

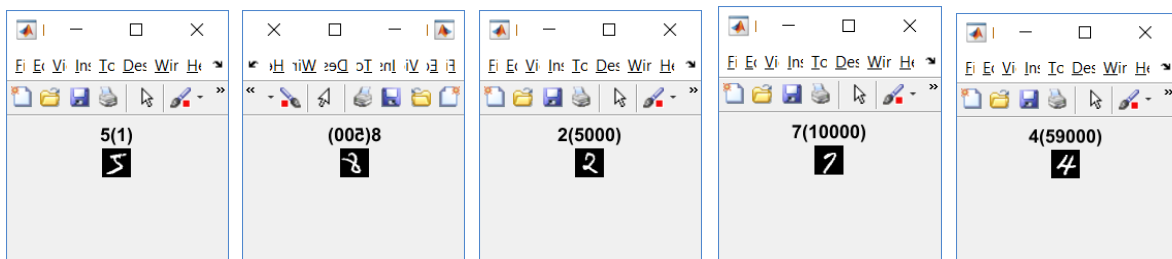
BÀI TẬP THỰC HÀNH 3

KHẢO SÁT BỘ DỮ LIỆU CHỮ SỐ VIẾT TAY

- **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)  
    imgs = loadMNISTImages ('train-images.idx3-ubyte');  
    labels = loadMNISTLabels ('train-labels.idx1-ubyte');  
  
    figure;  
    img = imgs(:, n);  
    img2D = reshape (img, 28, 28);  
  
    %get lable  
    strLabelImage = num2str(labels(n));  
    strLabelImage = [strLabelImage, '(', num2str(nNumber), ')'];  
  
    %show image  
    imshow(img2D);  
    title(strLabelImage);  
end  
-----
```

N = 1, 500, 5000, 10000, 59000



- **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)
    imgs = loadMNISTImages ('t10k-images.idx3-ubyte');
    labels = loadMNISTLabels ('t10k-labels.idx1-ubyte');

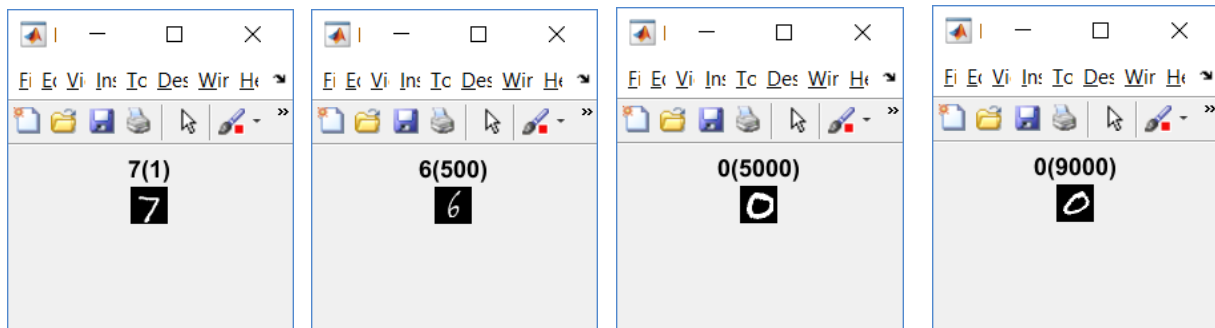
    figure;
    img = imgs(:, n);
    img2D = reshape (img, 28, 28);

    %get lable
    strLabelImage = num2str(labels(n));
    strLabelImage = [strLabelImage, '(', num2str(n), ')'];

    %show image
    imshow(img2D);
    title(strLabelImage);
end

```

N = 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).**
- **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: Chung cho Q3 và Q4

```

-----
function results = Q3_4(type, fileName)
    if strcmp(type, 'test')
        imgs = loadMNISTImages ('t10k-images.idx3-ubyte');
        lables = loadMNISTLabels ('t10k-labels.idx1-ubyte');
    elseif strcmp(type, 'train')
        imgs = loadMNISTImages ('train-images.idx3-ubyte');
        lables = loadMNISTLabels ('train-labels.idx1-ubyte');
    else
        throw('type is "train" or "test"');
    end
    sizeImgs = size(imgs,2);
    count = zeros(10,1);
    results = zeros(2, 10);
    for i = 1:sizeImgs
        label = lables(i) + 1;
        count(label,1) = count(label,1) + 1;
        results(1, label) = label;
        results(2, label) = results(2, label) + 1;
    end

    %export csv
    csvwrite(fileName,results);
end
-----
Run : Q3_4('test', 'test.csv')

```

- **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)  
  
    imgsTrain = loadMNISTImages('train-images.idx3-ubyte');  
    lablesTrain = loadMNISTLabels('train-labels.idx1-ubyte');  
    imgsTest = loadMNISTImages('t10k-images.idx3-ubyte');  
  
    Mdl = fitcknn(imgsTrain' , lablesTrain);  
  
    img = imgsTest(:, n);  
    lblPredictTest = predict(Mdl, img');  
    fprintf('With n = %d then it has the result :  %d\n', n,  
    lblPredictTest);  
end  
-----
```

Q5(5)

With n = 5 then it has the result : 4

Q5(500)

With n = 500 then it has the result : 6

Q5(900)

With n = 900 then it has the result : 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.**

```

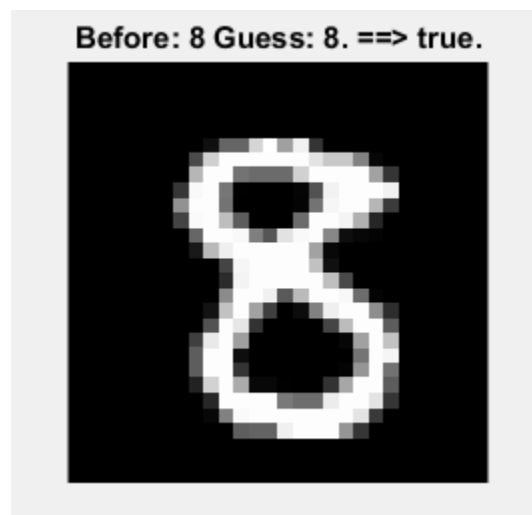
-----
function Q6(n)
    imgsTrain = loadMNISTImages('train-images.idx3-ubyte');
    labelsTrain = loadMNISTLabels('train-labels.idx1-ubyte');
    imgsTest = loadMNISTImages('t10k-images.idx3-ubyte');
    labelsTest = loadMNISTLabels('t10k-labels.idx1-ubyte');

    Mdl = fitcknn(imgsTrain' , labelsTrain);
    img = imgsTest(:, n);
    lblPredictTest = predict(Mdl, img);
    lblImageTest = labelsTest(n);

    figure;
    img2D = reshape(img, 28, 28);
    imshow(img2D);
    strLabelGuess = ['Before: ' , num2str(lblImageTest)];
    strLabelGuess = [strLabelGuess, ' Guess: '];
    strLabelGuess = [strLabelGuess, num2str(lblPredictTest), '.'];

    if(lblPredictTest == lblImageTest)
        strLabelGuess = [strLabelGuess, ' ==> true.'];
    else
        strLabelGuess = [strLabelGuess, ' ==> false.'];
    end
    title(strLabelGuess);
end
-----
Run : Q6(900)

```



- **Q6* - Hãy viết thêm phần giao diện cho bài trên - tham khảo code ở đây:**
<https://bitbucket.org/intelligenceagent/cudacnn-public/wiki/Home>
- **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)
    imgsTrain = loadMNISTImages('train-images.idx3-ubyte');
    labelsTrain = loadMNISTLabels('train-labels.idx1-ubyte');

    imgsTest = loadMNISTImages('t10k-images.idx3-ubyte');
    labelsTest = loadMNISTLabels('t10k-labels.idx1-ubyte');

    Mdl = fitcknn(imgsTrain', labelsTrain);

    sizeOfLabel = size(labelsTest, 1);
    results = 0;

    for i=1:sizeOfLabel
        if(labelsTest(i) == n)
            img = imgsTest(:, i);
            if(predict(Mdl, img') ~= labelsTest(i))
                results = results + 1;
            end
        end
    end
    fprintf('The lable %d has number of wrong : %d\n', n, results);
end
-----
Run : Q7(1)
The lable 0 has number of wrong : 7
The lable 1 has number of wrong : 6

```

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

```
-----
function Q7S()
    imgsTrain = loadMNISTImages('train-images.idx3-ubyte');
    labelsTrain = loadMNISTLabels('train-labels.idx1-ubyte');

    imgsTest = loadMNISTImages('t10k-images.idx3-ubyte');
    labelsTest = loadMNISTLabels('t10k-labels.idx1-ubyte');

    Mdl = fitcknn(imgsTrain', labelsTrain);

    mx = zeros(10, 10);
    lblImage = unique(labelsTest);

    for i = 1:numel(lblImage)
        n = lblImage(i);

        ind = find(labelsTest==n);
        for idx = 1:numel(ind)
            nNumber = ind(idx);
            img = imgsTest(:, nNumber);
            lblPredictTest = predict(Mdl, img');

            mx(n + 1, lblPredictTest + 1) = mx(n + 1, lblPredictTest + 1)
+ 1;
        end
    end
    %write csv
    csvwrite('ConfusionMatrix.csv', mx);
end
-----
Run : Q7S
```

- **Q8** (Tương đương 50% đề 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 Q8S()
    imgsTrain = loadMNISTImages('./train-images.idx3-ubyte');
    labelsTrain = loadMNISTLabels('./train-labels.idx1-ubyte');
    imgsTest = loadMNISTImages('./t10k-images.idx3-ubyte');
    labelsTest = loadMNISTLabels('./t10k-labels.idx1-ubyte');

    %NumNeighbors
    k = 5;
    %'Distance', 'minkowski'
    d = 'minkowski';
    Mdl = fitcknn(imgsTrain, labelsTrain, 'NumNeighbors', k, 'Distance',
d);

    labelsResult = predict(Mdl, imgsTest);
    result = (labelsResult == labelsTest);
    count = sum(result);

    acc = (count * 100) / size(imgsTest, 2);
    fprintf('\nWith NumNeighbors = %d ==> we have accuracy : %d \n', k,
acc);
end
-----
accuracy : 9.687000e+01
```

Hạn chót: 14/10/2017.
