**LA-SS Project Final Report**
**Compression and Head detection**
**15th October 2015**

## Project description

• Given a panaromic image , Compress the image using image compression techniques and Label the heads and count them Using Different Image processing techniques.

### WorkFlow

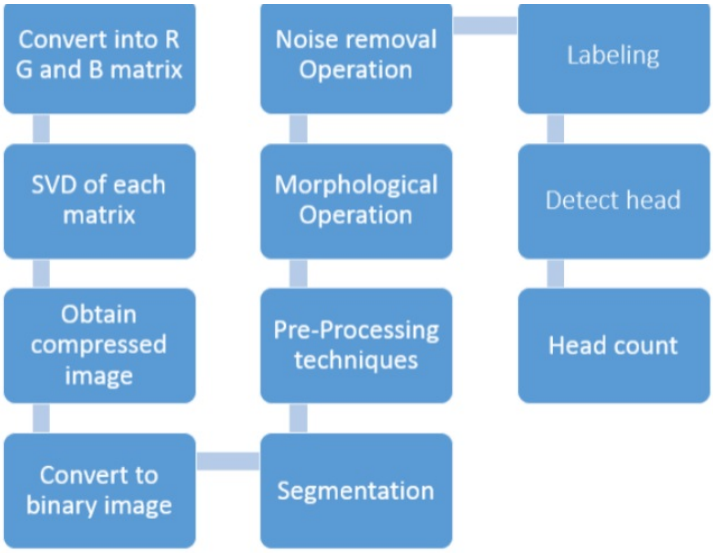



Image is a two dimensional function F(x,y),where x and y are the spatial co-ordinates, and the amplitude of F at any pair of co-ordinates is called the intensity of that level.If x and y and the amplitude values of F are finite and discrete quantities , we call it as a digital image.A digital image is composed of a finite number of elements called pixels,each of which has a particular location and value.

The 2D continuous image is divided into M rows and N columns and the intersection of a row and a column is termed as a Pixel. The value assigned to the integer coordinates $[m, n]$ with $m = 0, 1, 2, , (M - 1)$ and $n = 0, 1, 2, , (N - 1)$ is a[m,n].

$$0 < f(x,y) < \infty$$
$$f(x,y) = i(x,y) * r(x,y)$$
$0 < i(x,y) < \infty$ where, $i(x,y)$ is Illumination
$0 < r(x,y) < 1$ where, $r(x,y)$ is Reflection

$$P = \begin{bmatrix} f(0,0) & \cdots & f(0,N-1) \\ \vdots & \ddots & \vdots \\ f(M-1,0) & \cdots & f(M-1,N-1) \end{bmatrix}$$

This image can be represented as a matrix by placing the RGB values of each pixel in the matrix.
•RGB matrix representation:By the process of quantization we obtain the RGB value of the discrete image.The red, green and blue use 8 bits each, which have integer values from 0 to 255. This makes 256256256 = 16777216 possible colors.
•GrayScale Representation: GrayScale = 0.299R + 0.587G + 0.144B
•Pixel value range lies between 0 to 255

## Various Image Compression Techniques

• Lossy Compression : Lossy compression creates smaller files by discarding some information about the original image.
• Lossless Compression : Lossless compression is a method of data compression algorithms that allows the original data to be perfectly reconstructed from the compressed data.

## Methods of Lossy Compression

-Quantization(Uniform, NonUniform and Vector)
-Discrete Cosine Transform Used in JPEG
-Discrete Fourier transform
-Singular value decomposition

## Methods of Lossless Compression

-Huffman Coding: The data that occur more frequently are assign a smaller bits while the data that occur less frequently are assign the larger bit.
-Run Length Encoding: RLE is a very simple form of data compression in which runs of data that is, sequences in which the same data value occurs in many consecutive data elements are stored as a single data value and count, rather than as the original run.
For lossless compression we used this techniques
• $PDP^{-1}$ : For doing compression through $PDP^{-1}$ we first diagnolized the matrix into $PDP^{-1}$. Here P is made up of Eigen Vectors and D Matrix contains the Eigen Values of the Matrix.
The issue in using $PDP^{-1}$ is that it only works for square matrix. Mostly the image matrix is not square matrix.$P^{-1}$ requires more steps then $V^T$ and hence this method takes more time to implement
• SVD : Hence as $PDP^{-1}$ was not feasible , we use SVD (Singular Value Decompostion) to compress the image.It also takes too much time but as it converts into transpose it will take less time than$PDP^{-1}$.Hence this method can be used.
But finally as SVD was more feasible and took less time to compute we used this method for compressing the image.

$$A = U \sum V^T$$

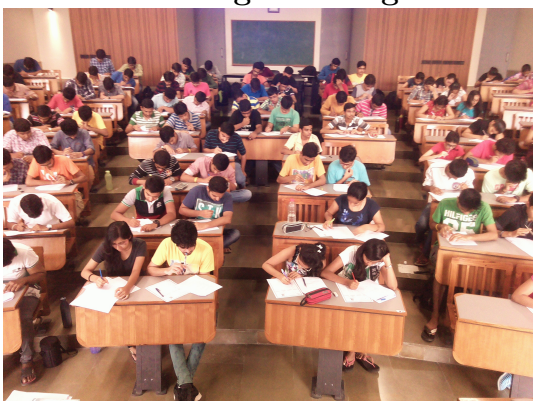
•SVD is based on a theorem from linear algebra which says that a rectangular matrix A can be broken down into the product of three matrices - an orthogonal matrix $U$, a diagonal matrix $S$, and the transpose of an orthogonal matrix $V$ .
The theorem is usually presented something like this: $Amn = UmmSmnV^Tnn$ where $U^TU = I$,$V^TV = I$; the columns of U are orthonormal eigenvectors of $AA^T$, the columns of V are orthonormal eigenvectors of $A^TA$, and S is a diagonal matrix containing the square roots of eigenvalues from U or V in descending order.

Here we take different number of eigen value and eigen vectors

| EigenValue | Compression Ration |
|---|---|
| 100 | 37.36 |
| 50 | 34.00 |
| 150 | 50.05 |

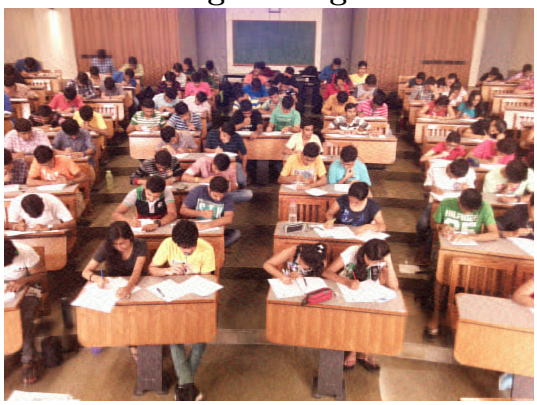
**Compression using different eigen value**

**Original Image**



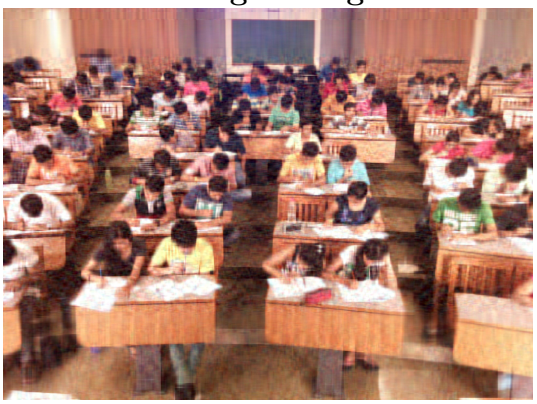
**1944 KB**

**Taking 100 eigen value**



**745KB**

**Taking 50 eigen value**



**661 KB**

**Taking 150 eigen value**

**973KB**

**Thresholding**

•OTSU Algorithm: It stores the intensities of the pixels in an array. The threshold is calculated by using total mean and variance. Based on this threshold value each pixel is set to either 0 or 1. i.e. background or foreground. Steps:
1.Compute histogram and probabilities of each intensity level

2. Set up initial i(0) and i(0)
3.Step through all possible thresholds t = 1... maximum intensity 1.Update iandi 2.Compute 2 b(t)
4.Desired threshold corresponds to the maximum 2 b(t)
5.You can compute two maxima (and two corresponding thresholds). 2 b1(t) is the greater max and 2 b2(t) is the greater or equal maximum
6.Desired threshold = threshold1+threshold2 The purpose of thresholding is to find the pixels which represent an object from some image which is known as foreground and background separation.
• Thresholding is used to segment an image by setting all pixels whose intensity values are above a threshold to a foreground value and all the remaining pixels to a background value.
•Adaptive thresholding: Adaptive thresholding typically takes a grayscale or color image as input and, in the simplest implementation, outputs a binary image representing the segmentation. For each pixel in the image, a threshold has to be calculated. If the pixel value is below the threshold it is set to the background value, otherwise it assumes the foreground value.

There are two main approaches to finding the threshold: (i) the Chow and Kaneko approach and (ii) local thresholding. The assumption behind both methods is that smaller image regions are more likely to have approximately uniform illumination, thus being more suitable for thresholding. Chow and Kaneko divide an image into an array of overlapping subimages and then find the optimum threshold for each subimage by investigating its histogram. The threshold for each single pixel is found by interpolating the results of the subimages.
•Multiple Threshold: In this method we take threshold value based on local property of each pixel.
•Local Thresholding: In this method we take taking threshold value based on pixel location and pixel intensity
•Basic Global Thresholding:In this method we take taking threshold value based on pixel location

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if } f(x,y) \leq T \end{cases}$$

We used Otsu's global thresholding for our image and by that we convert the image to binary image.
Output of thresholding process

**Thresholded image**

## Labeling

•Connected Components Labeling : Connected components labeling scans an image and groups its pixels into components based on pixel connectivity, i.e. all pixels in a connected component share similar pixel intensity values and are in some way connected with each other. Once all groups have been determined, each pixel is labeled with a graylevel or a color according to the component it was assigned to.
Connected component labeling works by scanning an image, pixel-by-pixel (from top to bottom and left to right) in order to identify connected pixel regions, i.e. regions of adjacent pixels which share the same set of intensity values V.

## Morphological Operation

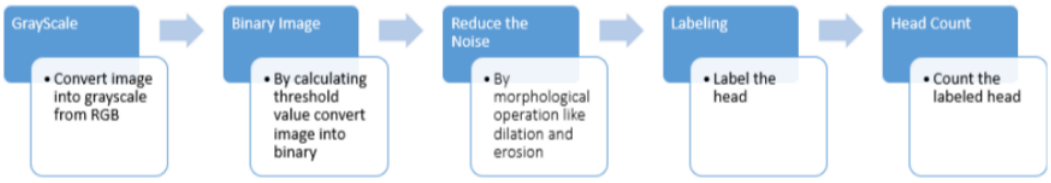
Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image.Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors.
The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes

pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the operation as a dilation or an erosion.

•erosion:Erosion is one of the two basic operators in the area of mathematical morphology, the other being dilation. It is typically applied to binary images, but there are versions that work on grayscale images. The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels. Thus areas of foreground pixels shrink in size, and holes within those areas become larger.

•dilation:The basic effect of the operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels (i.e. white pixels, typically). Thus areas of foreground pixels grow in size while holes within those regions become smaller.

**Face detection algorithm**




In this process we first convert compressed image into gray scale image and then find a threshold value. Acording to that threshold value we convert image into binary image.

For removing the noise of that thresholded image we apply morphological operation like dilation and erosion using disk and finally we get an image with lesser noise.

Then we searched for circle with specific radius range in image and label them and detect the heads.

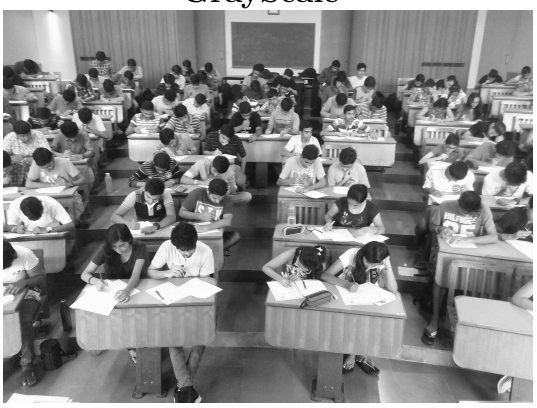
Then finally we count the labeled heads.

But this algorithm also detect some noise and for this reason we are not getting exect count.

We are trying to improve our algorithm to get accurate output.For that we are

trying to apply local thresholding by niblack alorithm.In this algorithm first we will split an image into four parts and we will apply different thresholed value to each subimage and will do all steps again on that new image.

**GrayScale**

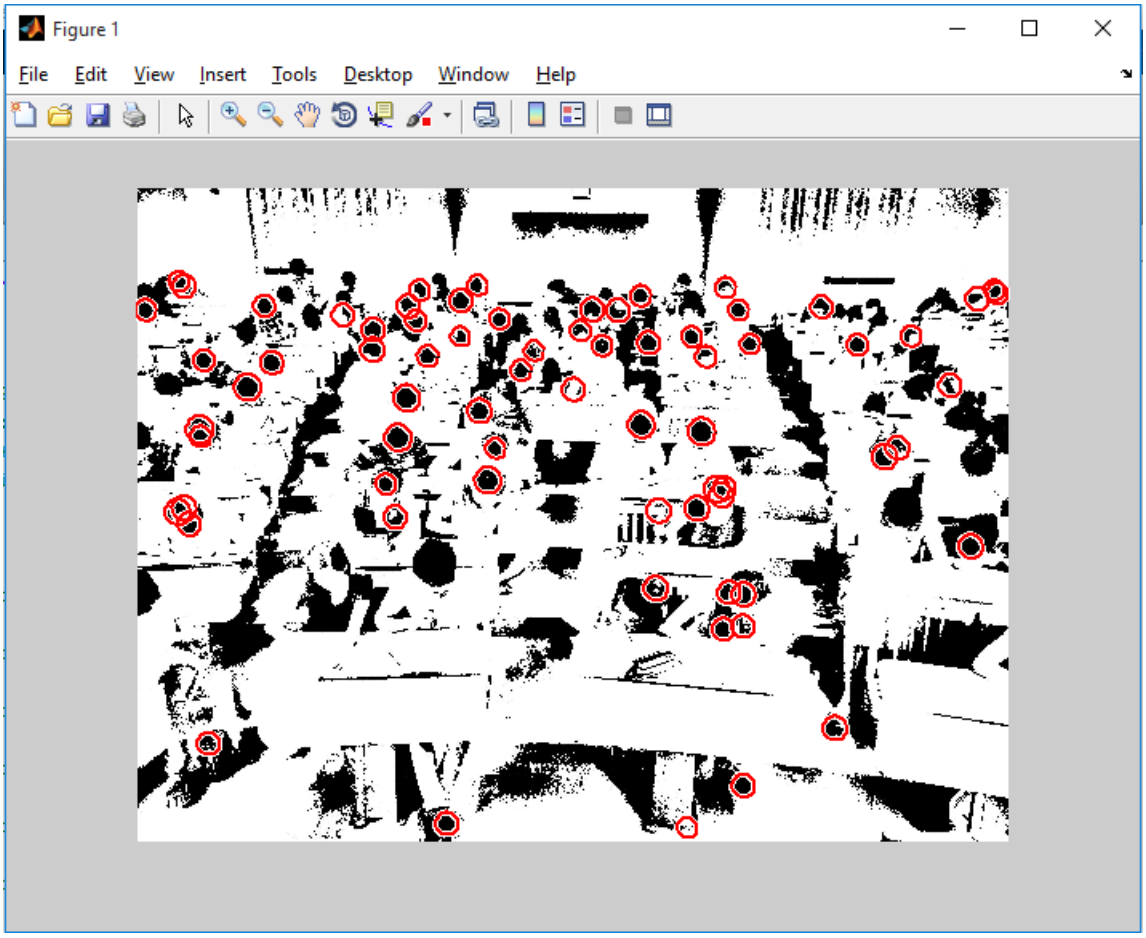


**Bianry Image**

**Erosion**

**Dilation**

**Head detected**




• From applying all the steps mentioned above we get the following output: The total number of object count is 71 but the accurate head count out of that object is 42.