

Họ và tên: Nguyễn Trọng Trí– MSHV: CH1602012

BÀI TẬP THỰC HÀNH 1

THIẾT LẬP MÔI TRƯỜNG

Yêu cầu:

1. Tạo tài khoản trên github.
<https://github.com/trint11/vra-k2.git>
2. Tạo tập tin bài tập thực hành VRA.HoVaTen.doc - ghi các thông tin Họ và Tên, MSSV, và đường dẫn tài khoản github.
3. Gỡ lại bài tập thực hành (matlab) đã được phát trong tài liệu.
4. Sync từ máy cục bộ (desktop, laptop) lên server.

BÀI TẬP THỰC HÀNH 2

CÁC THAO TÁC CƠ BẢN TRÊN MATLAB

Yêu cầu:

1. Trả lời các câu hỏi vào tập tin bài tập thực hành VRA.HoVaTen.doc - sau đó kết xuất ra tập tin VRA.HoVaTen-TH2.pdf.
2. Syn tập tin VRA.HoVaTen-TH2.pdf lên tài khoản github và nộp trong Google Classroom.

Các câu hỏi và phần trả lời.

- Q1 - Hãy cho biết câu lệnh tạo ngẫu nhiên một số trong đoạn [1 200].
`randi([1 200]);`
- Q2 - Hãy cho biết câu lệnh truy cập phần tử dòng 3, cột 5 của ma trận A.
`A(3,5)`
- Q3 - Hãy cho biết lệnh tạo ma trận A kích thước 100x200 và các giá trị đều là zero (0).
`A= zeros(100,200)`
- Q4 - Hãy cho biết lệnh lấy số dòng của ma trận A.
`size (A, 1)`
- Q5 - Hãy cho biết lệnh lấy vector cột 10 của ma trận A.
`A(:,10)`
- Q6 - Hãy cho biết lệnh lấy vector dòng 10 của ma trận A.
`A(10, :)`
- Q7 - Hãy cho biết lệnh chuyển vector dòng thành ma trận 28x28.
`A= randi([-10 10], 1, 784)`
`reshape (A, 28, 28)`

Hạn chót: 02/10/2017

BÀI TẬP THỰC HÀNH 3

KHẢO SÁT BỘ DỮ LIỆU CHỮ SỐ VIẾT TAY




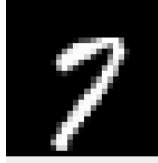
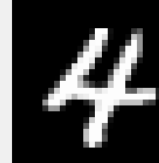
Yêu cầu:

1. Trả lời các câu hỏi vào tập tin bài tập thực hành VRA.HoVaTen.doc - sau đó kết xuất ra tập tin VRA.HoVaTen-TH3.pdf.
2. Syn tập tin VRA.HoVaTen-TH3.pdf lên tài khoản github và nộp trong Google Classroom.

Các câu hỏi:

- Q1 - Hãy viết function hiển thị ảnh có thứ tự là n (n là tham số) cùng label tương ứng trong tập huấn luyện (train) của tập dữ liệu MNIST. Paste code vào bài thực hành và lập bảng cho biết kết quả khi chạy với n=1, 500, 5000, 10000, 59000.

```
1. function Q1_TH3(number)
2.   strData='train-images.idx3-ubyte';
3.   strDataLabel='train-labels.idx1-ubyte';
4.   [allTrainImages,allTrainLabels]=LoadData(strData,strDataLabel);
5.   figure;
6.   img=allTrainImages(:,number);
7.   img2D=reshape(img,28,28);
8.   lblData=num2str(allTrainLabels(number));
9.   lblData=[lblData,'(',num2str(number),')'];
10.  imshow(img2D);
11.  title(lblData);
12.  end
```

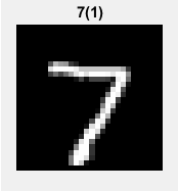

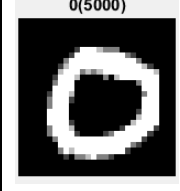
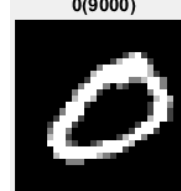
n	1	500	5000	10000	59000
Kết quả	<div>5(1) </div>	<div>8(500) </div>	<div>2(5000) </div>	<div>7(10000) </div>	<div>4(59000) </div>

- Q2 - Hãy viết function hiển thị ảnh có thứ tự là n (n là tham số) cùng label tương ứng trong tập test của tập dữ liệu MNIST. Paste code vào bài thực hành và lập bảng cho biết kết quả khi chạy với n=1, 500, 5000, 9000.

```

1. function Q2_TH3(number)
2. testData='t10k-images.idx3-ubyte';
3. testDataLabel='t10k-labels.idx1-ubyte';
4. [allTestImages,allTestLabels]=LoadData(testData, testDataLabel);
5. figure;
6. img=allTestImages(:, number);
7. img2D=reshape(img,28,28);
8. lblData=num2str(allTestLabels(number));
9. lblData=[lblData, '(', num2str(number), ')'];
10. imshow(img2D);
11. title(lblData);
12. end

```

n	1	500	5000	9000
Kết quả	 <p>7(1)</p>	 <p>6(500)</p>	 <p>0(5000)</p>	 <p>0(9000)</p>

- Q3 - Hãy viết function thống kê số lượng các ảnh tương ứng với các label trong tập huấn luyện (train) của tập dữ liệu MNIST. Paste code vào bài thực hành và lập bảng kết quả khi chạy (nên xuất dưới dạng csv để tiện import thành bảng).

```

1. function Q3_TH3()
2. allTrainLabels = loadMNISTLabels('./train-labels.idx1-ubyte');
3. nTrainLabels = size(allTrainLabels);
4.
5. nResult = zeros([10 1]);
6.
7. for i = 1 : nTrainLabels
8.     k = allTrainLabels(i);
9.     nResult(k+1) = nResult(k+1) + 1;

```

```

10.         end
11.
12.         fprintf('Thong ke tap train:\n');
13.         bangKQ=zeros([10 2]);
14.         for i = 1:10
15.             bangKQ(i,1)=i-1;
16.             bangKQ(i,2)=nResult(i);
17.         end
18.         bangKQ
19.         filename="Q3_TH3.csv";
20.         csvwrite(filename,bangKQ);
21.     end

```

```

Thong ke tap train:

bangKQ =

     0     5923
     1     6742
     2     5958
     3     6131
     4     5842
     5     5421
     6     5918
     7     6265
     8     5851
     9     5949

```

- Q4 - Hãy viết function thống kê số lượng các ảnh tương ứng với các label trong tập test của tập dữ liệu MNIST. Paste code vào bài thực hành và lập bảng kết quả khi chạy (nên xuất dưới dạng csv để tiện import thành bảng).

Code: Tương tự Q3.

```

1. function Q4_TH3()
2. allTestLabels = loadMNISTLabels('./t10k-
   labels.idx1-ubyte');
3. nTestLabels = size(allTestLabels);
4. nResult = zeros([10 1]);
5. for i = 1 : nTestLabels
6.     k = allTestLabels(i);
7.     nResult(k+1) = nResult(k+1) + 1;
8. end
9. fprintf('Thong ke tap test:\n');
10. bangKQ=zeros([10 2]);
11. for i = 1:10
    o bangKQ(i,1)=i-1;
    o bangKQ(i,2)=nResult(i);

```

```

12. end
13. bangKQ
14. filename="Q4_TH3.csv";
15. csvwrite(filename,bangKQ);
16. end

```

```

>> Q4_TH3
Thong ke tap test:

bangKQ =

         0         980
         1        1135
         2        1032
         3        1010
         4         982
         5         892
         6         958
         7        1028
         8         974
         9        1009

```

Q5 - Hãy viết function trả về kết quả nhận dạng của ảnh trong tập test có thứ tự là n (n là tham số, nằm trong đoạn [1, 10000]). Paste code vào bài thực hành và lập bảng kết quả khi chạy với n = 5, 500, 900.

```

1. function Q5(n)
2. imgTrainAll = loadMNISTImages('train-images.idx3-
   ubyte');
3. lblTrainAll = loadMNISTLabels('train-labels.idx1-
   ubyte');
4. Mdl = fitcknn(imgTrainAll', lblTrainAll);
5. imgTestAll = loadMNISTImages('t10k-images.idx3-ubyte');
6. imgTest = imgTestAll(:, n);
7. lblPredictTest = predict(Mdl, imgTest');
8. fprintf('Ket qua nhan dang anh thu %d trong tap test
   la: %d\n', n, lblPredictTest);
9. end

```

```
>> Q5(5)
Ket qua nhan dang anh thu 5 trong tap test la: 4
>> Q5(500)
Ket qua nhan dang anh thu 500 trong tap test la: 6
>> Q5(900)
Ket qua nhan dang anh thu 900 trong tap test la: 8
```

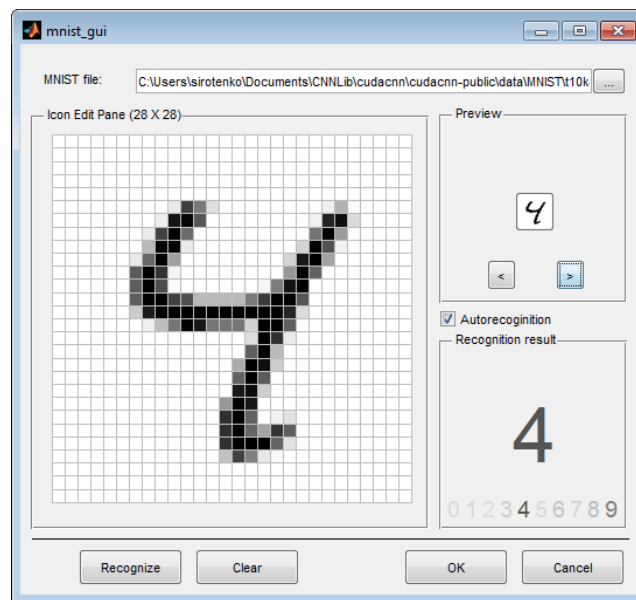
- Q6 - Hãy viết function với tham số đầu vào n là thứ tự của ảnh trong tập test - sau đó hiển thị ảnh tương ứng - rồi hiển thị kết quả nhận dạng - rồi cho biết kết quả nhận dạng là đúng hay sai khi so khớp với label của tập test.

```
1. function Q6(n)
2.     imgTrainAll = loadMNISTImages('train-images.idx3-ubyte');
3.     lblTrainAll = loadMNISTLabels('train-labels.idx1-ubyte');
4.     Mdl = fitcknn(imgTrainAll', lblTrainAll);
5.
6.     imgTestAll = loadMNISTImages('t10k-images.idx3-ubyte');
7.     lblTestAll = loadMNISTLabels('t10k-labels.idx1-ubyte');
8.     imgTest = imgTestAll(:, n);
9.     lblPredictTest = predict(Mdl, imgTest');
10.    lblImageTest = lblTestAll(n);
11.
12.    figure;
13.    img2D = reshape(imgTest, 28, 28);
14.    imshow(img2D);
15.    strLabelImage = 'Anh Test: ';
16.    strLabelImage = [strLabelImage, num2str(lblImageTest),
17.        '.'];
17.    strLabelImage = [strLabelImage, 'Du doan:'];
18.    strLabelImage = [strLabelImage, num2str(lblPredictTest),
19.        '.'];
19.    if(lblPredictTest == lblImageTest)
20.        strLabelImage = [strLabelImage, 'Anh duoc Nhan dang
21.            dung'];
21.    else
22.        strLabelImage = [strLabelImage, 'Anh bi nhan dang
23.            sai.'];
23.    end
24.    title(strLabelImage);
25. end
```

Anh Test:9.Du doan:9.Anh duoc Nhan dang dung



- Q6* - Hãy viết thêm phần giao diện cho bài trên - tham khảo code ở đây:
<https://bitbucket.org/intelligenceagent/cudacnn-public/wiki/Home>



Hình minh họa giao diện nhận dạng chữ số viết tay - trong đó người dùng vẽ lên màn hình và hệ thống hiển thị kết quả nhận dạng.

- Q7 - Hãy viết function đếm số lượng các ảnh có label là n (n là tham số) bị nhận dạng sai theo thuật toán knn. Paste code của function đã chạy được vào bài thực hành và lập bảng kết quả khi chạy với n= 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

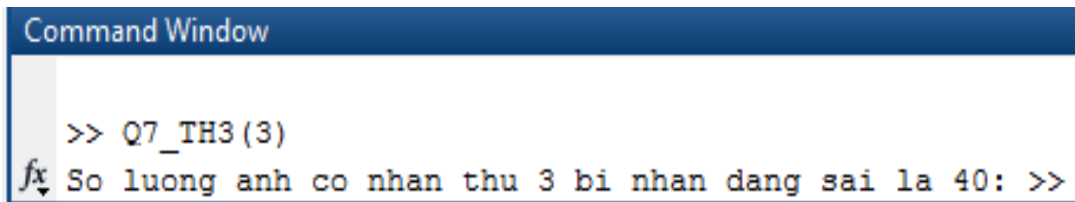
```
1. function Q7_TH3(number)
2. imgTrainAll= loadMNISTImages('./train-images.idx3-ubyte');
3. lblTrainAll= loadMNISTLabels('./train-labels.idx1-ubyte');
4.
5. Mdl=fitcknn(imgTrainAll',lblTrainAll);
6.
7. imgTestAll=loadMNISTImages('./t10k-images.idx3-ubyte');
8. lblTestAll=loadMNISTLabels('./t10k-labels.idx1-ubyte');
9.
10. nLabelTest=size(lblTestAll,1);
```



```

11.     count=0;
12.         for i=1:nLabelTest
13.             if(lblTestAll(i) == number)
14.                 imgTest = imgTestAll(:, i);
15.                 lblPredictTest=predict(Mdl,imgTest');
16.                 if(lblPredictTest ~= lblTestAll(i))
17.                     count = count + 1;
18.                 end
19.             end
20.         end
21.
22.         fprintf('So luong anh co nhan thu %d bi nhan dang sai
    la %d: ',number,count);
23.
24.     end

```



Command Window

```

>> Q7_TH3(3)
fx So luong anh co nhan thu 3 bi nhan dang sai la 40: >>

```

- Q7* - Tương tự bài tập Q7 - nhưng lập bảng confusion matrix.

```

1. function Q7(n)
2.     fprintf('\n Load du lieu train');
3.     imgTrainAll = loadMNISTImages('train-images.idx3-ubyte');
4.     lblTrainAll = loadMNISTLabels('train-labels.idx1-ubyte');
5.
6.     Mdl = fitcknn(imgTrainAll', lblTrainAll);
7.
8.     fprintf('\n Load du lieu test');
9.     imgTestAll= loadMNISTImages('t10k-images.idx3-ubyte');
10.    lblTestAll = loadMNISTLabels('t10k-labels.idx1-ubyte');
11.
12.    numLabel = size(lblTestAll, 1);
13.    result = 0;
14.
15.    for i=1:numLabel
16.        if(lblTestAll(i) == n)
17.            imgTest = imgTestAll(:, i);
18.            if(predict(Mdl, imgTest') ~= lblTestAll(i))
19.                result = result + 1;
20.            end
21.        end

```

```

22.     end
23.     fprintf('\nSo luong anh co label %d bi nhan dang sai la:
    %d\n', n, result);
24. end
25. function Q7(n)
26.     fprintf('\n Load du lieu train');
27.     imgTrainAll = loadMNISTImages('train-images.idx3-ubyte');
28.     lblTrainAll = loadMNISTLabels('train-labels.idx1-ubyte');
29.
30.     Mdl = fitcknn(imgTrainAll', lblTrainAll);
31.
32.     fprintf('\n Load du lieu test');
33.     imgTestAll = loadMNISTImages('t10k-images.idx3-ubyte');
34.     lblTestAll = loadMNISTLabels('t10k-labels.idx1-ubyte');
35.
36.     numLabel = size(lblTestAll, 1);
37.     result = 0;
38.
39.     for i=1:numLabel
40.         if(lblTestAll(i) == n)
41.             imgTest = imgTestAll(:, i);
42.             if(predict(Mdl, imgTest') ~= lblTestAll(i))
43.                 result = result + 1;
44.             end
45.         end
46.     end
47.     fprintf('\nSo luong anh co label %d bi nhan dang sai la:
    %d\n', n, result);
48. end

```

- Q8** (Tương đương 50% đồ án môn học nếu kết hợp với Q7*) - Hãy viết function tính độ chính xác của thuật toán knn với các tham số khác nhau của hàm knn (ví dụ số lượng nearest neighbors, độ đo distance). Paste code vào bài thực hành và lập bảng kết quả.

```

1. function Q8_Advance(k)
2. imgTrainAll = loadMNISTImages('./train-images.idx3-ubyte');
3. lblTrainAll = loadMNISTLabels('./train-labels.idx1-ubyte');

4. Mdl = fitcknn(imgTrainAll' , lblTrainAll, 'NumNeighbors', k);

5. imgTestAll = loadMNISTImages('./t10k-images.idx3-ubyte');

```

```

6. lblTestAll = loadMNISTLabels('./t10k-labels.idx1-ubyte');

7. lblResult = predict(Mdl,imgTestAll');
8. result = (lblResult == lblTestAll);
9. nCount = sum(result);

10.    accuracy = nCount*100/size(imgTestAll, 2);
11.    fprintf('\nDo chinh xac cua thuat toan knn voi %d nearest
    neighbors la %d\n', k, accuracy);
12.    end

```

BÀI TẬP THỰC HÀNH 4

KHẢO SÁT BỘ DỮ LIỆU KHÁC - VÍ DỤ KHUÔN MẶT

Yêu cầu:

1. Trả lời các câu hỏi vào tập tin bài tập thực hành VRA.HoVaTen.doc - sau đó kết xuất ra tập tin VRA.HoVaTen-TH4.pdf.
 2. Syn tập tin VRA.HoVaTen-TH4.pdf lên tài khoản github và nộp trong Google Classroom.
- Q1 - Thay thế bộ dữ liệu chữ số viết tay bằng bộ dữ liệu khuôn mặt tại địa chỉ sau: <http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html>. Tổ chức dữ liệu các tập train và test tương tự bộ dữ liệu chữ số viết tay.
 - Q2* (Tương đương 50% đồ án môn học) - Thực hiện các bài tập tương tự bài tập thực hành 3 trên bộ dữ liệu mới.

Các câu hỏi:

- Q1 - Hãy viết function hiển thị ảnh có thứ tự là n (n là tham số) cùng label tương ứng trong tập huấn luyện (train) của tập dữ liệu MNIST. Paste code vào bài thực hành và lập bảng cho biết kết quả khi chạy với n=1, 5, 50, 100, 190.

```
function Q1(n)

    fprintf('\n Load du lieu train');

    imgTrainAll = loadData('DatabaseofFace/imgTrainImagesAll.mat',
'imgTrainImagesAll');

    lblTrainAll = loadData('DatabaseofFace/lblTrainLabelsAll.mat',
'lblTrainLabelsAll');

    nTrainImages = size(imgTrainAll, 2);

    fprintf('\n So hinh train %d', nTrainImages);






    fprintf('\n Hinh thu %d\n', n);

    figure;
```

```

img = imgTrainAll(:, n);
img2D = reshape(img, 112, 92);
strLabelImage = num2str(lblTrainAll(n));
strLabelImage = [strLabelImage, ' (' , num2str(n), ') '];
imshow(img2D);
title(strLabelImage);
end

```

n	1	5	50	100	190	
Kết quả	1 (1) 	1 (5) 	8 (50) 	15 (100) 	28 (190) 	

- Q2 - Hãy viết function hiển thị ảnh có thứ tự là n (n là tham số) cùng label tương ứng trong tập test của tập dữ liệu MNIST. Paste code vào bài thực hành và lập bảng cho biết kết quả khi chạy với n=1, 500, 5000, 9000.

```

function Q2(n)
    fprintf('\n Load du lieu test');
    imgTestAll = loadData('DatabaseofFace/imgTestImagesAll.mat',
    'imgTestImagesAll');
    lblTestAll = loadData('DatabaseofFace/lblTestLabelsAll.mat',
    'lblTestLabelsAll');

    nTrainImages = size(imgTestAll, 2);
    fprintf('\n So hinh test %d', nTrainImages);

    fprintf('\n Hinh thu %d\n', n);

```





```

figure;

img = imgTestAll(:, n);
img2D = reshape(img, 112, 92);
strLabelImage = num2str(lblTestAll(n));
strLabelImage = [strLabelImage, ' (' , num2str(n), ') '];
imshow(img2D);
title(strLabelImage);

end

```

n	1	5	50	90
Kết quả	1 (1) 	2 (5) 	17 (50) 	30 (90) 

- Q3 - Hãy viết function thống kê số lượng các ảnh tương ứng với các label trong tập huấn luyện (train) của tập dữ liệu MNIST. Paste code vào bài thực hành và lập bảng kết quả khi chạy (nên xuất dưới dạng csv để tiện import thành bảng).

```

function Q3()

fprintf('\n Load du lieu train\n');

lblTrainAll = loadData('DatabaseofFace/lblTrainLabelsAll.mat',
'lblTrainLabelsAll');

indexs = unique(lblTrainAll(1,:));
indexs = indexs';
numLabel = size(lblTrainAll, 2);
numIndex = size(indexs, 1);
results = zeros(numIndex, 2);

for i=1:numIndex
    results(i, 1) = indexs(i);

```

```

end

for i=1:numLabel
    for j=1:numIndex
        if(lblTrainAll(i) == indexs(j))
            results(j, 2) = results(j, 2) + 1;
            break;
        end
    end
end

csvwrite('Q3.csv', results);
fprintf('Ket qua:\n');
fprintf([repmat('%d\t', 1, size(results, 2)) '\n'], results');
end

```

Label	Count
1	7
2	7
3	7
4	7
5	7
6	7
7	7
8	7
9	7
10	7
11	7
12	7

13	7
14	7
15	7
16	7
17	7
18	7
19	7
20	7
21	7
22	7
23	7
24	7
25	7
26	7
27	7
28	7
29	7
30	7
31	7
32	7
33	7
34	7
35	7
36	7
37	7
38	7

39	7
40	7

- Q4 - Hãy viết function thống kê số lượng các ảnh tương ứng với các label trong tập test của tập dữ liệu MNIST. Paste code vào bài thực hành và lập bảng kết quả khi chạy (nên xuất dưới dạng csv để tiện import thành bảng).

Code: Tương tự Q3.

Label	Count
1	3
2	3
3	3
4	3
5	3
6	3
7	3
8	3
9	3
10	3
11	3
12	3
13	3
14	3
15	3
16	3
17	3
18	3

19	3
20	3
21	3
22	3
23	3
24	3
25	3
26	3
27	3
28	3
29	3
30	3
31	3
32	3
33	3
34	3
35	3
36	3
37	3
38	3
39	3
40	3

- Q5 - Hãy viết function trả về kết quả nhận dạng của ảnh trong tập test có thứ tự là n (n là tham số, nằm trong đoạn [1, 10000]). Paste code vào bài thực hành và lập bảng kết quả khi chạy với n = 5, 500, 900.

```

function Q5(n)

    assert(n>=1 && n<=120, 'n nam trong khoang 1 den 120');

    fprintf('\n Load du lieu train');

    imgTrainAll = loadData('DatabaseofFace/imgTrainImagesAll.mat',
'imgTrainImagesAll');

    lblTrainAll = loadData('DatabaseofFace/lblTrainLabelsAll.mat',
'lblTrainLabelsAll');


    fprintf('\n Load du lieu test');

    imgTestAll = loadData('DatabaseofFace/imgTestImagesAll.mat',
'imgTestImagesAll');


    %% Extract Features

    featuresDataTrain = ExtractFeaturesPixel(imgTrainAll);


    %% Build model KNN

    Mdl = fitcknn(featuresDataTrain, lblTrainAll);

    imgTest = imgTestAll(:, n);

    lblPredictTest = predict(Mdl, imgTest');


    strLabelImage = 'Du doan: ';
    strLabelImage = [strLabelImage, num2str(lblPredictTest)];


    figure;

    img2D = reshape(imgTest, 112, 92);

    imshow(img2D);

    title(strLabelImage);


    fprintf('\n Ket thuc. \n');

end

```

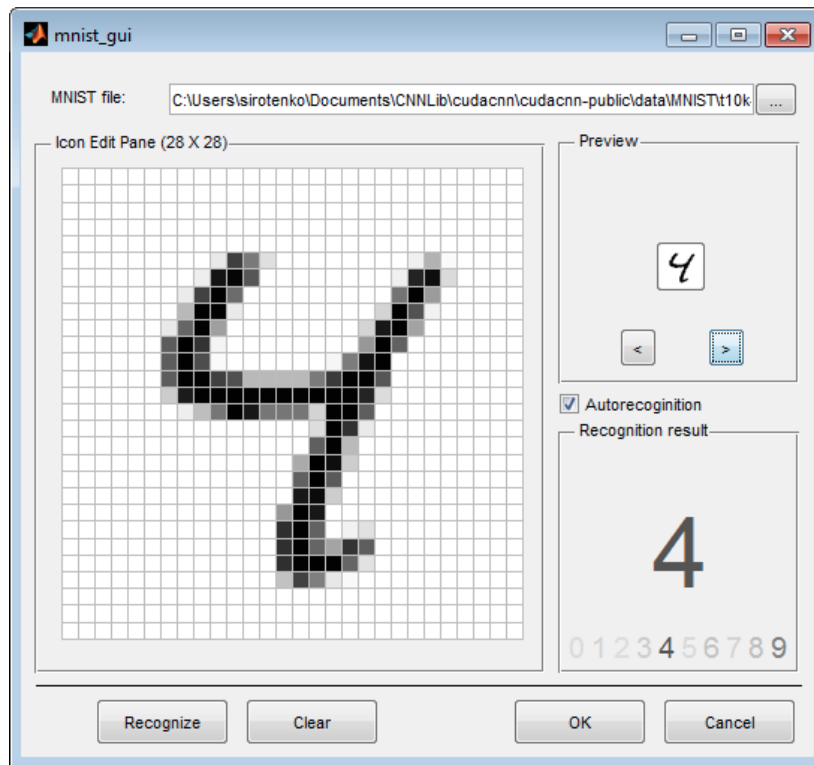
- Q6 - Hãy viết function với tham số đầu vào n là thứ tự của ảnh trong tập test - sau đó hiển thị ảnh tương ứng - rồi hiển thị kết quả nhận dạng - rồi cho biết kết quả nhận dạng là đúng hay sai khi so khớp với label của tập test.

```
function Q6(n)
    imgTrainAll = loadMNISTImages('train-images.idx3-ubyte');
    lblTrainAll = loadMNISTLabels('train-labels.idx1-ubyte');
    Mdl = fitcknn(imgTrainAll, lblTrainAll);

    imgTestAll = loadMNISTImages('t10k-images.idx3-ubyte');
    lblTestAll = loadMNISTLabels('t10k-labels.idx1-ubyte');
    imgTest = imgTestAll(:, n);
    lblPredictTest = predict(Mdl, imgTest);
    lblImageTest = lblTestAll(n);

    figure;
    img2D = reshape(imgTest, 28, 28);
    imshow(img2D);
    strLabelImage = 'Anh Test: ';
    strLabelImage = [strLabelImage, num2str(lblImageTest), '.'];
    strLabelImage = [strLabelImage, 'Du đoán:'];
    strLabelImage = [strLabelImage, num2str(lblPredictTest), '.'];
    if(lblPredictTest == lblImageTest)
        strLabelImage = [strLabelImage, 'Anh duoc Nhan dang dung'];
    else
        strLabelImage = [strLabelImage, 'Anh bi nhan dang sai.'];
    end
    title(strLabelImage);
end
```

- Q6* - Hãy viết thêm phần giao diện cho bài trên - tham khảo code ở đây: <https://bitbucket.org/intelligenceagent/cudacnn-public/wiki/Home>



Hình minh họa giao diện nhận dạng chữ số viết tay - trong đó người dùng vẽ lên màn hình và hệ thống hiển thị kết quả nhận dạng.

- Q7 - Hãy viết function đếm số lượng các ảnh có label là n (n là tham số) bị nhận dạng sai theo thuật toán knn. Paste code của function đã chạy được vào bài thực hành và lập bảng kết quả khi chạy với n= 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

```
function Q7(n)
    %Load du lieu train
    imgTrainAllLoad = load('data/imgTrainImagesAll.mat');
    lblTrainAllLoad = load('data/lblTrainLabelsAll.mat');
    imgTrainAll = imgTrainAllLoad.imgTrainImagesAll;
    lblTrainAll = lblTrainAllLoad.lblTrainLabelsAll;
    %Load du lieu test
    imgTestAllLoad = load('data/imgTestImagesAll.mat');
    lblTestAllLoad = load('data/lblTestLabelsAll.mat');
    imgTestAll = imgTestAllLoad.imgTestImagesAll;
    lblTestAll = lblTestAllLoad.lblTestLabelsAll;

    Mdl = fitcknn(imgTrainAll' , lblTrainAll);
    arrImagesNTest = [];
    for i = 1:size(imgTestAll, 2)
        if (lblTestAll(i) == n)
            arrImagesNTest = [arrImagesNTest, imgTestAll(:, i)];
        end
    end
end
```

```

        end
    end
    countFail = 0;
    for i = 1:size(arrImagesNTest,2)
        imgTest = arrImagesNTest(:, i);
        lblPredictTest = predict(Mdl, imgTest');
        if (lblPredictTest ~= n)
            countFail = countFail + 1;
        end
    end

    fprintf('So anh co label %d bi nhan dang sai la %d\n', n,
countFail);
end

```

- Q7* - Tương tự bài tập Q7 - nhưng lập bảng confusion matrix.

```

function Q7_Advance()
    imgTrainAll = loadMNISTImages(' ../lecture02/data/train-
images.idx3-ubyte');
    lblTrainAll = loadMNISTLabels(' ../lecture02/data/train-
labels.idx1-ubyte');

    Mdl = fitcknn(imgTrainAll', lblTrainAll);

    imgTestAll = loadMNISTImages(' ../lecture02/data/t10k-
images.idx3-ubyte');
    lblTestAll = loadMNISTLabels(' ../lecture02/data/t10k-
labels.idx1-ubyte');

    matrix = zeros(10, 10);
    lblImage = unique(lblTestAll);

    for val = 1:numel(lblImage)
        n = lblImage(val);
        fprintf('Anh so %d\n', n);
        indices = find(lblTestAll==n);
        for idx = 1:numel(indices)
            nNumber = indices(idx);
            imgTest = imgTestAll(:, nNumber);
            lblPredictTest = predict(Mdl, imgTest');
            matrix(n+1, lblPredictTest+1) =
matrix(n+1, lblPredictTest+1) + 1;
        end
    end
end

```

```
end
csvwrite('Confusion_matrix.csv', matrix);
end
```

- Q8** (Tương đương 50% đồ án môn học nếu kết hợp với Q7*) - Hãy viết function tính độ chính xác của thuật toán knn với các tham số khác nhau của hàm knn (ví dụ số lượng nearest neighbors, độ đo distance). Paste code vào bài thực hành và lập bảng kết quả.

```
function Q8_Advance(k)
    imgTrainAll = loadMNISTImages('./train-images.idx3-ubyte');
    lblTrainAll = loadMNISTLabels('./train-labels.idx1-ubyte');

    Mdl = fitcknn(imgTrainAll, lblTrainAll,
        'NumNeighbors', k);

    imgTestAll = loadMNISTImages('./t10k-images.idx3-ubyte');
    lblTestAll = loadMNISTLabels('./t10k-labels.idx1-ubyte');

    lblResult = predict(Mdl, imgTestAll);
    result = (lblResult == lblTestAll);
    nCount = sum(result);

    accuracy = nCount*100/size(imgTestAll, 2);
    fprintf('\nDo chính xác của thuật toán knn voi %d
nearest neighbors la %d\n', k, accuracy);
end
```

Hạn chót: 21/10/2017.

BÀI TẬP THỰC HÀNH 5

TBA

Yêu cầu:

1. Trả lời các câu hỏi vào tập tin bài tập thực hành VRA.HoVaTen.doc - sau đó kết xuất ra tập tin VRA.HoVaTen-TH5.pdf.
2. Syn tập tin VRA.HoVaTen-TH5.pdf lên tài khoản github và nộp trong Google Classroom.

Các câu hỏi:

- Q1 - TBA.

Hạn chót: TBA.