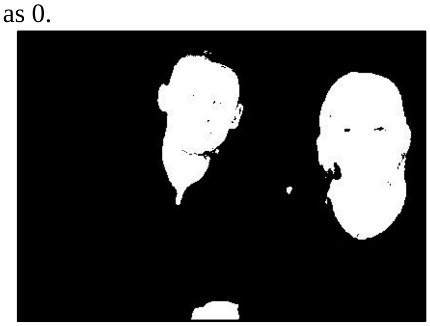


(b)

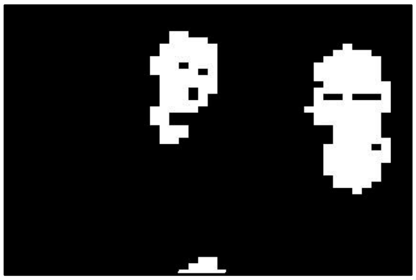
Fig. 6. (a) Input Image, (b) RGB threshold image

B. Block Based Grouping

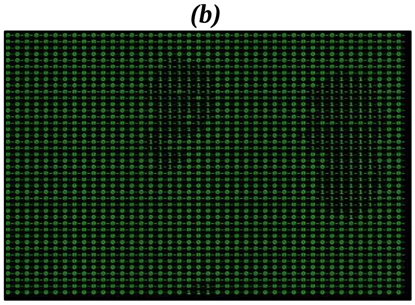
In this stage, the threshold frame is converted into blocks. The size of the blocks depend on size of the frame. Number of skin pixels present in each block is calculated and the blocks having count greater than a threshold value is set as 1, else set as 0.



(a)



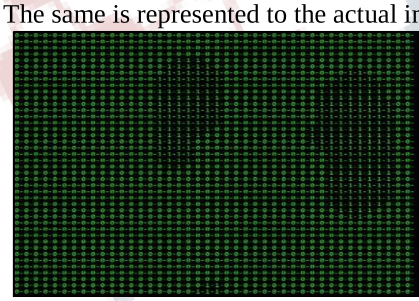
(b)



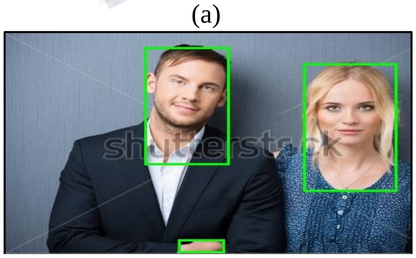
(c)

Fig.7. (a) Skin segmented image, (b) Image divided into blocks (c) Representation of Fig (b) in form of a matrix containing 0's and 1's.

The entire image is divided into blocks as it will be useful to group them. Each block contains either 0's, if there are no skin pixels, or 1's if there are skin pixels. To group the blocks which have connected regions, the matrix is smoothed. From the obtained matrix, blocks which form connected regions are grouped and a bounding box is draw around it. The same is represented to the actual image.



(a)



(b)

Fig. 8. (a) Matrix obtained after smoothing, (b) Output of block based grouping

C. Elimination of False Objects

The area of those connected regions which is considerably less than that of area of the largest connected region is eliminated. In ideal case the aspect ratio of a human face is 1 and the Golden ratio is 1.6. So any connected region whose aspect ratio is less than 1 and greater than 1.6 is eliminated. For the remaining regions, the aspect ratio is recalculated as 1.3 and a bounding box is drawn.

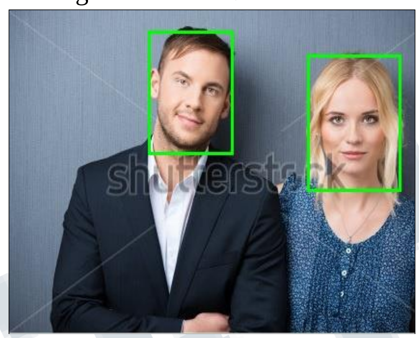
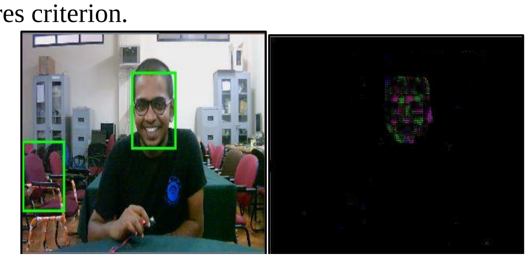


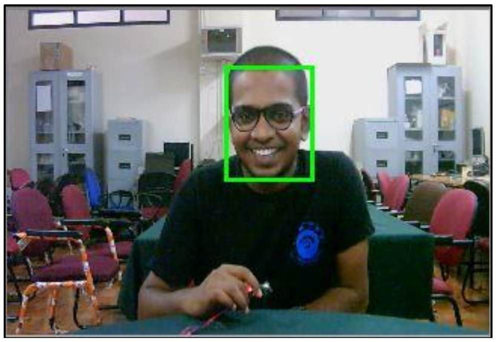
Fig. 9. Bounding box obtained after eliminating false objects.

D. Applying Optical Flow

Optical Flow is defined as the change of structured light in the image, due to a relative motion between the observer and the scene. Optical Flow is mainly used for tracking purposes. In this paper we make use of Optical Flow to find the exact location of face in a scene. After the previous step is completed, there is a still a possibility of non-face object to be present along with the face. To differentiate between these two, we apply Optical Flow as the non-face object is more likely to be the background of the scene which will remain constant throughout the scene. We have developed Lucas-Kanade optical flow method [6], in C++, which is a widely used differential method. It assumes that the flow is essentially constant in a local neighborhood of the pixel under consideration, and solves the basic optical flow equations for all the pixels in that neighborhood, by the least squares criterion.



(a) *(b)*



(c)

Fig. 10. (a) Output without Optical Flow, (b) Optical Flow for the scene, (c) Output after applying Optical Flow

IV. RESULTS

In this section we have presented the results which contain detection of human faces in a meeting using audio sensor camera. Input data is read from the Audio Sensitive Camera which is discussed in Section II. Four different video sequences are recorded from the camera. For these input data, Face Detection Algorithm discussed in Section III is implemented. In Fig. 11 (a), (b), (c), (d), (e) corresponds to input video frames of video sequence 1 and (f), (g), (h), (i), (j) are the corresponding face detection outputs. Similar results are shown in Fig. 12, 13 and 14 for video sequence 2, 3 and 4 respectively.

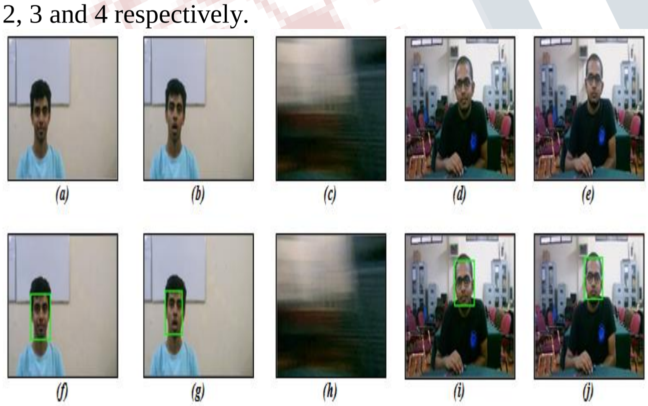


Fig. 11. Detection of Human Faces in a meeting using Audio Sensor Camera – Video Sequence 1 (a) Frame No 13 (b) Frame No 14 (c) Frame No 15 (d) Frame No 16 (e) Frame No 17 (f) Face detection corresponding to Frame No 13 (g) Face detection corresponding to Frame No 14 (h) Face detection corresponding to Frame No 15 (i) Face detection corresponding to Frame No 16 (j) Face detection corresponding to Frame No 17

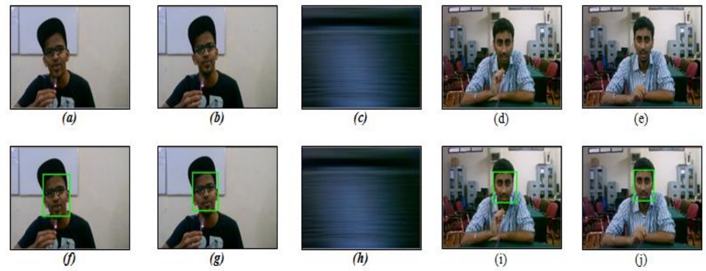


Fig. 12. Detection of Human Faces in a meeting using Audio Sensor Camera – Video Sequence 2 (a) Frame No 43 (b) Frame No 44 (c) Frame No 45 (d) Frame No 46 (e) Frame No 47 (f) Face detection corresponding to Frame No 43 (g) Face detection corresponding to Frame No 44 (h) Face detection corresponding to Frame No 45 (i) Face detection corresponding to Frame No 46 (j) Face detection corresponding to Frame No 47

Fig. 13. Detection of Human Faces in a meeting using Audio Sensor Camera – Video Sequence 3 (a) Frame No 51 (b) Frame No 52 (c) Frame No 53 (d) Frame No 54 (e) Frame No 55 (f) Face detection corresponding to Frame No 51 (g) Face detection corresponding to Frame No 52 (h) Face detection corresponding to Frame No 53 (i) Face detection corresponding to Frame No 54 (j) Face detection corresponding to Frame No 55

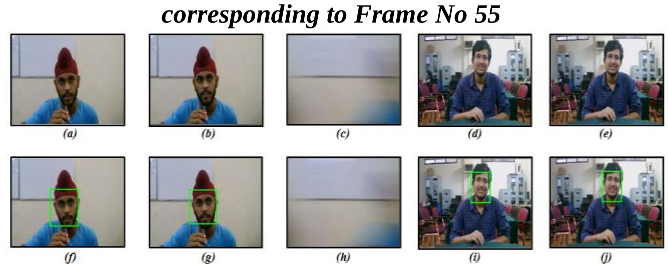


Fig. 14. Detection of Human Faces in a meeting using Audio Sensor Camera – Video Sequence 4 (a) Frame No 71 (b) Frame No 72 (c) Frame No 73 (d) Frame No 74 (e) Frame No 75 (f) Face detection corresponding to Frame No 71 (g) Face detection corresponding to Frame No 72 (h) Face detection corresponding to Frame No 73 (i) Face detection corresponding to Frame No 74 (j) Face detection corresponding to Frame No 75

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 11, November 2017

V. CONCLUSION

A meeting is where a group of people come together to discuss issues, to deal with any matters that are put on the agenda. These minutes of meetings serve as record of what was discussed and decided. The system that we have developed records the video of people speaking in the meeting. To provide a summary of the meeting, the video will go through face detection algorithm discussed in section III and will extract the faces of people present in the meeting. From this many useful information can be extracted. This can be extended to detect the facial expression of a person to know his/her state during the meeting.

VI. ACKNOWLEDGEMENT

The authors express their sincere gratitude to Prof N.R Shetty, Advisor and Dr H.C Nagaraj, Principal, Nitte Meenakshi Institute of Technology for giving constant encouragement and support to carry out research at NMIT. The authors extend their thanks and gratitude to the Vision Group on Science and Technology (VGST), Government of Karnataka to acknowledge their research and providing financial support to setup the infrastructure required to carry out the research.

REFERENCES

[1] Initializing and configuring the embedded board, https: // www.udoo.org/udoo-dual-and-quad

[2] Understanding the working of audio sensors, http:// www.electronics-tutorials.ws/io/io_8.html

[3] Guidelines on usage and handling of the camera. http://www.logitech.com/en-us/product/hd-pro-webcam-c920

[4] Bash shell scripting. https: //www .tutorialspoint.com/ unix/unix-what-is-shell.html

[5] Dynamixel information page, http: //www .robotis .us/ dynamixel

[6] Barron, J. L., D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt. "Performance of optical flow techniques". CVPR, 1992.

[7] Frontal View Human Face Detection and Recognition, L. S. Balasuriya and Dr. N. D. Kodikara, Department of Statistics and Computer Science, University of Colombo, Sri Lanka

[8] Two-Level Multiple Face Detection Algorithm Based on Local Feature Search and Structure Recognition Methods. Avenir K.
Troitsky

[9] S.Padmapriya & Esther Annlin KalaJames „Real Time Smart Car Lock Security System Using Face Detection and Recognition‟, 2012 International Conference on Computer Communication and Informatics (ICCCI - 2012), Jan. 10 – 12, 2012, Coimbatore, INDIA.

[10] Robust Real-Time Object Detection, Paul Viola and Michael Jones

[11] Comparative Study of Skin Color Detection and Segmentation in HSV and YCbCr Color Space, Khamar Basha Shaika, Ganesan P, V.Kalist, B.S.Sathish, J.Merlin Mary Jenith