ISSN 2277-2685
IJESR/Jan-Mar. 2025/ Vol-15/Issue-1s/366-373
A.Akshitha *et. al.,* / International Journal of Engineering & Science Research

# Smart Tv Controlling System For Child Eyes Safety Using Image Processing

[1]Mariam, [2]A.Akshitha, [3]M.Akshitha, [4]A.Malavika
[1]Assistant professor, Electronics and Communication Engineering, BRECW
[2,3,4]B.Tech Student, Department of Electronics and Communication Engineering, BRECW

***ABSTRACT***

*In a human and computer interaction based system, distance estimation by computer vision between camera and human face is a vital operation. To calculate the distance between the camera and face, an estimation method based on feature detection is proposed in this project, where detection of eye and face in an image sequence is described. From the estimated iris and the distance between the centroid of the iris, an algorithm is proposed to determine the distance from camera to face. An architecture for face detection based system on algorithm using Haar features and Canny and Hugo Transform for edge and circular iris estimation is presented here. From the estimated face, canny and Hugo transform is used to determine the iris, and to calculate the distance between the centroid of iris. Later Pythagoras and similarity of triangles are used for distance estimation and based on that, a arduino controller is connected to the system/ laptop so that it will send the commands to the controller to turn ON and turn OFF of the TV for the safety of the children's eye. A buzzer and leds are connected to the controller so based on the distance they will turn on and off.*

## 1-INTRODUCTION

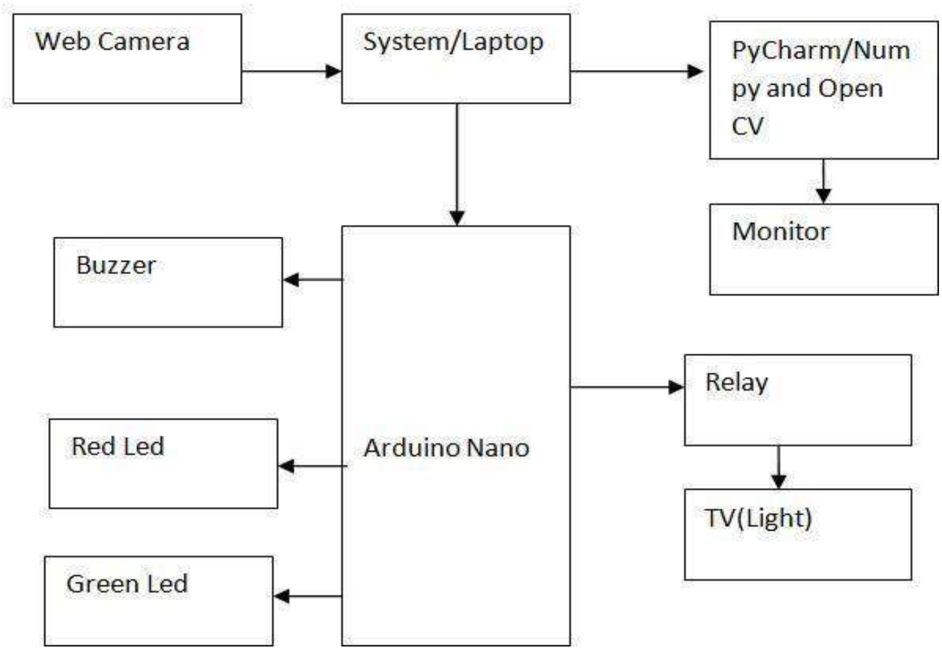
Dynamic development in smart machine vision, robotics, vehicle guidance and other computer vision based applications, distance measurement for computer vision is very important component of modem smart, dynamic and autonomous systems. In a Human and computer interaction based system, distance estimation for computer vision between camera and human face is vital. There have been very limited researches on single camera based distance estimation, which will be faster, reliable and can be used on real time with less computational burden. A 3-D position estimation for human face was proposed on the paper "Face distance estimation from a monocular camera", where location of human's head and face was determined by using motion detection, Hough transform and a statistical color model [4].

Changing pan, tilt and zoom of camera, the face is put on the center of the cameras field of view. Then to measure the distance between human face and an autofocus camera, information taken from focusing the ring are used. A novel method is emphasized in [1] to use devices with monocular camera to determine the depth between the user and the front camera using a back propagation neural network (BPNN). This depth is successively used to calculate the zooming factor for a legible view and to read a document on the display of the mobile device. It is proposed to use frontal facial features acquired from the monocular camera to find the depth information with the use of supervised leaming algorithm. An image processing algorithm has been proposed by [2] to measure the distance from the background. A single camera is fixed at a stationary position to capture the real time image of targeted object and determine its distance in contrast to existing and most common vision algorithms of stereo vision. The proposed method is a statistical method. A two-camera system was proposed in [3] to detect the face from a fixed, wideangle camera, which estimates a rough location for the eye region using an eye detector based on topographic features, and directs another active pan-tilt-zoom camera to focus in on this eye region. In [5], faces are detected in 2D images with a rapid object classifier based on haar-like features and principle component analysis to create an Eigen space. These algorithms are limited to the computational burden and hard to realize in real time. Moreover some of them need extra feature from the cameras. Therefore, we are proposing a new technique to measure the distance from camera based on iris estimation and the distance between them. In Section II, the human face estimation including the Haar Classifier, Integral Image and Training algorithm is described. In section III, the iris estimation algorithm is presented. TO measure the distance between the face and camera has been described in Section IV. Finally, section V gives the outline of the implementation of the algorithm and a developed software.

When photographing a human head, the subject's appearance can vary dramatically depending on the camera's distance from the subject. This variation is caused by perspective distortion, and for 3D objects cannot be undone by simply adjusting focal length; see Figure 1 for an illustration using a synthetic head1. This distortion presents a problem for automatic crosscondition face recognition, e.g. webcam-based recognition from social media images. Even humans find such recognition difficult [1,2]. It is also a source of information, allowing camera pose estimation in cases where the subject is known [3]. This information could potentially be used to un distort the images and improve recognition results. In this paper, we show camera distance estimation from 2D images is possible even when the subject is previously unseen. Our technique replaces the known subject assumption with knowledge of the general distribution of fiducials across people. This distribution turns out to be sufficiently tight to allow surprisingly accurate distance estimation using only a small training set. The paper is organized as follows. Section 2 covers related work from psychology and computer vision. Section 3 explains our method. In Section 4, we validate our method on the Texas 3D Face Recognition Database. Section 5 is the discussion and conclusion.A synthetic head viewed from different camera distances, illustrating projective distortion. Camera distance decreases in the first eight images, indexed in row-major order. Here, the focal length (zoom) is adjusted to keep the figure at a constant size. Fiducials are shown as red dots. In the first image, the camera is far away, resulting in near orthographic projection. In the eighth image (bottom row, middle column) the camera is very close to the human head. The last image is the same as the first, but with fiducial markers from all images. This illustrates the migration of fiducials as a function of camera distance and focal length.

## 2 -BLOCK DIAGRAM AND IMPLEMENTATION

### BLOCK DIAGRAM




**Implemantation:**

The iris image should be rich in iris texture as the feature extraction stage depends upon the image quality. To verify and check the dependency of implemented algorithms for face and iris detection, both high resolution and low resolution images are used here. For the experimental setup, a built in 2MP camera from the laptop and an external 720 P HD web cam were used. The approximate distance between the human face and the camera started from 10 cm. The image acquisition setup is given in Figure 6. To interface this implemented software directly to the webcam JMF tool was used [11] Fig. 3: Image acquisition Implementation steps of proposed algorithm are:

1) Initially the software gets connected with the webcam, reads the input image as described in V and takes the snap shot from live video. It is possible to select the video or image resolution based on the requirement.

2) In the next step, it removes noise using Gaussian filter.

3) Then threshold is applied based on the range of RGB pixels. If pixels value is greater than threshold value, then the background is selected and threshold value is 0 otherwise value is 1.

4) After this preprocessing of image, Harr-like classifier is used with Ad-boost algorithm to estimate human face. If the human face is absent then it requests for another frame form the video stream. If it estimates a face, it passes the image to the next step.

5) From the estimated face area, canny edge detection algorithm gets the edges of the objects. Afterwards, Hugo transform gets the circular iris. If it does not have both the iris or if eye is not present, it discard the image and asks for another instant.

## 3- HARDWARE DESCRIPTION

### Arduino Nano

The Arduino nano is a micro controller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the micro controller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The nano differs from all preceding boards in that it does not use the FTDI USB- to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

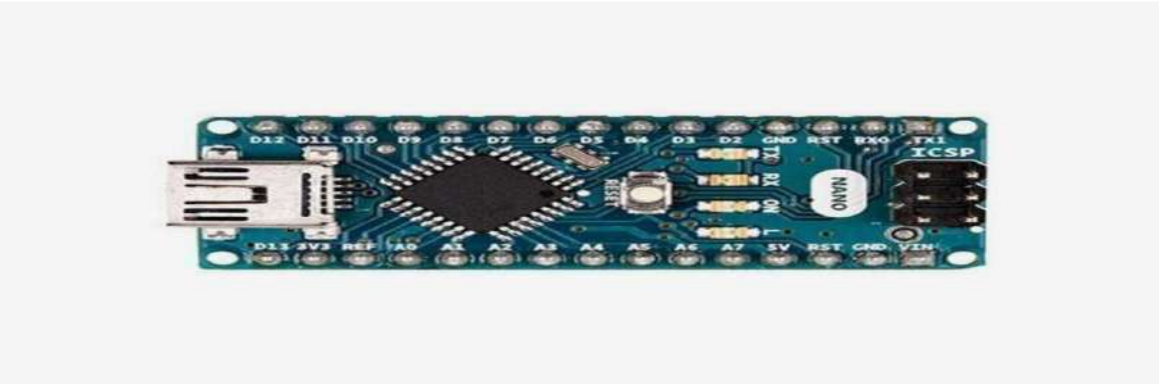


Fig 3.1: Arduino Nano

| | |
|---|---|
| Clock Speed | 16 MHz |

### Power

The Arduino Uno can be powered via the USB connection or with an external power supply. External (non-USB) power can come either from an AC-to-DC adapter (wallwart) or battery.The adapter can be connected by plugging a 2.1mm centerpositive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supplyless than five volts and the board may be unstable. If using more than 12V, thevoltage regulator may overheat and damage the board. The recommended range is 7to 12 volts. The power pins are as follows VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

### Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pin Mode(), digital Write(), and digital Read()_functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 KOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data.

These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip. External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attach Interrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analog Write() function. SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

## 4- SOFTWARE DESCRIPTION

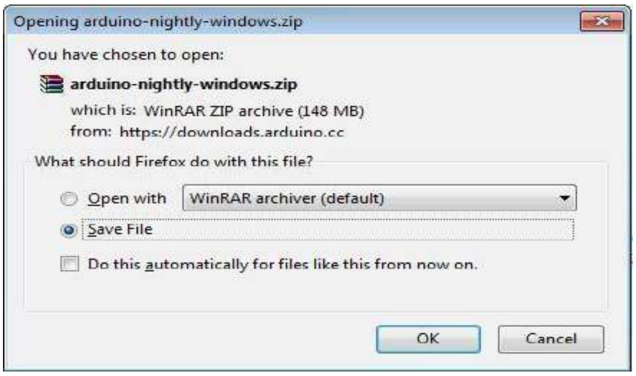
4.1 Creating Project In Arduino 1.7.11 Version Arduino Ide Installation:

In this we will get know of the process of installation of Arduino IDE and connecting Arduino uno to Arduino IDE.

Step 1 - First we must have our Arduino board (we can choose our favorite board) and a USB cable. In case we use Arduino UNO, Arduino Duemilanove Nano, Arduino Mega 2560, or Diecimila, we will need a standard USB cable t In case we use Arduino Nano, we will need an A to Mini-B cable.

Step 2 − Download Arduino IDE Software. We can get different versions of Arduino IDE from the Download page on the Arduino Official website. We must select wer software, which is compatible with wer operating system (Windows, IOS, or Linux).

After wear file download is complete, unzip the file.

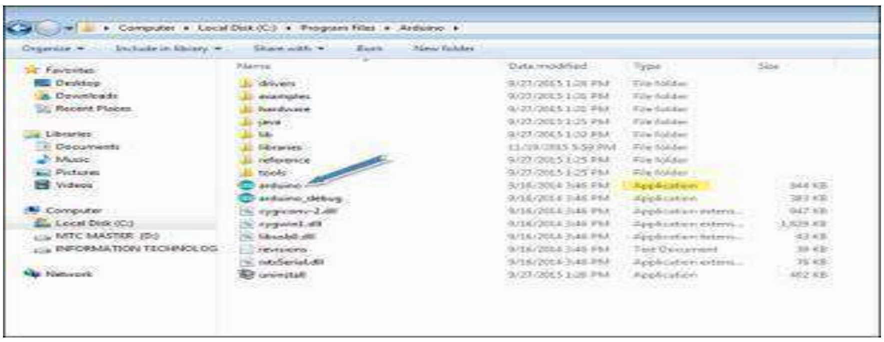



Step 3 − Power up our board.

The Arduino Uno, Mega, Duemilanove and Arduino Nano automatically draw power from either, the USB connection to the computer or an external power supply. If we are using an Arduino Diecimila, we have to make sure that the board is configured to draw power from the USB connection. The power source is selected with a jumper, a small piece of plastic that fits onto two of the three pins between the USB and power jacks. Check that it is on the two pins closest to the USB port.

Connect the Arduino board to wer computer using the USB cable. The green power LED (labeled PWR) should glow.

Step 4 − Launch Arduino IDE.



After our Arduino IDE software is downloaded, we need to unzip the folder. Inside the folder, we can find the application icon with an infinity label (application.exe).

**5-ADVANTAGES AND APPLICATIONS**

1. It is economically cheap.
2. It is fast.
3. It reduces man error and increases proficiency.
4. It can be used easily.
5. Less error is directly proportional to more profit.

**APPLICATIONS:**

1 It's is used in defence.
2 Used in laboratories.
3 Used in manufacturing industries.
4 Used in textile industry 5 Used in aerospace system.

**6-RESULTS**

## 7-CONCLUSION

In this project, a new approach for face to camera distance measurement has been proposed. The estimation method is simple but faster and gives reliable and accurate result for real-time application. With algorithm, the classifier estimates the face and Hugo transform with canny edge detection gives the estimated eye. The classifier is a onetime factory implement. Therefor, the approach is fast enough for real-time application and reduces computational burden. From the estimated iris centroid and the distance between them, the proposed method can estimate the distance from camera to face up to 40 cm. After that, the linearity relation could not be maintained and the error increases. The future work should focus on this. The specific distance and system adaptive experiment verify the feasibility for real-time application.

## REFERENCES

**[1]** Reale, M.; Hung, T.; Lijun Yin, "Pointing with the eyes: Gaze estimation using a static/active camera system and 3D iris disk model;' Multimedia and Expo (ICME), 2010 IEEE International Conference on , vol., no., pp.280,285, 19-23 July 2010 doi: 1O.1109IlCME.201O.5583014

**[2]** Ghidary, S.S.; Nakata, Y.; Takamori, T.; Hattori, M., "Human detection and localization at indoor environment by home robot," Systems, Man, and Cybernetics, 2000 IEEE International Conference on , vo1.2, no., pp.1360,1365 vo1.2, 2000 doi: 1O.1109IlCSMC.2000.886043

**[3]** Kosov, S.; Scherbaum, K.; Faber, K.; Thormahlen, T.; Seidel, H.-P., "Rapid stereo-vision enhanced face detection;' Image Processing (IClP), 2009 16th IEEE International Conference on , vol., no., pp.1221,1224, 7-10 Nov. 2009 doi: 1O.1109IlCIP.2009.5413675

**[4]** Kumar, M.S.S.; Vimala, K.S.; Avinash, N., "Face distance estimation from a monocular camera," Image Processing (IClP), 2013 20th IEEE International Conference on , vol., no., pp.3532,3536, 15-18 Sept. 2013 doi: 1O.1109IlCIP.2013.6738729

**[5]** Ali, A.M., "A 3D-Based Pose Invariant Face Recognition at a Distance Framework," Information Forensics and Security, IEEE Transactions on , vo1.9, no.12, pp.2158,2169, Dec. 2014 doi: 1O.1109ITIFS.2014.2362299

**[6]** Mita, T.; Kaneko, T.; Hori, 0., "Joint Haar-like features for face detection," Computer Vision,
2005. ICCV 2005. Tenth IEEE International Conference on , vo1.2, no., pp.l619,1626 Vol. 2, 1721 Oct. 2005 doi: 10. 1109IlCCV.2005.129

**[7]** Viola, P.; Jones, M., "Robust real-time face detection," Computer Vision, 2001. ICCV 2001.
Proceedings. Eighth IEEE International Conference on , vo1.2, no., pp.747,747, 2001 doi: 1O.1109IlCCV.2001.937709

**[8]** Richard Szeliski. 2010. Computer Vision: Algorithms and Applications (1st ed.).

SpringerVerlag New York, Inc., New York, NY, USA.

**[9]** Dirnitrios Ioannoua, Walter Hudab, Andrew F. Laine, "Circle recognition through a 2D Hough Transform and radius histogramming" Image and Vision Computing 17 (1999) 1526, Elsevier

**[10]** Naveen Singh, Dilip Gandhi, Krishna Pal Singh, "Iris recognination system using a Canny edge detection and a circular Hough Transfrorm ", International Journal of Advances in Engineering &
Technology, May 2011, ISSN: 2231-1963

**[11]** JMP developed by ORACLE,officile website,
http://www.oracle.comltechnetworklarticles/javase/index-jsp140239.html