Part f)

i)

As I was following the notation of jiang2000.pdf, all my feature matrices are transpose of what is asked in project instructions.

```
[m,n]=size(I_code3b);
%considering bifurcations and endpoints
[x1,y1]=find(I_code3b==1);
[x3,y3]=find(I_code3b==3);
r=[x1;x3];
c=[y1;y3];
F=nan(4,numel(r));
F(1,:)=r;
F(2,:)=c;
F(4,:)=3;
F(4,1:numel(x1))=1;

rmin1=r-3; rmax1=r+3; cmin1=c-3; cmax1=c+3;
rmin1(rmin1<1)=1; cmin1(cmin1<1)=1;
rmax1(rmax1>m)=m; cmax1(cmax1>n)=n;

%building branch angle matrix. in this case the whole branch is considered for linear regression because it is just a means to make invariant properties
Unq=unique(I_branches3b);
Unq(1)=[];
I_Sep_branches3b=double(repmat(I_branches3b,1,1,numel(Unq))==repmat(reshape(Unq,1,1,[]),size(I_branches3b)));
I_branch_angles=zeros(size(I_Sep_branches3b));
for i=1:size(I_Sep_branches3b,3)
   [x,y]=find(I_Sep_branches3b(:,:,i));
   p=polyfit(x,y,1);
   Angle=atan(p(1));
   I_branch_angles(:,:,i)=I_Sep_branches3b(:,:,i)*Angle;
end
I_branch_angles=sum(I_branch_angles,3);

%filling third row of feature matrix
for i=1:numel(r)
    select_window=zeros(size(I_code3b));
    select_window(rmin1(i):rmax1(i),cmin1(i):cmax1(i))=1;
    Unq=unique(select_window.*I_branch_angles);
    Unq(Unq==0)=[];
    F(3,i)=mean(Unq);
end
index=find(isnan(F(3,:)));
r(index)=[];
c(index)=[];
F(:,index)=[];
disp(F(:,1:10)')
```

```
   287.0000    7.0000    0.6435    1.0000
   190.0000    9.0000   -0.7383    1.0000
   205.0000   10.0000   -0.7863    1.0000
   172.0000   12.0000    0.9456    1.0000
   236.0000   13.0000   -0.6959    1.0000
   267.0000   13.0000   -0.7264    1.0000
   250.0000   14.0000   -0.7289    1.0000
   150.0000   16.0000   -0.5635    1.0000
   296.0000   16.0000    0.6435    1.0000
   125.0000   18.0000   -0.7574    1.0000
```

As it was required to repeat previous parts in order to get feature matrices and local feature matrices in (ii), a function **[F3,F]=feature_matrix(I_bin)**, which is presented at the end of this PDF file, was defined and used to reduce calculation time.

ii)

```
L=3;
F3=nan(13,size(F,2));
for i=1:size(F,2)
    X=repmat(F(1,i),1,size(F,2));
    Y=repmat(F(2,i),1,size(F,2));
    dX=F(1,:)-X;
    dY=F(2,:)-Y;
    theta=atan2(dY,dX);
    theta(isnan(theta))=pi/2;
    d=sqrt((dX).^2+(dY).^2);
    F1=[d;theta;F];
    %sorting in order to get "L" nearest minutia
    F1=sortrows(F1')';
    F3(1:L,i)=F1(1,2:1+L);
    %function dPhi is equal to equation (2) in Jian2000.pdf and is presented at the end of this PDF file
```

```
    F3(4:3+L,i)=dPhi(F1(2,2:1+L),F1(5,1));
    F3(7:6+L,i)=dPhi(F1(5,1),F1(5,2:1+L));
    F3(10:end,i)=F1(6,1:1+L);
end

disp(F3(:,1:10)')
```

```
   12.0830   12.7279   14.4222    0.5007    0.1419    1.5153    1.3462         0    1.3832    1.0000    1.0000    1.0000    3.0000
   12.5300   14.7648   14.8661    1.8098    1.8152    1.5713   -0.0283   -0.1046    0.0480    1.0000    3.0000    3.0000    3.0000
   11.1803   11.6619   11.7047    2.8207    1.8167    2.7058         0    0.0406         0    1.0000    3.0000    3.0000    3.0000
   10.0000   13.9284   18.2483    1.5525    0.2580   -1.1107    1.5783    1.2118    1.6839    1.0000    1.0000    1.0000    1.0000
   14.0357   16.0312   17.0294    0.7672   -2.3833   -2.3870    0.0330    0.1310    0.1310    1.0000    1.0000    3.0000    3.0000
   11.6619   12.5300   13.4164    1.2668    1.2258    1.1901    0.0133    0.0133    0.0133    1.0000    3.0000    3.0000    3.0000
   14.0357   17.0294   27.4591   -2.3414    0.6701    0.9120   -0.0330   -0.0024    0.0109    1.0000    1.0000    1.0000    3.0000
   14.1421   14.7648   14.8661    0.7054    3.0720    2.9211    0.0692    0.1109    0.1109    1.0000    1.0000    3.0000    3.0000
    4.4721   12.1655   12.7279    2.0344    0.7621   -2.9997    1.3462    1.2542         0    1.0000    1.0000    1.0000    1.0000
   11.4018   11.7047   12.0416    1.0236    1.1061    2.2450   -0.0829   -0.0829    0.0107    1.0000    3.0000    3.0000    1.0000
```

iii)

sl=similarity_level(Fli,Flk) is presented at the end of this PDF file

```
Temp=cell(10,1);
Input=cell(10,7);
for i=1:10
    Bin=imread([cd '\Binary\' num2str(100+i) '_1.png']);
    Bin(Bin~=0)=1;
    [Temp{i},~]=feature_matrix(Bin);
   for j=2:8
       Bin=imread([cd '\Binary\' num2str(100+i) '_' num2str(j) '.png']);
       Bin(Bin~=0)=1;
       [Input{i,j-1},~]=feature_matrix(Bin);
   end
end
Sim=cell(10,7,10);
for i=1:10
   for j=1:10
       for k=1:7
           Sim{j,k,i}=similarity_level(Input{j,k},Temp{i});
       end
   end
end

sl=Sim{1,1,1};
disp(sl(1:10,1:10))
```

```
         0         0         0         0         0         0         0         0         0         0
         0    0.1297         0         0         0    0.3605         0         0         0    0.1892
         0    0.2973    0.4003         0         0    0.4461         0    0.1446         0    0.5732
         0    0.1020    0.1530         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0
         0    0.2762         0         0         0    0.4359         0         0         0    0.2754
         0         0    0.1388         0         0         0         0         0         0         0
```

```
[loc1,loc2]=find(sl==max(max(sl)));
disp(['Max similarity level is equal to ' num2str(max(max(sl))) ' at: ' num2str([loc1,loc2])])
```

```
 Max similarity level is equal to 0.99421 at: 356  362
```

iv)

```
%cell array for Fgs for Templates
Fg_input=cell(10,7,10);
%cell array for Fgs for Inputs
Fg_temp=cell(10,7,10);
for i=1:10
   for j=1:10
       for k=1:7
           [r,c]=find(Sim{j,k,i}==max(max(Sim{j,k,i})));
           r(2:end)=[]; c(2:end)=[];

           F=Feature{i,1};
           Fg=zeros(3,size(F,2));
           X=repmat(F(1,c),1,size(F,2));
           Y=repmat(F(2,c),1,size(F,2));
           dX=F(1,:)-X;
           dY=F(2,:)-Y;
           Fg(1,:)=sqrt((dX).^2+(dY).^2);
```

```
            theta=atan2(dY,dX);
            theta(isnan(theta))=pi/2;
            Fg(2,:)=dPhi(theta,F(3,c));
            Fg(3,:)=dPhi(F(3,:),F(3,c));
            Fg_temp{j,k,i}=Fg;

            F=Feature{j,k+1};
            Fg=zeros(3,size(F,2));
            X=repmat(F(1,r),1,size(F,2));
            Y=repmat(F(2,r),1,size(F,2));
            dX=F(1,:)-X;
            dY=F(2,:)-Y;
            Fg(1,:)=sqrt((dX).^2+(dY).^2);
            theta=atan2(dY,dX);
            theta(isnan(theta))=pi/2;
            Fg(2,:)=dPhi(theta,F(3,r));
            Fg(3,:)=dPhi(F(3,:),F(3,r));
            Fg_input{j,k,i}=Fg;
        end
    end
end

Fg=Fg_temp{1,1,1};
disp(Fg(:,1:10)')
```

```
   270.4145   -0.9464    0.6347
   225.0000   -1.2958   -0.7471
   228.7116   -1.2315   -0.7950
   217.7016   -1.3713    0.9368
   238.3799   -1.1047   -0.7046
   254.0551   -0.9959   -0.7352
   244.2335   -1.0518   -0.7376
   210.2617   -1.4699   -0.5723
   268.7787   -0.8996    0.6347
   207.0097   -1.5892   -0.7661
```

v)

ml=matching_level(Fgi,Fgk) is presented at the end of this PDF file

there is an extra variable in my function definition because according to equation (11) similarity matrix is also needed to calculate matching level matrix

```
ml=cell(10,7,10);
for i=1:10
    for j=1:10
        for k=1:7
            ml{j,k,i}=matching_level(Fg_input{j,k,i},Fg_temp{j,k,i},Sim{j,k,i});
        end
    end
end

Ms=nan(10,7,10);
for i=1:10
    for j=1:10
        for k=1:7
            %equation (12)
            Ms(j,k,i)=100*sum(sum(ml{j,k,i}))/max(size(ml{j,k,i}));
        end
    end
end

disp(Ms(:,1,1))
```

```
   103.1372
    99.5816
   103.7940
   213.3494
    82.9818
    85.2102
   128.8758
    73.5613
    70.0555
   151.5161
```

Functions

```
function dp=dPhi(t1,t2)
dp=t1-t2;
dp(dp<-pi)=dp(dp<-pi)+2*pi;
dp(dp>pi)=-dp(dp>pi)+2*pi;
end
```

```
function sl=similarity_level(Fli,Flk)
W=[ones(1,3) repmat(.3*180/pi,1,6) 3*ones(1,4)];
bl=6*size(Fli,1);
sl=zeros(size(Fli,2),size(Flk,2));
for i=1:size(Fli,2)
   for j=1:size(Flk,2)
      if W*abs(Fli(:,i)-Flk(:,j))<bl
         sl(i,j)=(bl-W*abs(Fli(:,i)-Flk(:,j)))/bl;
      end
   end
end
end
```

```
function ml=matching_level(Fgi,Fgk,sl)
Bg=[8 pi/6 pi/6]';
ml=zeros(size(Fgi,2),size(Fgk,2));
for i=1:size(Fgi,2)
    for j=1:size(Fgk,2)
        if sum(abs(Fgi(i)-Fgk(j))<Bg)==3
            %equation (11)
           ml(i,j)=0.5+0.5*sl(i,j);
        end
    end
end
%implementing paragraph between (11) and (12)
for i=1:size(ml,1)
    for j=1:size(ml,2)
        if sum(sum(ml(i,j)<ml(i,:)))>0 || sum(sum(ml(i,j)<ml(:,j)))>0
            ml(i,j)=0;
        end
    end
end
end
```

```
function [F3,F]=feature_matrix(I_bin)
I_skel=double(bwmorph(I_bin,'skel',inf));
I_code=code_branches(I_skel);
I_branches=I_skel;
I_branches(I_code>2)=0;
I_branches=bwlabel(I_branches);
Unq=unique(I_branches);
Unq(1)=[];
Rep_branches=repmat(I_branches,1,1,numel(Unq));
Rep_values=repmat(reshape(Unq,1,1,[]),size(I_branches));
I_branches3a=double(Rep_branches==Rep_values);
Modfr=sum(sum(I_branches3a));
Modfr(Modfr<4)=0;
Modfr(Modfr~=0)=1;
Modfr=repmat(Modfr,size(I_branches));
I_branches3a=sum((I_branches3a.*Modfr),3);
I_skel3a=I_branches3a+double(I_code>2);
I_code3a=code_branches(I_skel3a);
I_branches3a=bwlabel(I_branches3a);
lambda=11;
[m,n]=size(I_code3a);
[r,c]=find((I_code3a==3)+(I_code3a==4));
rmin1=r-3; rmax1=r+3; cmin1=c-3; cmax1=c+3;
rmin2=r-lambda-3; rmax2=r+lambda+3; cmin2=c-lambda-3; cmax2=c+lambda+3;
rmin1(rmin1<1)=1; cmin1(cmin1<1)=1;
rmin2(rmin2<1)=1; cmin2(cmin2<1)=1;
rmax1(rmax1>m)=m; cmax1(cmax1>n)=n;
rmax2(rmax2>m)=m; cmax2(cmax2>n)=n;
I_branches3b=I_branches3a;
for i=1:numel(r)
    select_window=zeros(size(I_code3a));
    select_window(rmin1(i):rmax1(i),cmin1(i):cmax1(i))=1;
    select_branch=select_window.*I_branches3a;
    Unq=unique(select_branch);
    Unq(Unq==0)=[];
    if numel(Unq)==3
        analysis_window=zeros(size(I_code3a));
        analysis_window(rmin2(i):rmax2(i),cmin2(i):cmax2(i))=1;
        analysis_branch=analysis_window.*I_branches3a;
        Npix(1)=sum(sum(I_branches3a==Unq(1)));
        Npix(2)=sum(sum(I_branches3a==Unq(2)));
        Npix(3)=sum(sum(I_branches3a==Unq(3)));
        sNpix=sort(Npix);
```

```
        [x,y]=find(analysis_branch==Unq(1));
        p1=polyfit(x,y,1);
        u1=[p1(2)/p1(1),p1(2)]/sqrt(p1(2)^2/p1(1)^2+p1(2)^2);
        [x,y]=find(analysis_branch==Unq(2));
        p2=polyfit(x,y,1);
        u2=[p2(2)/p2(1),p2(2)]/sqrt(p2(2)^2/p2(1)^2+p2(2)^2);
        [x,y]=find(analysis_branch==Unq(3));
        p3=polyfit(x,y,1);
        u3=[p3(2)/p3(1),p3(2)]/sqrt(p3(2)^2/p3(1)^2+p3(2)^2);
        theta(1)=dot(u2,u3);
        theta(2)=dot(u1,u3);
        theta(3)=dot(u1,u2);
        theta=abs(theta);
        thetac=theta;
        thetac(theta==max(theta))=[];
        index=find(theta==max(theta));
        if sNpix(1)>=lambda
            if (max(theta)>0.85) && ((thetac(1)<=0.55) || (thetac(2)<=0.55))
                if numel(I_branches3b==Unq(index))<=round(5*lambda/6)
                    I_branches3b(I_branches3b==Unq(index))=0;
                end
            end
            if (max(theta)>0.85) && ((thetac(1)>0.55 && thetac(1)<0.85) || (thetac(2)>0.55 && thetac(2)<0.85))
                if numel(I_branches3b==Unq(index))<=round(3*lambda/2)
                    I_branches3b(I_branches3b==Unq(index))=0;
                end
            end
        end
        if (sNpix(1)<lambda)% && (sNpix(2)>=lambda) && (sNpix(3)>=lambda)
            if (max(theta)>0.8) && (thetac(1)<0.8) && (thetac(2)<0.8)
                I_branches3b(I_branches3b==Unq(index))=0;
            end
        end
    end
end
I_skel3b=double((I_branches3b~=0)+(I_code3a>2));
I_code3b=code_branches(I_skel3b);
[m,n]=size(I_code3b);
[x1,y1]=find(I_code3b==1);
[x3,y3]=find(I_code3b==3);
r=[x1;x3];
c=[y1;y3];
F=nan(4,numel(r));
F(1,:)=r;
F(2,:)=c;
F(4,:)=3;
F(4,1:numel(x1))=1;
rmin1=r-3; rmax1=r+3; cmin1=c-3; cmax1=c+3;
rmin1(rmin1<1)=1; cmin1(cmin1<1)=1;
rmax1(rmax1>m)=m; cmax1(cmax1>n)=n;
Unq=unique(I_branches3b);
Unq(1)=[];
I_Sep_branches3b=double(repmat(I_branches3b,1,1,numel(Unq))==repmat(reshape(Unq,1,1,[]),size(I_branches3b)));
I_branch_angles=zeros(size(I_Sep_branches3b));
for i=1:size(I_Sep_branches3b,3)
   [x,y]=find(I_Sep_branches3b(:,:,i));
   p=polyfit(x,y,1);
   Angle=atan(p(1));
   I_branch_angles(:,:,i)=I_Sep_branches3b(:,:,i)*Angle;
end
I_branch_angles=sum(I_branch_angles,3);
for i=1:numel(r)
    select_window=zeros(size(I_code3b));
    select_window(rmin1(i):rmax1(i),cmin1(i):cmax1(i))=1;
    Unq=unique(select_window.*I_branch_angles);
    Unq(Unq==0)=[];
    F(3,i)=mean(Unq);
end
F(:,isnan(F(3,:)))=[];
L=3;
F3=nan(13,size(F,2));
for i=1:size(F,2)
    X=repmat(F(1,i),1,size(F,2));
    Y=repmat(F(2,i),1,size(F,2));
    dX=F(1,:)-X;
    dY=F(2,:)-Y;
    theta=atan2(dY,dX);
    theta(isnan(theta))=pi/2;
    d=sqrt((dX).^2+(dY).^2);
    F1=[d;theta;F];
    F1=sortrows(F1')';
```

```
    F3(1:L,i)=F1(1,2:1+L);
    F3(4:3+L,i)=dPhi(F1(2,2:1+L),F1(5,1));
    F3(7:6+L,i)=dPhi(F1(5,1),F1(5,2:1+L));
    F3(10:end,i)=F1(6,1:1+L);
end
end
```