Cara Evangeline Chandra Mohan
190539421

MACHINE LEARNING FOR VISUAL DATA ANALYSIS LAB – 4

1. Calculate the Euclidean matrix between the dictionary elements and the descriptors extracted in an image sequence.

```
function D = compute_dist(X,B)
nframe=size(X,1);
nbase=size(B,1);
XX = sum(X.*X, 2);
BB = sum(B.*B, 2);
D  = repmat(XX, 1, nbase)-2*X*B'+repmat(BB', nframe, 1);
D = D';
```

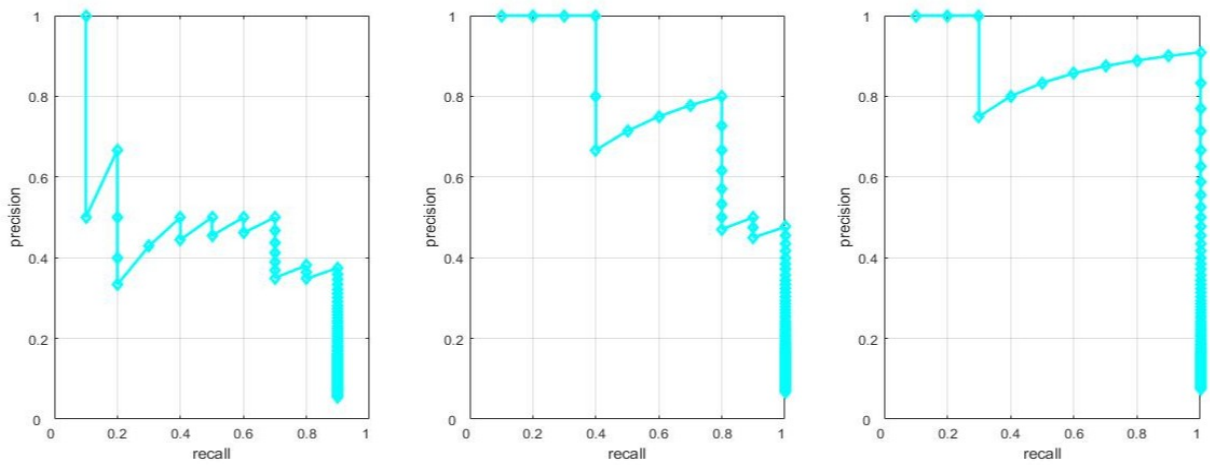
The above compute_dist() function calculates the square of euclidean distance between the codeword and the features.

2.Write a function that implements the voting scheme for the following properties: a) the spatial centre of the action at the current frame, b) the start and the end of action and c) the width and the height of the bounding box of the action in the current frame.

```
index = 1;
for i = 1:size(patches,2)
   for j = 1:size(flag_mat,1)
      match = sum(flag_mat(:,i));
      if flag_mat(j,i) == 1
         z = struct_cb.offset(j).tot_cnt;
         for k = 1:z
            hough_array(1,index) = position(1,i) - (struct_cb.offset(j).spa_offset(1,k) * spa_scale(i));
            hough_array(2,index) = position(2,i) - (struct_cb.offset(j).spa_offset(2,k) * spa_scale(i));
            hough_array(3,index) = frame_num(i) - (tem_scale(i) * struct_cb.offset(j).st_end_offset(1,k)) ;
            hough_array(4,index) = frame_num(i) - (tem_scale(i) * struct_cb.offset(j).st_end_offset(2,k));
            hough_array(5,index) = (1/z)*(1/match);
            hough_array(6,index) = struct_cb.offset(j).hei_wid_bb(1,k) * spa_scale(i) ;
            hough_array(7,index) = struct_cb.offset(j).hei_wid_bb(2,k) * spa_scale(i);
            index = index +1;
         end
      end
   end
end
```

hough_array is the voting map for the test samples. It is calculated based on the trained DataStructureVotemap and test samples struct_feat. The loop runs for all Spatio Temporal Interest points of the test data and compares it against every codeword of every class and hence we obtain hough_array. From the hough_array we obtain TP_FP_mat which helps in finding out the precision and recall values for all 3 classes and the graph is as shown below.

Cara Evangeline Chandra Mohan
190539421



3.Assign each sequence to a class according to which hypothesis received the higher number of votes (hint: use the values of the matrix TP_FP_mat). Report the misclassification error, or build the confusion matrix.

```
function mis_class_error = confusion()
load('struct_TP_FP');
incorrect = 0;
for i =1:3
   for j = 1:10
      flag =0;
      a = (struct_TP_FP.class(i).seq(j).array(1,1)~=1);  %check whether there is a overlap with GT[ie=1]
      b = (struct_TP_FP.class(i).seq(j).array(3,1)~=i);   %check whether it belongs to its respective
class i
      if (a)  %if it doesn't overlap with GT, go across the array to find whether there is some other
column which overlaps
         for k = 2:size(struct_TP_FP.class(i).seq(j).array,2) %Run the iteration across the array
            c = (struct_TP_FP.class(i).seq(j).array(1,k)==1);
            d = (struct_TP_FP.class(i).seq(j).array(3,k)); %Get the class value
            if (c && d ~= i) %if there is a overlap and if the predicted class is different than actualthen
ignore
               flag = 1;
               break;
            end
            if (c && d == i) %if there is a overlap and if the predicted class is same as actual class then
reduce the incorrect values by 1which is given below
               flag = 2;
               break;
            end
         end
      end
      if flag == 2
         incorrect = incorrect - 1;
      end
      if( a || b) %if there is no overlap or if the predicted doesn't match the actual class then it's
```

Cara Evangeline Chandra Mohan
190539421

```
incorrect classification
          incorrect = incorrect + 1;
       end

   end
end
%%------------------Confusion matrix------------%%
confusion = zeros(3,3);
for i = 1:3  %For all classes
   for j = 1:10 %for all test samples belonging to a class
      for k = 1:size(struct_TP_FP.class(i).seq(j).array,2)
         a = (struct_TP_FP.class(i).seq(j).array(1,k)); %Overlap of ground truth(0/1)
         b = (struct_TP_FP.class(i).seq(j).array(3,k)); %Predicted label
         if( a == 1) %if there is a overlap of ground truth, then the heighest weighed class is the
predicted class
            confusion(i,b) = confusion(i,b) + 1;
            break;
         end
      end
   end
end
save('confusion','confusion');
mis_class_error = incorrect/30;
fprintf('Misclassification rate  = %f\n', mis_class_error);
```

Output:

Misclassification rate = 0.16667

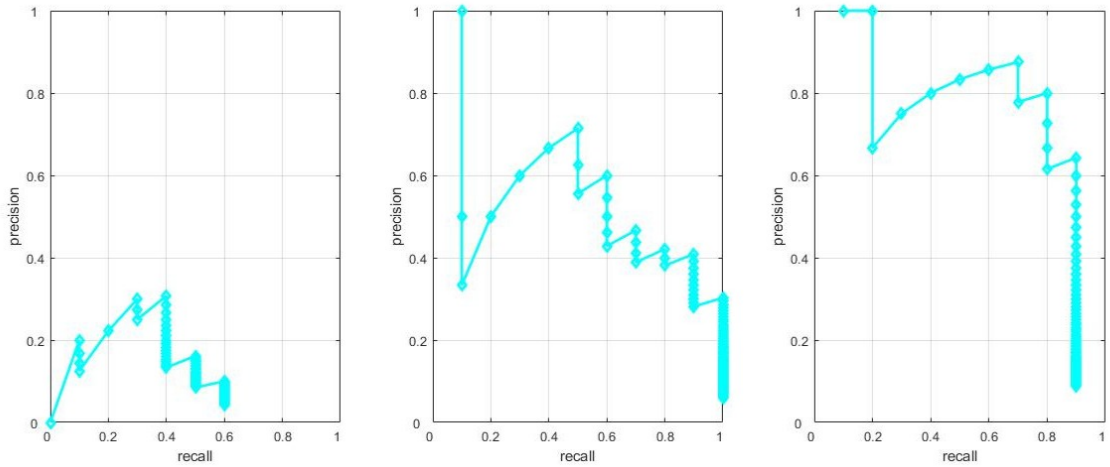
	Predicted Class 1	Class 2	Class 3
Actual Class 1	*5*	*1*	*4*
Class 2	*0*	*10*	*0*
Class 3	*0*	*0*	*10*

From Confusion Matrix : Misclassification rate = (1+4)/30 = 0.16667

4.Perform the localisation experiment using a very small dictionary and report the precision – recall curves. Hint: Cluster the descriptors into a small number of clusters and Explain the drop in the performance.

When I consider a dictionary size of 20 for all 3 classes I get the below graph :

Cara Evangeline Chandra Mohan
190539421



And,
Misclassification rate = 0.26667

There is drastic increase in the error rate and hence sufficient dictionary size should be given to reduce the error rate.