Assignment 4: CS 663, Fundamentals of Digital Image Processing

Mikhil Gupta-203079016
Nitish Gangwar-203050069

October 26, 2021

Answer 1:

As we calculate the SVD of the any matrix $A_{m\times n}$, we need to calculate the $U_{m\times m}$, $S_{m\times n}$, $V_{n\times n}$ matrix corrosponding to A ,
and

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

The right singular vectors of A are the eigenvectors of A'*A, and the left singular vectors of A are the eigenvectors of A*A'.

Similarly the singular values of A are the square root of the eigenvalues of A*A' (or A'*A, the eigenvalues of those are just the same).

since we define SVD for the decomposition for any arbitrary-size matrices (rectangular or square), while EVD ,eigenvalue decomposition ,i.e. **eig** routine applies only to square matrices., whereas in this case as we are directly apply the svd to. This given method used by the student may work in case of the square matrix only if he aligns the eigenvalues and the corresponding eigenvectors of the right and the left singular in decreasing order. , in order to generalise the process

To correct his mistake he needs to do the following , which will generalise the process for calculations of the right and left singular decomposition's . In order to calculate the singular value decomposition (SVD) components $\left(U, \Sigma, V^T\right)$, for a given matrix A,
we need to first calculate either of AA^T or A^TA, calculate their eigen vectors and eigen values.
Here, we calculate first the eigen vectors (V) and eigen values (Λ) of A^TA, sort in descending order and calculate the other unitary vector matrix U from the relation $AV = \Sigma U$.

The MATLAB code implementation of the same is give here in the file **Q1HW4.m**

Answer 2:

We have set of N vectors $\mathcal{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, ..., \boldsymbol{x_N}\}$ each in $\mathbb{R}^d$, with average vector $\bar{\boldsymbol{x}}$, We need to prove that , $\boldsymbol{f}^t\boldsymbol{C}\boldsymbol{f}$ will be maximise if $f \perp e$ and corresponds to eigen vector of second highest value after e ,
i.e. Given constrains are,

- $\boldsymbol{f}$ is perpendicular to $\boldsymbol{e}$.
- $\|\boldsymbol{f}\| = 1$
- $\boldsymbol{e^t}\boldsymbol{C}\boldsymbol{e} = \lambda_1$
- $\boldsymbol{e^t}\boldsymbol{f} = 0$

Our Cost function lagrangian is given as for the above condition , as

$$J(\boldsymbol{f}) = \boldsymbol{f^t}\boldsymbol{C}\boldsymbol{f}$$

we have given the constrains as $\boldsymbol{f^t}\boldsymbol{f} = 1$,and $\boldsymbol{f} \perp \boldsymbol{e}$ i.e. $\boldsymbol{f^t}\boldsymbol{e} = 0$, Therefore our modified cost function will be,

$$J(\boldsymbol{f}) = \boldsymbol{f^t}\boldsymbol{C}\boldsymbol{f} + \lambda(\boldsymbol{f^t}\boldsymbol{f} - 1) + \mu(\boldsymbol{f^t}\boldsymbol{e})$$

Differentiating the $J(\boldsymbol{f})$ with respect to $\boldsymbol{f^t}$, we get ,

$$J'(\boldsymbol{f}) = 2\boldsymbol{C}\boldsymbol{f} - 2\lambda\boldsymbol{f} - \mu\boldsymbol{e}$$

Solving for constrain in μ , since wkt, $\boldsymbol{C}\boldsymbol{e} = \lambda_1\boldsymbol{e}$,
which can be written as $\boldsymbol{e^t}\boldsymbol{C^t} = \boldsymbol{e^t}\boldsymbol{C} = \lambda_1 e^t$, therefore, $J'(\boldsymbol{f})$ expression will be ,

$$\boldsymbol{e^t}J'(\boldsymbol{f}) = 2\boldsymbol{e^t}\boldsymbol{C}\boldsymbol{f} - 2\lambda\boldsymbol{e^t}\boldsymbol{f} - \mu\boldsymbol{e^t}\boldsymbol{e} \qquad = 2\lambda_1\boldsymbol{e^t}\boldsymbol{f} - 2\lambda\boldsymbol{e^t}\boldsymbol{f} - \mu\boldsymbol{e^t}\boldsymbol{e}$$

for maximizing making $J'(\boldsymbol{f}) = 0$, therefore , we get $\mu = 0$,

Using this value in derivative of lagrange's multiplier i.e. $J'(\boldsymbol{f})$,we get ,

$$J'(\boldsymbol{f}) = 2\boldsymbol{C}\boldsymbol{f} - 2 \tag{1}$$

maximising and equating $J'(\boldsymbol{f}) = 0$,we get

$$\boldsymbol{C}\boldsymbol{f} = \lambda\boldsymbol{f}$$

multiply both side by $\boldsymbol{f^t}$, we get ,

$$\boldsymbol{f^t C f} = \lambda$$

Now , we got in above equation as the given relation , therefore to maximize $\boldsymbol{f^t C f}$, we will need maximize λ , but since we know that the maximum $\lambda = \lambda_1$, which is corresponding to eigenvector $\boldsymbol{e}$,
and as given as $rank(\boldsymbol{C}) > 2$, therefore the value corresponding to maximisation of the $\boldsymbol{f^C f}$ will be the second highest eigenvalue λ_2 , which is corresponding to the eigenvector $\boldsymbol{f}$,

Answer 3:

Here we are given with the two images ,

g_1 when with the focus f_1 , with outside scene focus ,

$$g_1 = f_1 + h_2 * f_2 \tag{2}$$

and the g_2 as the focus f_2 ,given as ,

$$g_2 = h_1 * f_1 + f_2 \tag{3}$$

Where , h_1 ,h_2 are the blurr kernel acting on the original image f_1 and f_2 respectively .,

Taking the fourier transform of g1 and g2 on both side , we get ,

$$G_1 = F_1 + H_2 F_2 \tag{4}$$

and

$$G_2 = H_1 F_1 + F_2 \tag{5}$$

As we have known the blurr kernels h_1 and h_2 , therefore the H_1 and H_2 can be known to us ,
and the final image captured will be available to us i.e. G_1 and G_2,
Solving the above linear equations for F_1 and F_2 , we get ,

$$F_1 = \frac{G_2 H_2 - G_1}{H_1 H_2 - 1} \tag{6}$$

and ,

$$F_2 = \frac{G_1 H_1 - G2}{H_1 H_2 - 1} \tag{7}$$

Therefore takin the inverse fourier transform we will get ,

$$f_1 = \mathcal{F}^{-1}(F_1)$$

$$f_2 = \mathcal{F}^{-1}(F_2)$$

As we can see the denominator of the F_1 and F_2 , we have $(H_1 H_2 - 1)$, i.e. this term will tend to 0 if the term $H_1 H_2$ will be 1 , and the resulting value of the value of the F_1 or F_2 will $\longrightarrow \infty$, this things makes the restoration of the two original images makes this formula less useful.
As it is given that the h_1 ,h_2 are blur kernels , i.e. the LPF therefore it can be considered as the corresponding estimates of the H_1 , H_2 will be very large value even for the small amount of the low frequency noise ,so it is possibility that the noise value in the resultant image may be greater than the original image value .

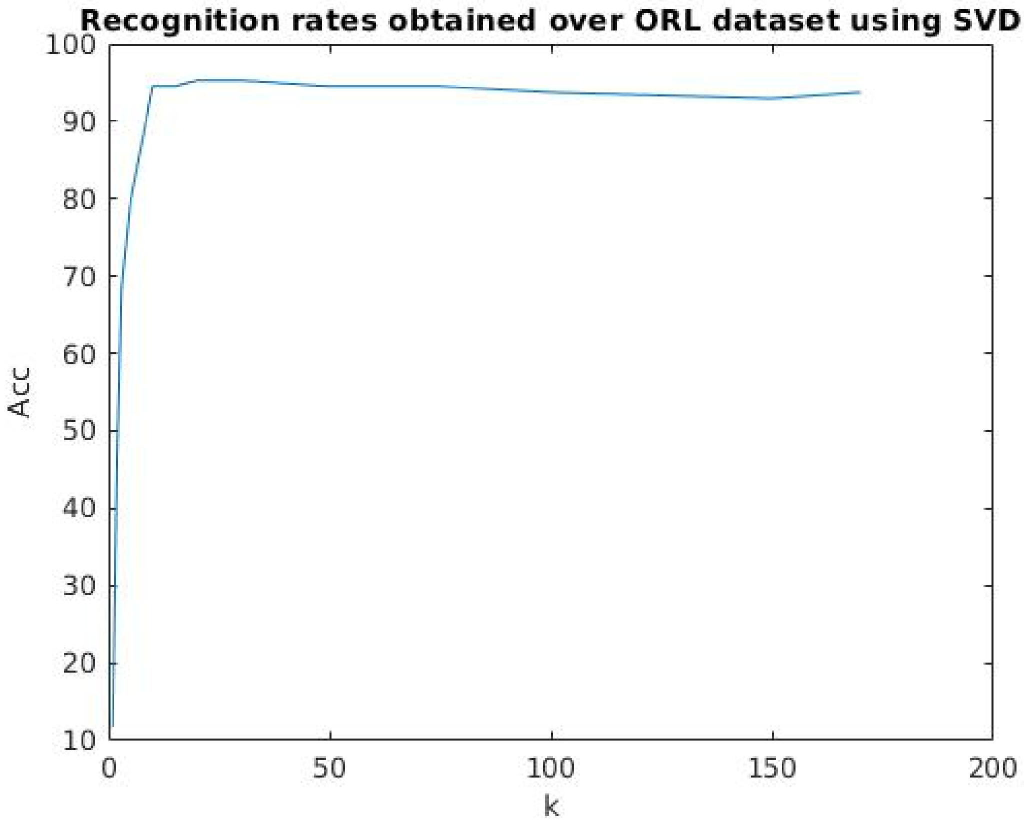


Figure 1: Figure represents the recognition rates obtained at different values of k={1, 2, 3, 5, 10, 15, 20, 30, 50, 75, 100, 150, 170} for 128 test images using **SVD**.Code for this plot lies inside q4.m file.

Answer 4:

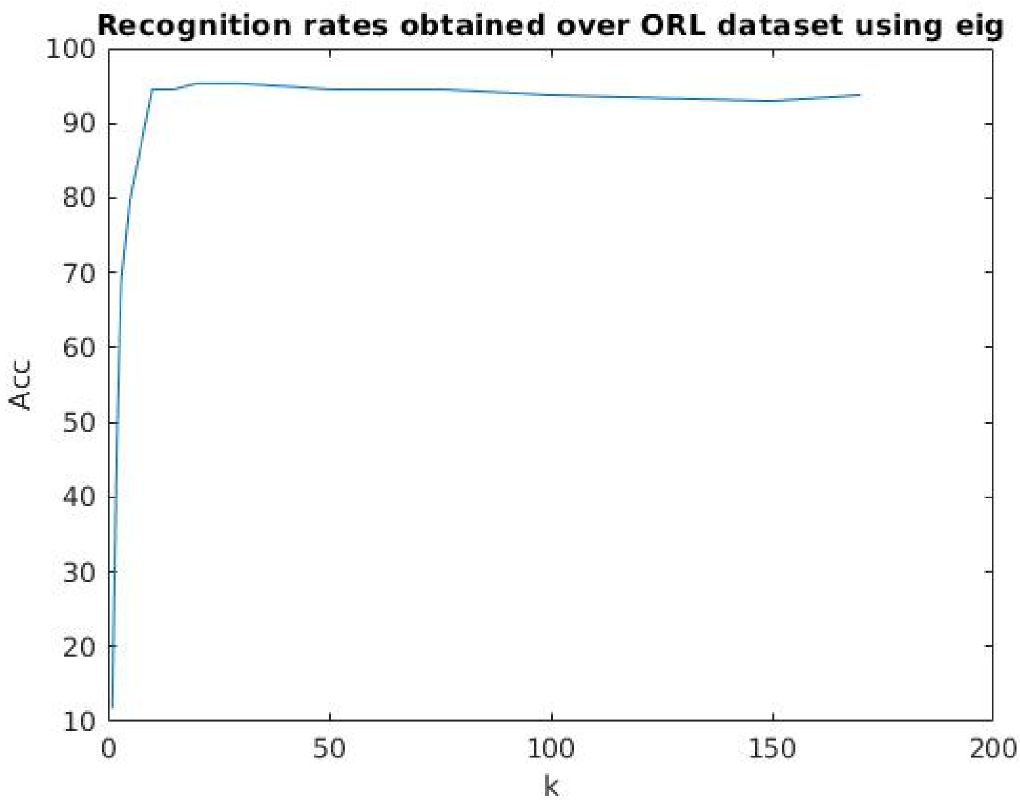
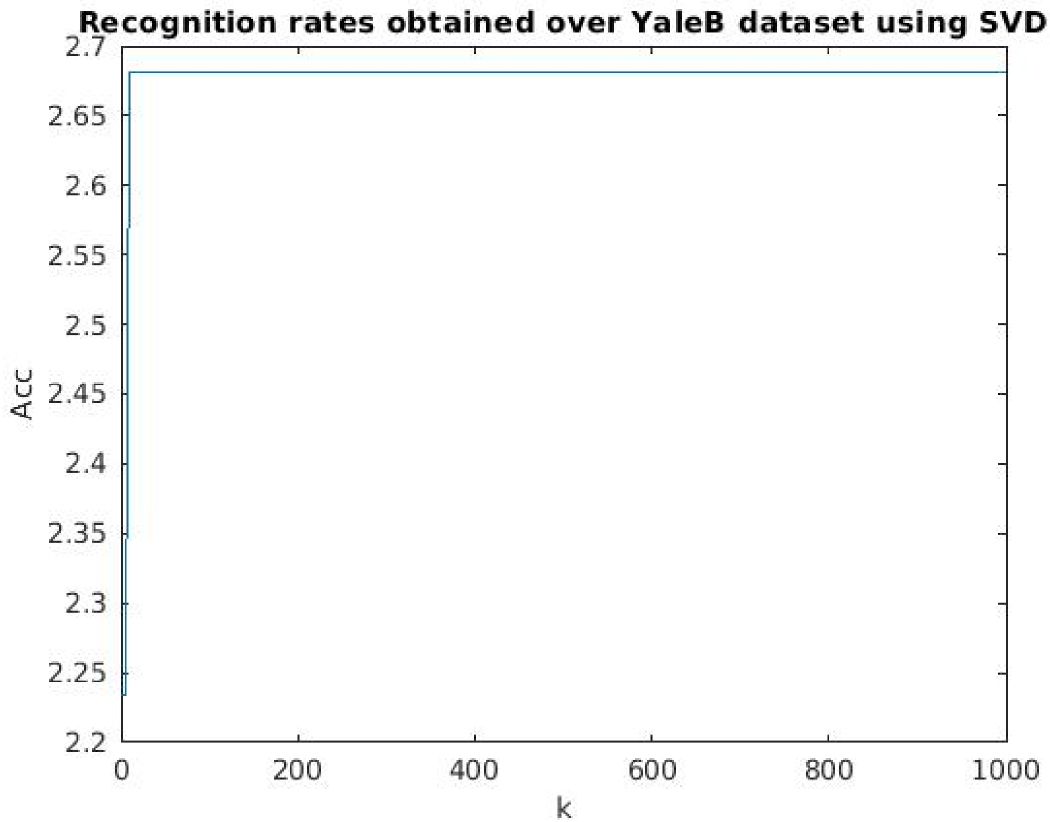
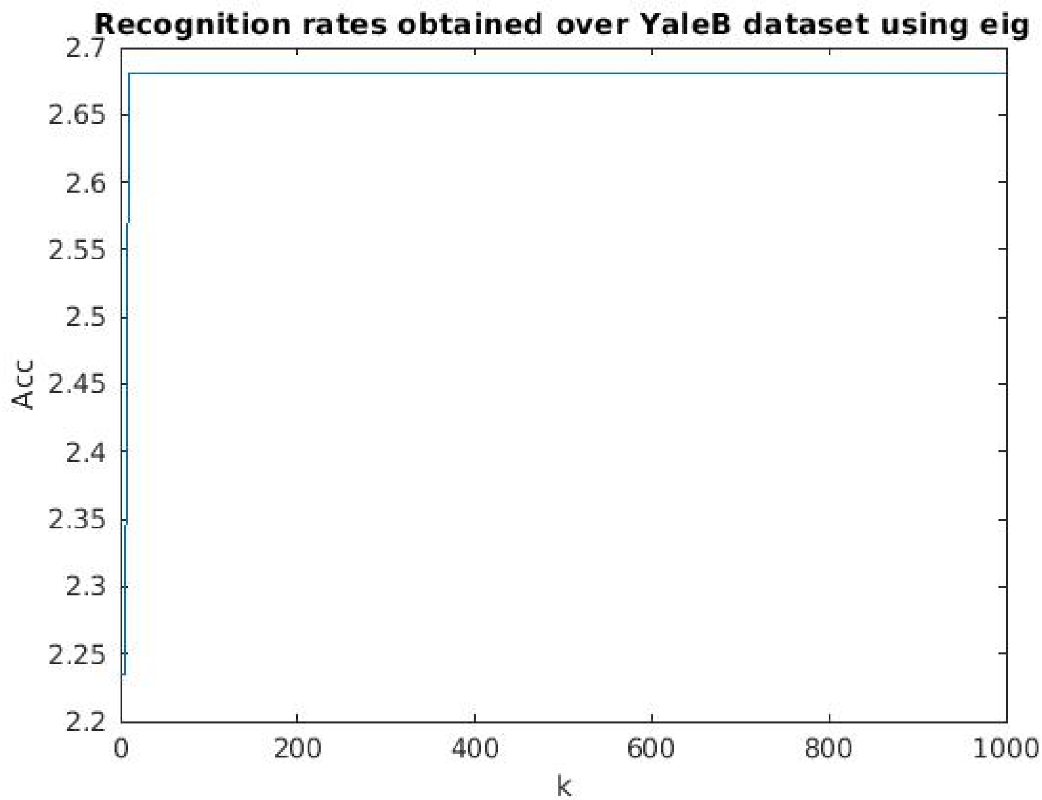
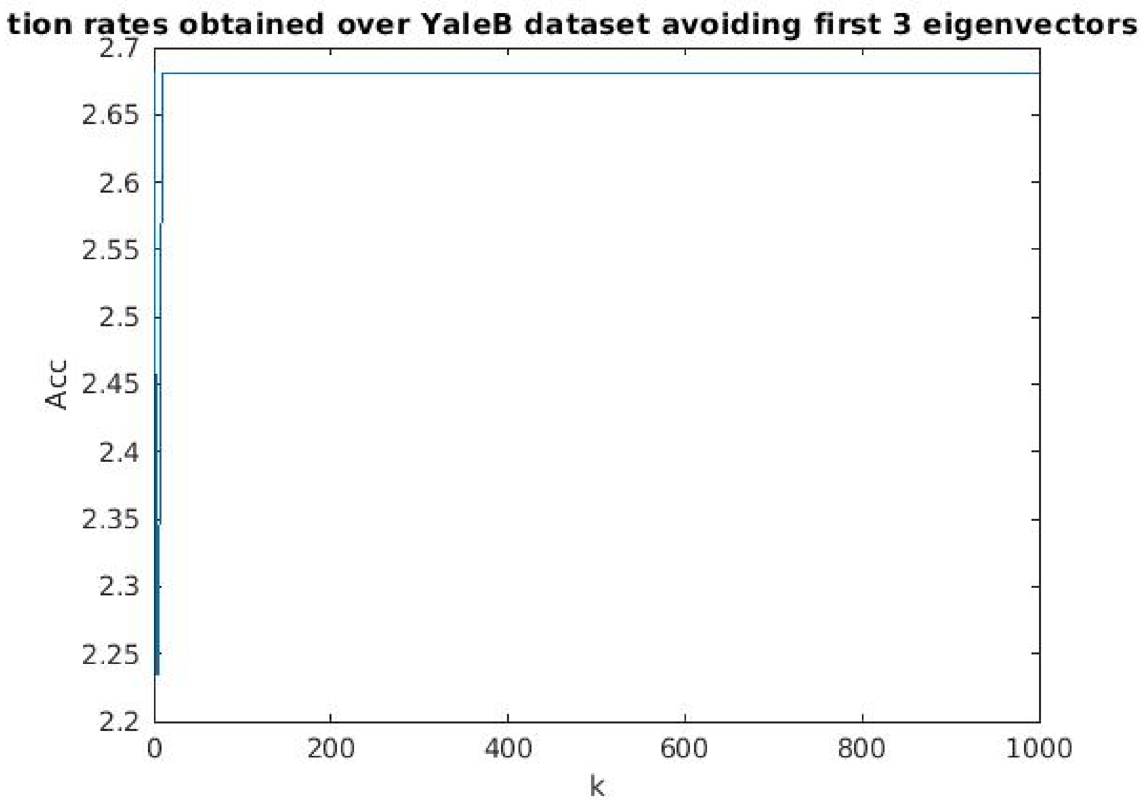


Figure 2: Figure represents the recognition rates obtained at different values of k={1, 2, 3, 5, 10, 15, 20, 30, 50, 75, 100, 150, 170} for 128 test images using **eig**.Code for this plot lies inside q4.m file.



Recognition rates obtained over YaleB dataset using SVD
Acc
2.7
2.65
2.6
2.55
2.5
2.45
2.4
2.35
2.3
2.25
2.2
0
200
400
600
800
1000
k


Recognition rates obtained over YaleB dataset using eig
2.7
2.65
2.6
2.55
2.5
2.45
2.4
2.35
2.3
2.25
2.2
Acc
0
200
400
600
800
1000
k


tion rates obtained over YaleB dataset avoiding first 3 eigenvectors
Acc
2.7
2.65
2.6
2.55
2.5
2.45
2.4
2.35
2.3
2.25
2.2
0
200
400
600
800
1000
k

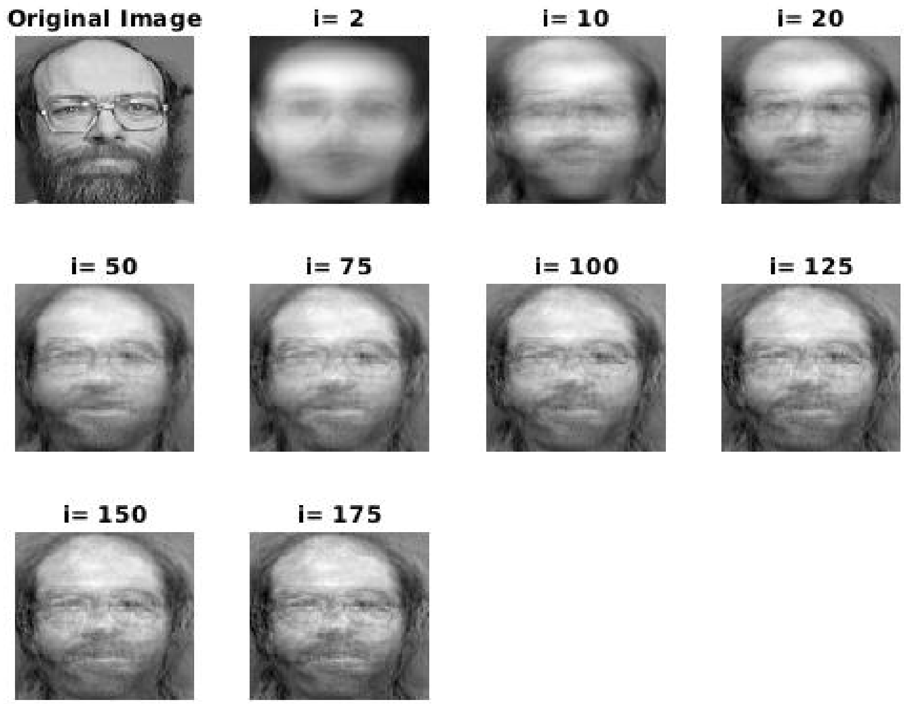


Figure 3: Figure represents the reconstruction done for an test image (ORL/s37/1.pgm) for the different values of k={2, 10, 20, 50, 75, 100, 125, 150, 175}

.

Answer 5:

Figure 3 and 4 represents the test image reconstruction and first 25 eigen faces.

Figure 4: Figure represents the first 25 eigenvectors corresponding to the decreasing eigenvalues

.

Answer 6:

We have already trained our model over the images present inside folders from s1 to s32 from ORL dataset. Now for testing the model over the images placed inside folders from s33 to s40. We will not be able to find least square difference as we do not have the eigen coefficients for these images. We thought of some threshold based method like setting some threshold from training set and finding out the false negatives first for the images which were treated as test images i.e. from s1 to s32 the last 4 images. we set the threshold by looking out the output and then same threshold was used to find out the false positive if the error computed using least square methods in eigen coefficients is less than the threshold then that image is contributing to the false positive result and this is how we are analysing the result for test images.
Final output obtained is:
false negative=26
false positive=10
where the optimal value of k used is 75.