

Melanoma Recognition Code Manual

Jiexin Guo, Junjian Xie
ID: 14213839, 14213834

2014.11.29

Contents

1	Overview	3
1.1	Introduction	4
1.2	Goal	4
1.3	Environment	5
1.4	Execution	5
2	Code View	6
2.1	main.m	7
2.1.1	Function	7
2.1.2	Accelerating the program	7
2.1.3	Parameters	7
2.1.4	Output	8
2.1.5	Code of main.m	8
2.2	trainMyNetwork.m	10
2.2.1	Function	10
2.2.2	Accelerating the program	10
2.2.3	Input parameters	10
2.2.4	Internal parameters	10
2.2.5	Output parameters	11
2.2.6	Code of trainMyNetwork.m	12
2.3	getSegment.m	14
2.3.1	Function	14
2.3.2	Accelerating the program	14
2.3.3	Input parameters	14
2.3.4	Output parameters	14
2.3.5	Output	14
2.3.6	Code of getSegment.m	16
2.4	featureExtra.m	18
2.4.1	Function	18
2.4.2	Input parameters	18
2.4.3	Output	18
2.4.4	Code of featureExtra.m	19
3	Application	20
3.1	Image pipeline	21
3.1.1	Median filter	21
3.1.2	Segmentation with Active contour without edges	21

3.1.3	Feature extracion	22
3.2	Dataset	22
3.3	Neural Network	23
3.3.1	Network definition	23
3.3.2	Network input and output	23
3.4	Testing	24
3.4.1	Changing dataset	24
3.4.2	Using default network	24
3.4.3	Train your network	25
3.4.4	Results	25
4	Future work	26
4.1	What we can improve:	27
4.1.1	Data	27
4.1.2	Speed	27

Chapter 1

Overview

1.1 Introduction

Melanoma is a type of skin cancer which forms from melanocytes (pigment-containing cells in the skin) and it is one kind of the skin cancer.

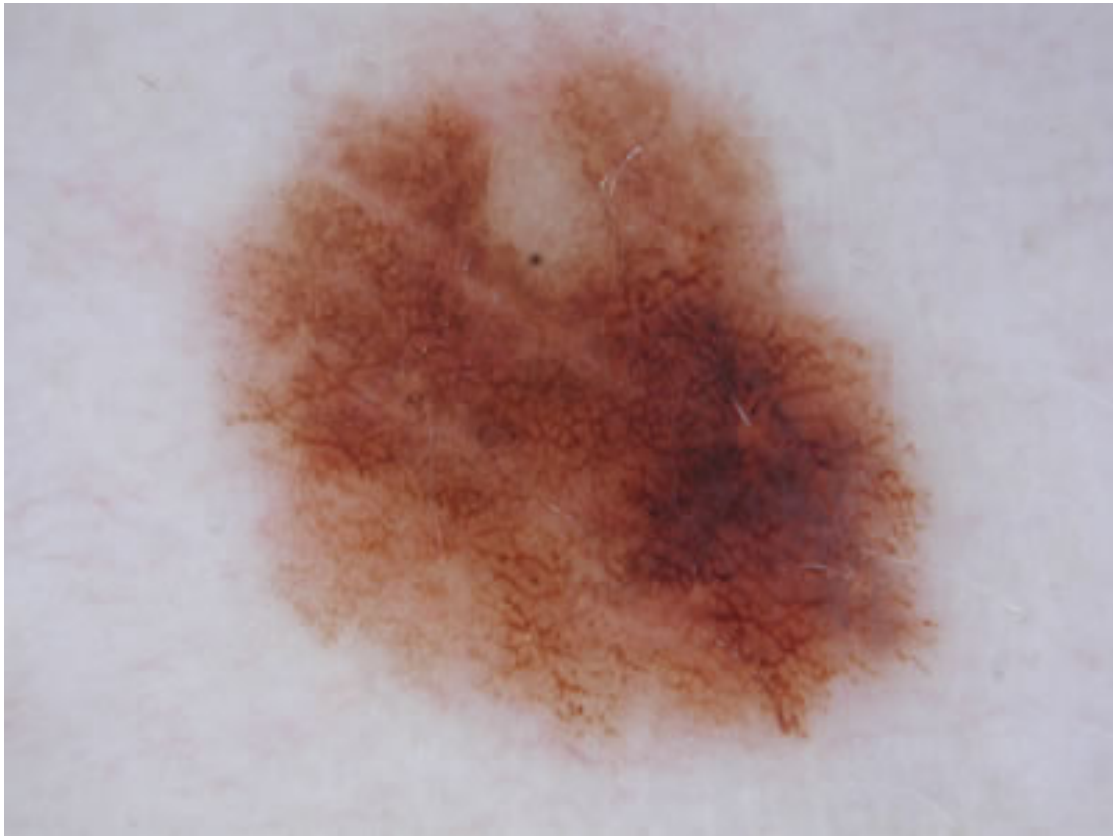


Figure 1.1: Melanoma

If melanoma is found early, while it is still small and thin, and if it is completely removed, then the chance of cure is high. However, it is much more dangerous if it is not found in the early stages. It causes the majority (75%) of deaths related to skin cancer. So we try to find a way of image recognition to identify this disease and try to help.

In the following passage we will use Melanoma or cancer refers to Melanoma.

1.2 Goal

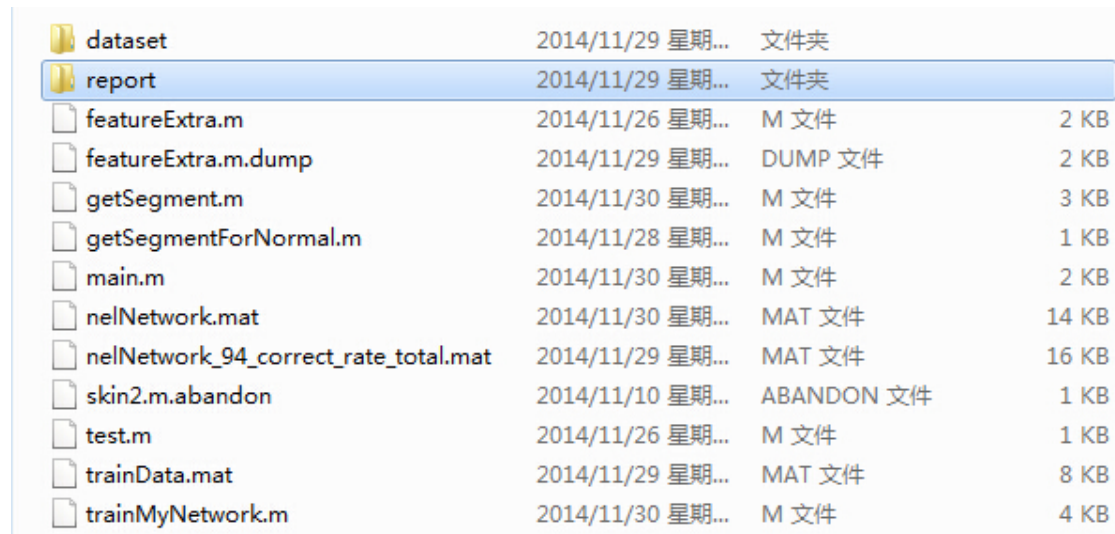
Our goal is to develop a tool that can automatically recognize whether an image is of Melanoma, and its correct rate should be more than 80%.

1.3 Environment

Our program is developed using Matlab 2014a, with 64bit windows. Since we have used the Neural Network Toolbox to set up and train our neural network, you should run this code with a Matlab version(maybe 2009 or higher) that has the Neural Network Toolbox(nntool).

1.4 Execution

Unzip the source file, do not change the directory structure, and you should see this:



File/Folder	Date	Type	Size
dataset	2014/11/29 星期...	文件夹	
report	2014/11/29 星期...	文件夹	
featureExtra.m	2014/11/26 星期...	M 文件	2 KB
featureExtra.m.dump	2014/11/29 星期...	DUMP 文件	2 KB
getSegment.m	2014/11/30 星期...	M 文件	3 KB
getSegmentForNormal.m	2014/11/28 星期...	M 文件	1 KB
main.m	2014/11/30 星期...	M 文件	2 KB
nelNetwork.mat	2014/11/30 星期...	MAT 文件	14 KB
nelNetwork_94_correct_rate_total.mat	2014/11/29 星期...	MAT 文件	16 KB
skin2.m.abandon	2014/11/10 星期...	ABANDON 文件	1 KB
test.m	2014/11/26 星期...	M 文件	1 KB
trainData.mat	2014/11/29 星期...	MAT 文件	8 KB
trainMyNetwork.m	2014/11/30 星期...	M 文件	4 KB

Figure 1.2: Directory structure

Then open your Matlab and change the directory to this so that you can directly run the main.m . Next we will talk about every file's function in chapter 2.

Chapter 2

Code View

2.1 main.m

2.1.1 Function

This is the main entrance of the program.

2.1.2 Accelerating the program

Please see this part in the section of [Accelerating the program](#) in the [trainMyNetwork.m](#) section and also [Accelerating the program](#) in the [getSegment.m](#).

2.1.3 Parameters

The parameters are set in this .m file's line 4 to line 7, like below:

```
1  numberOfCancer=56;%use how many cases of cancer to make up the data ,should be ...  
    equal to the number of jpgs in dataset\cancer\  
2  numberOfOther=32;%use how many cases of other to make up the data ,should be ...  
    equal to the number of jpgs in dataset\normal\  
3  reComputeFeature=false;  
4  reTrainNetwork=true;
```

numberOfCancer

This is the total number of cancer image, usually it should be equal to or less than the number of jpg files in `dataset/cancer/`.

numberOfOther

This is the total number of not cancer image(the opposite of our recognition,maybe normal skin or other skin disease), usually it should be equal to or less than the number of jpg files in `dataset/normal/`.

reComputeFeature

A boolean, if it is true, the program will recompute all the features of the first 'numberOfCancer' files in `dataset/cancer/` and all the first 'numberOfOther' files in `dataset/normal/`.

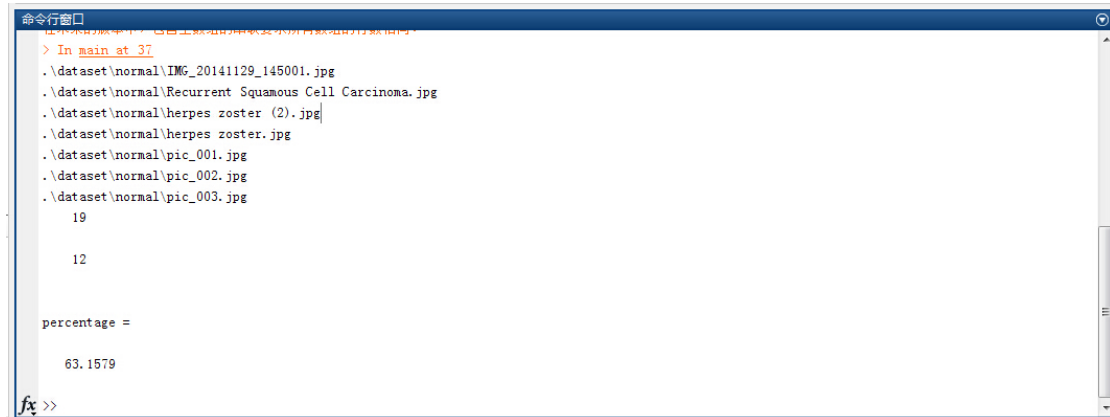
For example, if reComputeFeature is true, numberOfCancer is 56 and numberOfOther is 32, then the program will compute the feature of the first 56 jpgs in `dataset/cancer/` and first 32 jpgs in `dataset/normal/` and save them into a file:'trainData.mat'.

If it is set to be false, then the program will not recompute the feature but to load the features from file:'trainData.mat'.

reTrainNetwork

Also a boolean.True tells the function 'trainMyNetwork' to retrain the network. False will not retrain the network but to load the network from file: 'nelNetwork.mat'.

2.1.4 Output



```

命令窗口
> In main at 37
.\dataset\normal\IMG_20141129_145001.jpg
.\dataset\normal\Recurrent Squamous Cell Carcinoma.jpg
.\dataset\normal\herpes zoster (2).jpg
.\dataset\normal\herpes zoster.jpg
.\dataset\normal\pic_001.jpg
.\dataset\normal\pic_002.jpg
.\dataset\normal\pic_003.jpg
19
12
percentage =
63.1579
fx >>

```

Figure 2.1: Output of main.m

This is the console output of running main.m.

The first several lines of file path are those recognition failed files' path.

The next number 19 is the total test case number.

The next number 12 is the test case number of correct recognition.

The last number is the correct rate.

Here is another example of our best correct rate using 'nelNetwork_94_correct_rate_total.mat' (please change this mat file's name to nelNetwork.mat and run):



```

命令窗口
82 83 84 85 86 87 88
7
警告: 此串联操作包含行数不正确的空数组。
在未来的版本中, 包含空数组的串联要求所有数组的行数相同。
> In main at 41
23
22
percentage =
95.6522
fx >>

```

Figure 2.2: Our best output of main.m

2.1.5 Code of main.m

```

1 clc;
2 clear;
3

```

```

4  numberOfCancer=56;%use how many cases of cancer to make up the data ,should be ...
    equal to the number of jpgs in dataset\cancer\
5  numberOfOther=32;%use how many cases of other to make up the data ,should be ...
    equal to the number of jpgs in dataset\normal\
6  reComputeFeature=false;
7  reTrainNetwork=true;
8
9  [ myNet,trainNumberOfCancer,trainNumberOfOther ...
    ]=trainMyNetwork(numberOfCancer,numberOfOther,reComputeFeature,reTrainNetwork);
10 % return;
11 load trainData;
12 ptestRange=1+trainNumberOfCancer:numberOfCancer;
13 ptest=p(:,ptestRange);
14 ttest=zeros(13,0);
15 hitnumber=0;
16 testsize=size(ptest,2);
17 for i=1:size(ptest,2)
18     ttest=[ttest sim(myNet,ptest(:,i))];
19     if(ttest(1,i)>ttest(2,i))
20         if t(1,trainNumberOfCancer+i)==0.9
21             hitnumber=hitnumber+1;
22         else
23             disp(files{1,trainNumberOfCancer+i});
24         end
25     else
26         if t(2,trainNumberOfCancer+i)==0.9
27             hitnumber=hitnumber+1;
28         else
29             disp(files{1,trainNumberOfCancer+i});
30         end
31     end
32 end
33 disp(size(ptest,2));
34 ptestRange=numberOfCancer+trainNumberOfOther+1:numberOfCancer+numberOfOther;
35 disp(ptestRange);
36 ptest=p(:,ptestRange);
37 ttest=zeros(13,0);
38 testsize=testsize+size(ptest,2);
39 disp(size(ptest,2));
40 for i=1:size(ptest,2)
41     ttest=[ttest sim(myNet,ptest(:,i))];
42     if(ttest(1,i)>ttest(2,i))
43         if t(1,numberOfCancer+trainNumberOfOther+i-1)==0.9
44             hitnumber=hitnumber+1;
45         else
46             disp(files{1,numberOfCancer+trainNumberOfOther+i-1});
47         end
48     else
49         if t(2,numberOfCancer+trainNumberOfOther+i-1)==0.9
50             hitnumber=hitnumber+1;
51         else
52             disp(files{1,numberOfCancer+trainNumberOfOther+i-1});
53         end
54     end
55 end
56 disp(testsize);
57 disp(hitnumber);
58 percentage=hitnumber/testsize*100%;

```

2.2 trainMyNetwork.m

2.2.1 Function

This .m file can recompute all the image files' feature (by calling getSegment.m) and set up and retrain the Neural Network(using some of the images as the training data set).

2.2.2 Accelerating the program

The computation of feature may cost a lot of time, you can accelerate this procedure if you have a multiprocessor computer.

You can change the code in this file at line 23 and 39(please open the matlabpool first: `matlabpool ... open 8;`):

```
for i=1:numberofCancer
for i=1:numberofOther
change these above two into:
parfor i=1:numberofCancer
parfor i=1:numberofOther
```

Then you can use all the cpu to compute the features of image. But please ensure that your cpu will not get overheat.

2.2.3 Input parameters

numberOfCancer

It is the same as the parameter `numberOfCancer` in main.m.

numberOfOther

It is the same as the parameter `numberOfOther` in main.m.

retrainFea

A boolean the same as the parameter `reComputeFeature` in main.m.

retrainNetwork

A boolean the same as the parameter `reTrainNetwork` in main.m.

2.2.4 Internal parameters

There are also some parameters can be set in this file, in line 7 to line 8:

```
1 trainNumberOfCancer=40;%means using how many images of cancer to train the network
2 trainNumberOfOther=20; %means using how many images of other to train the network
```

and also line 74 to line 76(network training parameter):

```
1 myNet.trainParam.lr=0.05;
2 myNet.trainParam.goal = 0.01;
3 myNet.trainParam.epochs = 10000;
```

`trainNumberOfCancer`

Using the first 1 to `trainNumberOfCancer` of the cancer images' features to train the network.

`trainNumberOfOther`

Using the first 1 to `trainNumberOfOther` of the none cancer images' features to train the network.

`myNet.trainParam.lr`

This defines the Neural Network's learning rate. If the rate is large, the training speed may increase but the network maybe hard to converge. If the rate is small, the speed of training will be slow but the network would be converged and stable.

We set it to a default value 0.05.

`myNet.trainParam.goal`

This is the target error of neural network training. Its default value is 0.01.

`myNet.trainParam.epochsl`

This is the maximum value of training iteration. Default value is 10000.

2.2.5 Output parameters

`myNet`

Returns the Neural Network.

`trainNumberOfCancer`

The same as [trainNumberOfCancer](#).

`trainNumberOfOther`

The same as [trainNumberOfOther](#).

2.2.6 Code of trainMyNetwork.m

```

1 function [ myNet,trainNumberOfCancer,trainNumberOfOther ] = ...
    trainMyNetwork(numberofCancer,numberofOther,retrainFea,retrainNetwork)
2 %%
3 % This .m file can recompute all the image files' feature (by calling getSegment.m)
4 % and set up and retrain the Neural Network(using some of the images as the ...
    training data set).
5 %
6
7 trainNumberOfCancer=40;%means using how many images of cancer to train the network
8 trainNumberOfOther=20;%means using how many images of other to train the network
9
10 p=zeros(13,0);
11 t=zeros(1,0);
12 files=cell(1,numberofCancer+3);
13 oriimgs=cell(1,numberofCancer+3);
14 imgs=cell(1,numberofCancer+3);
15 if ~retrainFea && exist('trainData.mat','file')
16     load trainData p t files;
17 else
18     readDir = './dataset/cancer';
19     readPath = [readDir '\*.jpg']
20     readList = dir(readPath);
21     %     matlabpool open 8;
22     tic
23     for i=1:numberofCancer
24         picName= readList(i, 1).name;
25         readPath = [readDir '\' picName];
26         files{1,i}=readPath;
27         oriimgs{1,i}=imread(readPath);
28         tocom=getSegment(readPath,false);
29         imgs{1,i}=tocom;
30         fea=featureExtra(tocom);
31         p=[p fea];%p(:,i)=fea;
32         t=[t [0.9;0.1]]; %t(1,i)=1;
33     end
34     toc
35     readDir = './dataset/normal';
36     readPath = [readDir '\*.jpg']
37     readList = dir(readPath);
38     tic
39     for i=1:numberofOther
40         picName= readList(i, 1).name;
41         readPath = [readDir '\' picName];
42         files{1,i+numberofCancer}=readPath;
43         oriimgs{1,i+numberofCancer}=imread(readPath);
44         tocom=getSegment(readPath,false);
45         imgs{1,i+numberofCancer}=tocom;
46         fea=featureExtra(tocom);
47         p=[p fea];%p(:,i)=fea;
48         t=[t [0.1;0.9]]; %t(1,i)=1;
49     end
50     toc;
51     save trainData p t files;
52 end
53 % disp(p);
54 % disp(t);
55 % return;
56 % pr=[0 400;

```

```
57 %      0 100;
58 %      0 50;
59 %      0 400;
60 %      0 100;
61 %      0 50;
62 %      0 400;
63 %      0 100;
64 %      0 50;
65 %      0 10;
66 %      0 10;
67 %      0 10;
68 %      0 10;];
69 if retrainNetwork
70     %      load nelNetwork myNet;
71     %      [p,ps] = mapminmax(p);
72     %      disp(p);
73     myNet=newff(minmax(p),[50,2],{'tansig' 'tansig'},'trainlm');
74     myNet.trainParam.lr=0.05;
75     myNet.trainParam.goal = 0.01;
76     myNet.trainParam.epochs = 10000;
77     [myNet,tr]=train(myNet,p(:, [1:trainNumberOfCancer ...
        numberOfCancer+1:numberOfCancer+trainNumberOfOther]),t(:, [1:trainNumberOfCancer ...
        numberOfCancer+1:numberOfCancer+trainNumberOfOther]));
78     save nelNetwork myNet;
79 else
80     load nelNetwork myNet;
81 end
82 end
```

2.3 getSegment.m

2.3.1 Function

Use median filter to filter the image first. Then we use the active contour without edges algorithm to get the lesion area curve. After these, this function will set the area out of the curve to be blank and return this image.

2.3.2 Accelerating the program

The computation of feature may cost a lot of time, you can accelerate this procedure by decreasing the iteration time of this file in line 39:

```
1 for n=1:700
```

You can change the 700 to a smaller value, it can reduce the time but may also reduce the correct rate.

2.3.3 Input parameters

imgFilename

The path of the image.

printImage

True if the user want to see the segment result.

2.3.4 Output parameters

X

Returns the image after segmentation.

2.3.5 Output

If you set printImage=true, then you will see the next picture:

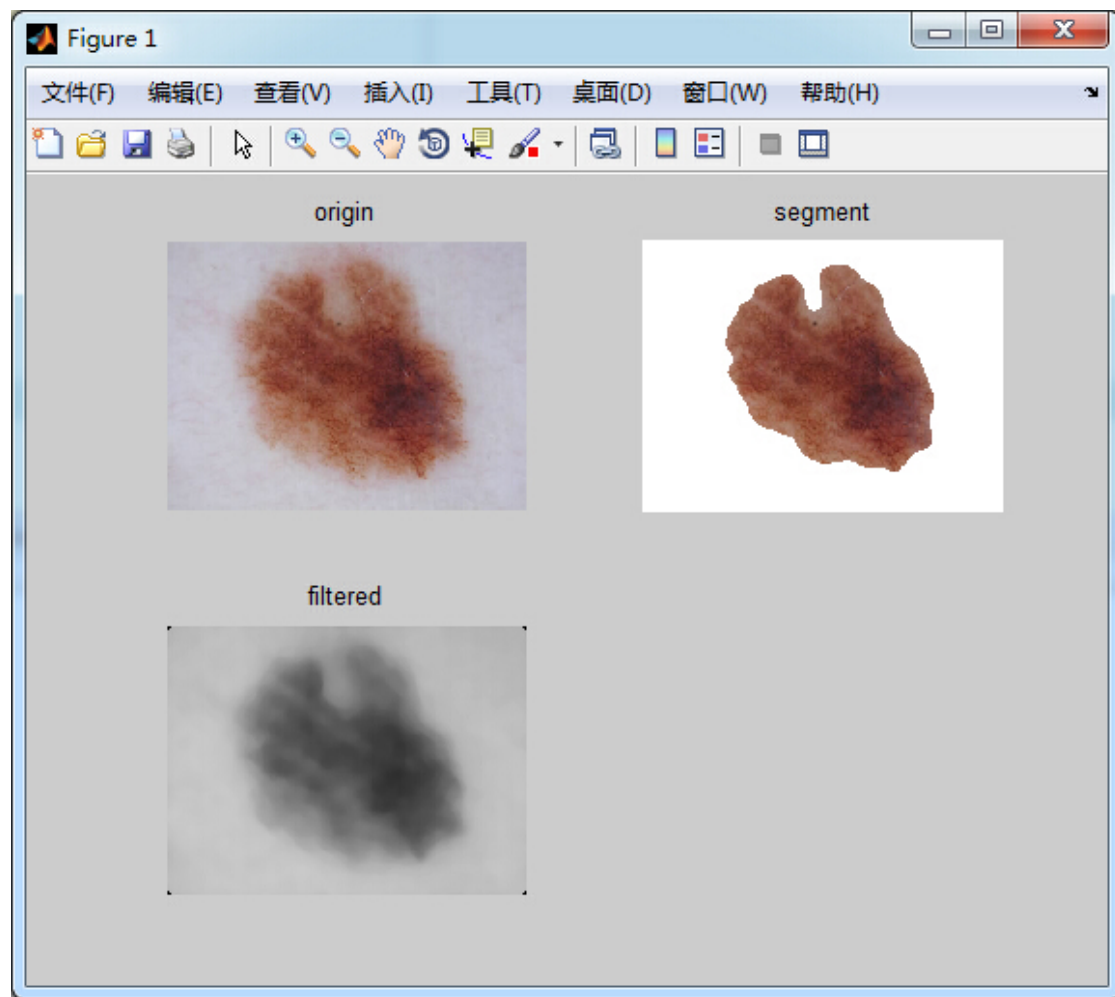


Figure 2.3: getSegment.m output picture

The image on the top left is the origin image. The one on the bottom left is the filtered image. And the last one is the segment result.

2.3.6 Code of getSegment.m

```

1 function [X] = getSegment( imgFilename, printImage )
2 %
3 %   Use median filter to filter the image first. Then we use the active contour ...
4 %   without edges algorithm to get the lesion area curve.
5 %   After these, this function will set the area out of the curve to be blank ...
6 %   and return this image.
7
8 % tic
9 Img=imread(imgFilename);
10 ImgBak=Img;
11 Img=double(rgb2gray(Img));
12 Img=medfilt2(Img,[15,15]);
13 % toc
14 % tic
15 [nx,ny]=size(Img);
16 %%- set init curve C as a circle
17 ic=floor(nx/2);           % compute init circle center
18 jc=floor(ny/2);
19 r=ic/2;                   % radius
20
21 %%- init u as the distance function
22 u = zeros([nx,ny]);
23 for i=1:nx
24     for j=1:ny
25         u(i,j)= r-sqrt((i-ic).^2+(j-jc).^2);
26     end
27 end
28 %%- draw init curve on the picture
29 % figure(2);
30 % imshow(uint8(Img));
31 % hold on;
32 % [c,h] = contour(u,[0 0],'r');
33
34 %%- init parameter
35 epsilon=1.5;              % Heaviside parameter setting
36 nu=2500;
37 Δ_t=0.1;
38
39 nn=0;
40 %%- iter begin
41 for n=1:700
42     %%- regularization of Heaviside
43     H_u = 0.5*(1+(2/pi)*atan(u/epsilon));
44
45     %%- compute c1 and c2 according to u
46     c1=sum(sum(H_u.*Img))/sum(sum(H_u));
47     c2=sum(sum((1-H_u).*Img))/sum(sum(1-H_u));
48     %nu=1/min(c1,c2);
49     %c1=c1.^0.5;c2=c2.^0.5;
50
51     %%- use c1 and c2 to update u
52     Δ_H = (1/pi)*epsilon./(epsilon^2+u.^2);
53     m=Δ_t*Δ_H;
54     C_1 = ...
55         1./sqrt(eps+(u(:,[2:ny,ny])-u).^2+0.25*(u([2:nx,nx],:)-u([1,1:nx-1],:)).^2);
56     C_2 = ...
57         1./sqrt(eps+(u-u(:,[1,1:ny-1])).^2+0.25*(u([2:nx,nx],[1,1:ny-1])-u([1,1:nx-1],[1,1:ny-1])).^2);

```

```

54     C_3 = ...
        1./sqrt(eps+(u([2:nx,nx],:)-u).^2+0.25*(u(:,[2:ny,ny])-u(:,[1,1:ny-1])).^2);
55     C_4 = ...
        1./sqrt(eps+(u-u([1,1:nx-1],:)).^2+0.25*(u([1,1:nx-1],[2:ny,ny])-u([1,1:nx-1],[1,1:ny-1])).^2);
56     C = 1+nu*m.*(C_1+C_2+C_3+C_4);
57     u = ...
        (u+nu*m.*(C_1.*u(:,[2:ny,ny])+C_2.*u(:,[1,1:ny-1])+C_3.*u([2:nx,nx],:)+C_4.*u([1,1:nx-1],:)) ...
        )+...
58     m.*((Img-c2).^2-(Img-c1).^2))./C;
59
60     %%- show the curve and image
61     %     if mod(n,500)==0
62     %         nn=nn+1;
63     %         f=Img;
64     %         f(u>0)=c1;
65     %         f(u<0)=c2;
66     %         figure(nn); subplot(1,2,1);imshow(uint8(f));
67     %         subplot(1,2,2); imshow(uint8(Img));
68     %         hold on;
69     %         [c,h] = contour(u,[0 0],'r');
70     %         hold off;
71     %     end
72 end
73 % toc
74 f=ImgBak;
75 for i=1:nx
76     for j=1:ny
77         if(u(i,j)<0)
78             f(i,j,:)=255;
79         end
80     end
81 end
82 % row=find(sum(f,2));
83 % col=find(sum(f,1));
84 % row=row(1):row(end);
85 % col=col(1):col(end);
86 % a=a(row,col)
87 if printImage
88     figure(1);
89     subplot(2,2,1);imshow(ImgBak);title('origin');
90     subplot(2,2,2);imshow(f);title('segment');
91     subplot(2,2,3); imshow(uint8(Img));title('filtered');
92 end
93 X=f;
94 end

```

2.4 featureExtra.m

2.4.1 Function

Compute the feature vector of an image. Using 9 color features (3 for each color planes R G B) and 4 texture features.

2.4.2 Input parameters

`inputImage`

The RGB image matrix after median filtered and segmented.

2.4.3 Output

`outputX`

Returns a 13×1 feature vector.

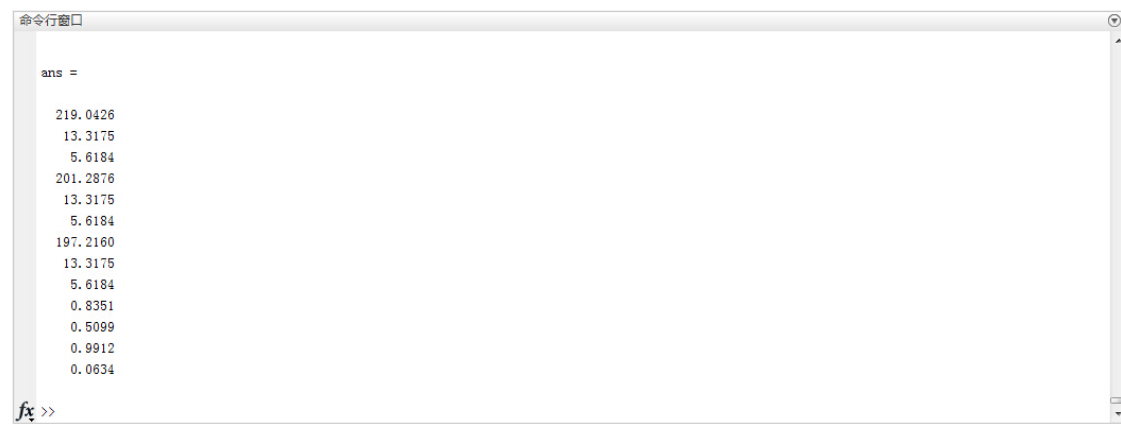


Figure 2.4: Features

2.4.4 Code of featureExtra.m

```

1 function [outputX] = featureExtra( inputImage )
2 % featureExtra Extra the features of an image
3 %   return a 13*1 vector of the feature
4
5 % testPic=imread(imgFileName);
6 testPic=inputImage;
7 pic2=rgb2gray(testPic);
8 pic2=im2double(pic2);
9 for i=1:3
10     testPic(:,:,i)=medfilt2(testPic(:,:,i), [15 15]);
11 end
12 [M,N,Z2]=size(testPic);
13 meanC=zeros(3,1);
14 covC=zeros(3,1);
15 thirdCovC=zeros(3,1);
16 for z=1:3
17     %         meanC(z,1)=sum(sum(testPic(:,:,z)));
18     %         meanC(z,1)=meanC(z,1)/M/N;
19     meanC(z,1)=mean(mean(testPic(:,:,z)));
20     %         disp(meanC(z,1));
21     covC(z,1)=sum(sum((testPic(:,:,z)-meanC(z,1)).^2));
22     covC(z,1)=nthroot((covC(z,1)/M/N),2);
23     %         disp(covC(z,1));
24     thirdCovC(z,1)=sum(sum((testPic(:,:,z)-meanC(z,1)).^3));
25     thirdCovC(z,1)=nthroot((thirdCovC(z,1)/M/N),3);
26     %         disp(thirdCovC(z,1));
27 end
28 pic2=rgb2gray(testPic);
29 [glcms,SI]=graycomatrix(pic2);
30 stats = graycoprops(glcms, 'Contrast Homogeneity Energy');
31 % save temp glcms;
32 outputX=zeros(13,1);
33 count=1;
34 for i=1:3
35     outputX(count,1)=meanC(i,1);
36     count=count+1;
37     outputX(count,1)=covC(i,1);
38     count=count+1;
39     outputX(count,1)=thirdCovC(i,1);
40     count=count+1;
41 end
42 outputX(count,1)=entropy(glcms);
43 outputX(count+1,1)=stats.Energy;
44 outputX(count+2,1)=stats.Homogeneity;
45 outputX(count+3,1)=stats.Contrast;
46 disp(outputX);
47 end

```

Chapter 3

Application

3.1 Image pipeline

3.1.1 Median filter

We first use the median filter to reduce the effect of noise or hear on the feature extraction(see [getSegment.m](#)).

3.1.2 Segmentation with Active contour without edges

Then we use the active contour without edges algorithm to segment the image and set the outter area of normal skin(we assume that the lesion area should be in the middle orof the image) to be blank so that it won't have the effect on the feature extraction.

And the result is like (or see [getSegment.m](#)):

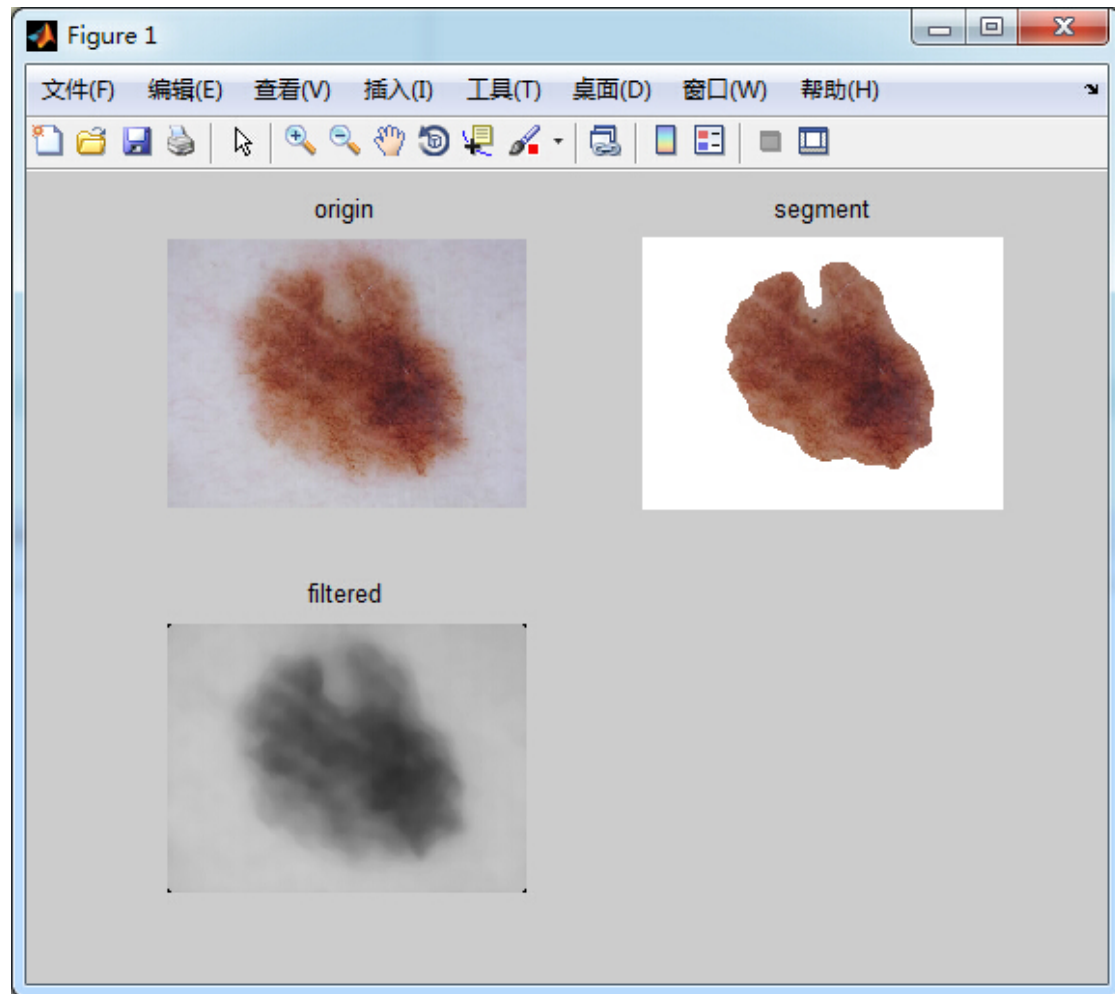


Figure 3.1: `getSegment.m` output picture

3.1.3 Feature extracion

We use the three color moments eq. (3.1) eq. (3.2) eq. (3.3) for each color plane(R G B),and form the first 9 feature:

$$\mu_c = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N p_{ij}^c \quad (3.1)$$

$$\sigma_c = \left[\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (p_{ij}^c - \mu_c)^2 \right]^{1/2} \quad (3.2)$$

$$\theta_c = \left[\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (p_{ij}^c - \mu_c)^3 \right]^{1/3} \quad (3.3)$$

Then we use another 4 features eq. (3.4) eq. (3.5) eq. (3.6) eq. (3.7) associated with gray level co-occurrence matrix $C(i,j)$:

$$Entropy = \sum_i \sum_j C(i,j) \log(C(i,j)) \quad (3.4)$$

$$Energy = \sum_i \sum_j C^2(i,j) \quad (3.5)$$

$$Contrast = \sum_i \sum_j (i-j)^2 C(i,j) \quad (3.6)$$

$$Homogeneity = \sum_i \sum_j \frac{C(i,j)}{1 + |i-j|} \quad (3.7)$$

3.2 Dataset

We divide the total dataset into two classes : the Melanoma class and the none Melanoma class.

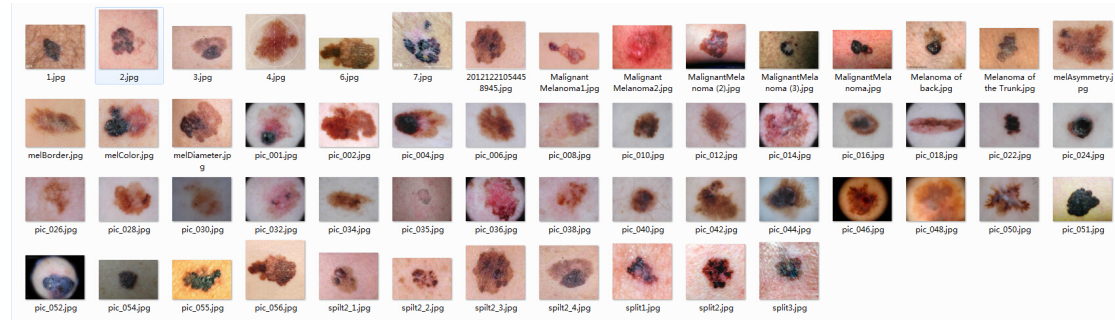


Figure 3.2: Images of Melanoma class

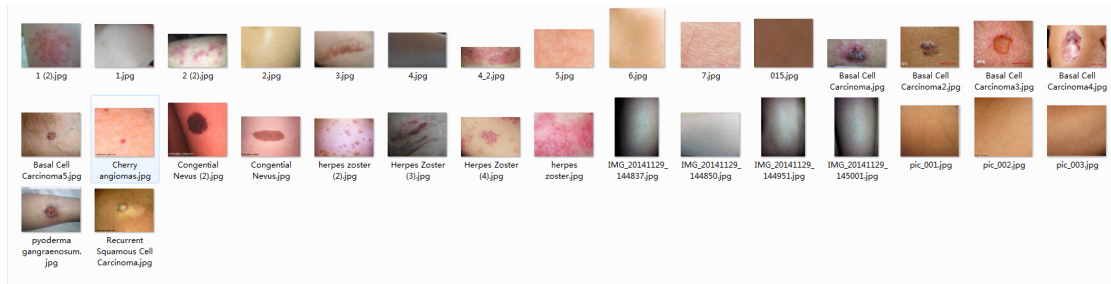


Figure 3.3: Images of none Melanoma class

There are 56 images in the `/dataset/cancer/` directory and 32 in `/dataset/normal/` directory.

3.3 Neural Network

3.3.1 Network definition

We use a Neural Network with 13 input , one hidden layer of 50 nodes and an output layer with 2 nodes.

3.3.2 Network input and output

We use a 13×1 vector to represent an image's feature, and a 2×1 vector to represent which class this image belongs to. The image feature is the input and the class label is the output.

Input definition

The input feature vector is like below:

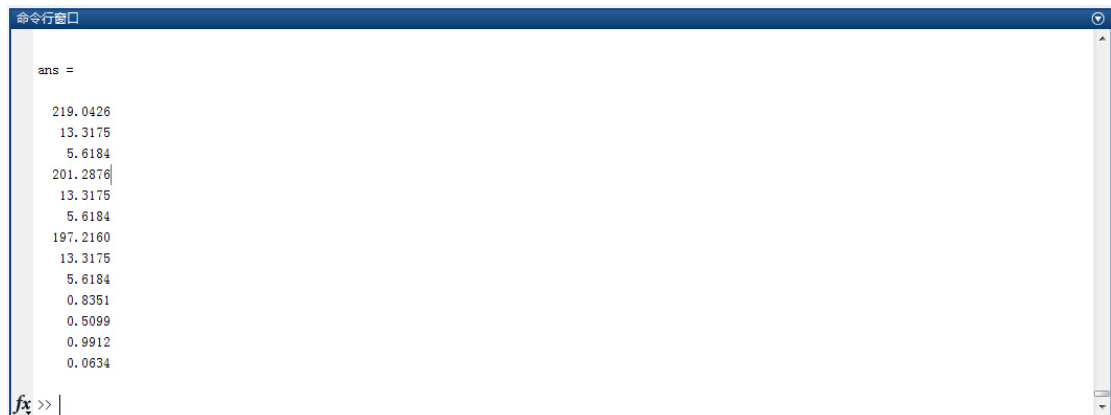


Figure 3.4: Network input vector

Output definition

For example, when training we will have the 2*1 vector(value of min 0.1 and max 0.9) like this:

$$\begin{bmatrix} 0.9 \\ 0.1 \end{bmatrix} \quad (3.8)$$

Vector 3.8 means that this image belongs to the first class(the Melanoma class).

$$\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix} \quad (3.9)$$

And this above vector 3.9 means that this image belongs to the second class(the none Melanoma class).

3.4 Testing

3.4.1 Changing dataset

After adding new images to the dataset directory, you should set in main.m:

```
numberOfCancer=new number of cancer images;
numberOfOther=new number of none cancer images;
reComputeFeature=true;
reTrainNetwork=true;
```

Then run the main.m, it may take a long time to recompute all the features.

3.4.2 Using default network







 nelNetwork_83_train_30_15.mat	2014/11/30 星期...	MAT 文件	16 KB
 nelNetwork_85_train_35_25.mat	2014/11/30 星期...	MAT 文件	16 KB
 nelNetwork_87_train_40_15.mat	2014/11/30 星期...	MAT 文件	16 KB
 nelNetwork_90_train_35_20.mat	2014/11/30 星期...	MAT 文件	16 KB
 nelNetwork_92_train_40_20.mat	2014/11/30 星期...	MAT 文件	16 KB
 nelNetwork_95_train_40_25.mat	2014/11/30 星期...	MAT 文件	16 KB

Figure 3.5: Networks

We have setup some networks like the above picture, user can use them to do the test.

For example the file: 'nelNetwork_95_train_40_25.mat', means that this network's test correct rate is 95%, and it is trained using the first 40 of 56 cancer images, and also the first 25 of the none cancer images.

If you want to use this network, just change this file's name to 'nelNetwork.mat', and change the parameters in file: trainMyNetwork.m, the trainNumberOfCancer and trainNumberOfOther parameters to those of the network file name.

Like the file nelNetwork_95_train_40_25.mat, you should set

```
trainNumberOfCancer=40
trainNumberOfOther=25
```

Then inside the main.m, set :

```
reComputeFeature=false
reTrainNetwork=false
```

Finally, you can run the main.m and get the correct rate.

3.4.3 Train your network

Change the parameter `trainNumberOfCancer` and `trainNumberOfOther` to what you are going to do experiment on (no larger than the total number of cancer image and none cancer image) inside the `trainMyNetwork.m`. And set:

`reComputeFeature=false`

`reTrainNetwork=true`

in the `main.m`.

3.4.4 Results

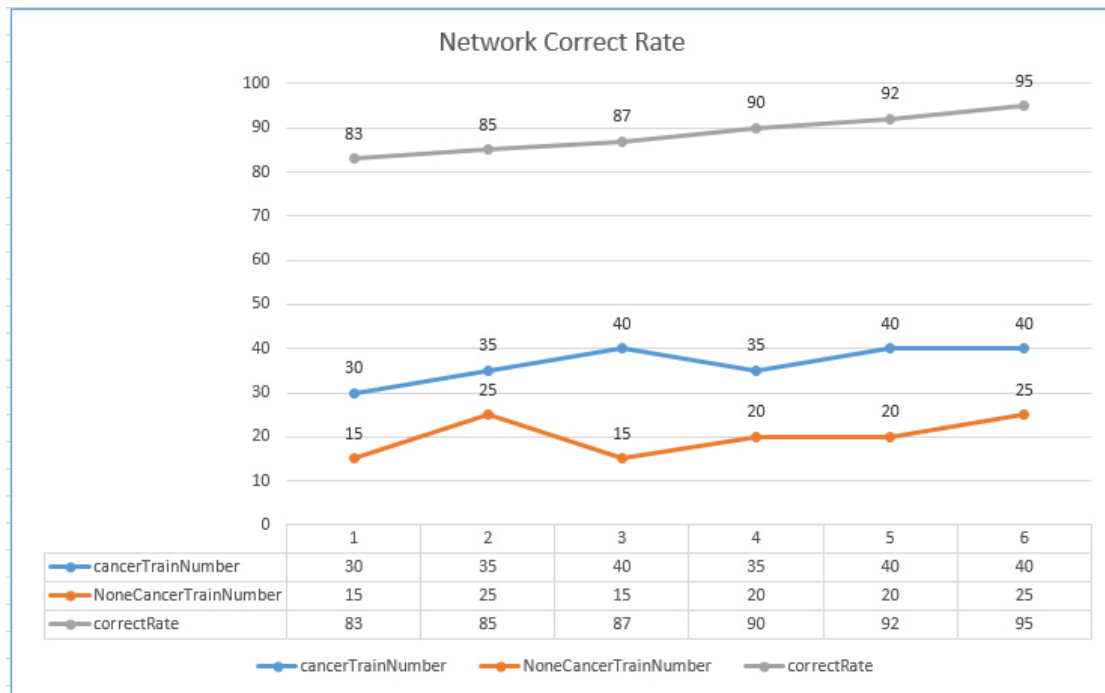


Figure 3.6: Network Correct ResultTable

From the above table, we can know that the more training data we use , the more accurate we get. And the correct rates are all more than 80%.

Chapter 4

Future work

4.1 What we can improve:

4.1.1 Data

In this case we have totally 80 images to test and train on, and it is a small dataset since it is hard to find many Melanoma images online. So we can get much improvement if we can get more images to train and to test.

4.1.2 Speed

It now takes more than 300 seconds to computed all the images' features, and it is quite slow. So maybe we can accelerate by making a better tradeoff between the iteration number of segment algorithm and the correct rate, and make it faster.