**function [AUC,AUCnor] = cal\_AUC(det\_map,GT,mode\_tau,mode\_eq)**

% det\_map; the detection result N\*k, N is the number of pixels in the detection result,

% k is the number of detection results with different detector

% GT: the ground truth vector N\*1

% mode\_tau: if mode\_tau=1,the tau is sample value; else the tau is uniform step 0.01

% mode\_eq: if mode\_eq=1. the equation (7) in the paper is used; the equation (9) is used;

%%%%%

num\_map = size(det\_map,2);

for i = 1:num\_map

det\_map(:,i) = (det\_map(:,i) - min(det\_map(:,i))) /(max(det\_map(:,i))-min(det\_map(:,i)));

end

**% PD and PF**

if mode\_tau == 1

tau = sort(det\_map,'descend');

else

for k = 1:num\_map

tau(:,k) = [0:0.01:1]';

end

tau = sort(tau,'descend');

end

for k = 1:num\_map

for i = 1: length(tau)

AD\_bw =det\_map(:,k);

if mode\_eq==1

AD\_bw(det\_map(:,k)>=tau(i,k))=1;

AD\_bw(det\_map(:,k)<tau(i,k))=0;

else

AD\_bw(det\_map(:,k)>tau(i,k))=1;

AD\_bw(det\_map(:,k)<=tau(i,k))=0;

end

[PD(i,k),PF(i,k)] = cal\_pdpf(AD\_bw,GT);

end

end

**% Compute a1, a0, b1 and b0 and normalized PD PF**

a1 = min(PD(1,:));

a0 = max(max(PD));

b1=min(PF(1,:));

b0 = max(max(PF));

AUCnor.a1 = a1;

AUCnor.a0 = a0;

AUCnor.b1 = b1;

AUCnor.b0 = b0;

**% AUC (PF, PD) and Nor\_AUC (PF, PD)**

for i = 1:num\_map

AUC0 = trapz(PF(:,i),PD(:,i));

AUC.PFPD(i,1) = round(AUC0,6);

AUCnor.PFPD(i,1) =round((AUC0-a1)/((a0-a1)\*(b0-b1)),6);

end

**% AUC (PD, tau) and Nor\_AUC (PD, tau)**

for i = 1:num\_map

AUC0 = -trapz(tau(:,i),PD(:,i));

AUC.tauPD(i,1) = round(AUC0,6);

AUCnor.tauPD(i,1) = round((AUC0-a1)/(a0-a1),6);

end

**% AUC (PF, tau) and Nor\_AUC (PF, tau)**

for i = 1:num\_map

AUC0 = abs(trapz(tau(:,i),PF(:,i)));

AUC.tauPF(i,1) =round(AUC0,6);

AUCnor.tauPF(i,1) =round((AUC0-b1)/(b0-b1),6);

end

**function plot\_3DROC(det\_map,GT,detec\_label,mode\_eq)**

% det\_map: the detection result N\*k, N is the number of pixels in the detection result,

% GT: Ground truth

% detec\_label: the name of detector for legend

% mode\_eq: if mode\_eq==1. the equation (7) in the paper is used;the equation (9)is used;

num\_map = size(det\_map,2);

for i = 1:num\_map

det\_map(:,i) = (det\_map(:,i) - min(det\_map(:,i))) /(max(det\_map(:,i))-min(det\_map(:,i)));

end

**%PD and PF based on uniform step and sample value**

for k = 1:num\_map

tau1(:,k) = [0:0.01:1]';

end

tau1 = sort(tau1,'descend');

for k = 1:num\_map

for i = 1: length(tau1)

map =det\_map(:,k);

if mode\_eq==1

map(det\_map(:,k)>=tau1(i,k))=1;

map(det\_map(:,k)<tau1(i,k))=0;

else

map(det\_map(:,k)>tau1(i,k))=1;

map(det\_map(:,k)<=tau1(i,k))=0;

end

[PD1(i,k),PF1(i,k)] = cal\_pdpf(map,GT);

end

end

map = [];

tau2 = sort(det\_map,'descend');

for k = 1:num\_map

for i = 1: length(tau2)

map =det\_map(:,k);

if mode\_eq==1

map(det\_map(:,k)>=tau2(i,k))=1;

map(det\_map(:,k)<tau2(i,k))=0;

else

map(det\_map(:,k)>tau2(i,k))=1;

map(det\_map(:,k)<=tau2(i,k))=0;

end

[PD2(i,k),PF2(i,k)] = cal\_pdpf(map,GT);

end

end

%

a11 = min(PD1(1,:));

a10 = max(max(PD1));

b11=min(PF1(1,:));

b10 = max(max(PF1));

a21 = min(PD2(1,:));

a20 = max(max(PD2));

b21=min(PF2(1,:));

b20 = max(max(PF2));

PD1nor = (PD1-a11)/(a10-a11);

PF1nor = (PF1-b11)/(b10-b11);

PD2nor = (PD2-a21)/(a20-a21);

PF2nor = (PF2-b21)/(b20-b21);

**% show ROC (PF, PD)**

figure,plot(PF1,PD1,'LineWidth',2)

hold on

plot(PF2,PD2,'LineWidth',2,'linestyle','--')

axis([0,1,0,1])

set(gca,'XTick',(0:0.2:1),'fontsize',16)

set(gca,'YTick',(0:0.2:1),'fontsize',16)

set(gca, 'XDir','reverse')

xlabel('P\_F','fontsize',18) ; ylabel('P\_D','fontsize',18)

grid on

for i = 1:num\_map

name1(i) =strcat(detec\_label(i),',','{\Delta}','=0.01');

name2(i) = strcat(detec\_label(i),',','sample values');

end

legend([name1,name2])

legend boxoff

**% show normalized ROC (PFnor, PDnor)**

figure,plot(PF1nor,PD1nor,'LineWidth',2)

hold on

plot(PF2nor,PD2nor,'LineWidth',2,'linestyle','--')

axis([0,1,0,1])

set(gca,'XTick',(0:0.2:1),'fontsize',16)

set(gca,'YTick',(0:0.2:1),'fontsize',16)

set(gca, 'XDir','reverse')

xlabel('P\_F','fontsize',18) ; ylabel('P\_D','fontsize',18)

grid on

legend([name1,name2])

hold off

legend boxoff

**% show ROC (PD, Tau) based on uniform step and sample value**

figure,plot(tau1,PD1,'LineWidth',2)

hold on

plot(tau2,PD2,'LineWidth',2,'linestyle','--')

axis([0,1,0,1])

set(gca,'XTick',(0:0.2:1),'fontsize',16)

set(gca,'YTick',(0:0.2:1),'fontsize',16)

xlabel('\tau','fontsize',18) ; ylabel('P\_D','fontsize',18)

grid on

hold off

legend([name1,name2])

legend boxoff

**% show normalized ROC (PDnor, Tau)**

figure,plot(tau1,PD1nor,'LineWidth',2)

hold on

plot(tau2,PD2nor,'LineWidth',2,'linestyle','--')

axis([0,1,0,1])

set(gca,'XTick',(0:0.2:1),'fontsize',16)

set(gca,'YTick',(0:0.2:1),'fontsize',16)

xlabel('\tau','fontsize',18) ; ylabel('P\_D','fontsize',18)

grid on

legend([name1,name2])

hold off

legend boxoff

**% show ROC (PF, Tau) based on uniform step and sample value**

figure,plot(tau1,PF1,'LineWidth',2)

hold on

plot(tau2,PF2,'LineWidth',2,'linestyle','--')

axis([0,1,0,1])

set(gca,'XTick',(0:0.2:1),'fontsize',16)

set(gca,'YTick',(0:0.2:1),'fontsize',16)

xlabel('\tau','fontsize',18) ; ylabel('P\_F','fontsize',18)

grid on

hold off

legend([name1,name2])

legend boxoff

**% show normalized ROC (PFnor, Tau)**

figure,plot(tau1,PF1nor,'LineWidth',2)

hold on

plot(tau2,PF2nor,'LineWidth',2,'linestyle','--')

axis([0,1,0,1])

set(gca,'XTick',(0:0.2:1),'fontsize',16)

set(gca,'YTick',(0:0.2:1),'fontsize',16)

xlabel('\tau','fontsize',18) ; ylabel('P\_F','fontsize',18)

grid on

legend([name1,name2])

hold off

legend boxoff

**% 3D ROC**

figure,plot3(PF1,tau1,PD1,'LineWidth',2)

hold on

plot3(PF2,tau2,PD2,'LineWidth',2,'linestyle','--')

hold off

axis([0, 1, 0, 1, 0, 1])

set(gca,'XTick',(0:0.2:1),'fontsize',16)

set(gca,'YTick',(0:0.2:1),'fontsize',16)

set(gca,'ZTick',(0:0.2:1),'fontsize',16)

xlabel('P\_F','fontsize',18) ; ylabel('\tau','fontsize',18); zlabel('P\_D','fontsize',18)

grid on

ax = gca;

ax.BoxStyle = 'full';

box on

legend([name1,name2],'fontsize',12)

legend boxoff

**% Normalized 3D ROC**

figure,plot3(PF1nor,tau1,PD1nor,'LineWidth',2)

hold on

plot3(PF2nor,tau2,PD2nor,'LineWidth',2,'linestyle','--')

hold off

legend([name1,name2],'fontsize',12)

axis([0, 1, 0, 1, 0, 1])

set(gca,'XTick',(0:0.2:1),'fontsize',16)

set(gca,'YTick',(0:0.2:1),'fontsize',16)

set(gca,'ZTick',(0:0.2:1),'fontsize',16)

xlabel('P\_F','fontsize',18) ; ylabel('\tau','fontsize',18); zlabel('P\_D','fontsize',18)

grid on

ax = gca;

ax.BoxStyle = 'full';

box on

legend boxoff

**function [PD, PF] = cal\_pdpf(result,GT)**

BKG = find(GT == 0);

g\_result = find(GT == 1);

d\_result = find(result(:) == 1);

PD(:) = length(intersect(d\_result,g\_result))/length(g\_result);

PF(:) = length(intersect(d\_result,BKG))/length(BKG);