**Face recognition project report**

1、 Project background and principle

This experiment includes two parts. The first part is face detection, that is, through a series of means, the computer can recognize the face area on a picture. The second part is face recognition, which can be divided into early PCA dimension reduction algorithm and recent deep learning algorithm based on feedforward neural network and convolution neural network.

**The first part is: face detection based on skin color model.**

Digital pictures in daily life are actually made up of matrices, and the elements of the matrix represent the brightness values of corresponding pixels. The gray image is composed of a matrix, and the color image is composed of three matrices representing red, green and blue components respectively. Face detection is to detect the location of the face image. The input of the face detection algorithm is an image, and the output is a sequence of face frame coordinates. The specific result is 0 or 1 or more face frames. The output face coordinate frame can be square, rectangle, etc. The principle of face detection algorithm is simply a process of "scanning" and "judging". Firstly, the whole image is scanned, and then whether the candidate region is a face is determined one by one.

The face detection based on skin color model can be roughly divided into the following steps: (1) get the average skin color of a person's face. (2) Based on the average skin color, the whole image is retrieved to recognize the face region.

In computer implementation, the average value of these matrices is calculated by manually intercepting a few pictures (i.e. matrix) containing only faces. Then a certain threshold range is set artificially to retrieve the required image. If the image contains the region within the threshold range and meets a series of other conditions, it is identified as the face region. Finally, the obtained face region is divided and the face area is framed by rectangle or circle frame, and the work of face detection is completed.

**Secondly, face recognition based on PCA dimension reduction algorithm**

Face recognition is a kind of biometric recognition technology based on human face feature information. A series of related technologies, usually called portrait recognition and face recognition, are used to collect images or video streams containing faces with cameras or cameras, and automatically detect and track faces in the images, and then recognize the detected faces.

Since the earliest paper published in 1964, face recognition technology has experienced a series of technological revolution. The earliest technology in 1964-1990 was based on feature, mainly including face recognition based on silhouette analysis. Then, from 1991 to 1997, the problem of face recognition on small and medium-sized face databases was gradually solved under the ideal conditions, that is, the detection cooperated with the detection. The linear subspace analysis and statistical pattern recognition methods of 2D face image based on appearance are the mainstream. Representative technology, eigenface, feature face, etc. This experiment is based on the eigenface face face recognition. The main idea is as follows: (1) get the mean value of the image, use PCA algorithm to de average the image, and then calculate the sample covariance matrix. Take the first n large eigenvalues corresponding to the eigenvector to form the transformation matrix W, and use the transformation matrix w to reduce the data of the image after averaging, so as to extract the image features, namely "feature face". (2) The extracted image features are used to retrieve the input face in turn, and find out the two images with the minimum distance between the corresponding feature vectors, then these two images belong to the same face.

In the specific implementation, the main task is to find the average face of the training set image first, and then get a set of feature vectors through the average face. This set of eigenvectors is used to calculate the Euclidean distance from the eigenvector of the test set image. The one with the smallest distance is regarded as the same face.

**The third part: face recognition based on fully connected feedforward neural network**

Neural network is a model in machine learning. It is a mathematical model that imitates the behavior characteristics of human visual system and processes information. This kind of network depends on the complexity of the system, by adjusting the relationship between a large number of internal nodes, as well as the weight and bias of each node, so as to achieve the purpose of processing information.

Feedforward neural network is the most common type of neural network in practical application. The first layer is input, and the last layer is output. If there are multiple hidden layers, it is called "deep" neural network, and the "depth" of deep learning lies in the fact that the neural network has many hidden layers. By calculating a series of transformations to change the similarity of samples. The activity of neurons in each layer is a nonlinear function of the activity of the previous layer. In the output layer, the soft maxloss function is used to evaluate the classification results of the neural network, and the weights and bias of the network are adjusted accordingly according to the evaluation results.

In this experiment, a classification task is realized by fully connected feedforward neural network. Face recognition. The specific implementation is to build a fully connected neural network. After the weights of the neural network are trained with the training set pictures and the verification set is tested, the images of the test set are input to test the training results, and the picture with the largest output probability is regarded as the same face. Finally, the correct rate was counted.

**The last part is the fourth part: face recognition based on convolutional neural network**

Due to a series of defects of fully connected neural network, such as the number of connections is too much, resulting in too many parameters to be trained, easy to produce over fitting, and limit the increase of the number of layers, affecting the fitting effect. A convolution neural network is proposed. Convolution neural network and fully connected neural network have many similarities. The difference between convolutional neural network and fully connected neural network is that convolution neural network is weight sharing and locally connected.

In image processing, it is usually two-dimensional convolution. Its general process is to check with a convolution. An image (i.e. matrix) is sampled in a certain step. Each sampling will get a value, which contains part of the information of the original image in the sampling area. After sampling the whole image, the sampling results form a new matrix called feature map feature map. Each convolution kernel maps a feature map. In the first layer of the local network, each of the neurons in the first layer of the local network contains the information of a new part of the local network. The parameters of convolution kernel are shared for the whole image. In this way, the training parameters are greatly reduced and the possibility of expanding the depth of neural network is increased. However, although the convolution layer can significantly reduce the number of connections, the number of neurons per feature map is not significantly reduced. In this way, if a classifier is followed, the input dimension of the classifier is still very high, and it is easy to over fit. In order to solve this problem, in convolutional neural network, a pooling operation, namely subsampling, is added after the convolution layer to form a subsampling layer. Sub sampling layer can greatly reduce the dimension of features and avoid over fitting. The last layer is still fully connected and output to the loss function softmaxloss.

It can be realized only by modifying the specific parameters of the neural network. For example, given the size, number, dimension of convolution kernel and the step size of pooling layer. The rest of the operation is the same as full connection.

2、 Completion and code analysis

(1) Face detection based on skin color model

Completion status:

By calculating the average value of skin color, the average pixel value of skin color is obtained, and then the face region is searched in the neighborhood of the average value, and the face area is binarized, and the corrosion expansion is carried out to find the connected region that conforms to the length width ratio of the face as the face, so as to realize the face detection based on the skin color model. But it can't be applied to areas with two or more faces.

Code analysis:

%-------------------test function-----------------------------------%

path\_test = 'D:\WeChatFiles\zzz15911721117\FileStorage\File\2019-07\D1\dht.jpg';

ptest = imread(path\_test);%原图

temtest = ptest;%识别后1

temtest2 = ptest;%识别后2

t = rgb2ycbcr(ptest);

tem = t;

tcb = tem(:,:,2);

tcr = tem(:,:,3);

[l,r] = size(tcb);

[yu\_cb,yu\_cr]=yuzhi(ptest);

for i = 1:l

for j = 1:r

if tcb(i,j)<(yu\_cb-20) || tcb(i,j)>(yu\_cb+20)

temtest(i,j,:) = 0;

temtest2(i,j,:) = 0;

else

temtest(i,j,:) = 255;

end

if tcr(i,j)>(yu\_cr-20) && tcr(i,j)<(yu\_cr+20)

temtest(i,j,:) = 0;

temtest2(i,j,:) = 0;

else

temtest(i,j,:) = 255;

end

end

end

%

temtest;%二值化后

se=strel('square',3);%3×3方阵内粗处理

im\_Region=imopen(temtest,se);%开运算（先膨胀后腐蚀）

im\_Region=imclose(im\_Region,se);%闭运算（先腐蚀后膨胀）

im\_Region=imfill(im\_Region,'holes');%填洞

se1=strel('square',8);%8×8方阵内精处理

im\_Region=imerode(im\_Region,se1);%腐蚀

im\_Region=imdilate(im\_Region,se1);%膨胀

imgr = rgb2gray(im\_Region);

[im\_Face,num]=bwlabel(imgr,8);%获取所有4连通区域

[r c]=find(im\_Face~=0);

r\_min=min(r);

r\_max=max(r);

c\_min=min(c);

c\_max=max(c);

wid = c\_max-c\_min;%宽

len = r\_max-r\_min;%长

len = wid \* (1/0.8);

%显示框选人脸后的图像

figure(2)

subplot(2,2,4);

imshow(ptest);%左上角图片

title('基于肤色的人脸检测');

rectangle('Position',[c\_min r\_min wid len],'EdgeColor','r');

subplot(2,2,1);

imshow(ptest);%右下角图片

subplot(2,2,2);

imshow(temtest);%右上角图片

subplot(2,2,3);

imshow(im\_Face);%左下角图片

%--------------------yuzhi函数--------------------------------------%

function [outcb,outcr] = yuzhi(img)

y = rgb2ycbcr(img);

y1 = y(:,:,2);

y2 = y(:,:,3);

outcb = mean(y1(:));

outcr = mean(y2(:));

End

(1) Face recognition using Eigenface feature combined with PCA dimension reduction algorithm

Completion:

After a long period of inspection, it seems that the problem could not be solved at all.At first, the correct rate was less than 10%, and then an error was detected: there was a problem with the algorithm when converting the smallest distance vector sequence number to the face sequence number. After correcting, the correct rate was 22%, but obviously there was still a problem.During debugging, the sequence number of the feature vector with the smallest distance will obviously change as the loop progresses, but it will not change in the featureface function, causing the accuracy rate to never increase.Best of all, I can't really check it out.

Code parsing:

%----------------------featureface function-------------------------%

file\_path = 'E:\matlab\bin\D2\Yale2/';% 图像文件夹路径

img\_path\_list = dir(strcat(file\_path,'\*.bmp'));%获取该文件夹中所有bmp格式的图像

img = zeros(10000,90);

tem=zeros(10000,1);

for i = 1:15%读样本人数

for j = 1:6 %逐一读取图像（某个人的样本数量）

image\_name = ['subject',num2str(i,'%02d'),'\_',num2str(j),'.bmp'];% 图像名

tem\_img = imread(strcat(file\_path,image\_name));

tem\_img = tem\_img';

n=1

for o=1:100

for p=1:1000

tem(n,1)=tem\_img(o,p);

n=n+1;

end

end

img(:,i) = tem;

end

end

[A,avt] = fuse(img);

[x,y] = ev(A);%y是特征向量，important

feature = y'\*A;

%特征脸可视化

for o = 1:6 %6个特征向量

img1 = y(:,o);

img1 = reshape(uint8(img1),100,100);

subplot(2,3,o)

imshow(img1);%显示特征脸

end

counter = 0;

for m = 1:15%读样本人数

for k = 7:11 %逐一读取图像（某个人的样本数量）

test\_name = ['subject',num2str(m,'%02d'),'\_',num2str(k),'.bmp'];% 图像名

test\_tem\_img = imread(strcat(file\_path,test\_name));

test\_img = zeros(10000,1);

n = 1;

for i = 1:100

for j = 1:100

test\_img(n,1) = test\_tem\_img(i,,j);

test\_img(n,1) = test\_img(n,1) - avt(n,1);%测试集减平均脸

n = n+1;

end

end

test\_feature = y' \* test\_img;

res = distance(test\_feature,feature);

if res == m

counter = counter + 1;

end

end

end

counter

%--------------------------------------------------------fuse函数------------------------------------------------------%

% The fuse function inputs a matrix of 10000\*90 and returns a de-averaged matrix A and an average matrix AVT

function [A,avt] = fuse(bigimg)

AVG=zeros(100,100);

av = mean(bigimg');

n = 1;

for i=1:100%AVG是平均值矩阵

for j=1:100

AVG(i,j)=av(n);

n=n+1;

end

end

avt=av';

A=zeros(10000,90);

for i=1:10000

for j=1:90

A(i,j)=bigimg(i,j)-av(i);% A is the input matrix minus its own average

end

end

end

%-----------------------------------------------------ev function--------------------------------------------------------%

% Input de-averaging matrix, output diagonal matrix Ni and eigenvector outvt

function [ni,outvt] = ev(avimg)

%n = 6;

temvt = avimg' \* avimg;

[y,x] = eig(temvt);%特征向量，特征值

temx = x;

ni = zeros(1,n);%放特征值最大的几个

for i = 1:n

[ni(i),tem] = find(temx==max(max(temx)));%找出最大值的位置

temx(find(temx==max(max(temx)))) = -1;

end

outvt = [avimg\*y(:,ni(1)) avimg\*y(:,ni(2)) avimg\*y(:,ni(3)) avimg\*y(:,ni(4)) avimg\*y(:,ni(5)) avimg\*y(:,ni(6))];

end

%--------------------------------------------------distance函数-------------------------------------------------------%

% Enter a vector a of length 6 and a matrix B of 6\*90 containing the eigenvectors of the training set, returning the ordinal R

function [r] = distance(a,b)

d=+inf;

flag=0;

for i=1:90

tmp = b(:,i);

td=norm(a-tmp);

if td < d

d = td;

flag=i;

end

end

if rem(flag,6)==0

r=flag/6;

else

r=floor((flag-1)/6)+1;

end

（3）、Face Recognition Based on Fully Connected Feedforward Neural Network

Completion:

The main work is to preprocess the data and adjust the parameters.Pre-processing, it is not very easy to use the structure of MATLAB at first, ask the teacher to know that the original definition of the structure of MATLAB and the form of invocation is almost identical.Parameter adjustment basically uses the simplest layer of full connection, no hidden layer.The experimental results show that the accuracy of Yale2 dataset and ORL dataset reaches 98.3% and 99.4% respectively after the image is reduced in size.To summarize, I did not intend to change the size of the picture before, but I have been working on the number of layers, adjusting parameters, the accuracy can no longer be improved up to about 95%, until I later changed the size of the picture, the accuracy can only be improved, it should be because the reduction of the picture enlarges the field of perception, and the accuracy increases.

Code parsing:

%-----------------------------------------------------train function-----------------------------------------------------%

run(fullfile('D:\WeChat Files\zzz15911721117\FileStorage\File\2019-07\matconvnet-1.0-beta25\matconvnet-1.0-beta25\matlab\vl\_setupnn.m')) ;

opts = init\_parameters( );%初始化

imdb = load\_data(opts);%读数据

net = cnn\_network\_init(opts.nclass);

net.meta.normalization.averageImage = imdb.images.data\_mean ;%存图像均值

trainfn = @cnn\_train ;

[net, info] = trainfn(net, imdb, getBatch(), ...

'expDir', opts.expDir, ...

'val', find(imdb.images.set == 2),'numEpochs',150) ;

%-----------------------------init\_parameters fuction----------------------------%

function [opts] = init\_parameters()

opts.nclass =15 ;%人脸类别数

opts.expDir = 'E:\matlab\bin\D3\model\' ;%保存训练的网络模型

opts.basePath = 'E:\matlab\bin\D3\' ;

opts.modelDir = opts.expDir ;

opts.modelName = 'net-epoch-150.mat' ;

end

%--------------------------------------------------load\_data function-------------------------------------------------%

%读数据到imdb

function imdb=load\_data(opts)

sum=single(zeros(28,28));

sett=1;

for i=1:15

for j=1:10

image\_name1=['subject',num2str(i,'%02d'),'\_',num2str(j,'%d'),'.bmp'];

img\_tem = imread(strcat('E:\matlab\bin\mycnn\Yale2\',image\_name1)) ;

img\_tem = imresize(img\_tem,[28 28]) ;

imdb.images.data(:,:,1,sett) = img\_tem;

if j<7

imdb.images.label(1,sett)=i;%属于哪张人脸

imdb.images.set(sett)=1;

end

if j==7||j==8

imdb.images.label(1,sett)=i;

imdb.images.set(sett)=2;

end

if j==9||j==10

imdb.images.label(1,sett)=i;

imdb.images.set(sett)=3;

end

sett=sett+1;%训练集标识

end

end

imdb.images.data=single(imdb.images.data);%

for i=1:90

sum(:,:)=sum(:,:)+imdb.images.data(:,:,1,i);%求和

end

imdb.images.data\_mean=single(sum./90);%取平均

for i=1:150

imdb.images.data(:,:,1,i)=imdb.images.data(:,:,1,i)-imdb.images.data\_mean;%减平均

end

save('imdb','imdb');

%----------------------------------------cnn\_network\_init function-------------------------------------------------%

%设置网络结构

function net=cnn\_network\_init(nclass)

rng('default');

rng(0);

f=1/100;

net.layers={};

net.layers{end+1}=struct('type','conv','weights',{{f\*randn(28,28,1,nclass,'single'),zeros(1,nclass,'single')}});

net.layers{end+1}=struct('type','softmaxloss');

net.meta.inputSize=[28,28,1];

net.meta.trainOpts.learningRate=0.01;

net.meta.trainOpts.numEpochs=150;

net.meta.trainOpts.batchSize=64;

net=vl\_simplenn\_tidy(net);

%----------------------------------------------------get\_single\_data function---------------------------------------%

function [single\_im,imdb] = get\_single\_data(opts)

test\_path = 'E:\matlab\bin\D3\Yale2\' ;

imdb = load(fullfile(opts.basePath,'imdb.mat'));

test\_list = dir(strcat(test\_path,'\*.bmp')) ;

single\_im = [ ];

tem = 1;

for i = 1:15

for j=11

image\_name2 = ['subject',num2str(i,'%02d'),'\_',num2str(j),'.bmp'] ;% 取训练集图片

img\_tem = imread(strcat(test\_path,image\_name2)) ;

img\_tem = imresize(img\_tem,[28 28]) ;

single\_im(:,:,1,tem) = img\_tem;

single\_im(1,tem) = i ;

tem = tem + 1;

end

end

for i = 1:15

single\_im(:,:,1,i) = single\_im(:,:,1,i) - imdb.imdb.images.data\_mean(:,:);

end

single\_im = single(single\_im) ;

end

(4) Convolution neural network for face recognition

Completion:

The main difference between convolution network and full connection is weight sharing and local connection, which is mainly reflected in the difference of network structure.So the convolution network and the fully connected code are just different from the definition of the network structure.Only some of the functions used are listed here.

This time, the picture size is still not reduced at the beginning. After adjusting the number of network layers several times, the accuracy can only reach 92%.Based on previous experience, another attempt was made to modify the image size, and the accuracy rate was greatly improved. The accuracy rate of ORL dataset and Yale2 dataset was 98.9% and 98.7%, respectively.Another unresolved problem is that in the last layer, when dropout function was expected to be added to alleviate the over-fitting phenomenon, the error function is large and fluctuating no matter what the discard rate is set.Finally, dropout has to be abandoned, but the fitting phenomenon can be observed from the error function curve.

Code parsing:

%--------------------------cnn\_network\_init function----------------%

% Set up network structure

function net=cnn\_network\_init(nclass)

rng('default');

rng(0);

f=1/100;

net.layers={};

net.layers{end+1}=struct('type','conv','weights',{{f\*randn(5,5,1,5,'single'),zeros(1,5,'single')}},'stride',1,'pad',0);

net.layers{end+1}=struct('type','relu');

net.layers{end+1}=struct('type','pool','method','max','pool',[2 2],'stride',2,'pad',0);

net.layers{end+1}=struct('type','conv','weights',{{f\*randn(5,5,1,5,'single'),zeros(1,5,'single')}},'stride',1,'pad',0);

net.layers{end+1}=struct('type','relu');

net.layers{end+1}=struct('type','pool','method','max','pool',[2 2],'stride',2,'pad',0);

net.layers{end+1}=struct('type','conv','weights',{{f\*randn(5,5,1,nclass,'single'),zeros(1,nclass,'single')}},'stride',1,'pad',0);

%net.layers{end+1}=struct('type','dropout','rate',0.5);

net.layers{end+1}=struct('type','softmaxloss');

net.meta.inputSize=[32,31,1];

net.meta.trainOpts.learningRate=0.01;

net.meta.trainOpts.numEpochs=150;

net.meta.trainOpts.batchSize=64;

net=vl\_simplenn\_tidy(net);

3. Summary of Experience

\* In face detection based on skin color models, an attempt has been made to compute and divide the average skin color directly within the RGB color space, but the divisions always show the red component separately, while the green and blue components can overlap normally.Later, I listened to the advice of my assistant teacher and converted the picture into YCbCr color space. There are only two color components to process, and the result is normal.Medical studies have shown that the human eye is more sensitive to the Y component of video, so changes in image quality will not be noticed by the naked eye after subtracting the chroma component to reduce the chroma component.In fact, in the field of digital video picture transmission, YCbCr color space is one of the most commonly used continuous image display processing spaces. In the future, I will also give priority to processing in YCbCr space when processing image.

In addition, in the binary image after processing, the process of determining the face area can try to corrode and expand first, and then determine the face by finding the connected areas that meet the requirements of face length-width ratio, area size, etc. This can eliminate some fine disconnected areas, which is good for detecting the correct results.In addition, in the detection of a single face, you can start traversing from the top of the picture, because the faces are located on the top of the body.

\* Because the fully connected network has enough adjustable parameters, it limits the expansion of layers, but also ensures the accuracy of some simple classification tasks.In the future, when doing some simple classification tasks, you can give priority to the simple and intuitive fully connected neural network, while in the face of some complex classification tasks, you need to design a convolution neural network with multiple hidden layers.

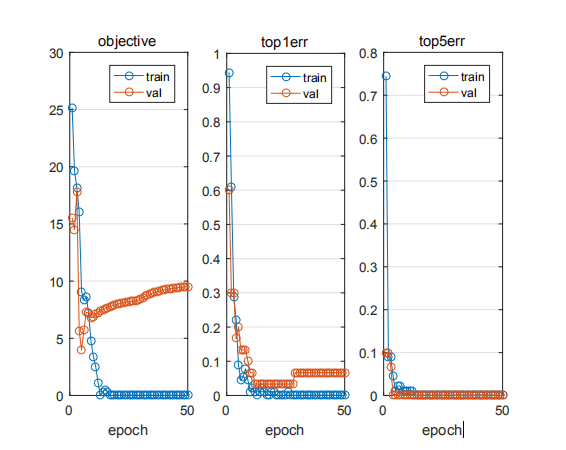
\* In the face recognition process using the neural network, the effect of multi-layer network input is not as good as that of one-layer network input.However, the size of the input picture is neither the smaller the better nor the larger the better.Experiments show that there should be a dimension with the highest accuracy.(Experiments have tried to reduce 100\*100 pictures to 50\*50, 40\*40, 32\*32, 28\*28, 24\*24, 20\*20, respectively, while the highest accuracy group is 28\*28, see below).Personally, whether convoluting or pooling the original image is a sampling process, a means to increase the efficiency of the classifier by filtering out important information and discarding secondary information, and only processing important information.Sampling below Nyquist sampling rate will lose information, which limits the classification performance of convolution neural networks to some extent.Reducing the size of a picture using the imresize function may result in less loss of important information than convolution, pooling, and expanding the field of perception, thereby optimizing the performance of the classifier.However, since there should be an optimal picture input size for different classification tasks, the disadvantage of this is that users must make manual attempts to find an optimal input size based on different classification tasks.This greatly reduces the usability and reliability of the classifier in dealing with different classification tasks.In the future, when designing the classifier, the structure of the neural network and whether the input information is processed manually can be determined according to the actual needs.

Fig. 1. Input picture size 32\*32, one layer full connection, accuracy 90.6%

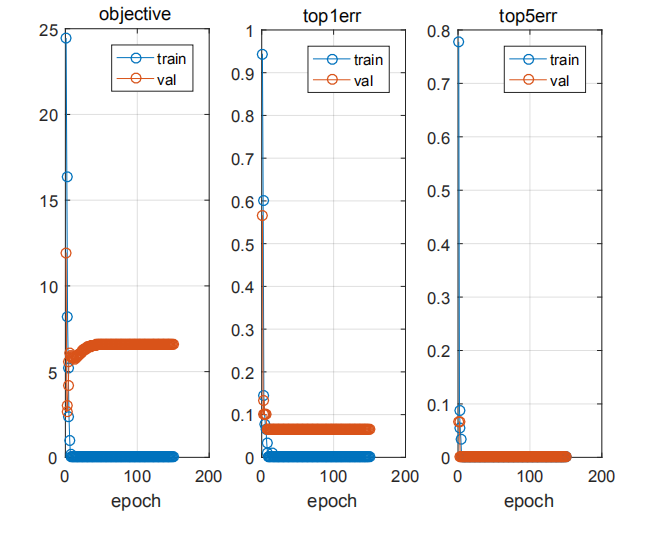


Fig. 2. Input size 28\*28, one layer full connection, accuracy 98.3%

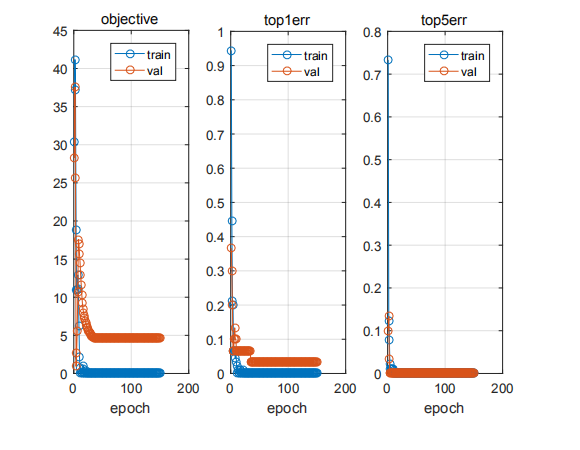


Fig. 3. Input size 24\*24, one layer full connection, accuracy 94.0%

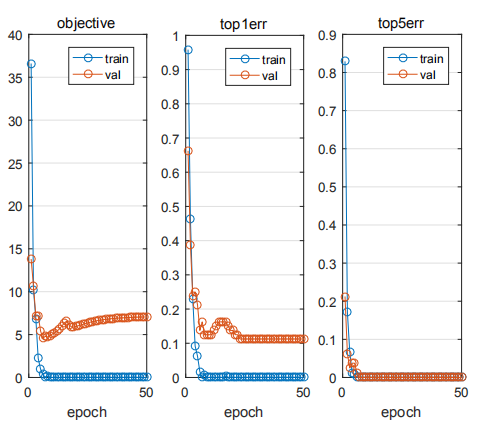
\* For pictures with an aspect ratio other than 1:1, scaling at equal scales is not as effective as scaling directly at 1:1. The exact reason is not clear.The results are as follows:

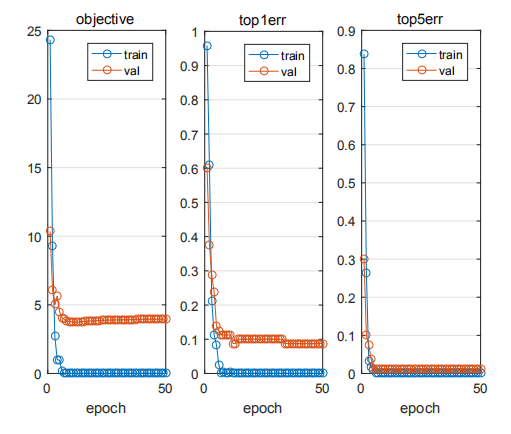
Figure a. For ORL datasets, the scale is reduced to 46\*56, with an accuracy of 87.4%.

Figure B. For ORL datasets, 1:1 is reduced to 32\*32 with 96% accuracy

\* Last lesson, when you encounter some functionality that you need to implement later in your programming, you can first query whether there are ready-made functions to call.Don't just write your own code with a dull head. This is not only inefficient, but also increases the likelihood of making a mistake.

4. Summary and Feeling

(1) What has been done

First day: Grouping cooperation uses YCbCr space. I proposed to retrieve face using face length-width ratio, area size, put forward the idea of retrieving face from the top, and wrote the code of picture reading, preprocessing and so on.Skin color model face detection in RGB space is done privately and independently, and face classification based on skin color mean is achieved. However, the red component will always be displayed separately from the blue-green component instead of overlapping, and it will be final.

Next day: Complete picture reading, preprocessing, averaging face, and code for covariance matrix part

Day 3: Write the structure of the read-in data, but use

"Field=xxx; vaue=xxx; imdb=struct(field, value);"

This complex and tedious form does not work well, and mapping relationships between fields cannot be established. Finally, it uses

"Imdb.images.data=xxxx;Imdb.images.labels=xxxxIn this form.

In addition, referring to the assistant teacher's general framework, I have written several required functions, such as load\_Data, init\_Parameters, cnn\_Network\_Functions such as init.Under the framework set up by the teacher, I have not actually completed any core work.After the experiment, I found some learning materials, combined with the teacher's courseware and online data, to have a deeper understanding of the neural network.The third day of the experiment was completed independently.Also try the picture input size with the highest accuracy and the number of network layers.

Day 4: On the basis of fully connected neural networks, two and three layers of convolution neural networks are built independently.At the end of the experiment, the optimal picture size, other activation functions, dropout function, were tried.

(2) What difficulties were encountered

\* First day problems:

1. At the beginning, we chose to detect the connected areas in the binary image, and divide these connected areas into face areas that match the proportion of face length to width.However, the binary image has many fragmented and disconnected white areas, and the program can not accurately divide the face area.

Solution: Corrode the binary image, expand it, and eliminate the small, disconnected white pixel areas.

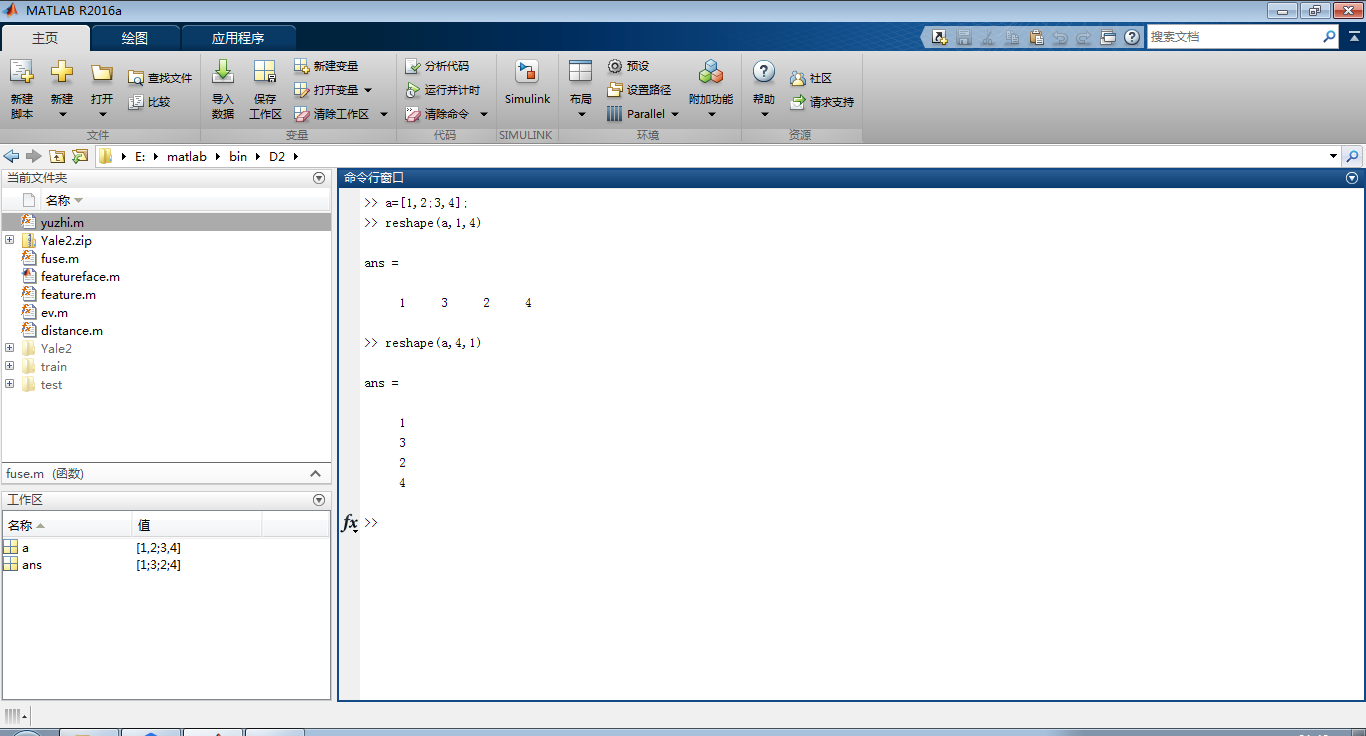
2. However, the results are still inaccurate and do not work well for images with multiple body parts.

Solution: Considering that faces are located on the upper part of the body, we try a new method: first find all the white pixel points with the find function, then start traversing from the upper left corner of the white pixel point area, find the maximum length and width of the white area, and divide the face area with a rectangular box based on this set of lengths and widths.Disadvantages: Can not be applied to more complex images with more than one face.

3. Unfamiliar with batch read and write operations for multiple files.Then the strcat function was used to connect the strings, stitch out the image names, and read the pictures in the folder into a large matrix in batch with a for loop to complete the reading of the pictures.

\* The next day's difficulties:

1. When you average the training set pictures, the results are always strange and you can't get the right results.

Solution: The previous operation was to convert 90 100\*100 training set pictures into 10,000\*90 vectors using the reshape function, and then found the problem with the reshape function, as shown in the following figure:

The expected result is a column vector whose values are in rows for matrix a, but the reshape function is in columns.Finally, you can only write your own code to solve the problem by choosing the value of a large matrix of 100\*100 by row.

1. The test result accuracy is 0.

Solution: Previously only know to reduce the dimension of training set mean, not know to test set mean also to reduce dimension

Dimension, after the test set is reduced by means, the accuracy is no longer 0.

2. Not familiar with the various built-in functions of matlab.Previously, all code was written by oneself, including two loops written by oneself for averaging; for eigenvalues, the eigenvectors were traversed by loops written by oneself; and for Euclidean distances, the loops were also written by oneself.Causes the program BUG many, also wastes a lot of time and energy.

Solution: mean function; eig function for eigenvalue and eigenvector; mean function

Distance uses the norm function, which makes the program more rigorous and saves a lot of time.

3. Accuracy to date is only 22.2%

Solution: An algorithmic problem was found: finding the corresponding face from the serial number of 1-90 pictures

The serial number is not correct, it was calculated with r=i/6 before, and later found that it is incorrect, changed to:

if rem(flag,6)==0

r=flag/6;

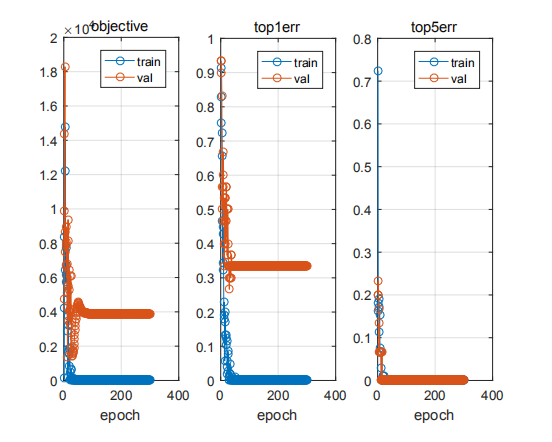
else

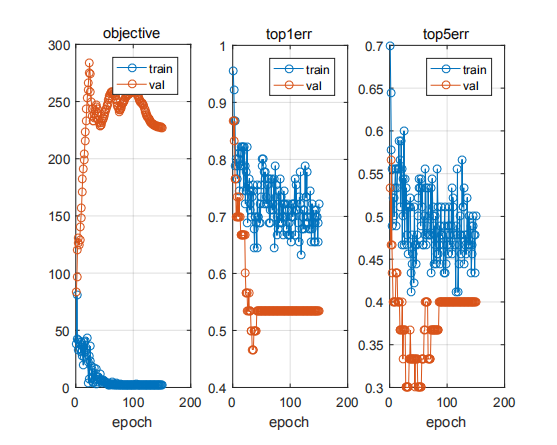
r=floor((flag-1)/6)+1;

end

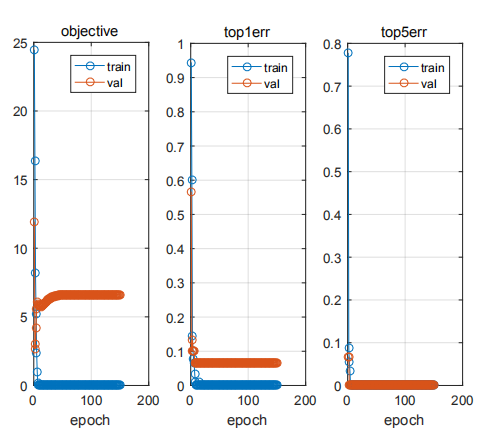
However, the accuracy increased from 13.3% to 22.2%.This accuracy is still not enough to explain the problem, but unfortunately, despite a lot of time and effort, I really can't find out what the error is, and I consulted the assistant teacher to find out what the problem is.Unfortunately, this problem had to be put on hold in the end.

1. \* Problems on day 3:
2. 1. Problems were encountered in defining a structure that contains an index of a picture matrix, a picture mean matrix, a class label, a validation set, and a training set.Only used before
3. "Field=xxx; vaue=xxx; imdb=struct(field, value);"
4. This format defines the structure.Therefore, the mapping relationship between fields cannot be achieved when defining a multidimensional structure in the face and when reading and writing it.
5. Solution: After consulting the assistant teacher, the teacher told me an easy way to define the structure unique to matlab.That is, it can be defined directly without prior declaration.For example, when I want to define a structure named IMDB and want it to have multiple dimensions, I can use it directlyImdb.xxx=xxxForm to define a new field for imdb.
6. When training pictures of training set, pictures cannot be input into the neural network
7. Solution: After consulting your teacher, you know that the neural network must input single data.
8. The data type of the picture matrix is uint8, which solves the problem after a forced type conversion of the picture.
9. When training the basic layer of the neural network at the beginning, the accuracy is very low, as shown in the following figure:



This is obviously a phenomenon of over-fitting, which is expected to be reduced by adding a dropout function, but it seems to be worse, as shown in the following figure:

The error function cannot be converged directly.

Solution: Discard dropout function to improve accuracy by changing picture size.After reducing the image size of both Yale2 and ORL datasets, the perception field was expanded and the results improved, as shown below:

\* Yale2 dataset, entering 28\*28 pictures, 98.3% accuracy

\* Problems encountered on day 4:

1. A CNN with two activation functions relu is built, but the accuracy is very low, as shown below:

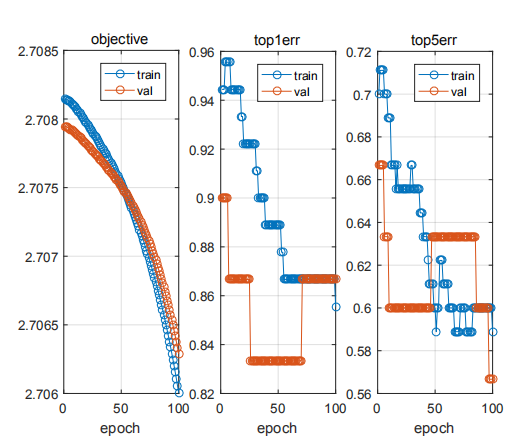


Figure a. Enter a 100\*100 picture with two layers of CNN and the accuracy is only 14%

Terms of settlement:

After removing relu, the accuracy rate reaches 40%, as shown in the following figure:

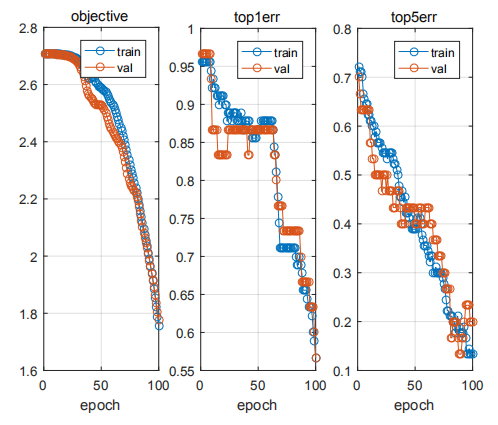


Figure B. Enter a 100\*100 picture with two layers of CNN and no relu, with 40% accuracy

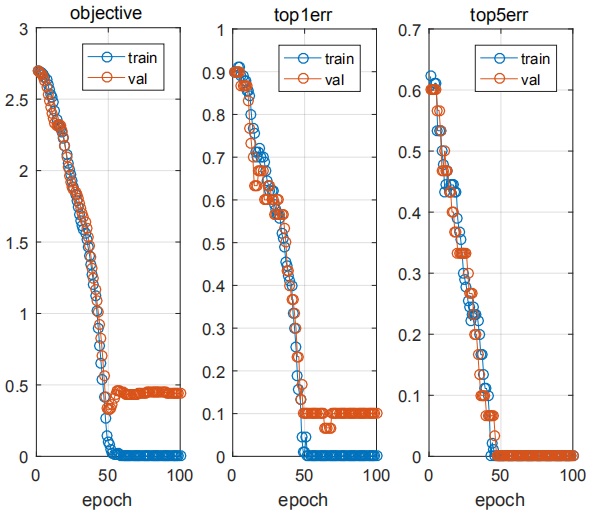
Obviously, this is still a failure. Considering that the error function is always decreasing, I consider increasing the learning rate. The result is as follows:

Figure C. Enter 100\*100 pictures, the learning rate is increased from 0.001 to 0.01, no relu, 90% accuracy

1. After two layers of CNN are built, the accuracy rate is only 90%, and we want to improve it further. We have tried to increase or decrease the number of layers, but the effect is not as good as reducing the size of the picture.

Solution: Draw on the experience of the third day, reduce the picture and re-enter the neural network, try again and again

28\*28 is the most accurate size, as shown below:

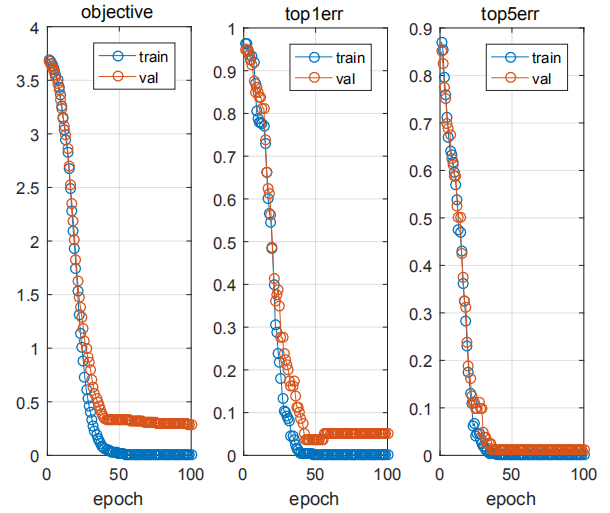


Figure a. Two-layer cnn, input 28\*28, accuracy 98%

1. Insight, harvest and inspiration
2. It was a challenge for me to participate in this project, but I also learned a lot of knowledge, was inspired and broadened my horizons in the process.
3. First, since it is a face recognition project, digital image processing will certainly be involved.Previously, I didn't know much about digital images. Through this project, I learned the essence of digital images: matrices.At the same time, a series of image processing operations involved in the experiment have also made me more familiar with digital images and basic image processing methods.
4. Next, through the teacher's explanations in this project, we have learned the development of face recognition technology, so that I have a certain understanding of the history of face recognition technology.It also inspired me to think about the research road and direction.
5. It is also important that I only heard about neural networks, in-depth learning and other terms before I participated in the project. I also tried to learn relevant knowledge, but the learning curve of most knowledge is too steep for me to have a systematic and comprehensive understanding of neural networks.The knowledge of neural network explained by the teacher in the course gives me a more systematic understanding of the research and development background, theoretical basis, basic principles, and basic building techniques of neural network.The experimental feed-forward neural network and convolution neural network have enhanced my understanding of the theoretical knowledge.Neural network can not only be used in face recognition, but also in a wide range of applications. It can be a tool to find the best solution, an ideal classifier and so on...
6. Secondly, participating in this project has fostered my spirit of cooperation and awareness of cooperation.Before I participated in this project, I did not really have a very positive attitude towards cooperation, and did not really realize the efficiency and necessity of team cooperation.In this project, my cooperation with four members of the group has helped me a lot.In the experiment, everyone will inevitably have something they don't understand, something they don't know, or something they don't know, or something they need to work together to accomplish.At this time, I realized the importance of efficient communication, patient explanation and tacit cooperation between the members.
7. Finally, through the training of MATLAB programming in this experiment, I am familiar with the characteristics and skills of MATLAB programming, and improve my programming ability.
8. The experiment is also inspired and may be useful for future research.
9. For example, the teacher has always emphasized that many times some new ideas, new technologies are actually derived from the sudden emergence of scientific researchers'brains.Take a pat on your head and find that you can solve problems like this.I think it reminds me that most of the work I did in my undergraduate and even before the primary and secondary stages of study was to simply learn the existing knowledge, mature theory, and few inspiring thinking and creative activities.In fact, the study of science and engineering emphasizes strict logical derivation and a solid and profound understanding of knowledge.As a result, I am always used to deduction, understanding and not willing to go out of the thinking mode when I am learning.If I have not always had the habit of thinking diligently, daring not to think, daring to break the original frame of courage, and having no fantastic and divergent thinking, in fact, it is not necessarily a good thing in the future scientific research work. I must remind myself from now on to think more, cultivate innovative and divergent thinking habits.
10. It is also very important that I think analogical thinking can be applied more on the road of scientific research.Take the example of a neural network.The invention of the neural network is actually inspired by the human visual system. By analogizing the working principle of the human visual system, the concept of deep learning is proposed.I was impressed by this example.I realized the importance of analogical thinking.There are many simple rules in life, familiar phenomena and well-known principles, but if you have analogous thinking, you will be able to abstract rules from some seemingly unrelated phenomena, thus inspiring your own thinking and even discovering new knowledge.