ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

Parallel Algorithm for Image Recognition and Retrieval from Human Frontal Face Image Database

Gaydhankar Roma Ravindra[1], Shaikh Zakir Mujeeb[2]

M.E Student, Dept. of Computer Science and Engineering, NKOCET, Solapur, Maharashtra, India[1],

Assistant Professor, Dept. of Computer Science and Engineering, NKOCET, Solapur, Maharashtra, India[2]

ABSTRACT: Face detection and recognition are high demanded research area. There are many techniques available to detect and recognize faces with its advantages and disadvantages. Everyday images are becoming larger in sizes because of increasing in its number of pixels. However the face detection and recognition takes more time to execute and produce results while dealing with large size image databases.
The objective of this paper is to increase the performance of algorithm for recognizing human frontal face. In proposed algorithm reduce the image searching time from the database. The parallel algorithm is used to increase the performance with B+ tree. The proposed system accepts person's name or the image containing human frontal face as an input. Then it searches images for the input and produces the desired output.

KEYWORDS: Face detection; face recognition; haar Cascade Classifier; image histogram comparison; parallel algorithm; OpenCV.

I. INTRODUCTION

The study of increasing the speed of searching images for the particular human face is one of the challenging domain in the research community. The goal of our project is to design and implement an application for automatic human face detection and recognition from the input image and searching the images with the same face given in input image from frontal human face image database in minimum time. The result of a proposed system is of great importance for broad area of applications like 'finding duplicate images in database', 'finding images from criminal records' etc. where time is important for retrieving images.

II. RELATED WORK

There are many applications available in the market for searching images of a particular person from the image database by giving input of his/her name or by uploading his/her image[6]. But the result of these applications contains irrelevant data also which is not needed to user. That means the images of other person whose name is not mentioned in input is also displayed as a result.

An image retrieval system is a technique used in computer system for searching and retrieving images from a large set of database of digital images. Most common and traditional methods of image retrieval used some techniques for adding metadata such as keywords, captions [2]. By doing this, the retrieval of image can be done based on these annotated words. Manual image annotation consumes larger time, laborious and expensive too. To address this issue, large number of research has been done on automatic image annotation. In addition, social web applications and semantic web increasing rapidly, this has been inspired the development of various web based image annotation tools. The rapid evolution of the Internet and the explosive growth of the visual contents of the Web, image search in large scale have attracted considerable attention. The ability of fast similarity search of an image in a large-scale dataset is a research issue which is under consideration [1].

The image retrieval in large database through various mechanisms are available [3][4][5]. The existing applications have some disadvantages. Some existing image search applications don't search the images by recognizing the persons

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

in the image but it recognizes the entire image as a whole. That means they search for the duplicate images as the input image. But some of the applications search the images by recognizing the persons in the image but at slow speed. Some applications are trying to increase the speed by various algorithms.

III. PROPOSED METHODOLOGY

Here, we are describing, some important methods used in proposed system. Then the overall architecture of a proposed system is described with the help of a diagram.

A. *Human Frontal Face Detection:*

There are many methods are available for face detection [7]. The proposed system uses Open Source Computer Vision Library (OpenCV) files which already contains many pre-trained classifiers for face, eyes, smile etc. The system uses the following emgu.cv.CascadeClassifier method for detecting Human Frontal Face.

```
CascadeClassifier _cascadeClassifier = new CascadeClassifier ("haarcascade_frontalface_default.xml");
```

B. *Histogram Comparison*

The system uses following histogram comparison method:

```
Emgu.CV.CvInvoke.CompareHist(H1, H2, Method);
```

In above method,
H1 - First compared histogram.
H2 - Second compared histogram of the same size as H1.
Method - Comparison method that could be following:

- Emgu.CV.CvEnum.HistogramCompMethod.Bhattacharyya - Bhattacharyya distance
- Emgu.CV.CvEnum.HistogramCompMethod.Chisqr - Chi-Square
- Emgu.CV.CvEnum.HistogramCompMethod.ChisqrAlt
- Emgu.CV.CvEnum.HistogramCompMethod.Correl - Correlation
- Emgu.CV.CvEnum.HistogramCompMethod.Hellinger - Synonym for CV_COMP_BHATTACHARYYA
- Emgu.CV.CvEnum.HistogramCompMethod.Intersect - Intersection

The functions compareHist compare two dense or two sparse histograms using the specified method.

The proposed system uses the Bhattacharyya distance method.

C. *Parallel Programming:*

Parallel programming [8] is programming that leverages multiple cores and all processors to improve processing speeds by partitioning work into small chunks, execute these chunks in parallel using multithreading and collating the results as they become available - in a thread-safe and performant manner.

PLINQ offers the richest functionality for automating all the steps of parallelization including partitioning work into tasks and collating the results into a single output sequence. PLINQ automatically parallelizes local LINQ queries. It offloads the burden of both partitioning and collating to the Framework. To use PLINQ, simply call AsParallel() on the input sequence and then continue the LINQ query as usual. AsParallel is an extension method in System.Linq.ParallelEnumerable. The Fig. 1 shows the execution of AsParallel( ).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

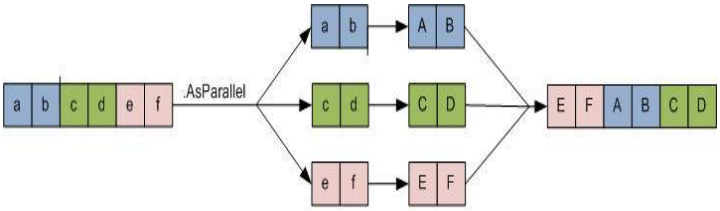

Fig. 1. AsParallel( ) Execution

D. *Design Methodology:*

In Fig. 2, the number of operations and workflow of a proposed system is shown. The description for the Fig. 2 is given below.

In Fig. 2, it is shown that only authorized user can access the proposed system. After successful login, the user can search images, view database, perform operations on database such as add, update, delete and also perform operations on resulted images such as zoom, crop and SaveAs.

Human Frontal Face Image Database Interface provides interface to view database contents, to add new record in that database, to update existing records and to delete existing records. To add new record into the Human Frontal Face Image Database the user has to give an input image which contains at least one human frontal face. Then the system will detect the Human Frontal Face in an input image. With the input image user has to input the Full Name of the person whose frontal face is detected in an input image. After this when user will submit this information to the system, it will add this information into the Human Frontal Face Image Database in form of one record.

Human Frontal Face Image Database contains the Original Color Image which contains at least on Human Frontal Face, Cropped GrayScale Image of Human Frontal Face detected in an input Original Image, Full Name of the person whose frontal face is detected in an input image in form of First Name, Middle Name, Last Name. Each record in Human Frontal Face Image Database is assigned unique identifier named as Image Id.

Image Search Interface provides interface to user for searching images. The proposed system provides users two ways for searching images. First one is Searching Images by Image and second is Searching Images by Name. In Searching Images by Image, the user has to submit input image in which at least one human frontal face is present and set the accuracy percentage for searching images. After submitting this information, the proposed system will detect and crop the Human Frontal Face Image in an input image. Then it calculates the histogram for the cropped Human Frontal Face Image. Then the proposed system compare histogram of cropped Human Frontal Face Image from input image with every Cropped GrayScale Image of Human Frontal Face Image saved in Human Frontal Face Image Database by applying Parallel Algorithm. For detection of Human Frontal Face in an input image, histogram calculation and histogram comparison of an image, the proposed system uses the OpenCV Library. Then the proposed system retrieve and shows only those images from Human Frontal Face Image Database which histogram comparison result matches the user accuracy percentage to prevent unwanted result. In searching images by name, the user can search images of a particular person by submitting his/her first name, middle name or last name. The user can give any of the name from first name, middle name and last name as an input or can give full name as an input for searching images by name. Then proposed system search images in Human Frontal Face Image Database for the given input name by applying Parallel Algorithm. The proposed system will retrieve images from Human Frontal Face Image Database and show to the user which names match with the input name.

In Image Editor Interface the image can be zoomed, cropped and SaveAs which are retrieved from Human Frontal Face Image Database and shown to the user.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

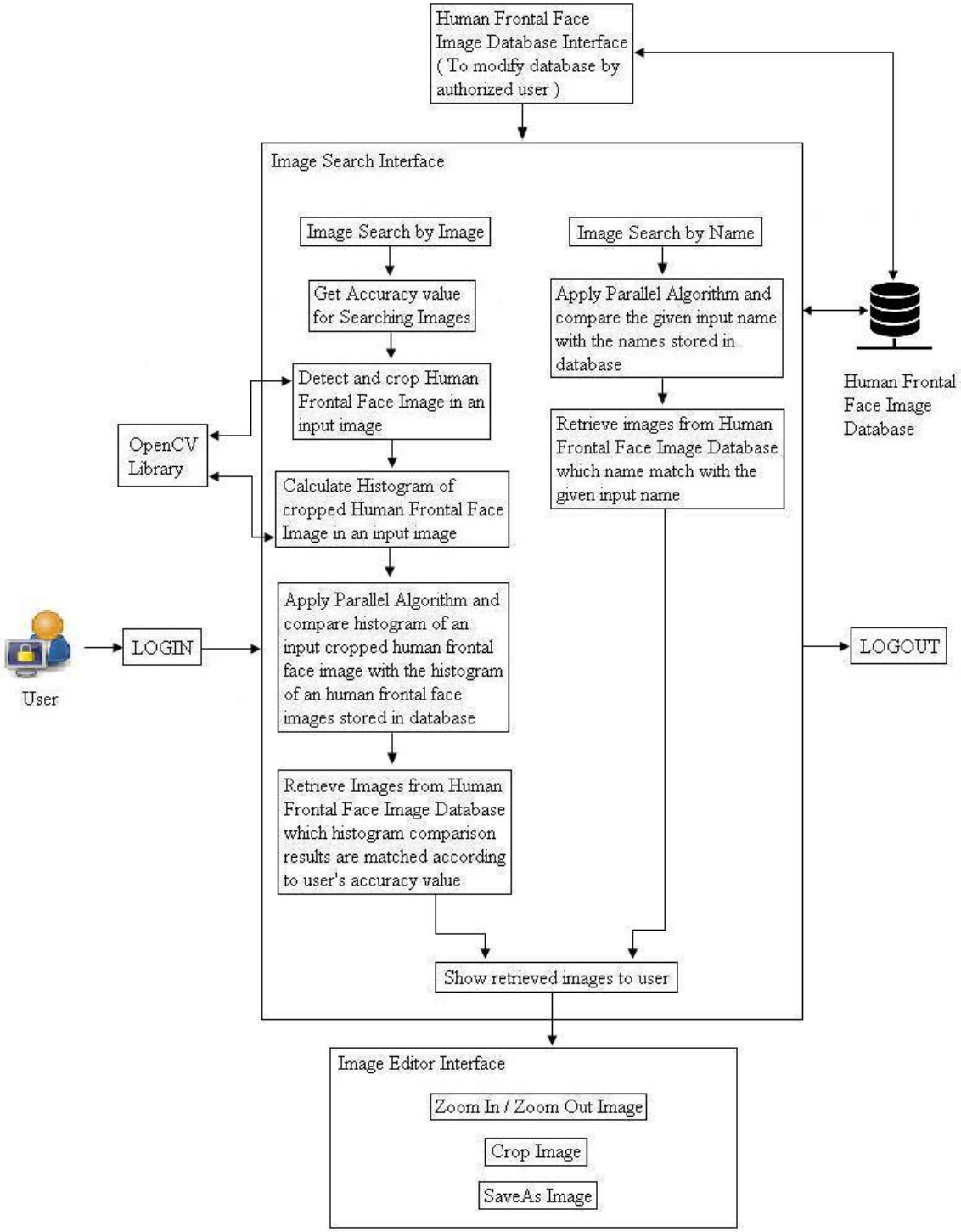

Fig. 2. Overall Architecture of Parallel Algorithm for Image Recognition and Retrieval from Human Frontal Face Image Database

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

IV. IMPLEMENTATION AND RESULT

A. *Pseudo Code for Searching Images By Image :*

```
public void searchByImageParallel()
{
if (imageBox1.Image != null)
{
for (int index = 0; index <noOfImages; index++)
{
pnlSearchedImages.Controls.Remove(picturebox[index]);
}
faceImages.Clear();

/*detect face of input image*/
_cascadeClassifier = new CascadeClassifier("haarcascade_frontalface_default.xml");
if (currentFrame1.Data == null) //currentFrame1 contains input image
return;
using (varimageFrame = currentFrame1)
{
if (imageFrame != null)
{
vargrayframe = imageFrame.Convert<Gray, byte>();
var faces = _cascadeClassifier.DetectMultiScale(grayframe, 1.1, 10, Size.Empty); //the actual face detection happens
//here
foreach (var face in faces)
{
imageFrame.Draw(face, new Bgr(Color.BurlyWood), 1); //the detected face(s) is highlighted here using a box that is
drawn around it them
count = count + 1;
result = imageFrame.Copy(face).Convert<Gray, byte>().Resize(100, 100, Emgu.CV.CvEnum.Inter.Cubic);
faceImages.Add(result.Bitmap);
}
}
}

for (int k = 0; k <faceImages.Count; k++)
{
Bitmap b = (Bitmap)faceImages[k];
result = new Image<Gray, byte>(b);

/* calculate input image histogram*/
hist1 = new DenseHistogram(256, new RangeF(0.0f, 255.0f));
hist1.Calculate<Byte>(new Image<Gray, byte>[] { result }, true, null);

List<Image>listOfImages = new List<Image>();

DataClasses1DataContext context=new DataClasses1DataContext();
var bytes = (from a in context.Test7s.AsParallel().WithExecutionMode(ParallelExecutionMode.ForceParallelism)
      select a).ToList();
foreach (varSampbytes in bytes)
{
byte[] trueBytes = Sampbytes.FImage.ToArray();
```

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

```
if(trueBytes != null)
{
using (varms = new MemoryStream(trueBytes))
{
using (Bitmap img = new Bitmap(Image.FromStream(ms)))
{
result3 = new Image<Gray, byte>(img);
hist2 = new DenseHistogram(256, new RangeF(0.0f, 255.0f));
hist2.Calculate<Byte>(new Image<Gray, byte>[] { result3 }, true, null);
Double compHistRes = Emgu.CV.CvInvoke.CompareHist(hist1, hist2,
      Emgu.CV.CvEnum.HistogramCompMethod.Bhattacharyya);

Double accValue = Convert.ToDouble(cmbbxImageSearchAccuracy.Text);
accValue = 100.0 - accValue;
accValue = accValue / 100;
if (compHistRes<accValue)
{
// add resulted images to list
byte[] b1 = Sampbytes.OImage.ToArray();
Bitmap Oimg = new Bitmap(Image.FromStream(new MemoryStream(b1)));
firstNameL.Add(Sampbytes.FirstName.ToString());
middleNameL.Add(Sampbytes.MiddleName.ToString());
lastNameL.Add(Sampbytes.LastName.ToString());
listOfImages.Add(Oimg);
}
}
}
}
}
pnlSearchedImages.AutoScroll = true;

picturebox = new PictureBox[listOfImages.Count];
int y = 0;
noOfImages = listOfImages.Count;
if (noOfImages == 0)
{
MessageBox.Show("No Similar images found in database!", "Search Images", MessageBoxButtons.OK,
      MessageBoxIcon.Error, MessageBoxDefaultButton.Button1);
}
else
{
for (int index = 0; index <listOfImages.Count; index++)
{
picturebox[index] = new PictureBox();
picturebox[index].BackgroundImageLayout = ImageLayout.Stretch;
picturebox[index].Image = (Bitmap)listOfImages[index];
// Following three lines set the images(picture boxes) locations
if (index % 10 == 0)
y = y + 15; // 3 images per rows, first image will be at (20,150)
picturebox[index].Location = new Point(index * 120 + 20, y);
picturebox[index].Name = "pb" + index;
picturebox[index].Size = new Size(100, 120);
```

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

```
picturebox[index].Click += new EventHandler(picturebox_Click);
pnlSearchedImages.Controls.Add(picturebox[index]);
}
}
}
}

else
{
MessageBox.Show("Please browse Input Image. Input Image is empty.", "Search Images", MessageBoxButtons.OK, MessageBoxIcon.Error, MessageBoxDefaultButton.Button1);
}
}
```

The result of the searchByImageParallel**( )** is shown in Fig. 3. In Fig. 3, **t**he Input for the proposed system contains Input Image and the searching accuracy percentage. Input Image contains single human frontal face given to the proposed system as an input and the proposed system detected human frontal face from Input Image and histogram of that human frontal face image is calculated. Then that calculated histogram is matched with the histogram of human frontal face images stored in Human Frontal Face Image Database. Then images are retrieved from the database which histogram comparison results match the user given accuracy and shown to the user.

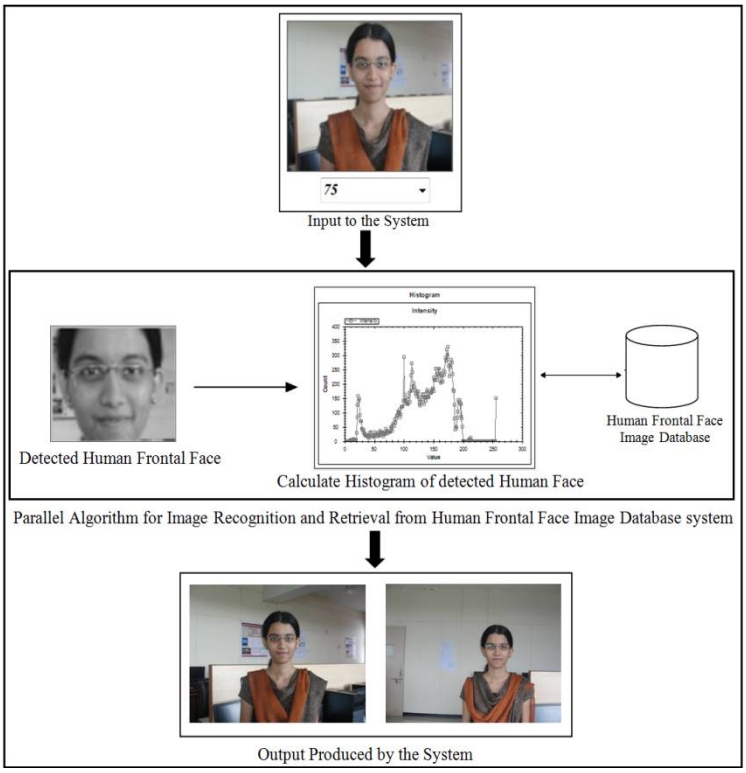

Fig. 3. Input and Output of searchByImageParallel( )

The Fig. 4 shows the Input and Output Image for the Zoom In and Crop Image operations to be performed on Input Image. The Output Image with Zoom and selected Area for Cropping shows Zoomed In Image with the selected area for Crop operation. The user can SaveAs that selected area with other name.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

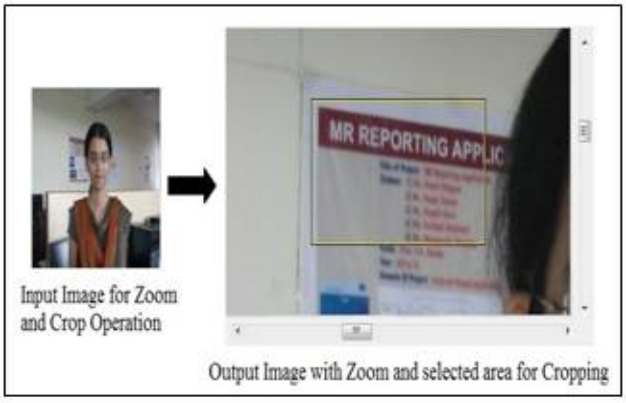

Fig. 4. Zoom and Crop Operation on Input Image

V. CONCLUSION AND FUTURE WORK

We implemented Parallel Algorithm for Image Recognition and Retrieval which can retrieve from Human Frontal Face Image Database system using OpenCV Library and Parallel Algorithm. The system provides authority to authorized user to make changes in the database. The user can Crop, Zoom In/Zoom Out, SaveAs operations on the retrieved images. The authorized user can add new human frontal faces in Human Frontal Face Image Database with its information such as First Name, Middle Name, Last Name. After adding new faces the user can modify its information that is First Name, Middle Name, Last Name and also can delete existing records from the database.

Parallel programming leveraging multicore processors to speed up computationally intensive code. The AsParallel() method of PLINQ is used for implementing Parallel Algorithm. The system is built and executed on Dell Inspiron 7520 having Intel(R) Core(T) i5-3210M CPU @ 2.50 GHz with 4.00 RAM and Windows7 Home Premium 64-bit Operating System. Its processor Quad Core Processor. Therefore our implemented system used four processors and produced results as shown in above Results section.

Now days everywhere CCTV cameras are used widely for the security purposed. The HD CCTV cameras capture 30 frames per second. So choosing the best human frontal face image from those 30 frames is challenging task. Real time human frontal face detection and recognition is a challenge even though we have i7 processor. The future scope of the system is identifying the human frontal face from motion picture and retrieving images for the indentified human frontal face.

VI. ACKNOWLEDGEMENT

This work was done when R. R. Gaydhankar was doing final year project of Master in Engineering in Computer Science and Engineering from Solapur University under the guidance of Z. M. Shaikh.

REFERENCES

1. Weiyan Wang, Yunquan Zhang, Shengen Yan, Ying Zhang, HaipengJia, “Parallelization and Performance Optimization on Face Detection Algorithm with OpenCL: A Case Study”, TSINGHUA SCIENCE AND TECHNOLOGY ISSN 1007-0214 08/18, Volume 17, Number 3, pp287-295, June 2012.
2. Yong Rui and Thomas S. Huang, “Image Retrieval: Current Techniques, Promising Directions, and Open Issues”, Journal of Visual Communication and Image Representation, 10, 39–62, (1999).
3. Ms. Apurva N. Ganar, Prof. C. S. Gode, Prof. Sachin M. Jambhulkar, “Enhancement of Image Retrieval by using Colour, Texture And Shape Features”, International Conference on Electronic Systems, Signal Processing and Computing Technologies, 2014.
4. Abby A. Goodrum, “Image Information Retrieval: An Overview of Current Research”, Informing Science, Special Issue on Information Science Research, Volume 3, No 2, 2000.
5. Bo Luo, Xiaogang Wang, and Xiaoou Tang, “A World Wide Web Based Image Search Engine Using Text and Image Content Features”, Internet Imaging IV, Simone Santini, RaimondoSchettini, Editors, Proceedings of SPIE-IS&T, Electronic Imaging, SPIE-IS&T,SPIE, Vol. 5018, 2003.
6. Yongzhong Lu, Jingli Zhou, Shengsheng Yu, “A Survey of Face Detection, Extraction And Recognition”, Computing and Informatics, Vol. 22, 2003.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 8, August 2016

7. Ms. Varsha Gupta, Mr. Dipesh Sharma, "A Study of Various Face Detection Methods", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 5, May 2014.
8. Zuomin Luo, Quanfa Zheng, Xinhong Hei, Nasser Giacaman, "Parallel Programming Based on Microsoft.NET TPL", Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013).

BIOGRAPHY

Roma Ravindra Gaydhankar is pursuing the Master in Engineering (M.E.) degree in Computer Science and Engineering Department from Nagesh Karajagi Orchid College of Engineering and Technology, Solapur, Solapur University, Solapur, Maharashtra, India. She received Bachelor of Engineering (B.E.) degree in Computer Science and Engineering in 2014 from Solapur University, Solapur, Maharashtra, India.

Zakir Mujeeb Shaikh received B.E. degree in Computer Science and Engineering in 2000 from Dr. Baba Saheb Ambedkar University, Aurangabad, Maharashtra, India and ME Computer from BVU Pune, Maharashtra, India in 2004. He is presently working as an Assistant Professor in Nagesh Karajagi Orchid College of Engineering and Technology, Solapur, Maharashtra, India. His research area includes Image Processing, Database and software framework