

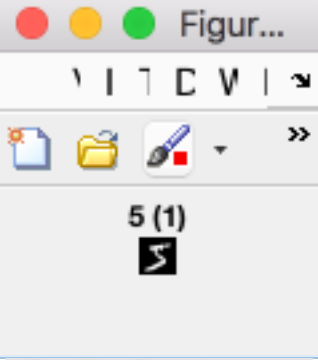
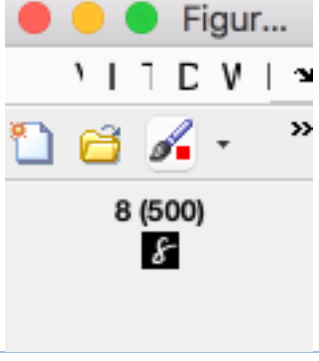
- Q1 - Hãy viết function hiển thị ảnh có thứ tự là n (n là tham số) cùng label tương ứng trong tập huấn luyện (train) của tập dữ liệu MNIST. Paste code vào bài thực hành và lập bảng cho biết kết quả khi chạy với n=1, 500, 5000, 10000, 59000.

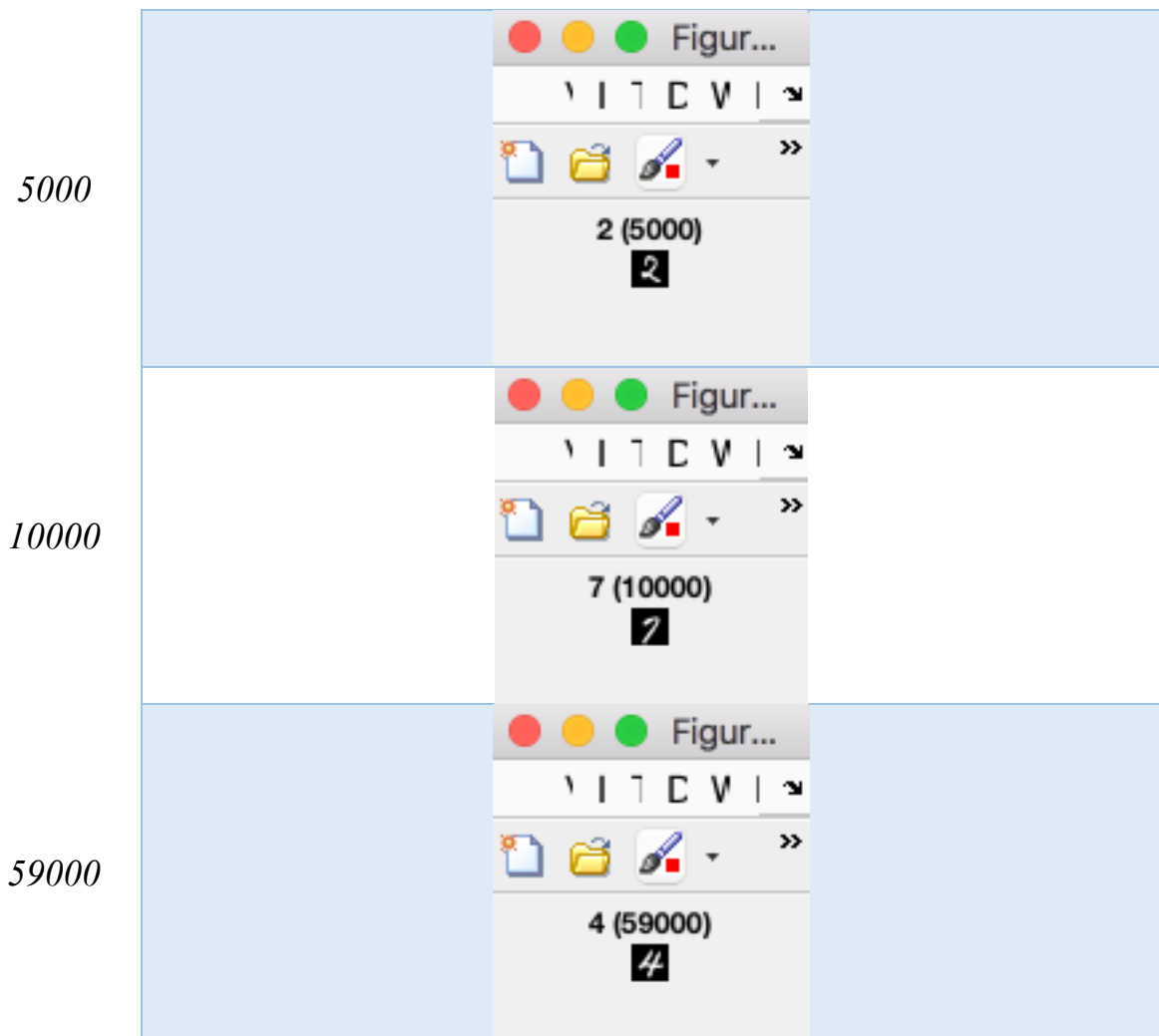
```
function Q1(n)
    fprintf('\n Load du lieu train');
    imgTrainAll = loadMNISTImages(' ../train-images.idx3-
ubyte');
    lblTrainAll = loadMNISTLabels(' ../train-labels.idx1-
ubyte');

    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, 28, 28);
    strLabelImage = num2str(lblTrainAll(n));
    strLabelImage = [strLabelImage, ' (' , num2str(n), ') '];
    imshow(img2D);
    title(strLabelImage);
end
```

		Kết quả	
<i>n</i>			
<i>1</i>			
500			



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

    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, 28, 28);
    strLabelImage = num2str(lblTestAll(n));
    strLabelImage = [strLabelImage, ' (' , num2str(n), ') '];
    imshow(img2D);
```

```
title(strLabelImage);
end
```

n	Kết quả	
1		
500		
5000		
9000		

- 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 = loadMNISTLabels(' ../train-labels.idx1-
ubyte');

    indexs = unique(lblTrainAll(:,1));
    numLabel = size(lblTrainAll, 1);
    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([repmat('%d\t', 1, size(results, 2)) '\n'],
results');

end

```

Label	Count
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).

```

function Q4()
    fprintf('\n Load du lieu test\n');
    lblTestAll = loadMNISTLabels(' ../t10k-labels.idx1-ubyte');

```

```

indexs = unique(lblTestAll(:,1));
numLabel = size(lblTestAll, 1);
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(lblTestAll(i) == indexs(j))
            results(j, 2) = results(j, 2) + 1;
            break;
        end
    end
end

csvwrite('Q4.csv', results);
fprintf('Ket qua:\n');
fprintf([repmat('%d\t', 1, size(results, 2)) '\n'],
results');
end

```

Label	Count
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.

```

function Q5(n)
    assert(n>=1 && n<=10000, 'n nam trong khoang 1 den 10000');
    fprintf('\n Load du lieu train');
    imgTrainAll = loadMNISTImages(' ../train-images.idx3-ubyte');

```

```

    lblTrainAll = loadMNISTLabels(' ../train-labels.idx1-
ubyte');

    Mdl = fitcknn(imgTrainAll', lblTrainAll);

    fprintf('\n Load du lieu test');
    imgTestAll = loadMNISTImages(' ../t10k-images.idx3-ubyte');

    imgTest = imgTestAll(:, n);
    lblPredictTest = predict(Mdl, imgTest');

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

    figure;
    img2D = reshape(imgTest, 28, 28);
    imshow(img2D);
    title(strLabelImage);

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

```

<i>n</i>	Kết quả		
5		Du doan: 4 	
500		Du doan: 6 	
900		Du doan: 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.

```

function Q6(n)
    assert(n>=1 && n<=10000, 'n nam trong khoang 1 den
10000');
    fprintf('\n Load du lieu train');
    imgTrainAll = loadMNISTImages(' ../train-images.idx3-
ubyte');
    lblTrainAll = loadMNISTLabels(' ../train-labels.idx1-
ubyte');

```

```

Mdl = fitcknn(imgTrainAll', lblTrainAll);

fprintf('\n Load du lieu test');
imgTestAll = loadMNISTImages(' ../t10k-images.idx3-ubyte');
lblTestAll = loadMNISTLabels(' ../t10k-labels.idx1-ubyte');

imgTest = imgTestAll(:, n);
lblPredictTest = predict(Mdl, imgTest');

lblImageTest = lblTestAll(n);

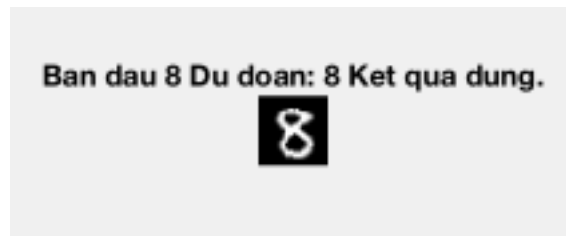
strLabelImage = 'Ban dau ' ;
strLabelImage = [strLabelImage, num2str(lblImageTest)];
strLabelImage = [strLabelImage, ' Du doan: '];
strLabelImage = [strLabelImage, num2str(lblPredictTest)];

if (lblImageTest == lblPredictTest)
    strLabelImage = [strLabelImage, ' Ket qua dung.'];
else
    strLabelImage = [strLabelImage, ' Ket qua sai.'];
end

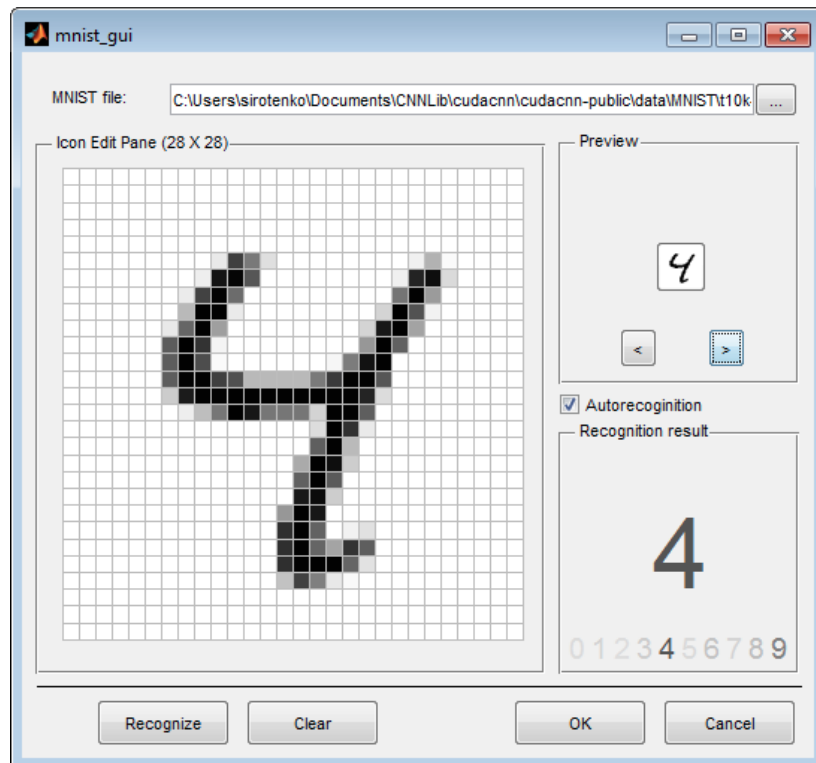
figure;
img2D = reshape(imgTest, 28, 28);
imshow(img2D);
title(strLabelImage);

fprintf('\n Ket thuc. \n');
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)
    assert(n>=1 && n<=10000, 'n nam trong khoang 1 den 10000');
    fprintf('\n Load du lieu train');
    imgTrainAll = loadMNISTImages(' ../train-images.idx3-ubyte');
    lblTrainAll = loadMNISTLabels(' ../train-labels.idx1-ubyte');

    Mdl = fitcknn(imgTrainAll', lblTrainAll);

    fprintf('\n Load du lieu test');
    imgTestAll = loadMNISTImages(' ../t10k-images.idx3-ubyte');
    lblTestAll = loadMNISTLabels(' ../t10k-labels.idx1-ubyte');

    numLabel = size(lblTestAll, 1);
    result = 0;

    for i=1:numLabel
        if(lblTestAll(i) == n)
            imgTest = imgTestAll(:, i);
            if(predict(Mdl, imgTest') ~= lblTestAll(i))
```



```

        result = result + 1;
        fprintf('\n Hình thu %d sai, tong so %d\n', i,
result);
    end
end
end
    fprintf('\n Ket qua: %d \n', result);
end

```

n	Count
0	7
1	6
2	40
3	40
4	38
5	32
6	14
7	36
8	54
9	42

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

```

function Q7_()
    fprintf('\n Load du lieu train');
    imgTrainAll = loadMNISTImages(' ../train-images.idx3-
ubyte');
    lblTrainAll = loadMNISTLabels(' ../train-labels.idx1-
ubyte');

    Mdl = fitcknn(imgTrainAll', lblTrainAll);

    fprintf('\n Load du lieu test');
    imgTestAll = loadMNISTImages(' ../t10k-images.idx3-ubyte');
    lblTestAll = loadMNISTLabels(' ../t10k-labels.idx1-ubyte');

    indexs = unique(lblTestAll(:,1));
    numIndex = size(indexs, 1);
    results = zeros(numIndex + 1, numIndex + 1);

    for i=1:numIndex
        results(i + 1, 1) = indexs(i);
        results(1, i + 1) = indexs(i);
    end
    predictResults = predict(Mdl, imgTestAll');
    resultCount = size(predictResults, 1);

    for i=1:resultCount

```

```

        label = lblTestAll(i);
        labelIndex = label + 2;
        result = predictResults(i);
        resultIndex = result + 2;
        results(labelIndex, resultIndex) = results(labelIndex,
resultIndex) + 1;
    end

    csvwrite('Q7*.csv', results);
    fprintf('\n Ket qua: \n');
    fprintf([repmat('%d\t', 1, size(results, 2)) '\n'],
results');

end

```

	0	1	2	3	4	5	6	7	8	9
0	973	1	1	0	0	1	3	1	0	0
1	0	1129	3	0	1	1	1	0	0	0
2	7	6	992	5	1	0	2	16	3	0
3	0	1	2	970	1	19	0	7	7	3
4	0	7	0	0	944	0	3	5	1	22
5	1	1	0	12	2	860	5	1	6	4
6	4	2	0	0	3	5	944	0	0	0
7	0	14	6	2	4	0	0	992	0	10
8	6	1	3	14	5	13	3	4	920	5
9	2	5	1	6	10	5	1	11	1	967

- 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(k, distance)
    fprintf('\n Load du lieu train');
    imgTrainAll = loadMNISTImages(' ../train-images.idx3-
ubyte');
    lblTrainAll = loadMNISTLabels(' ../train-labels.idx1-
ubyte');

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

    fprintf('\n Load du lieu test');
    imgTestAll = loadMNISTImages(' ../t10k-images.idx3-ubyte');
    lblTestAll = loadMNISTLabels(' ../t10k-labels.idx1-ubyte');

    lblResult = predict(Mdl, imgTestAll');

```

```

nResult = (lblResult == lblTestAll);
nCount = sum(nResult);
fprintf('\nSo luong mau dung: %d\n',nCount);
end

```

k	Distance	Số mẫu đúng
1	euclidean	9691
3	euclidean	9706
5	euclidean	9687
1	cosine	9723
3	cosine	9737
5	cosine	9729