Q6)

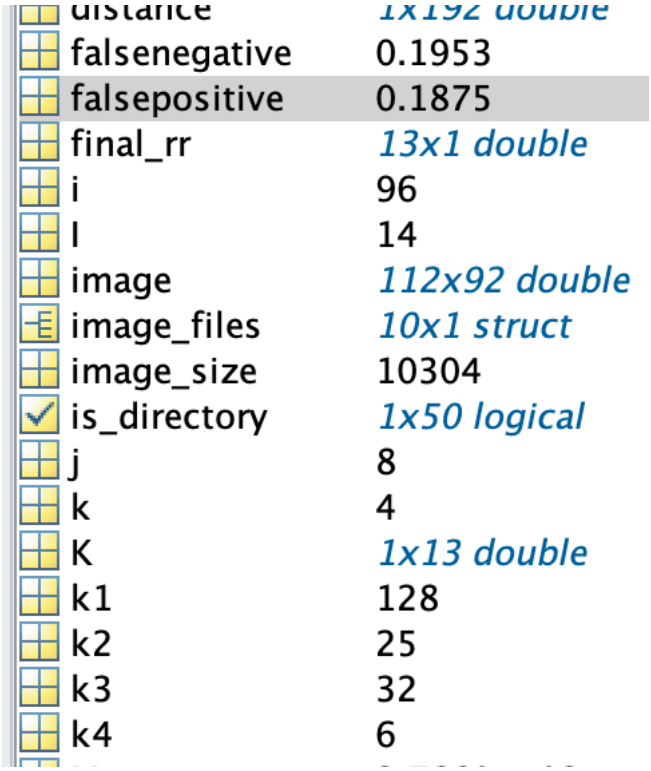
What we did in this question was, that we first found our svd from our training data set.

Then we used those parameters on our test model, in order to get the minimum errors of our test dataset.

Then we used those minimum values from our test model, in order to calculate a value of threshold, below which our model would classify an image as recognised, and above which our model would classify an image as not recognisable.

The metric we used was-
threshold= Mean error of our dataset + k*(max error in our test dataset- mean error of our dataset).

(Here we varied the values of k, such that we get the best values of false positive and false negative).



We then found the errors on our test dataset. Whenever the minimum error for an image was greater than the threshold, we classified it as unrecognisable. But as its part of our test dataset, we know that the image is recognisable. Thus we called it a false negative.

We then found the errors on our unseen images. Whenever the error for an image turned out lesser than the threshold, we classify them as recognisable. But since this is the unseen dataset, we know that these images are unclassifiable. Hence we term all these instances as false positives.

We varied k manually. As k increased, we saw that the amount of false positives increased and false negatives decreased. And vice-versa. As our k varied, we saw that either one of falsepositive or falsenegative became fairly large. For our value of k as 0.55, we see that we get very good values for both falsepositive and falsenegative.

Thus for k=0.55, we saw that 25 (k2) out of the 128 (k1) images were classified as unrecognised, hence false negatives. Thus our net rate of false negatives, I.e. falsenegative came out to be 0.1953.

We also see for the above k, that 6 (k4) out of the 32 (k3) unseen images were classified as recognisable, hence false positives. Thus rate of false positives was 0.1875.

(Rate of false positives= falsepositive= number of false positives/ total instances tested).
(Rate of false negatives= falsenegative= number of false negatives/ total instances tested).

Note: whatever I have referred to as unseen images means images of people that aren't in the dataset.

Code-

```
% INITIALIZATION

image_size = 112*92;
train_image_array = zeros(image_size,32,6);
test_image_array = zeros(image_size,32,4);
not_in_gallery_image_array = zeros(image_size,8,4);
K = [1 2 3 5 10 15 20 30 50 75 100 150 170];
final_rr = zeros([length(K),1]);

%FILE READING
directory_name = dir('ORL/');
is_directory = [directory_name.isdir]  & ~strcmp({directory_name.name},'.') &
~strcmp({directory_name.name},'..');
person_directory = directory_name(is_directory);

for i= 1:length(person_directory)
    image_files = dir(['ORL/' person_directory(i).name '/*.pgm']);
    for j=1:length(image_files)
        image = double(imread(['ORL/' person_directory(i).name '/' image_files(j).name]));
        if (i > 32) && (j<=4)
            not_in_gallery_image_array(:,i-32,j) = image(:);
        elseif (i <= 32) && (j>6)
            test_image_array(:,i,j-6) = image(:);
        elseif (i <= 32) && (j<=6)
            train_image_array(:,i,j) = image(:);
        end
    end
end

train_image_array = reshape(train_image_array,image_size,[]);
test_image_array = reshape(test_image_array,image_size,[]);
not_in_gallery_image_array = reshape(not_in_gallery_image_array,image_size,[]);
%imshow(reshape(train_image_array(:,1),112,92),[])

%MEAN
mean_images = mean(train_image_array(:,:),'all');
train_image_array= train_image_array-mean_images;

%SVD
[U,S,V] = svd(train_image_array);




%TESTING
thres= zeros(32*3);

U_k = U(:,1:50);
rr =0;
projected_train_image_array = transpose(U_k)*train_image_array;
for j= 1:32
   for k = 1:4
       test_image = test_image_array(:,32*(k-1)+j)-mean_images;
       projected_test_vector = transpose(U_k)*test_image(:);
       distance = vecnorm(projected_train_image_array-projected_test_vector,2,1);
       [M,I] = min(distance);
```

```
      if mod(l-1,32)+1 == j
        thres((j-1)*3 + k)= M;
      end
   end
end
```

```
num=0;
val=0;
max=0;
for i=1:length(thres)
    if thres(i)>0
        num=num+1;
        val=val+thres(i);
        if thres(i)>max
            max= thres(i);
        end
    end
end

meann= val*1.0/num;
threshold= meann + 0.55*(max- meann);
```

```
k1=0;
k2=0;
projected_train_image_array = transpose(U_k)*train_image_array;
for j= 1:32
    for k=1:4
      test_image = test_image_array(:,32*(k-1)+ j)-mean_images;
      projected_test_vector = transpose(U_k)*test_image(:);
      distance = vecnorm(projected_train_image_array-projected_test_vector,2,1);
      [M,l] = min(distance);
      k1= k1+1;
      if M>threshold
          k2= k2+1;
      end
    end
end

falsenegative= k2*1.0/k1;
```

```
k3=0;
k4=0;
projected_train_image_array = transpose(U_k)*train_image_array;

for j= 1:8
   for k = 1:4
      test_image = not_in_gallery_image_array(:,8*(k-1)+j)-mean_images;
      projected_test_vector = transpose(U_k)*test_image(:);
      distance = vecnorm(projected_train_image_array-projected_test_vector,2,1);
      [M,I] = min(distance);
      k3= k3+1;
      if M<threshold
         k4= k4+1;
      end
   end
end

falsepositive= k4*1.0/k3;
```