P

ROPOSED METHODOLOGY

**INTRODUCTION**

We intended to develop a system that can translate the sign

language of a deaf user so that an ordinary person can

understand.

A deaf user will express his gesture in front of Kinect device.

Then the system will detect the respective sign language for the gesture and will give the output in monitor as text.

Thus the ordinary user will understand the sign language.

The specific aims of this work are given below:

The specific aims of this work are given below:

The specific aims of this work are given below:

The specific aims of this work are given below:

The specific aims of this work are given below:

The specific aims of this work are given below:

The specific aims of this work are given below:

The specific aims of this work are given below:

The specific aims of this work are given below:

The specific aims of this work are given below:

The specific aims of this work are given below:

The specific aims of this work are given below:

• Use data provided by the Microsoft Kinect device.

• Use data provided by the Microsoft Kinect device.

• Use data provided by the Microsoft Kinect device.

• Use data provided by the Microsoft Kinect device.

• Use data provided by the Microsoft Kinect device.

* Use data provided by the Microsoft Kinect device.
* Recognize a list of basic signs. This list will contain key words. Using these words, the deaf user will be able to transmit what he/she needs and the communication between deaf and ordinary user will be possible.
* By executing a combination of several signs in a row, the user will be able to construct some basic sentences that will make the communication more suitable.
* Design an interactive user interface so that the user will be able to run the application without any previous knowledge.
* The system must work on real time and give an instantaneous output once the sign is executed.

**PROPOSED METHODOLOGY**

Sign language is expressed by gesture. Each gesture contains different word. By combining those words or gesture a full, meaningful sentence is found.

In this project the machine translates each gesture, word by word. Here user can also train the machine to learn gesture for new word which is not available in the database.

Then that word will be included in the database and user can use it later.

Here, two features is added: translation and training.

User first has to determine which feature he/she will use.

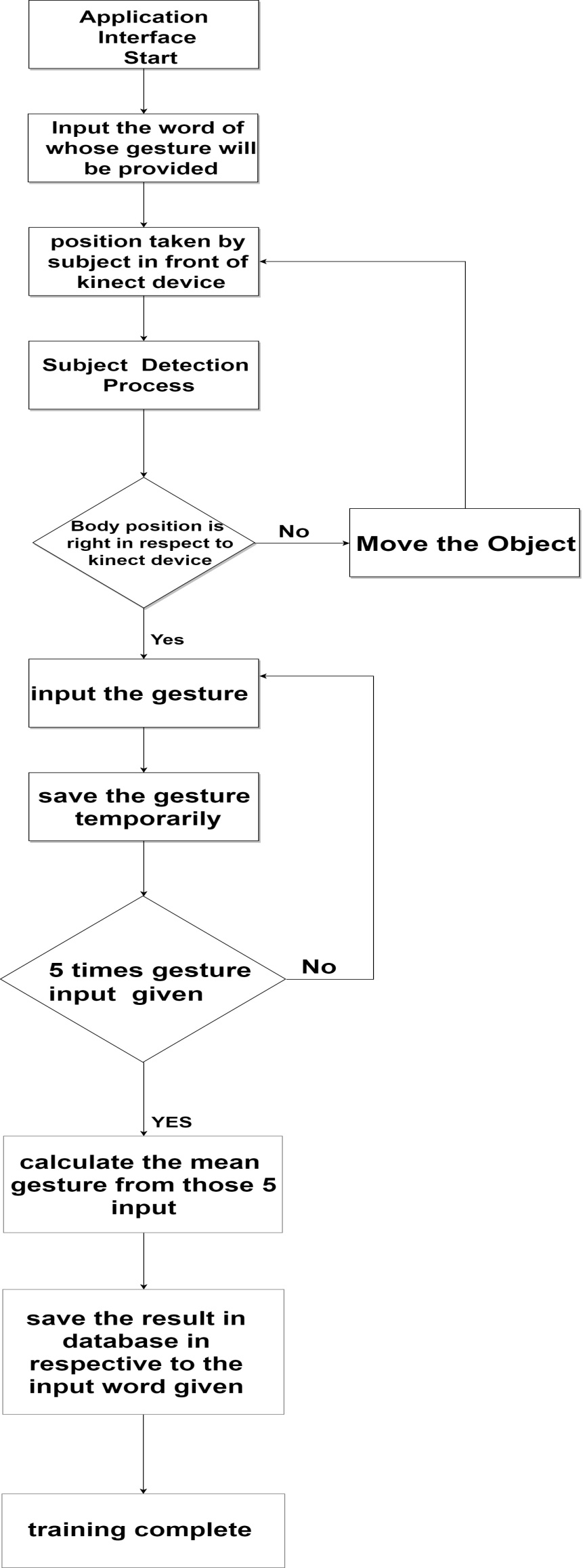
After selecting user have to stand in a pre-defined specific position in front of Kinect Motion Sensor Device.

Then, the initial position of human will be detected based on the depth image.

After that frame by frame position is updated according to the depth data which results in tracking the human.

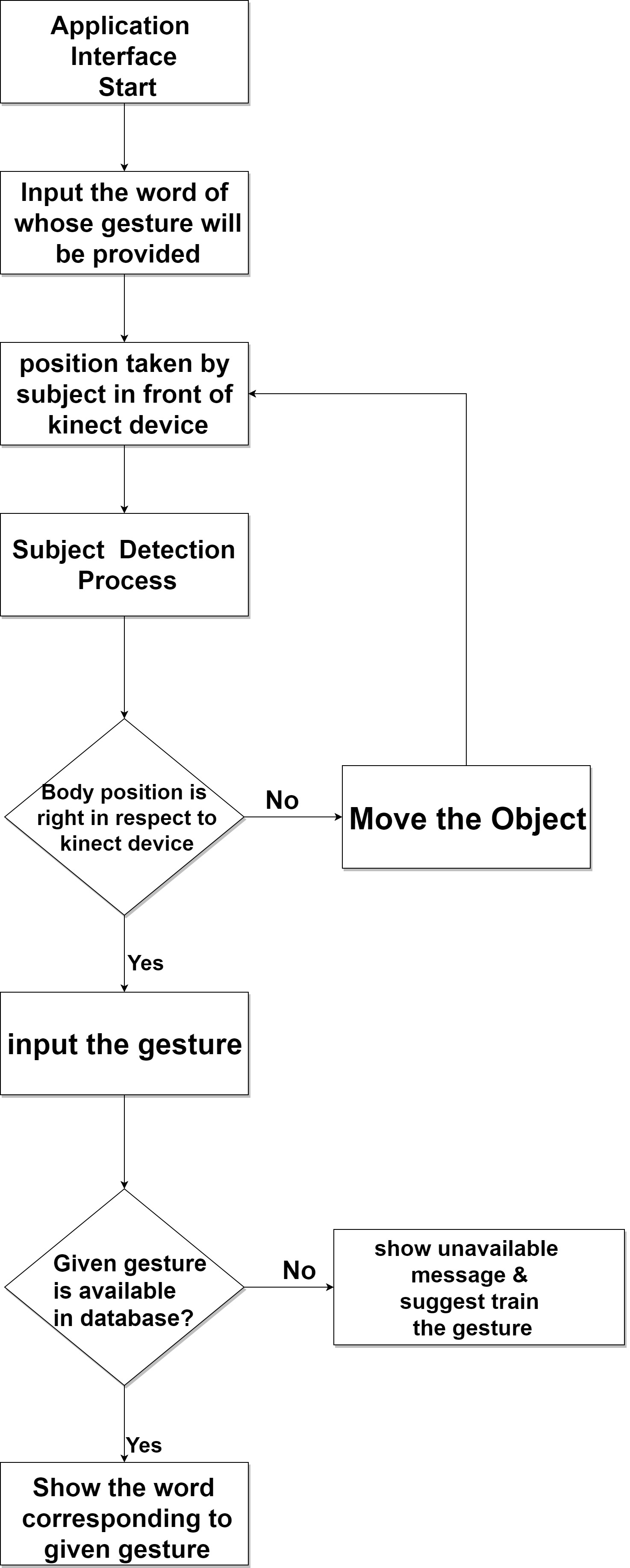
After detecting the position correctly system will instruct the user to provide the gesture.

Than the system will capture the gesture according to the algorithm and will show complete message after capturing the gesture.



Here, fig. 1 shows the proposed

framework for the training mode



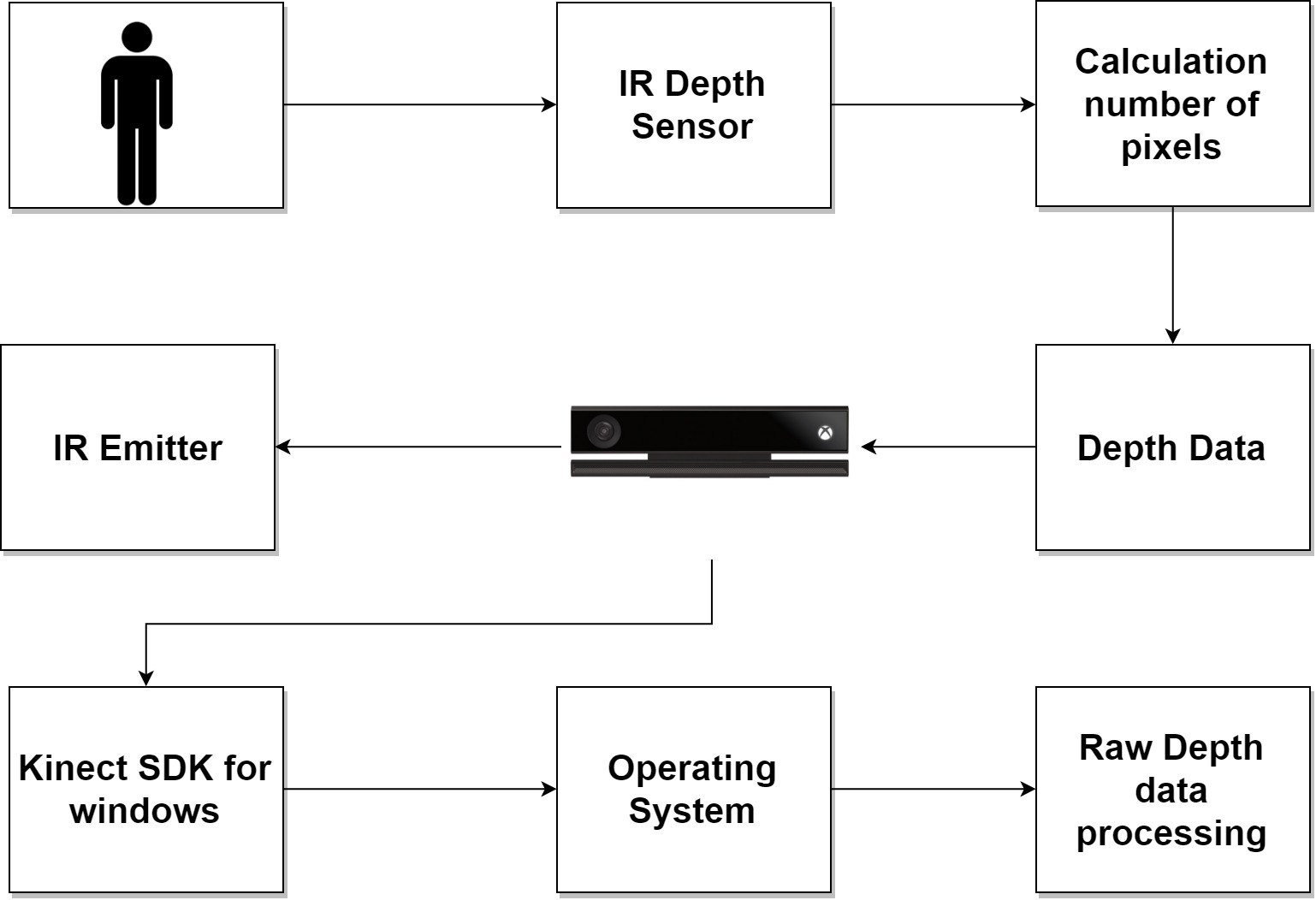
while fig. 2 describes the

translation mode framework.

**A.Subject Detection Process:**

Human detection is one of the internal features of Kinect Motion Sensor Device.

The machine does it by capturing the depth information of human body. The infrared (IR) emitter and infrared (IR) depth sensors of Kinect are the key point to capture the depth information of human. A depth image of human is produced by Kinect in front of it. Both IR emitter and the IR depth sensor use a combined function to obtain the X, Y and Z co-ordinate values on specific point of the detected human.



This figure shows the methodology of

determining a human in front of

the machine.

Here 10 joint of subject is considered. Those are Head and Neck and Shoulder, Elbow, Wrist and Palm of both hands

**B. Depth Data Processing:**

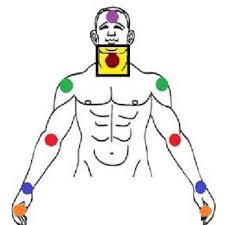
A model based method is used for processing the depth

data. Here, human is detected using depth information obtained by Kinect using a 2-stage head detection process. It includes a 2D edge detector along with a 3D shape detector.

**C.Human position and gesture detection :**

Next part is gesture detection and the user have to stand straightaway to the Kinect sensor. The correct position of human follows two conditions:

1. User have to stand in between 2-3 meters from the sensor. This distance can be measured by Z co-ordinate calculated by the machine.



1. The neck joint of the user should always be inside of a yellow box on the screen. The position of yellow box is pre-defined and fixed in the screen. The program can’t translate or train if the neck joint remain outside the yellow square box.

pre-defined and fixed in the screen. The program can’t translate

or train if the neck joint remain outside the yellow square box.

Once the user stood at the correct position the yellow box

will disappear and a 15\*15 virtual square graph will appear

behind the body in the screen

Once the user stood at the correct position the yellow box

will disappear and a 15\*15 virtual square graph will appear

behind the body in the screen

Once the user stood at the correct position the yellow box

will disappear and a 15\*15 virtual square graph will appear

behind the body in the screen.

One of the most important thing is that that the graph size varies upon the height of the user.

A person’s height can be approximately measured by the distance

between the end of middle finger of his/her two hand when

he/she stands by stretching out his/her both hand on two side.

In this project this concept is used to determine the height of

human using the following equation:-

L = (s1 ~ s2) + (s1 ~ p1) \* 2

Here, L=Length of human.

s1~s2= Distance between the two shoulder.

s1~p1=Distance between shoulder and palm of any hand.

After measuring the height the horizontal/vertical length of the whole graph and length of the each square in the graph is calculated. This calculation is only valid for this specific user.

For other user the graph length will vary following the same process.

Equation 5 and 6 defines the horizontal/vertical length

of total graph and square length respectively.

G1 = L / 15 \* 16 (5)

G2 = G1 / 15 (6)

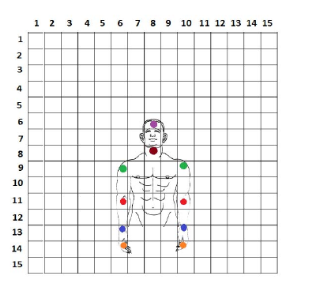
Here, L= Length of human

G1= Horizontal/vertical size of human

G2= Square length

Now user can provide the gesture.

This program can detect sign language word by word.

Even though for every different word of sign language gestures are different but the initial and final position of the body for each gesture should always be same for this project which is shown in the following figure :-

**D. Database Management** :-

The database contains the word with its sign language or

gesture. In case of translation mode the input gesture given by

the user is matched from the database and shows the respective word. In case of training mode the gesture input by the user is saved in the database corresponding to the given word.

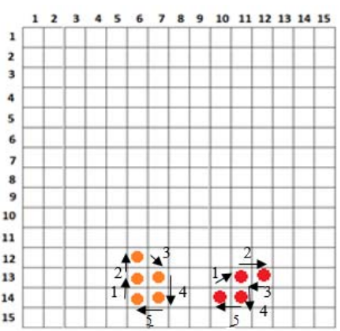
Here the sequence of each joint except head and neck is taken in consideration for creating the database. In training mode a user completes a gesture the sequence movement is saved in two matrixes in the database. These two matrixes are for separate hands. Each matrix consists of four row and variable number of column.

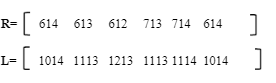
One row contains the movement sequence of one specific joints. The four row contains movement sequence of shoulder, elbow, wrist and palm from top to bottom.

In database for every word two matrix is stored. During storing the position of a joint both co-ordinate of x and y are combined by using following equation :-

M = X \* 100 + Y (7)

For example if the position of joint is captured in co-ordinate (6, 14) the combine position will be considered as 614. In these tow figures a sample movement of palm of both hand and the corresponding matrix in respect to the movement of the palm is showed. Here red highlighted mark for the left hand and orange for the right hand.





In this way the sequence for shoulder, elbow and wrist is

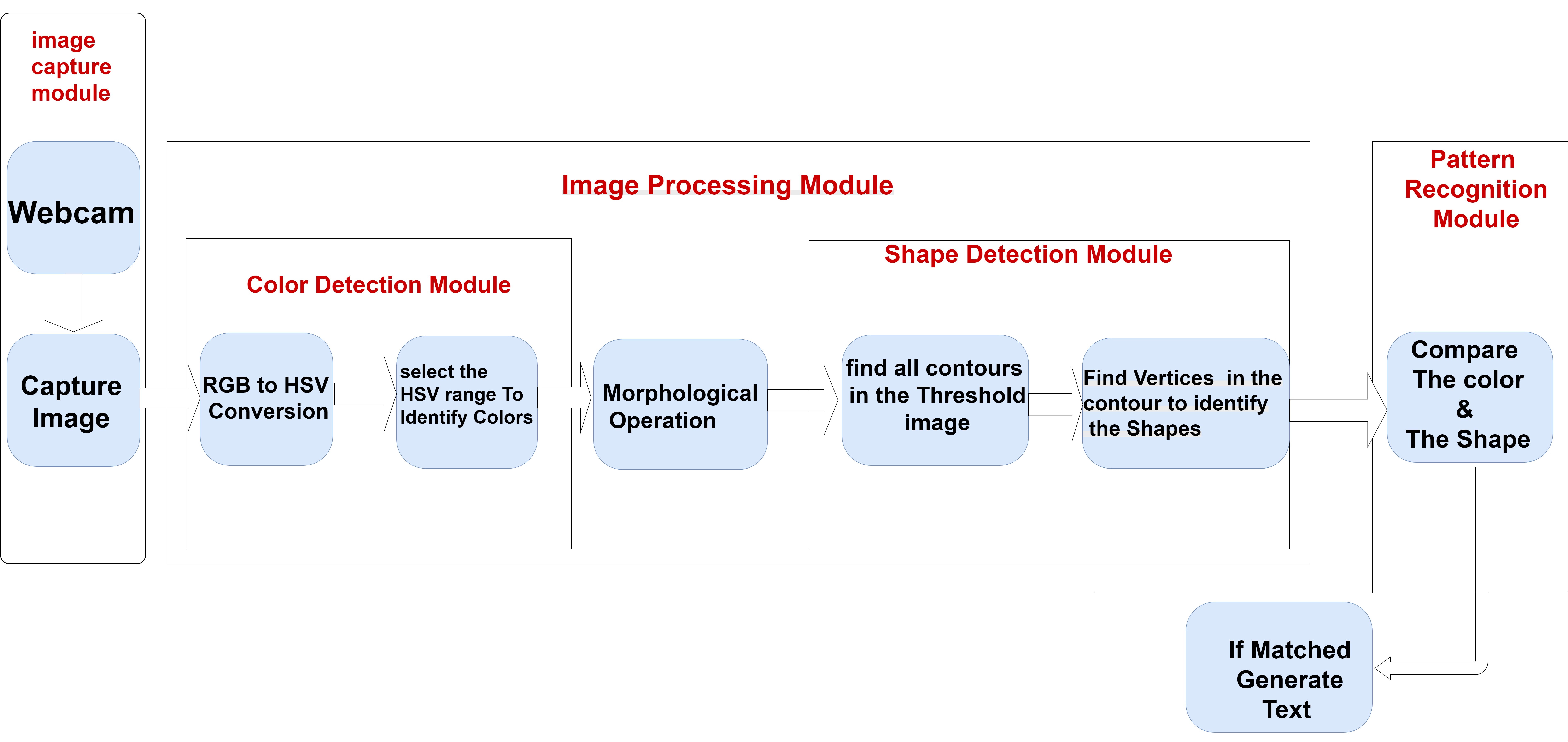
saved in row 1, 2 and 3 respectively in the matrix for both

hands. The two matrixes are saved corresponding to the input

word given.

**System Architecture**

* The specific aims of this work are given below:
* The specific aims of this work are given below:
* The specific aims of this work are given below:
* The specific aims of this work are given below:
* The specific aims of this work are given below:



**We have 3 modules:**

* **Image capture module.**
* **Image processing module:-**
* **Color detection module:-**
* **RGB to HSV Conversion.**
* **Select the HSV range to identify Colors.**
* **Morphological operation.**
* **Shape detection module.**
* **Find all contours in the threshold image.**
* **Find vertices in the contours to identify shapes.**
* **Pattern recognition module.**
* **Compare the color & the shape.**
* **If matched ,generate text.**

**Each module in details:**

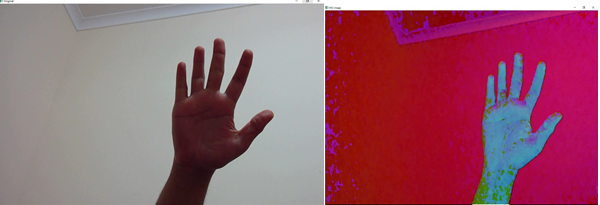
**-Image capture module:-**

Image taken by the webcam.

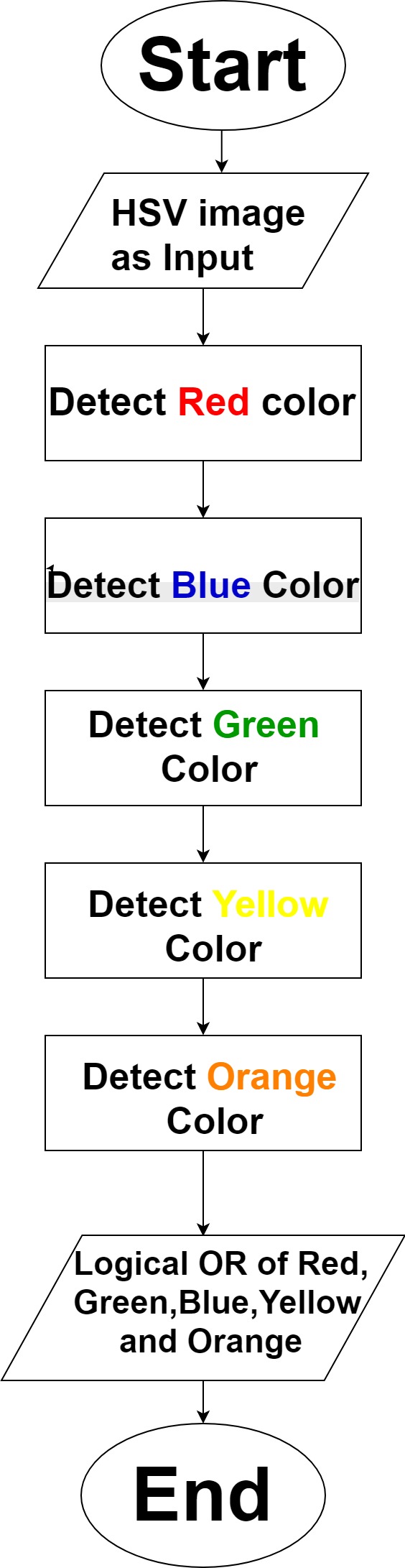
**-Image processing module:-**

-**Color detection module:-**

**1-RGB to HSV Conversion:-** In Open CV, value for ‘hue’, ’saturation’ and ‘value’ are respectively 0-179,0-255 and 0-255.



**2- Select the HSV range to identify Colors:-** each pixel of hsv image(src) is compared with the corresponding predefine lower (lowerb) and upper(uppeerb) HUE, saturation and VALUE values to detect a single color.



**Hue value for RED is (160-179)**

**Hue value for Blue is (105-120)**

**Hue value for Green is (45-65)**

**Hue value for Yellow is (23-38)**

**Hue value for Orange is (5-18)**

**Logical OR** of all the binary

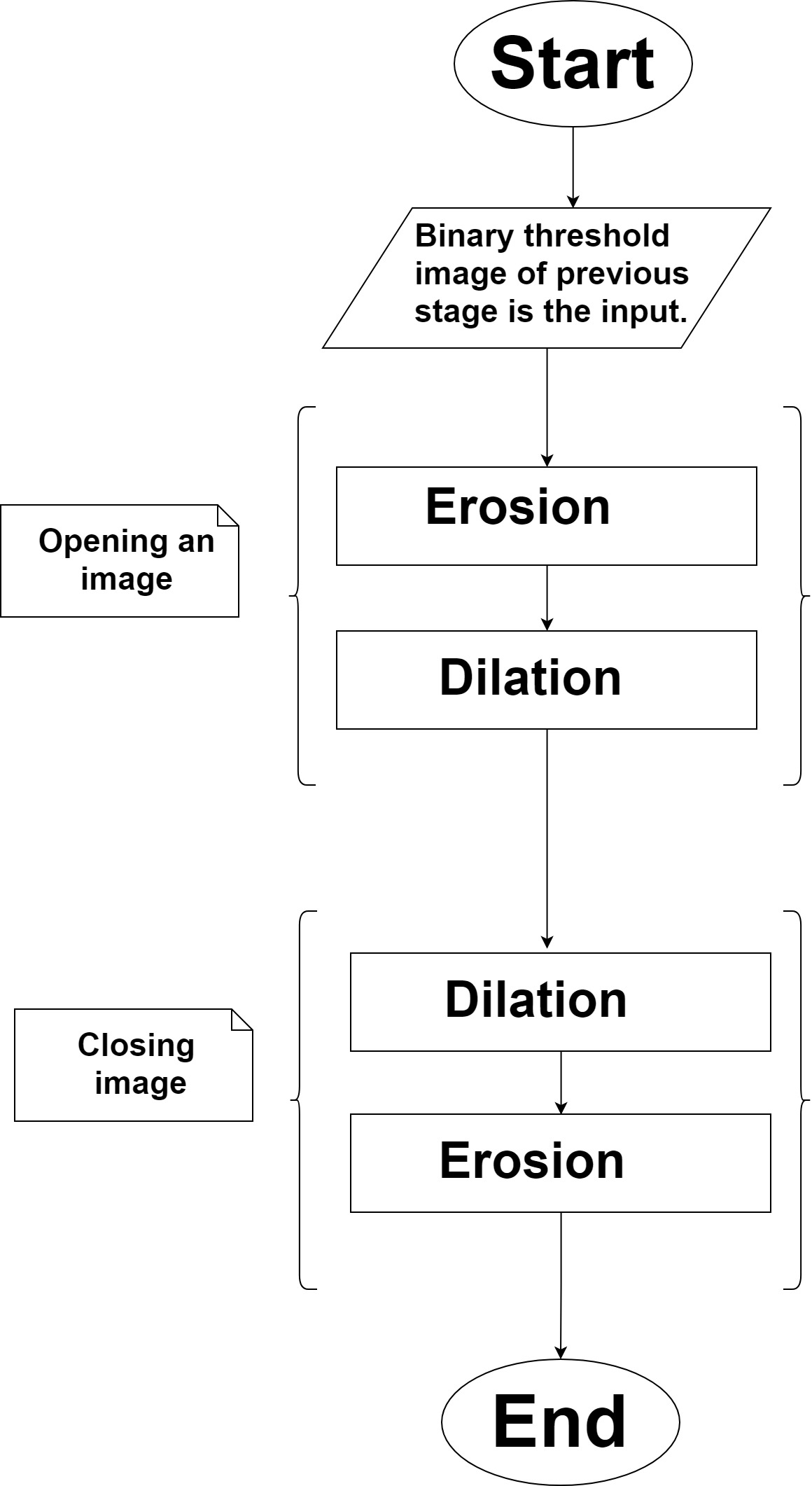
Output of all the detected

Color to make all the detected

Color in the same frame

**-Morphological operation:-**

**Erosion:** operation



examines the value of a

pixel and it’s neighbors

and sets the output

value equal to the

minimum of the input

pixel.

**Dilation:** operation

examines the value of a

pixel and it’s neighbors

and sets the output

value equal to the

maximum of these

pixels.

**-Shape detection module:**

**Find all contours in the threshold image:**

**Contour** is a list of points that represent a curve in image.

**Contours** are represented in openCV by sequences in which every entry in the sequence encodes information about the location of the next point on the curve.

The function **findContours()** compute contours from binary images.

**Find vertices in the contours to identify shapes:-**

