Precision-based Face Detection Algorithm Implementation on FPGA

Rohit Kumar Singh[1]
MTech, VLSI and Embedded Systems
Sardar Vallabhbhai National Institute of Technology
Surat, Gujarat, India

Hridayjit Nayak[2]
Senior year
Folsom High School
California, USA

Arti Singh[3]
MTech, Electronics and Communication Engineering
Nirma University
Ahmedabad, Gujarat, India

Abstract:- Face detection is a crucial step to implement a face recognition and tracking system which is used in security, surveillance, biometrics, artificial intelligence etc. Face detection is a technique in which face(s) in a image or video and its location in image/video is identified. Face detection can be implemented by using different algorithms which depends upon the accuracy and the processing capabilities of the system on which it is implemented. The accuracy of detection is highly influenced by the factors like illumination, head pose, occlusion etc. This paper talks about the implementation of Viola-Jones algorithm for face detection in each image. This algorithm works on Haar features extracted from a face. Viola-Jones algorithm is a highly accurate algorithm, but it requires large number of resources. The complexity level of this algorithm is very high and can be used in such places where the accuracy of the system is a major concern.

Keywords:- *Accuracy, Algorithm, Haar Features, Viola-Jones.*

I. INTRODUCTION

Face Detection is an application of object detection technique. Object detection is finding the location of objects of a particular class like car, building, face etc. in each image or video. So, face detection is a technique to find the location of face/faces in given image/video. The available algorithms for face detection focus on the detection of the frontal human face. Since lots of robust hardware and algorithms are available, it gives a motivation to use face detection in the wide range of applications. Although powerful algorithms and hardware are available today but still face detection systems are not 100% accurate because they work within some constraints. Illumination, pose, occlusion etc. are some factors which affect the performance of the system. An efficient face detection algorithm is the one which takes care of all the factors like pose, illumination, occlusion etc. But these much of factors will increase the complexity of the system as well as the time taken by the algorithm to detect the face. The response time of the face detection system acts as a bottleneck in the system development. Minimum response time of the face detection system is the need of the hour.

The detection of face in an input image/video is the first step in any face tracking system and face recognition system which plays an important role in a surveillance system that can be so much helpful in many cases like, finding suspects or convicts. An example of this is if a webcam is connected to a display, then it can detect any face that walks by in front of the webcam. Once this information is stored a number of operations can perform on it in order to detect gender/race/age. Face detection system also has many applications in the fields of Biometrics, Robotics, Human interface and other commercial use.

A. Factors affecting Face Detection: Below are some factors which can affect the result of face detection in an input image/video:

- *Head Pose –*

Due to the pose of the head some of the facial features like nose, eyes, cheeks or lips may be get blocked partially or fully. In an input image the location of the faces may vary due to the profile, half profile, frontal plane rotation and upside down. Figure [1] below shows different poses of head.

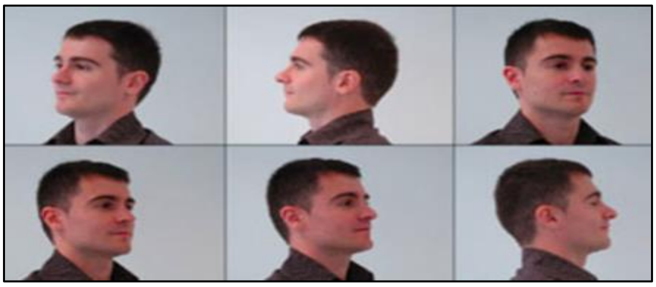

Fig 1 Different Poses of Head

- *Occlusion –*

Occlusion is a type of obstruction for a face in an input image which can cover the face either partially or fully by some other object. Figure [1] below shows some occluded face which can affect the result of face detection.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

Fig 2 Occluded Face Images

➢ *Illumination—*

Illumination determines the quality of images and affects the result of face detection algorithms. This illumination factor is correlated with the lighting and the angle of light at which it falls on the face in the image. Figure [1] below shows a face under different illumination conditions.

Fig 3 Same Faces Under Different Illumination Condition

➢ *Image Orientation—*

This factor depends upon the nature of the input image, which may appear upside down, rotated, inverted, or in the correct form.

➢ *Computation Time and Speed—*

For real-time applications, an algorithm's computation time plays an important and critical role. The computation time depends upon the algorithm's complexity and the availability of hardware resources.

➢ *Facial Expression—*

A human's emotion is expressed by facial expression. Emotions like angriness or happiness are directly related to facial expression and can directly impact an individual's facial expression [1].

B. Features of Human Faces:

A human face has many unique features, such as two eyes, eyebrows, nose, lips, mouth, cheek, skin colour, etc. These features have some patterns, like the skin region under the eye is darker than the cheek area, the nose's bridge is brighter than the remaining nose area, etc. Our face detection technique belongs to the pattern recognition problem class. So, to detect a face in an image, we need to use the features of a face model.

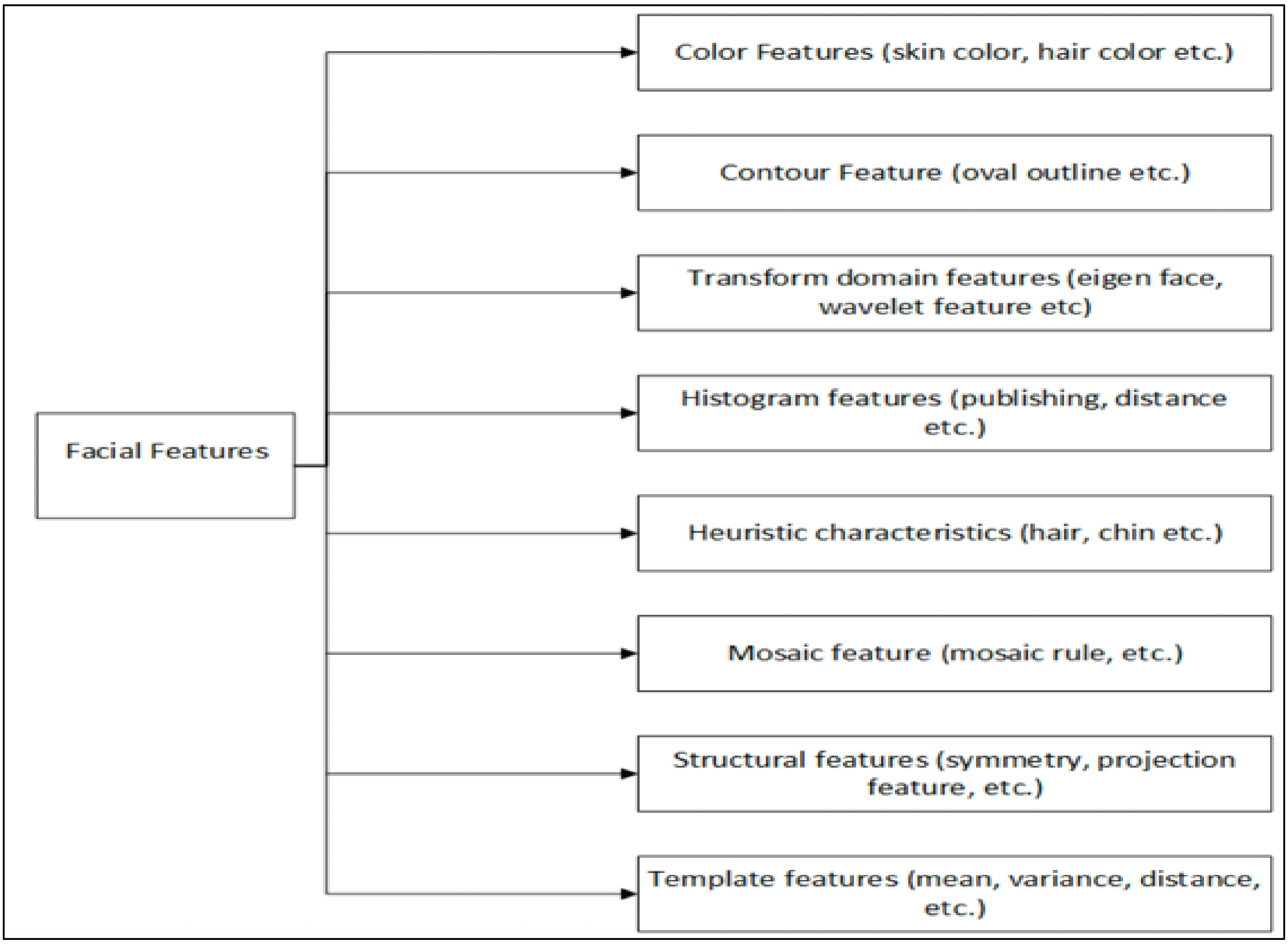

Fig 4 Features of Human Face

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

II. CLASSIFICATION OF FACE DETECTION TECHNIQUES

There are four main categories in which face detection techniques can be broadly divided: knowledge-based face detection technique, feature-based face detection technique, template-based face detection technique and statistic-based face detection technique. Each category is again subdivided into different techniques which are given in below figure [3]

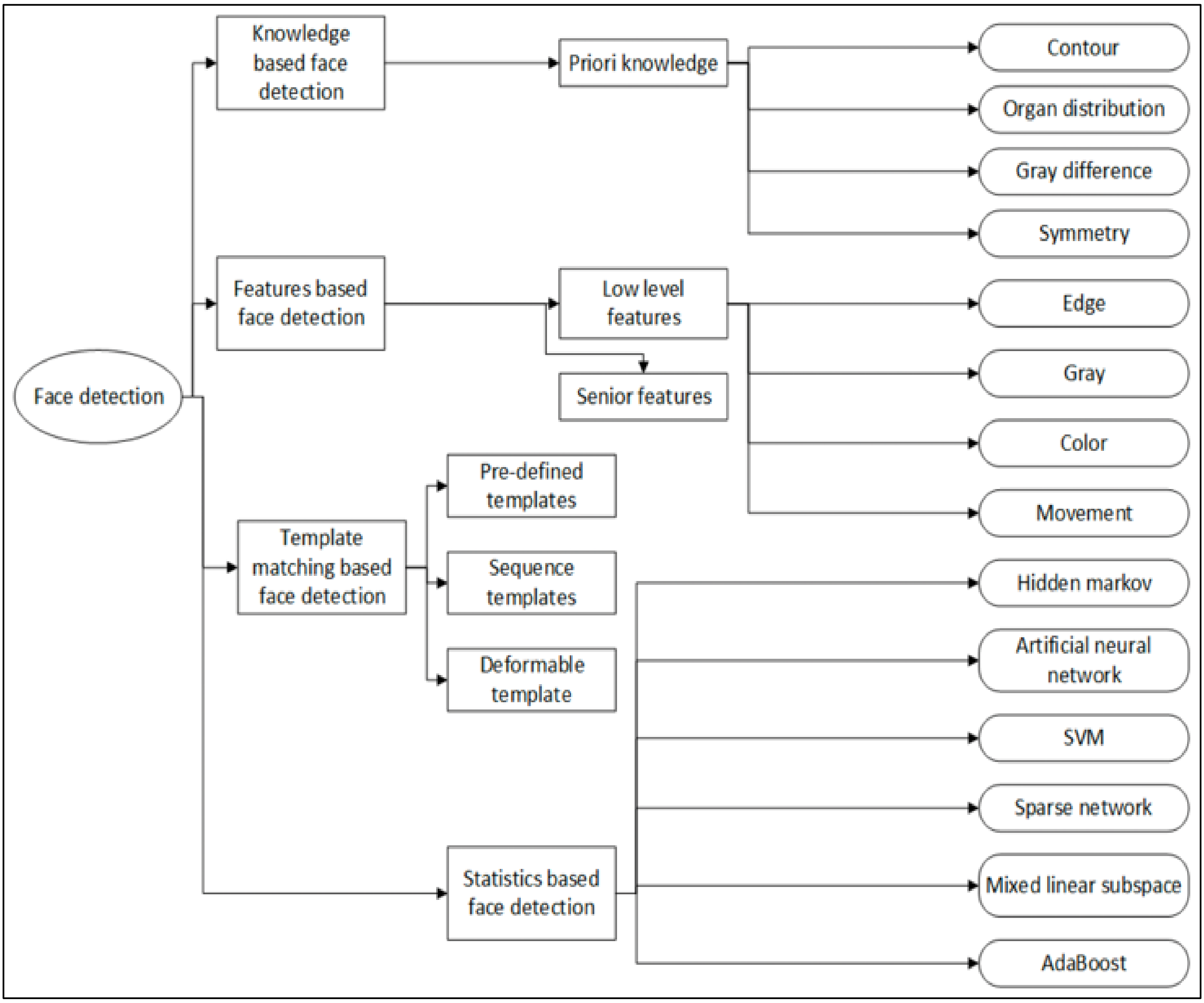

Fig 5 Classification of Face Detection Methods

➢ *Knowledge based Face Detection Technique:*

It is a rule-based face detection technique. In this technique, some rules are defined to detect the faces from the input image. These rules can be extended to detect faces from a complicated background. These rules are nothing but some features like 2 ears, 1 nose, 2 eyes, 1 mouth and other facial features. For example, a rule is like there are two symmetric eyes a face usually has, and the area under the eyes is darker than the cheeks. In the input image, firstly the facial features are extracted and then the face is detected according to the defined rules [4].

The knowledge-based face detection technique tries to capture knowledge of human faces and encode it into well-defined criteria. When the input image meets the criteria, it is declared a face. The difficult part of this method is building such appropriate rules.

➢ *Feature-based Face Detection Technique:*

This face detection technique depends on the features of human faces. A human face can be distinguished by other objects by using these features like the area under eye is darker than the cheeks area, the edge of the nose is brighter than the surrounding area etc. This technique depends on the features which are extracted from the human face and will not be undergone for any changes due to the factors like occlusion, illumination, pose etc. Skin colour, nose, eyes, ears, mouth, eyebrows and etc., are some features that can be used in face detection techniques.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

Generally, the edge detector is used to extract these features. Then a statistical model is created to provide a relationship between the characteristics which are extracted from the features and this model is used to determine the presence of a huma face [3]. Furthermore, some studies have proved that skin colour of human is an excellent feature of a human face and can be utilized to detect human faces from the other objects because the skin colour of different people is different [5]. Along with this, human faces can be differentiated from the other objects with the help of the textures of the human face because human faces have particular textures. Edges of the features can also help to detect the face from the other object.

- *Templet based Face Detection Technique:*

A template base face detection technique tries to define a function for a face. This technique tries to find a global function for all the faces. This function acts as a template. The features of a face act as a variable for the template and different features can be defined individually. For example, a human face can be divided into mouth, eyes, nose etc. For these different parts, a relationship can be defined in terms of brightness and darkness. For the face detection, this method uses the relation between the pattern present in the input image and the pattern which is defined for the face or for its features. This technique can be divided into two categories: predetermined template-based face detection and deformable template-based face detection.

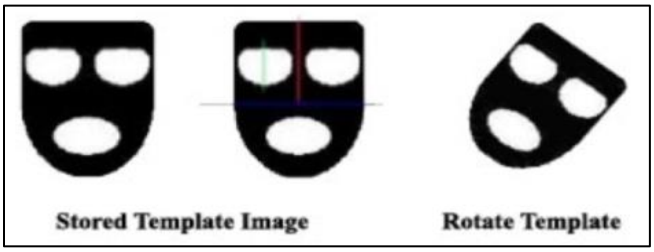

Fig 6 Sample of Defined Template's Images

In predetermined template-based technique a standard template is calculated, and then we calculate the associated value of detection area and templates. When the associated value is within the limits, the detection area is a human face. In the deformable template-based technique, firstly a template parameter is developed, and then according to the detection area, we modify the parameters until convergence, in order to achieve the purpose of face detection.

- *Statistics based Face Detection Technique:*

This technique depends on the statistical analysis and machine learning to find the features of the face. The feature of an image is a variable which have some probability for belonging to a face. This technique is a learning-based technique in which classifiers are trained by using a number of positive images (having faces) and negative images (don't have any face). By using AdaBoost (Adaptive Boosting) some weak classifiers are collected to create a stage and these stages are cascaded into multiple stages [8]. So, in this first weak classifier check each single window and if they pass the threshold value, they will be passed to the next classifiers. This action continues till the last pixel value. Above figure explain how we use weak classifiers to make a strong classifier.

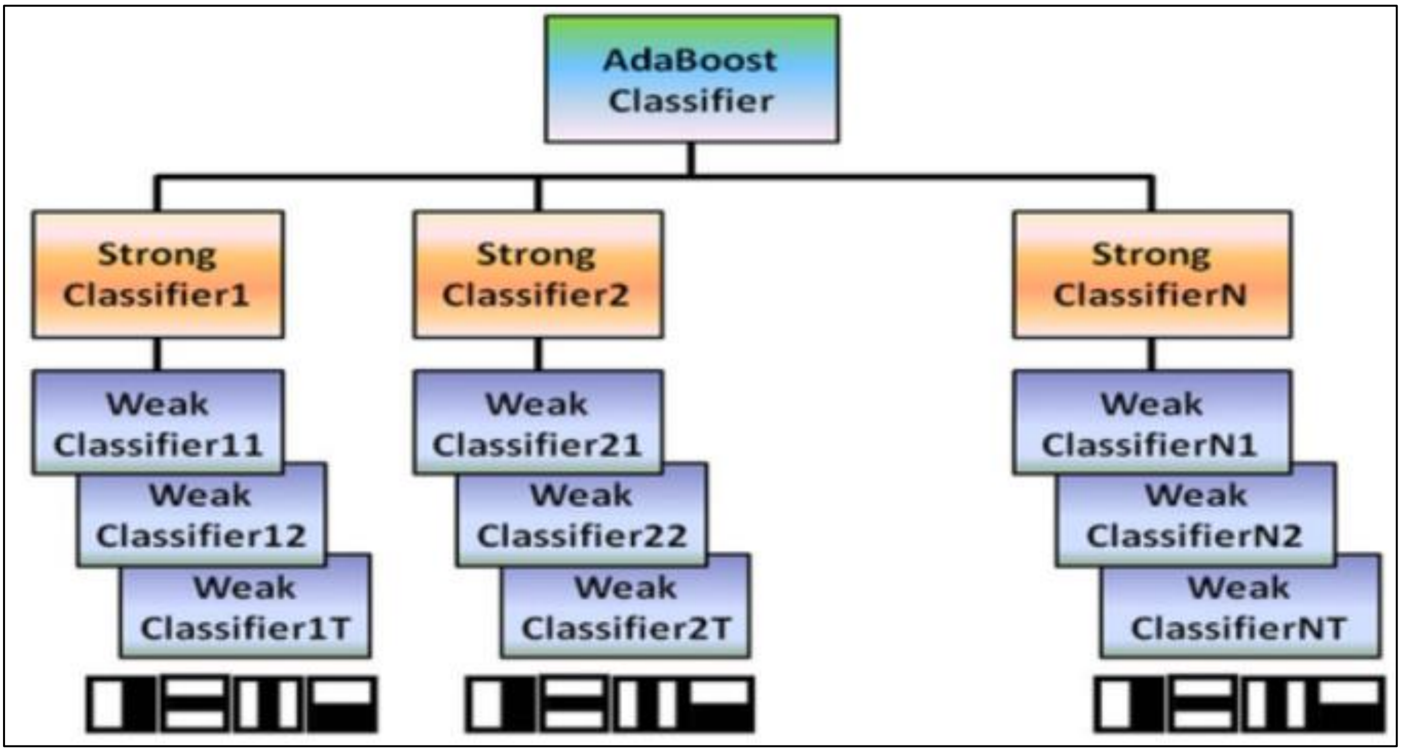

Fig 7 AdaBoost Cascade Classifiers

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

III. ADVANTAGES AND CHALLENGES IN FACE DETECTION

➢ *Advantages and Challenges in above-Described Face Detection Methods are given in Table:*

Table 1 Advantages and Challenges in Face Detection Methods

Methods	Advantages	Challenges
Feature Based	These techniques have Rotation independency, scale independency and execution time is less when contrasted with different strategies [7].	The main challenge is feature "restoration". This can happen when the algorithm tries to retrieve features that are imperceptible because of huge varieties, similar to head posture when we are coordinating a profile picture with a frontal picture [8].
Knowledge Based	Knowledge based algorithms are easy to implement.	Building an appropriate set of rules which can apply for all faces with the different condition. There could me many false positive if the rule were being so general, while there might be numerous false negative if the rules are too detailed. [9]
Statistic Based	In this technique a window with non-face will be dismisses through early stages and along these lines the execution time will be diminished and accuracy will be increased also [12].	In this method the system structure requires many adjustment (number of layers of nodes, learning rate, etc.) to acquire the desired performance [3].
Template Based	Template based algorithms are simple for implementation and we can take some assumption in advance [1].	These algorithms are dependent on size, scale, occlusion, and rotation. So every time need to take care that input image must be a frontal image and un-occluded [1].

IV. VIOLA JONES ALGORITHM

Viola-Jones face detection technique is one of the face detection techniques which can detect the presence of frontal face(s) in an input image and determine the location of that face. This face detection technique can scan the input image rapidly and give a high accuracy rate. Therefor the detection rate of viola-jones technique is high as well as the false positive rate is very small. There are three main properties of this algorithm which are characterized briefly as:

- The first one is the representation of the input image, in which the location of faces needs to determine, into a new format known as "Integral Image". This format of the image allows the detector to calculate the "features" rapidly. Along with the original image, this format also helps to calculate the features at different scale value. This format of the image can be obtained by only some mathematical operations per pixel. With the help of the integral image, we can obtain the "Haar" like features of an image very quickly and in constant time irrespective of the location of the pixel.
- The second property of this algorithm is the introduction of the "Adaptive boosting (AdaBoost)" to select the critical features of the face out of all the computed feature. The AdaBoost algorithm is a learning algorithm and after learning by the different examples of faces and non-faces it gives a classifier which can classify the face in an image. Out of all features, the irrelevant features must be rejected by the learning process to achieve fast classification and process only the critical features.
- The third and important contribution is the application of the cascaded structure of the strong classifiers. This cascaded structure enables the feature like the rejection of the background region quickly and spends most of the computation over the promising face like regions. Due to this structure the region of the image which is not containing the face.

➢ *Haar-Like Feature:*

Our face contains a number of features like nose, eyes, lips, cheek, eyebrows etc. In face detection scheme we prefer to use features of the face rather than the pixels directly. There are a number of reasons to use the features instead of the pixels directly for computation. The first reason of this is the speed of detection because the detection speed of a feature-based system is higher than the pixel-based system. The second reason is that the features can be used to encode ad-hoc domain knowledge that is hard to learn by using a finite quantity of training data. The feature used in this technique is same as Haar basis function which used by Papageorgiou et al (1998). Below figure shows some rectangular Haar like feature.

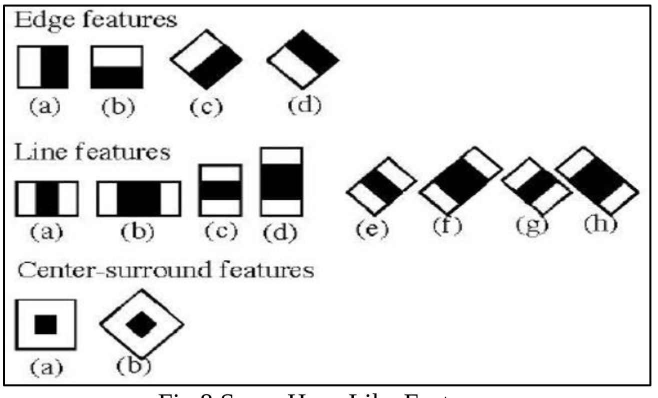

Fig 8 Some Haar-Like Features

The two, three and four rectangular haar-like feature is shown in figure below. This haar-like feature gives one value after the computation of the feature which can be used to categorize the subsections of an image.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

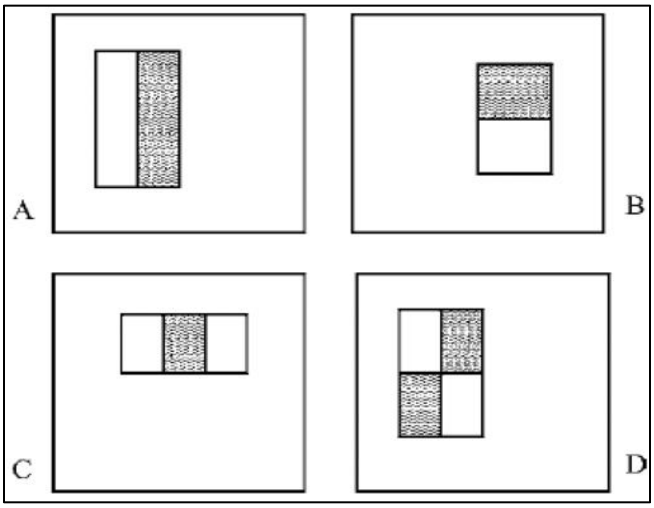

Fig 9 Rectangle Haar-Like Features

The computation of two rectangle features (figure 9 (A), (B)) can be done by taking the difference between the summation of the pixels under white region and black region. The computation of three rectangle feature (figure 9(C)) is done by taking the summation of the pixels coming under the two outside rectangles and then subtract it from the summation of the pixels under the centre rectangle. Similarly, for four rectangle feature (figure 9(D)) the value can be computed by taking the summation of the pixels under diagonal region and then take the difference of the two summed values.

➤ *How Haar-Like Feature Works*:

Haar-like features are nothing, but the adjacent rectangular region and its value is calculated by taking the difference between the sum of the pixel's intensity in the white rectangular region and black rectangular region. These types of features are basically used in the machine learning where a function is trained by a number of positive images (these images contain the human faces) and negative images (these images didn't contain any human face) which are utilized to detect the location of human faces in an image. The input image is scanned and searched for the Haar like feature of the current stage. The size and weight of every feature are computed by the machine learning algorithm like Adaboost. There are a number of features which can be applied on the face, and it is shown in the figure below:


Fig 10 Applying Haar-Like Feature on a Face

One of the common features of the human face is the region of eyes is darker than the region of the cheeks. So, during the training phase of classifier through the database of images, we make a set of two adjacent rectangles that lies above the eye and the cheek region and save the dataset for the testing phase. To compute these features, we compute the sum of the pixel's intensity under the black and white rectangle and take the difference. To compute this summation rapidly and in equal time we use a new type of image representation that is the integral image.

➤ *Integral Image:*

"Integral image" is the intermediate representation of the input image. The main objective of this type of representation is to compute the summation of the pixel's intensity under a rectangular region quickly and in constant time irrespective to the location where it is needed. This representation of the input image at pixel location (i,j) can be achieve by taking the summation of the pixel's intensity above and to the left of the location (i,j). Equation (3.1) gives an idea to calculate the integral image and also shown in figure below*:*

V. MATLAB IMPLEMENTATION AND RESULTS

Viola-Jones algorithm for face detection technique is a machine learning technique in which a cascaded function is trained through a number of positive (contains face/faces) and negative images (didn't contain any face/faces). Now, this function is able to classify the face(s) in other images also. This cascade classification function is obtained by taking the weighted sum of weak classifiers. These weak classifiers are made by the features which are extracted from the training data.

For the development of MATLAB implementation of the viola-jones algorithm, I have used a pre-trained classifier given by Dr. Rainer Lienhart professor at University at Augsburg in Computer Science department [10]. This is one of the best trained cascaded classifiers based on Viola-Jones approach which is widely used by all prominent companies such as Intel, Microsoft, Apple etc. for face detection applications. So, after the training of the classifiers, it get trained and then it can be used to classify/detect the objects. The flow of the implementation of the Viola-Jones algorithm is shown in figure below:

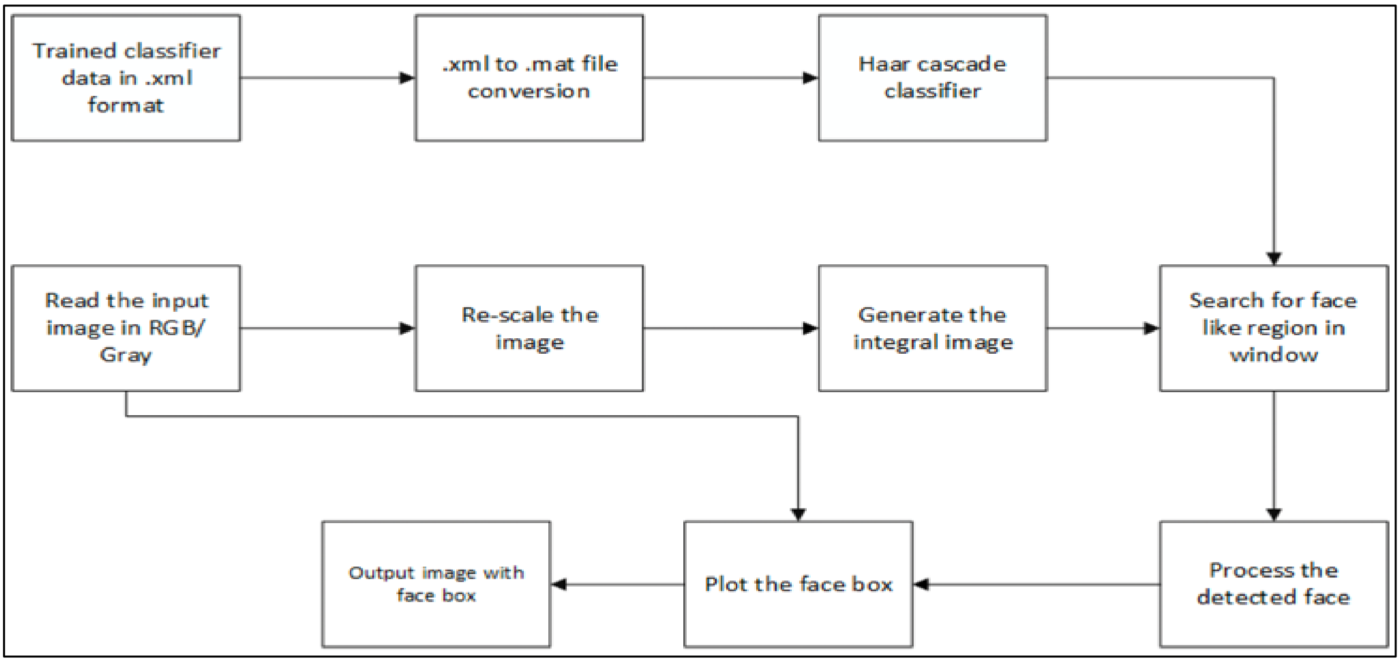

Fig 11 MATLAB Implementation Flow of Face Detector

The trained data which contains all the features at multiple stages is taken as a .xml format. This file is converted into .mat file which stores all the variables that contain the values of required constraints. Now all the features are used to make a cascaded structure of Haar features and every stage have its own threshold. This cascaded structure is used for the detection process. For the detection purpose, an input image in RGB/Grayscale is given to the algorithm. This input image is scaled at multiple values in order to detect the faces of any size. After the scaling process, the image is given to integral image generation unit to generate the integral image. With the help of this integral image, we can calculate the sum of pixels by using four-pixel value irrespective of the number of pixel's to be summed. Through this integral image computation of Haar feature in the image is done. This Haar feature is used to compare with the feature values taken from the trained data. If the computed feature value crosses the threshold value, then it is pass to the next stage of cascaded structure otherwise got rejected for the face detection process. If in an image face is found, then it returns the starting co-ordinate of the face and its width and height. With the help of co-ordinate, width, and height we can get extreme co-ordinates of the face. By using these co-ordinates face region can bound by a box. The bounding box can be drawn by changing the colour of the pixel of the original image to the colour of the bounding box.

- *Classifiers Details:*

The trained classifiers used here is taken from OpenCV to detect the frontal human face by using viola-jones algorithm. Training of this cascaded classifier is done by the frontal faces of size 20X20. The total number of stages used here is 22, the total number of Haar classifiers are 2135 and total number of features used is 4630. For each stage the number of classifiers used is shown in the table below. As shown in the table the number of classifiers in each stage is increasing thus the complexity of each stage is also increases.

Table 2 Number of Weak Classifiers in each Stage

Stage No.	No. of classifiers	Stage No.	No. of classifiers	Stage No.	No. of classifiers
0	3	8	56	16	140
1	16	9	71	17	160
2	21	10	80	18	177
3	39	11	103	19	182
4	33	12	111	20	211
5	44	13	102	21	213
6	50	14	135	Total	2135
7	51	15	137		

- *Performance of MATLAB Implementation:*

In order to obtain the accuracy performance measurement of implemented code has been done. Two different databases has been used in order to measure the performance of implemented viola-jones algorithm. *First database contains 100 images of different people and obtained by collecting images of 320x240 pixels from the internet. This database contains the frontal face images of different people in a complex background and different lightening conditions. Table below shows the accuracy of MATLAB implementation of Viola-Jones algorithm with this database.*

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

Table 3 Accuracy of MATLAB Implementation for Viola-Jones Algorithm

Total Images	Image Type	Correct detection	False Detection	Accuracy
100	Frontal face images	80	20	80%

Second database is given by Cambridge University named as "Pointing'04 ICPR Workshop" [11]. This database contains the face pointing images of 15 different people with different head poses. The angle of head poses varies from -90 to +90. In order to performance measurement only 0-to-45-degree variation in head pose is taken. Along with the head poses this database also contains the facial images of 15 people with and without spectacles. Accuracy of implemented algorithm is also tested for the images with and without spectacles. The image size in this database is 384x288 pixels.

Fig 12 Sample of Image Database

Performance measurement of implemented Viola-Jones algorithm with 'Pointing'04 ICPR workshop' [11] is tabulated in table below. This table shows the accuracy of face detection as the face is rotated by 0^o, 15^o, 30^o, 45^o. We can observe that as rotation angle increases the accuracy of detection decreases because increase in face rotation angle leads to decrement in visible facial features. In 45^o rotation of face the right eye is less than 50% visible than the 0^o rotation i.e. frontal face image. Therefore, with 45^o facial rotation the detection accuracy is minimum.

Table 4 Accuracy of MATLAB Implementation of Viola-Jones Algorithm with Database of Cambridge University

Person ID	Facial Rotation			
	0^o	15^o	30^o	45^o
Person-1	T	T	T	F
Person-2	T	T	T	T
Person-3	T	T	F	F
Person-4	T	T	T	T
Person-5	T	T	F	F
Person-6	T	T	T	F
Person-7	T	T	T	T
Person-8	F	F	F	F
Person-9	F	F	F	F
Person-10	T	T	T	F
Person-11	T	T	F	F
Person-12	T	T	T	F
Person-13	T	T	T	F
Person-14	T	T	T	F
Peron-15	T	T	F	F
Accuracy	86.67%	86.67%	60%	13.34%

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

Along with this database 'Pointing'04 ICPR workshop' [11] database also consists of frontal face images of 15 different people with and without spectacles. The implemented algorithm is also tested with this database. This gives less accuracy as compared to the image of people without spectacles because the features which are related to eyes are getting blocked due to spectacles. This comes under occlusion of facial features. Table below shows the comparison of implemented algorithm on the database which contains the frontal faces with and without spectacles. This table show with spectacles the accuracy of implemented algorithm deceases as some of the features of face get blocked.

Table 5 Accuracy of MATLAB Implementation of Viola-Jones Algorithm with and without Spectacles using Database of Cambridge University

Person ID	**Frontal Face**	
	With Spectacles	**Without Spectacles**
Person-1	T	T
Person-2	T	T
Person-3	F	T
Person-4	F	T
Person-5	F	T
Person-6	F	F
Person-7	T	T
Person-8	F	T
Person-9	F	T
Person-10	T	T
Person-11	T	T
Person-12	F	T
Person-13	T	T
Person-14	T	T
Peron-15	F	F
Accuracy	46.47%	86.67%

Therefore, this implementation giving 46.67% accuracy of detection when person wears spectacles and 86.67% with naked eyes.

- *Face Detection Result:*

For the detection of the face(s) in an image, we have given different images to the implemented algorithm. Some of these images contain single face and some of the images contain multiple faces. The complexity of background is also varied for different images.

For the single face the input image and its corresponding output is shown below:

Fig 13 Input and Output Image of Viola-Jones Algorithm with 1 Face

For the input image which contains multiple faces in RGB space, and its corresponding output is shown in figure below:

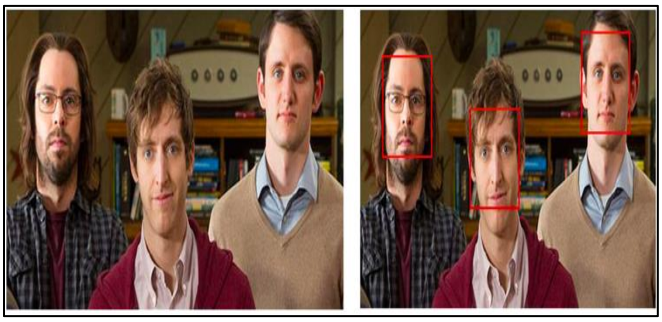

Fig 14 Input and Output Image of Viola-Jones Algorithm with 3 Face

VI. HARDWARE IMPLEMENTATION AND RESULTS

For the hardware implementation of Viola-Jones algorithm, FPGA is used as processing element, a camera is used to take the input image, and a display is used to display the result. The hardware components used in this project are:

- *OV7670 CMOS Camera Module:*

The OV7670 is a CMOS camera module which can operate at a maximum of 30 FPS and 640 x 480 resolution. It is equivalent to 0.3 MegaPixels. This camera is used to take input image to the Zedboard.

- *VGA Display:*

In this project displaying the images is done on a VGA (Video Graphics Array) display. The resolution of the display is 640x480. Like the other displays (ex. TFT) it has horizontal

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

rows which contains a fixed number of pixels which called as number of columns in the screen. At each pixel location, the RGB colour information in the video signal is used to control the colour of the pixel. By changing the analog levels of the three RGB signals all other colours are produced.

➢ *FPGA Development Board:*

The Zedboard development board was chosen for the development of our project. The Zedboard is an evaluation and development board which is based on Xilinx Zynq-7000 All Programmable SoC (AP-SoC). This development kit implements a Xilinx Zynq-7000 AP SoC XC7Z020-CLG484 which is having 4.9MB Block RAM, 106,400 numbers of flip-flops, 53200 number of LUTs (look up tables) and 85K of programmable logic cells. In this project the FPGA kit plays a role of heart of the entire system that captures images from the camera, process the captured image to get the facial features in the image, and display the faces on the VGA display monitor. The camera OV7670 is interfaced with the Zedboard via GPIO pins on the board and the VGA display is interfaced with the VGA connector available on board. A number of general purpose I/Os, switches and LEDs are used for the implementation of some user-controlled activity.

➢ *Hardware Setup used:*

The block diagram of hardware setup and connections of the Viola-Jones face detection system is shown in figure below. Camera OV7670 and a VGA display is connected with the Zedboard. In this setup an image as an input to the system is taken of the resolution of 320x240. Since the display is of resolution 640x480, only the left top corner is used to display the input image or video. VHDL language is used to develop the code and VHDL code is compile and synthesized on Xilinx ISE software and programmed onto the FPGA board. A VGA cable is used to connect the VGA display and GPIOs are used to connect the OV7670 camera. Four switches are used for the user control over the system. To reset the entire system [SW0] is used. In order to send the reconfigure the camera register [SW1] is used. [SW2] is used to define the capture mode like whether we want input as image or video. [SW3] is used to take the snapshot by the camera.

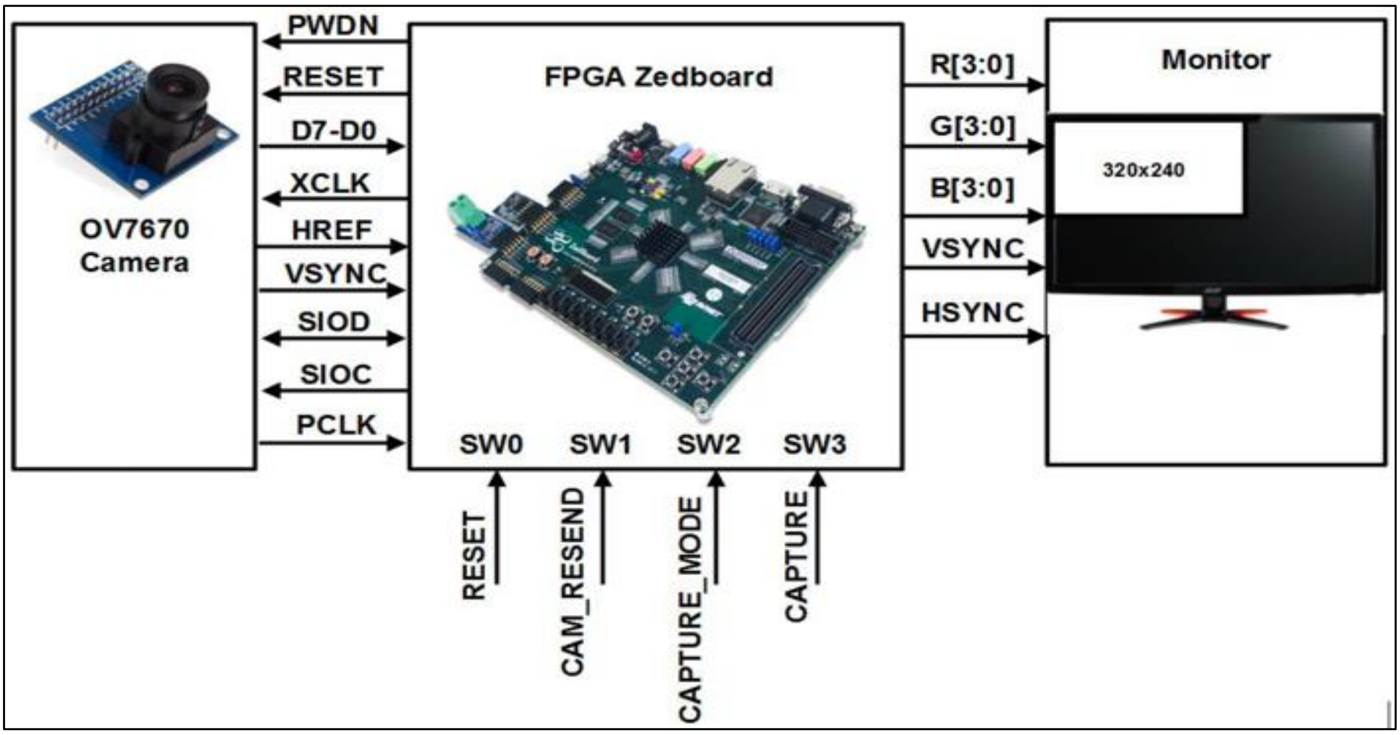

Fig 15 Block Diagram of Hardware Setup used

➢ *The Mapping of Switches on FPGA Board and Detailed Description is given in Table below.*

Table 6 Mapping of Switches used for user's Control

Function	Switch Name	Mapping to Zedboard	Description
RESET	[SW0]	F22	System Reset
Reconfigure Camera	[SW1]	G22	Resend the configuration data to the camera's pins.
Capture Mode	[SW2]	H22	Select the capture mode or video mode [0]: Video mode [1]: Capture mode
Capture	[SW3]	F21	Capture an image [0]: Snapshot [1]: Video

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

VII. VHDL IMPLEMENTATION

Implementation of the Viola-Jones algorithm for face detection system is done on Zedboard using VHDL. The entire implementation is divided into four parts.

- Capturing the input image using OV7670 camera and saving it into BRAM
- Generation of integral image of a part of the image
- Process the integral image for finding the face
- Face box creation over the face.

➢ *Top Level Implementation:*

To control the different modules of the system a top-level state machine has been implemented. The top level of the design consists of a number of modules. These modules are responsible for capturing the image, processing it and display it. All these top-level modules are synchronized with each other and controlled from a top-level entity. A PLL is used to generate the different clock frequencies (25 MHz, 40 MHz, 50 MHz and 80MHz) form the on-board crystal oscillator of frequency 50 MHz. Figure below shows the top-level interconnection between the different modules.

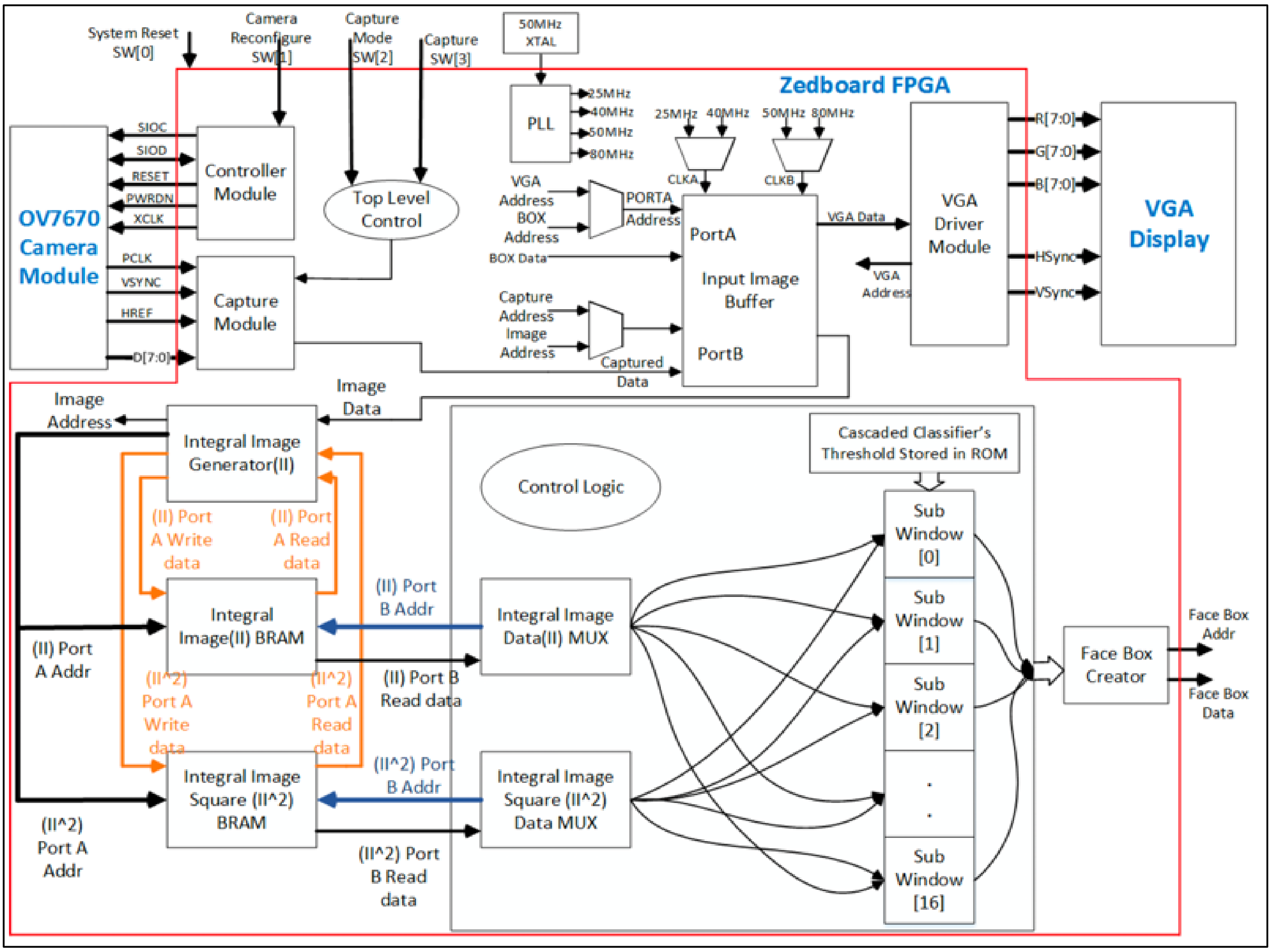

Fig 16 Block Diagram of Top level Hardware Implementation

The top-level entity controls the camera module, display module and the processing part of the input image. For processing the input image, the top level entity takes the source image from the image frame buffer and process it. The processing of source image is done by first generating the integral image and then the comparison of weak and strong stage threshold with pre-trained classifiers. Initially, an integral image of 40x60 pixels is generated. After this 17 parallel subwindow scans the integral image and evaluates this for faces. In parallel with subwindow scanning, the integral image for the next subwindow has been generated so that there will be no delay to give the new integral image to the subwindow evaluation. So there will be no any latency in between the integral image generation and subwindow evaluation. This integral image generation and subwindow evaluation process will continue till the entire current scaled image is processed. After the evaluation of the last subwindow the current image is further scaled down and the again the integral image generation and subwindow evaluation process take place. This system evaluates the source image for the face candidate at 4 scaled values and then create the red box around the face in the source image. The final processed image shown on the VGA display. The scanning of image by subwindow and integral image is shown in figure below:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

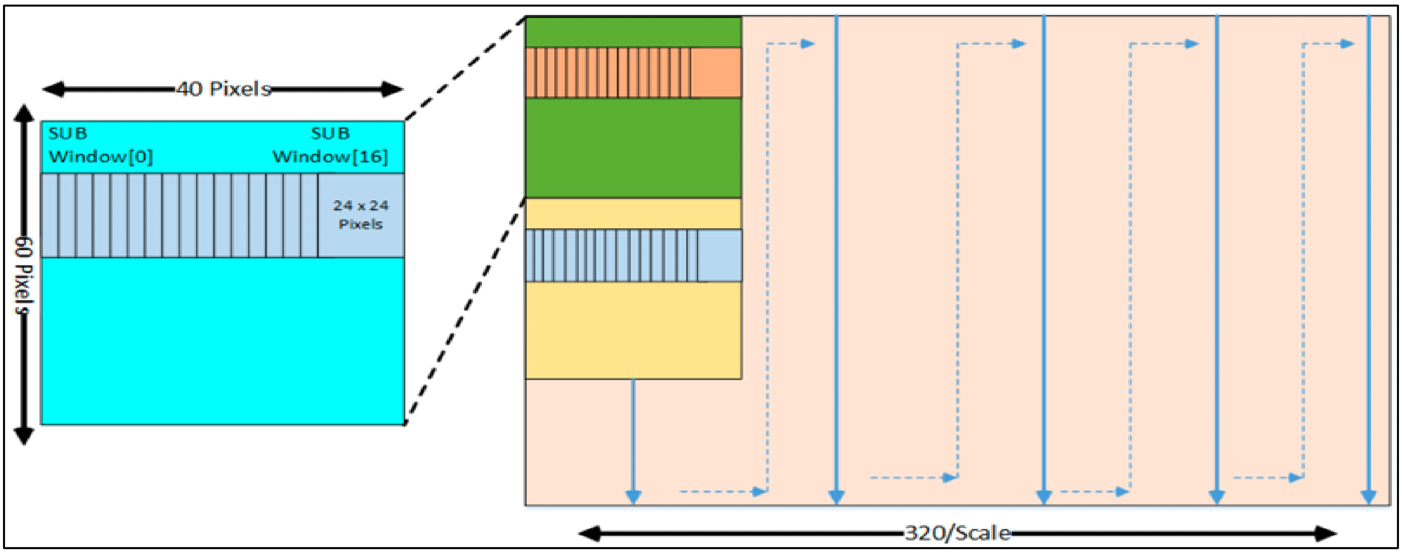

Fig 17 Sub-Window Scanning and Integral Image

➢ *Capturing of Input Image and Saving it into BRAM:*

The input to the system which is an image, and it is captured by the camera module OV7670. The image is captured by using an on-board switch which is connected to one of the pins of the FPGA. The captured image is saved into the Block RAM which have a 320X240 = 76800 addressable memory addresses and each memory location can save 12 bits of data (4: 4: 4 = R: G: B). Here dual port BRAM is instantiated in order to save the original image. Out of two ports, one port is used to just read the original image in order to convert it into the integral image and save this integral image in some other memory. Another port is used to make the box over the detected faces in the original image.

➢ *Generation of Integral Image:*

The next stage of this implementation is the generation of the integral image. The integral image is generated for a portion of the source image in order to use minimum memory resources. The integral image at any location (x,y) is the summation of gray scale pixels above and to the left of (x,y).

The 12 bit RGB input image is taken from the image buffer and convert it into grey image which is used for the generation of the integral image. The integral image square is also generated for the purpose of the calculation of variance normalization factor. This generator uses accumulators and recursive computation to obtain the resultant integral image. For the current row of location (x,y) an accumulator is used to compute the sum of gray scale pixels value. If the current row is not the first row of the source image then it must be added with the previous rows (x,y-1) integral image value in order to get the correct integral value at location (x,y). Here multiplexer is used to select the pixel summation of the first row of the image. After the first row of the image, integral image generator requires the summation of the pixels of the previous rows also so for that we take the data back from the memory and then add it to the summation of the pixels value of the current row. For the generation of the integral image the block diagram is shown in the figure:

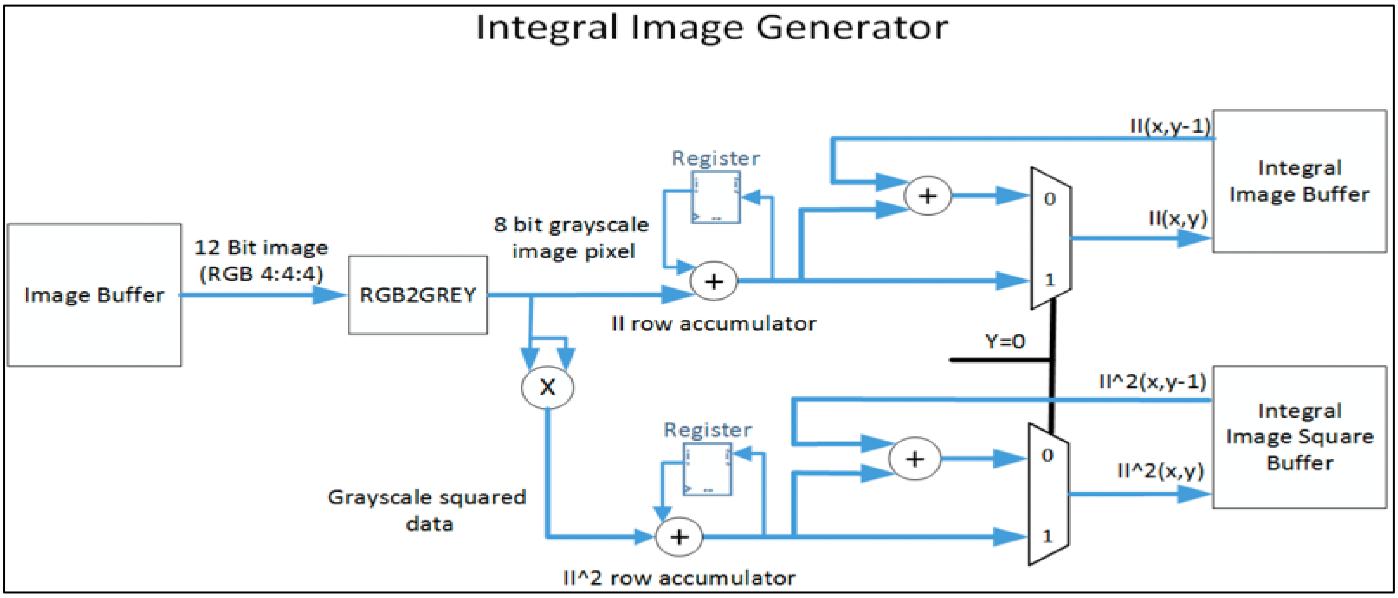

Fig 18 Block Diagram of Integral Image Generator

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

The simulation waveform is shown in figure below for the integral image generation. In this figure wrdata_buff_2A[19:0] shows the waveform for integral image and wrdata_buff_3A[27:0] shows the waveform of integral image square.

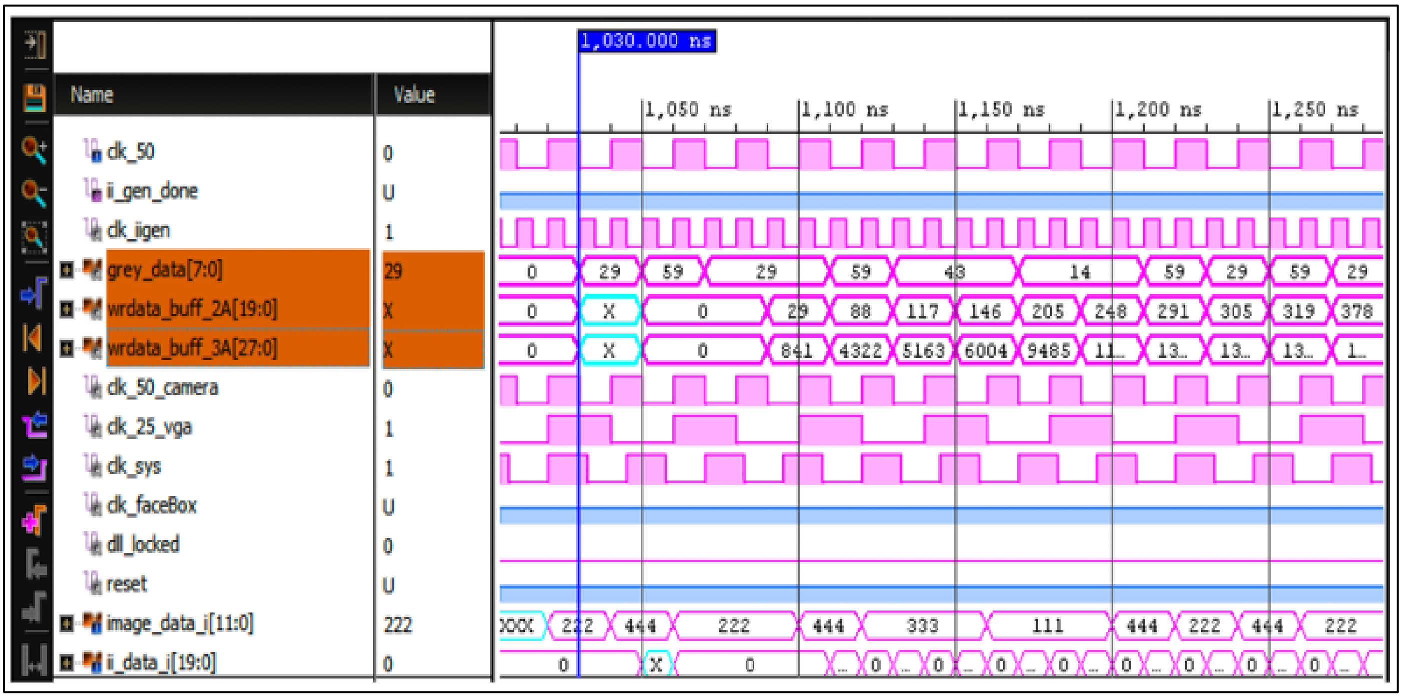

Fig 19 Simulation waveform of Intergral Image Generator

Figure below shows the computation time of an integral image. The high pulse shows that the computation of an integral image is done, and the low period of the pulse shows the computation of the integral image.

Fig 20 Integral Image done Signal and Computation Time of Integral Image

On Zedboard a GPIO pin is used to get the signal and displayed on a DSO. Table below shows the computation time of integral image used.

Table 7 Integral Image Computation Time

Integral image size	Integral image computation time
39X59	60 micro-sec

➢ *Processing of Sub-Window to Find Face*:

A cascaded classifier is used to remove the non-faces and detect the faces in an image. This cascaded classifier is a trained chain of facial features. Different feature evaluations have been done throughout 22 strong stages in a cascaded manner. The evaluation of feature is done in a 24x24 pixel subwindow area. Figure below shows the chosen window area for the subwindow processing. The sum of grey scale pixel values within these rectangular areas are basically used to obtain the difference between the dark and light regions in human faces.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

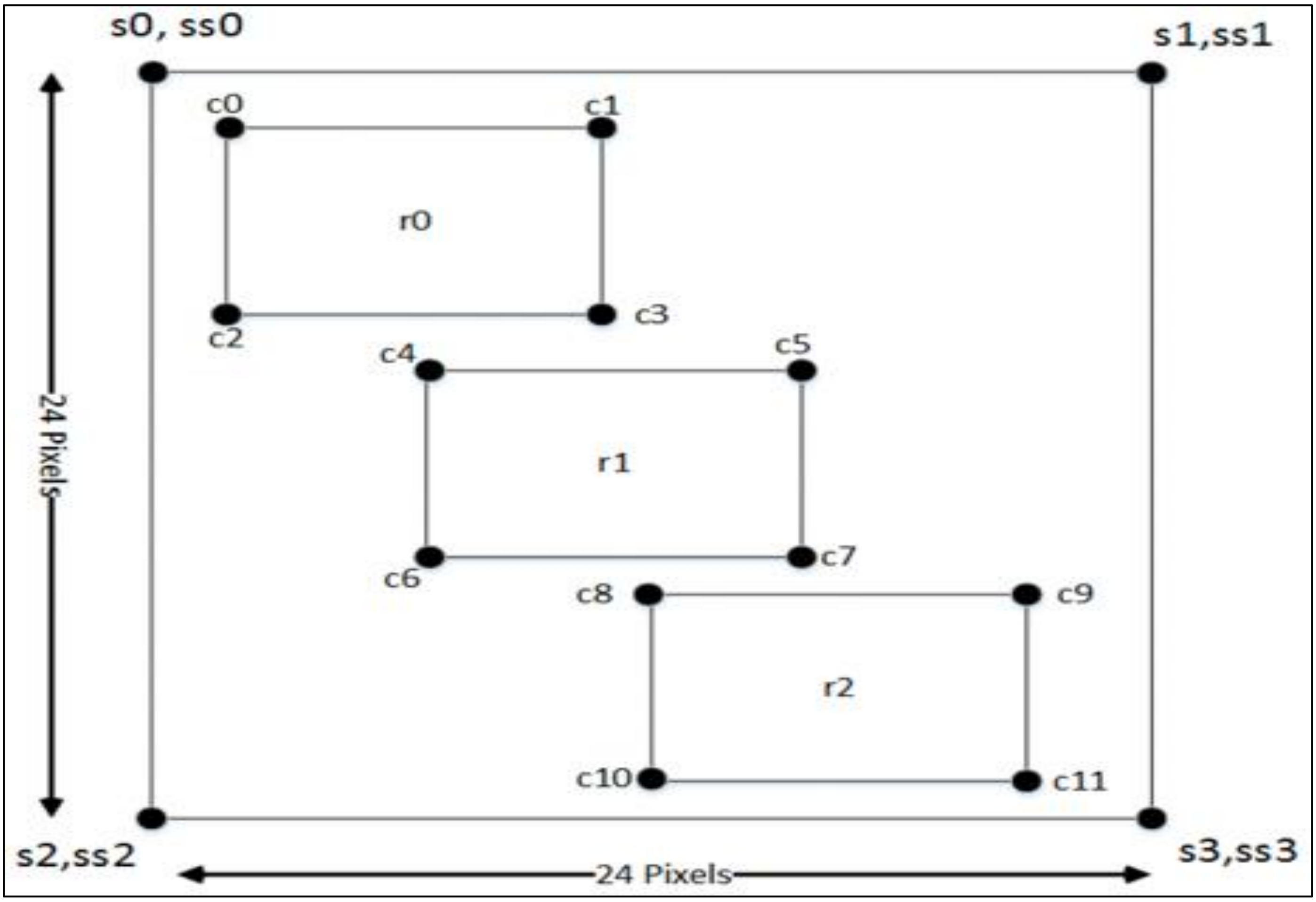

Fig 21 24x24 Window Area

After the features calculation, the accumulated values are compared with the threshold of the strong stage. If this threshold is crossed by the accumulated value than the currently evaluated subwindow is considered for having the face element and passed to the next stage for further processing. If this threshold is not crossed by the accumulated value than a non-face is detected and the currently evaluated subwindow is rejected and then the processing of next subwindow starts. If the subwindow crosses the last stage of the cascaded structure without detecting any non-face, then the subwindow is determined to contain a face. Figure below shows the sequential processing of a subwindow in a cascaded classifier structure.

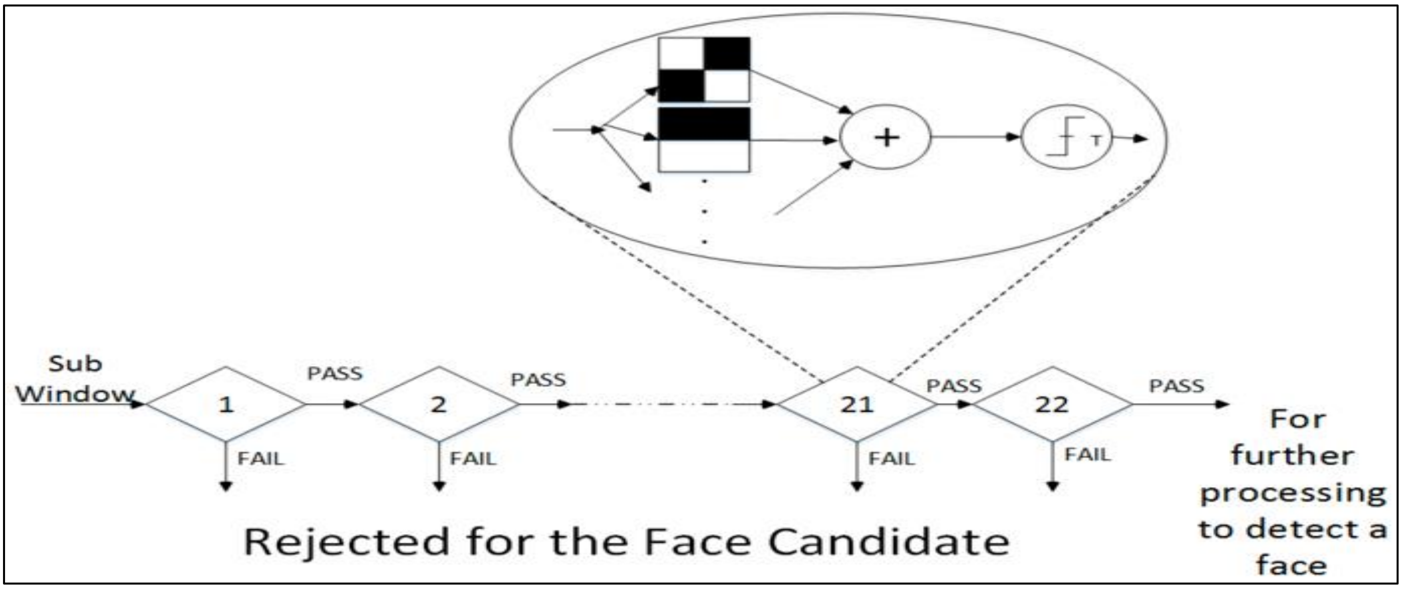

Fig 22 Cascade Classifier

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

Figure below shows the processing of subwindow in which calculation of features and its comparison with the weak threshold and the strong threshold is shown. This implementation shows two data paths, first is for variance normalization of subwindow and second is for feature evaluation.

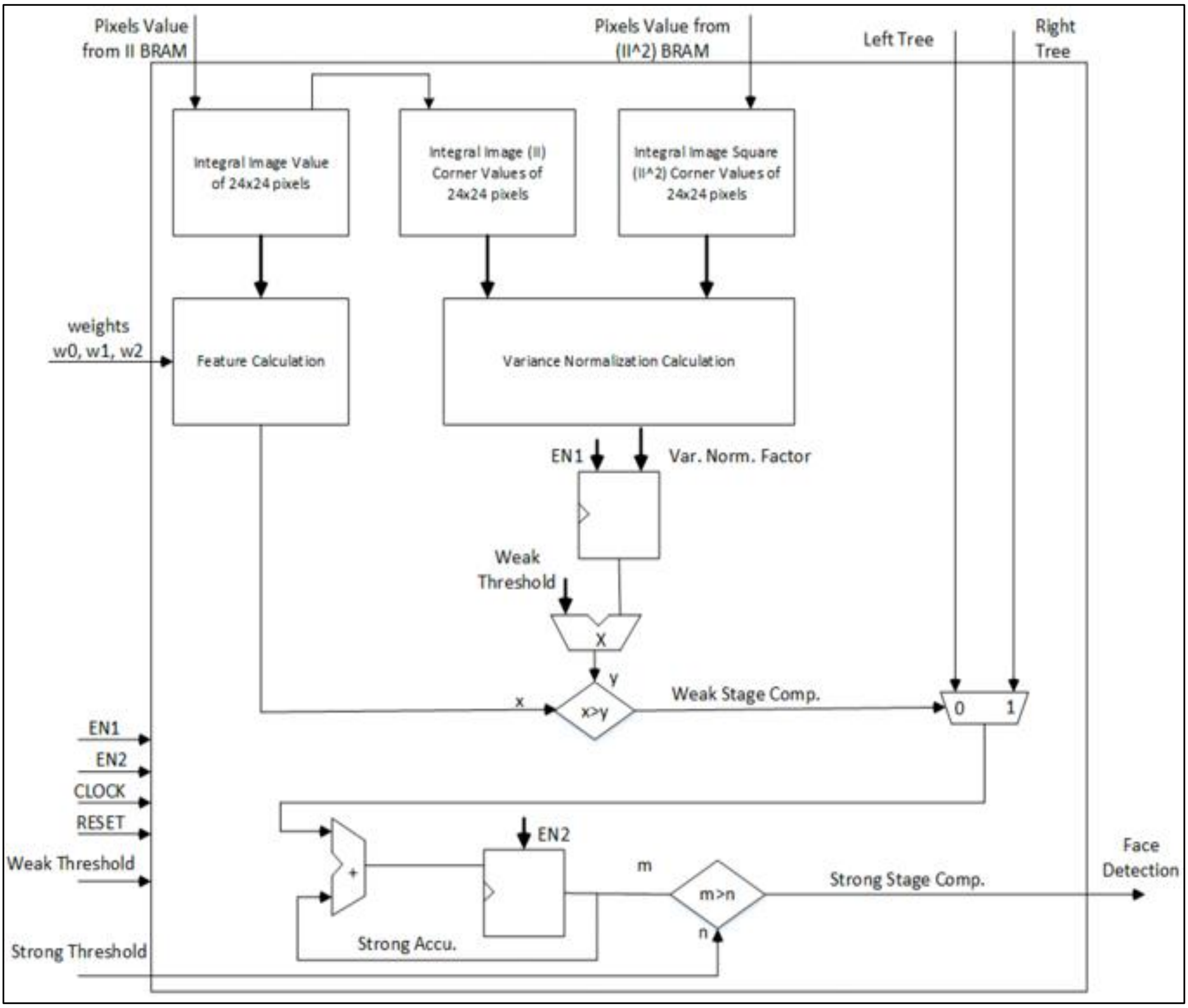

Fig 23 Processing of Sub-Window

In order to bring the light level of the subwindow to the light levels of the training images, variance normalization is used. The formula used to calculate the variance normalization factor (VNF) is given by the equation below:

$$VNF = \sqrt{m^2 - \frac{1}{N}\sum(p^2)}$$

$$VNF = \sqrt{[(s0 + s3) - (s1 + s2)]^2 - \frac{1}{512}[(ss0 + ss3) - (ss1 + ss2)]}$$

Here mean (m) is obtained from the integral image (s0 : s3) and the sum of the square of pixel's value (p2) is obtained from the integral image of squared pixels (ss0 : ss3). In one subwindow total pixels are 24 x 24 = 576 but for the value of N, we have taken 512 for ease in the division by using right shifting of bits by 9 places.

The second data path is for calculation of feature(f). Integral image representation helps to get the quick summation of the pixels value within a rectangle. Equations below shows how to calculate the feature(f) value.

$$f = w0 * \sum P_{r0} + w1 * \sum P_{r1} + w1 * \sum P_{r2}$$

$$f = w0 * [(c0 + c3) - (c1 + c2)] + w1 * [(c4 + c7) - (c5 + c6)] + w2 * [(c8 + c11) - (c9 + c10)]$$

According to the normalized weak threshold, a left tree and right tree value is collected into a register for strong threshold comparison at the end of a stage.

- *Face Box Creation:*

Face box creation stage draws the red box around the detected faces in the source image. To draw the box over face, the actual position (x and y position) and scale values are required. A red box can be drawn by easily changing the pixel colour of the desired pixel to the red. For this, a memory is used in first in first out manner. It stores the detection from all the 16 subwindow. When this subwindow process is completed for all the scales ofthe image, desired pixel value of source image in image buffer is changed to the red colour.

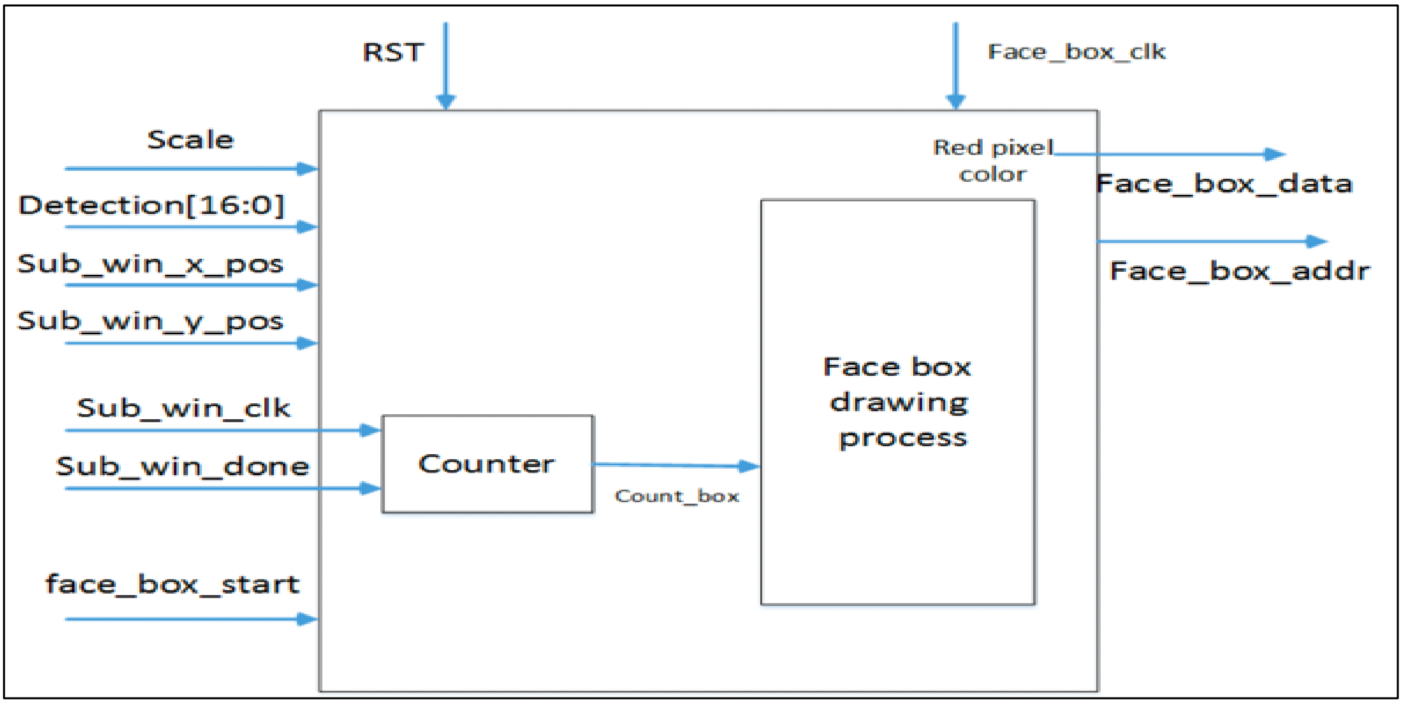

Fig 24 Face Box Creator

- *Performance Measurement of Implemented Face Detection System on FPGA:*

The performance of implemented system on FPGA is measured as the total time taken to detect the face. To obtain the performance of FPGA based face detection system FPS is measured. FPS is nothing but the number of frames detected per second. In order to compute the FPS of system, time duration between capturing an image to displaying the detected result is measured. More accurately the time between the starting of the frame capture to the end of face box drawing on the detected faces is measured. On Zedboard FPGA a register is used to get the FPS. The output of the register is mapped with a GPIO pin of the FPGA. A DSO is used to measure the time duration after which the output of the register become high. The waveform of face detection is shows in figure below which is taken from the GPIO pin. The low period of the signal shows the processing time of the face detection. When the system detects the face after processing then the signal become high.

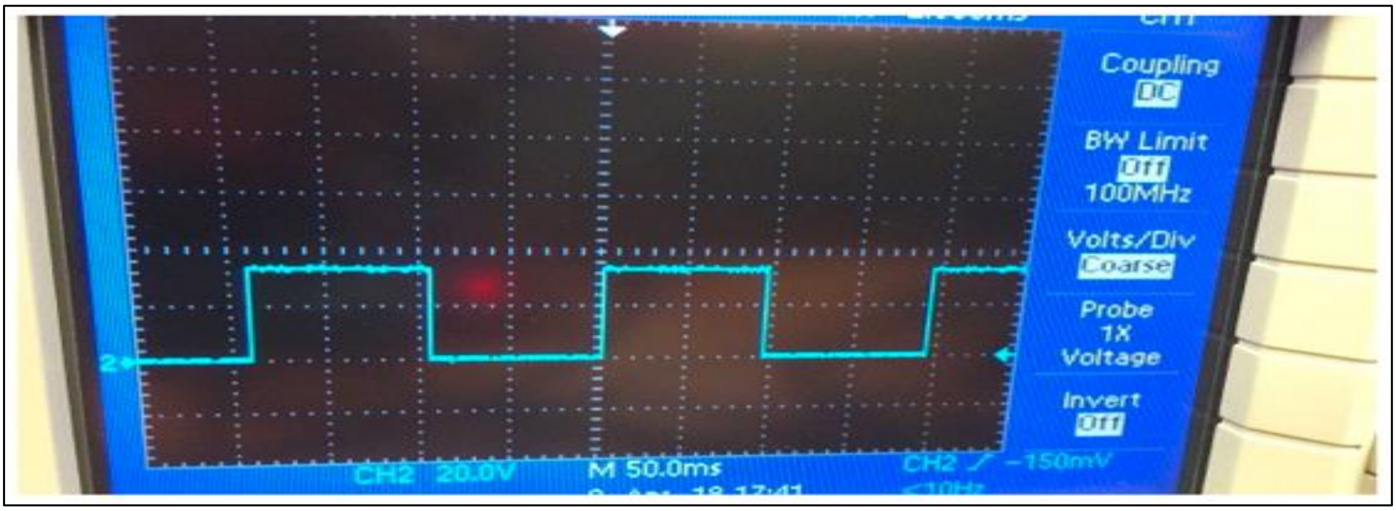

Fig 25 Waveform of Face Detected

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT808

In the above shown figure above one single square is of 60us. So, the total time for which the signal is low is approx. 130us. The time taken to detect the face, and the detection frequency is shown in the table below:

Table 8 Face Detection Time

Input image size	Detection time	Detection frequency	FPS
320x240	130ms	7.69Hz	7.69

VIII. CONCLUSION

There are several algorithms available for the face detection. Selection of the algorithm entirely depends upon the requirement of the time. If we require a system which can detect the faces in an image with high accuracy, then it leads to more computation and thus requires more powerful hardware. If the hardware is not powerful, then we need to compromise with the accuracy. In this paper Viola-Jones algorithm is used for the implementation which gives a high accuracy, but it requires more computation.

Viola-Jones algorithm is implemented for face detection on MATLAB and then by using VHDL it is implemented on Zedboard FPGA. By MATLAB implementation and simulations, we will verify that how accurate Viola-Jones algorithm can detect the faces. The accuracy of the MATLAB implementation of viola-jones algorithm is 86.67%. For the hardware implementation the algorithm is developed in VHDL and implemented on Zedboard FPGA. The detection rate given by the hardware implementation is measured in processed frames per second and its value is obtained as 7.69 FPS.

ACKNOWLEDGMENT

I would acknowledge my professor, guide and friends who has supported work.

REFERENCES

[1]. A. Sharifara, M. S. M. Rahim, and Y. Anisi, "A general review of human face detection including a study of neural networks and haar feature-based cascade classifier in face detection," in Biometrics and Security Technologies (ISBAST),2014 International Symposium on, Aug 2014, pp. 73–78.

[2]. Z. Guo, H. Liu, Q. Wang, and J. Yang, "A fast algorithm face detection and head pose estimation for driver assistant system," in 2006 8th international Conference on Signal Processing, vol. 3, 2006.

[3]. X. Zhu, D. Ren, Z. Jing, L. Yan, and S. Lei, "Comparative research of the common face detection methods," in Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on, Dec 2012, pp. 1528–1533.

[4]. D. Ghimire and J. Lee, "A robust face detection method based on skin color and edges," Journal of Information Processing Systems, vol. 9, no. 1, pp. 141–156, 2013.

[5]. Z. Li, L. Xue, and F. Tan, "Face detection in complex background based on skin color features and improved adaboost algorithms," in Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on, vol. 2, Dec 2010, pp.723–727.

[6]. X. Zhao, X. Chai, Z. Niu, C. Heng, and S. Shan, "Context constrained facial landmark localization based on discontinuous haar-like feature," in Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on, March 2011, pp. 673–678.

[7]. W.-C. Hu, C.-Y. Yang, D.-Y. Huang, and C.-H. Huang, "Feature-based face detection against skin-color like backgrounds with varying illumination," Journal of Information Hiding and Multimedia Signal Processing, vol. 2, no. 2, pp. 123–132, 2011.

[8]. D. N. Parmar and B. B. Mehta, "Face recognition methods & applications," arXiv preprint arXiv:1403.0485, 2014.

[9]. P. F. De Carrera and I. Marques, "Face recognition algorithms," Master's thesis in Computer Science, Universidad Euskal Herriko, 2010.

[10]. "Pre-trained classifier for face detection."

[11]. "Pointing'04 icpr workshop."