School of Electronic Engineering and Computer Science
ECS797 Machine Learning for Visual Data Analysis
Lab 4: Part-based Action localisation

Sandeep Kaur
190562692

1. Localising actions in image sequences

1. The function ism_test_voting calculates the voting maps for a single image sequence. Calculate the Euclidean matrix between the dictionary elements and the descriptors extracted in an image sequence.

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. The details are given in houghvoting.m

Ans: We can implement following MATLAB code as houghvoting.m

Write your code here to compute the matrix hough_array

```
cob_sz = size(flag_mat, 1);
[th_x, th_y, th_s, th_e, v, b1, b2] = deal([]);
for i=1:cob_sz
    is_act_cw = sum(flag_mat(i,:));
    if is_act_cw ~= 0
        act_cw_s = position(:, flag_mat(i,:));
        act_cw_t = frame_num(:, flag_mat(i,:));
        edge_num = struct_cb.offset(i).tot_cnt;
        vs = [];
        for l=1:edge_num
            th_x = [th_x act_cw_s(1,:) - spa_scale(1, flag_mat(i,:))  *
struct_cb.offset(i).spa_offset(1, l)];
            th_y = [th_y act_cw_s(2,:) - spa_scale(1, flag_mat(i,:))  *
struct_cb.offset(i).spa_offset(2, l)];
            th_s = [th_s act_cw_t - tem_scale(1, flag_mat(i,:))  *
struct_cb.offset(i).st_end_offset(1, l)];
            th_e = [th_e act_cw_t - tem_scale(1, flag_mat(i,:))  *
struct_cb.offset(i).st_end_offset(2, l)];
            vs = [vs repmat(1/(edge_num), is_act_cw, 1)'];
            b1 = [b1 struct_cb.offset(i).hei_wid_bb(1, l) * spa_scale(1,
flag_mat(i,:)) ];
            b2 = [b2 struct_cb.offset(i).hei_wid_bb(2, l) * spa_scale(1,
flag_mat(i,:)) ];

        end
        v = [v vs];
    end
end
```

```
hough_array = [th_x; th_y; th_s; th_e; v; b1; b2];
```

2. Evaluation

1. Using the provided code in ism_test_voting.m and recall_prec_curve.m, plot the Recall precision curves for each class.
2. 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.

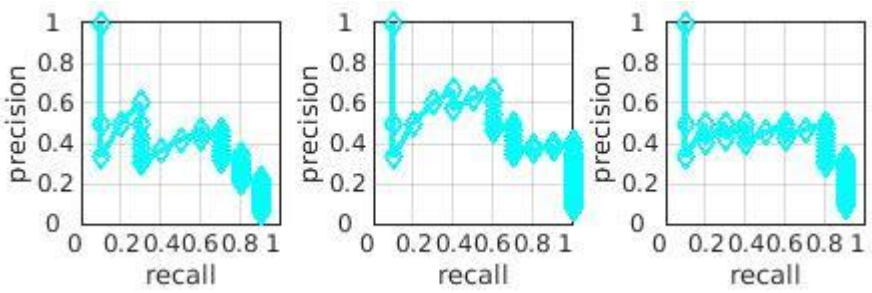
Ans: MATLAB code

```
function recall_prec_curve()
load('struct_TP_FP');
for i = 1:3
cl_sz = size(struct_TP_FP.class(i).seq,2);
arr_tp_fp = [];
for j = 1:cl_sz
temp_ind_pos = ((struct_TP_FP.class(i).seq(j).array(1,:)==1) &
(struct_TP_FP.class(i).seq(j).array(3,:)==i));
        cumsum_temp_ind_pos = cumsum(temp_ind_pos);
        temp_array = 1:length(cumsum_temp_ind_pos);
        temp_ind = temp_array(cumsum_temp_ind_pos==1);
        if(~isempty(temp_ind))
            arr_tp_fp = [arr_tp_fp
[struct_TP_FP.class(i).seq(j).array(2,temp_ind(1));1;0]];
        end
        temp_ind_neg = (struct_TP_FP.class(i).seq(j).array(3,:)~=i) +
((struct_TP_FP.class(i).seq(j).array(1,:)==0) &
(struct_TP_FP.class(i).seq(j).array(3,:)==i));
        log_temp_ind_neg = temp_ind_neg>0;
        if(~isempty(sum(log_temp_ind_neg)))
            arr_tp_fp = [arr_tp_fp
[struct_TP_FP.class(i).seq(j).array(2,log_temp_ind_neg);zeros(1,sum(log_tem
p_ind_neg));ones(1,sum(log_temp_ind_neg))]];
        end
    end
    confidence = arr_tp_fp(1,:);
    tp = arr_tp_fp(2,:);
    fp = arr_tp_fp(3,:);
    [~,si]=sort(-confidence);
    tp = tp(si);
    fp = fp(si);
    fp=cumsum(fp);
    tp=cumsum(tp);
    rec=tp/cl_sz;
    prec=tp./(fp+tp);
    TP_FP_mat.array(i).rec = rec;
    TP_FP_mat.array(i).prec = prec;
end
for i = 1:3
    subplot(1,3,i);
    plot(TP_FP_mat.array(i).rec,TP_FP_mat.array(i).prec,'-
c','LineWidth',2,'Marker','diamond');
    axis([0 1 0 1]);
```

```
    set(gca,'YTick',[0:0.2:1]);
    set(gca,'YTickLabel',char('0','0.2','0.4','0.6','0.8','1'));
    set(gca,'XTick',[0:0.2:1]);
    set(gca,'XTickLabel',char('0','0.2','0.4','0.6','0.8','1'));
    grid on;
    xlabel 'recall'
    ylabel 'precision'
end
f2 = figure(1);
set(f2,'Position',[10 300 1900 530]);
```

For Codebook size [500,500,560], we obtained following misclassification error and recall precision plot:

Missclassification rate = 0.066667

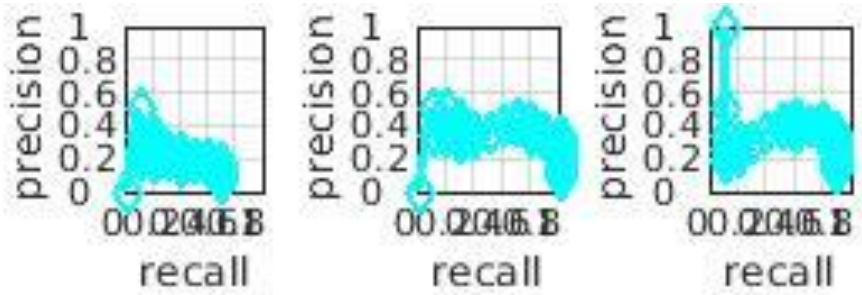


3. Dictionary size

1. 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 (e.g. 20).
2. Explain the drop in the performance.

For Codebook size [20,20,20], we obtained following misclassification error and recall precision plot:

Ans: Missclassification rate = 0.133333



As we can observe there is a drop in performance that is due to the underfitting and decreased number of clusters in k means.

Thus, we observe this drop in performance .