Họ và tên: Đặng Minh Tuấn
MSSV: 14521039
Đường dẫn vào github:
https://github.com/TuanDang1996/VRAClass

THỰC HÀNH 3

Q1: * Source:

```
function Quest1()
    imgTrainAll = loadMNISTImages('./train-images.idx3-ubyte');
    nNumber = 1;
    imgTrain = imgTrainAll(:,nNumber);
    img2D = reshape(imgTrain, 28, 28);
    imshow(img2D);
end
```

* Bảng kết quả:

n =	1	500	5000	10000	59000
Kết quả	5	8	2	7	4

Q2: * Source:

```
function Quest2()
    imgTestAll = loadMNISTImages('./t10k-images.idx3-ubyte');
    nNumber = 1;
    imgTrain = imgTestAll(:,nNumber);
    img2D = reshape(imgTrain, 28, 28);
    imshow(img2D);
end
```

* Bảng kết quả:

n =	1	500	5000	9000
Kết quả	7	6	0	0

Q3: * Source:

```
function Quest3()
  lblTrainAll = loadMNISTLabels('./train-labels.idx1-ubyte');
  nNumber = size(lblTrainAll,1);
  temp = 0;
  A = [];
  while temp <= 9
    A = cat(2, A, [temp;0]);
```

```
        temp = temp + 1;
    end
    i = 1;
    while i <= nNumber
        switch lblTrainAll(i)
            case 0
                A(2,1) = A(2,1) + 1;
            case 1
                A(2,2) = A(2,2) + 1;
            case 2
                A(2,3) = A(2,3) + 1;
            case 3
                A(2,4) = A(2,4) + 1;
            case 4
                A(2,5) = A(2,5) + 1;
            case 5
                A(2,6) = A(2,6) + 1;
            case 6
                A(2,7) = A(2,7) + 1;
            case 7
                A(2,8) = A(2,8) + 1;
            case 8
                A(2,9) = A(2,9) + 1;
            case 9
                A(2,10) = A(2,10) + 1;
        end
        i= i + 1;
    end
    disp(A);
    csvwrite("result.dat",A)
end
```

* Bảng kết quả:

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

Q4: * Source code:

```
function Quest4()
    lblTestAll = loadMNISTLabels('./t10k-labels.idx1-ubyte');
    nNumber = size(lblTestAll,1);
```

```
    temp = 0;
    A = [];
    while temp <= 9
        A = cat(2, A, [temp;0]);
        temp = temp + 1;
    end
    i = 1;
    while i <= nNumber
        switch lblTestAll(i)
            case 0
                A(2,1) = A(2,1) + 1;
            case 1
                A(2,2) = A(2,2) + 1;
            case 2
                A(2,3) = A(2,3) + 1;
            case 3
                A(2,4) = A(2,4) + 1;
            case 4
                A(2,5) = A(2,5) + 1;
            case 5
                A(2,6) = A(2,6) + 1;
            case 6
                A(2,7) = A(2,7) + 1;
            case 7
                A(2,8) = A(2,8) + 1;
            case 8
                A(2,9) = A(2,9) + 1;
            case 9
                A(2,10) = A(2,10) + 1;
        end
        i= i + 1;
    end
    disp(A);
    csvwrite("result.dat",A)
end
```

* Bảng kết quả:

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

Q5: * Source:

```
function Quest5()
    lblTrainAll = loadMNISTLabels('./train-labels.idx1-ubyte');
    imgTrainAll = loadMNISTImages('./train-images.idx3-ubyte');
    Mdl = fitcknn(imgTrainAll',lblTrainAll);
    imgTestAll = loadMNISTImages('./t10k-images.idx3-ubyte');
    lblTestAll = loadMNISTLabels('./t10k-labels.idx1-ubyte');
    nNumber = 900;
    img = imgTestAll(:,nNumber);
    lblPredictTest = predict(Mdl,img');
    lblImageTest = lblTestAll(nNumber);
    strLabelImage = 'Du Doan ';
    strLabelImage = [strLabelImage, num2str(lblPredictTest),'. '];
    strLabelImage = [strLabelImage, 'Ket qua: ', num2str(lblImageTest), '.'];
    fprintf(strLabelImage);
End
```

* Bảng kết quả:

n=	5	500	900
Dự đoán	4	6	8
Kết quả	4	6	8

Q6: *Source:

```
function Quest6()
    lblTrainAll = loadMNISTLabels('./train-labels.idx1-ubyte');
    imgTrainAll = loadMNISTImages('./train-images.idx3-ubyte');

    Mdl = fitcknn(imgTrainAll',lblTrainAll);

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

    nNumber = 2;
    img = imgTestAll(:,nNumber);
    lblPredictTest = predict(Mdl,img');
    lblImageTest = lblTestAll(nNumber);
    img2D = reshape(img,28,28);
    imshow(img2D);
    strLabelImage = 'Du Doan ';
    strLabelImage = [strLabelImage, num2str(lblPredictTest),'. '];
```

```
    strLabelImage = [strLabelImage, 'Ket qua: ', num2str(lblImageTest), '.'];
    title(strLabelImage);
end
```

Q7: *Source:

```
function Quest7()
    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');



    nTestAll = size(imgTestAll, 2);
    lblTest = 0;
    while lblTest <= 9
        i = 1;
        filtImageTest = [];
        filtLabelTest = [];
        while i <= nTestAll
            if(lblTestAll(i) == lblTest)
                img = imgTestAll(:,i);
                lbl = lblTestAll(i);
                filtImageTest = cat(2,filtImageTest,img);
                filtLabelTest = cat(1,filtLabelTest,lbl);
            end
            i = i + 1;
        end
        lblPredict = predict(mdl,filtImageTest');
        nResult = (lblPredict ~= filtLabelTest);
        count = sum(nResult);
        fprintf('\nSo %d bi nhan dang sai %d',lblTest,count);
        lblTest = lblTest + 1;
    end
end
```

* Bảng kết quả:

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

Q7*: * source:

```
function Quest8()
   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');

   matrix = [-1; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9];
   temp = 0;
   while temp <= 9
      A = [temp; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0];
      matrix = cat(2,matrix,A);
      temp = temp + 1;
   end
   lblPredict = predict(mdl,imgTestAll');
   i= 1;
   nSize = size(imgTestAll,2);
   while i <= nSize
      if( lblPredict(i) ~= lblTestAll(i))
         matrix(lblTestAll(i) + 2, lblPredict(i) + 2) = matrix(lblTestAll(i) + 2, lblPredict(i)
+ 2) + 1;
      end
      i = i + 1;
   end
   disp(matrix);
   csvwrite("result.dat",matrix)
end
```

* Bảng kết quả:

-1	0	1	2	3	4	5	6	7	8	9
0	0	1	1	0	0	1	3	1	0	0
1	0	0	3	0	1	1	1	0	0	0
2	7	6	0	5	1	0	2	16	3	0
3	0	1	2	0	1	19	0	7	7	3
4	0	7	0	0	0	0	3	5	1	22
5	1	1	0	12	2	0	5	1	6	4
6	4	2	0	0	3	5	0	0	0	0
7	0	14	6	2	4	0	0	0	0	10
8	6	1	3	14	5	13	3	4	0	5
9	2	5	1	6	10	5	1	11	1	0

Q8**: * Source:

```
function Quest9()
    imgTrainAll = loadMNISTImages('./train-images.idx3-ubyte');
    lblTrainAll = loadMNISTLabels('./train-labels.idx1-ubyte');

    mdl = fitcknn(imgTrainAll',lblTrainAll,'NumNeighbors',4,'Standardize',1);

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

    matrix = [-1; 0; 1; 2; 3; 4; 5; 6; 7; 8; 9];
    temp = 0;
    while temp <= 9
        A = [temp; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0];
        matrix = cat(2,matrix,A);
        temp = temp + 1;
    end
    lblPredict = predict(mdl,imgTestAll');
    i= 1;
    nSize = size(imgTestAll,2);
    while i <= nSize
        if( lblPredict(i) ~= lblTestAll(i))
```

```
            matrix(lblTestAll(i) + 2, lblPredict(i) + 2) = matrix(lblTestAll(i) + 2, lblPredict(i)
+ 2) + 1;
        end
        i = i + 1;
    end
    disp(matrix);
end
```

*****Bảng kết quả:

** k = 2:

-1	0	1	2	3	4	5	6	7	8	9
0	0	1	0	1	2	5	10	2	0	0
1	0	0	4	0	0	0	1	0	0	0
2	20	13	0	8	10	0	7	18	7	0
3	0	4	10	0	5	34	0	19	4	2
4	0	13	3	0	0	2	6	4	2	10
5	5	0	3	16	8	0	8	2	2	5
6	6	4	1	0	7	11	0	0	0	0
7	0	26	9	2	10	2	0	0	2	14
8	19	4	11	31	12	55	4	10	0	4
9	7	4	7	11	34	9	0	60	3	0

** k = 3:

-1	0	1	2	3	4	5	6	7	8	9
0	0	0	0	2	2	4	6	2	0	0
1	0	0	3	0	1	0	3	0	1	0
2	12	9	0	15	5	0	5	15	9	2
3	0	4	4	0	2	19	0	15	8	2
4	1	9	4	2	0	2	7	4	3	27
5	5	0	3	18	4	0	10	2	7	8
6	10	4	2	0	3	10	0	0	2	0
7	0	22	12	3	7	2	0	0	1	27
8	12	3	8	15	11	32	5	7	0	7
9	5	6	5	10	18	7	0	29	3	0

** k = 4:

-1	0	1	2	3	4	5	6	7	8	9
0	0	1	2	2	1	6	7	0	0	0
1	0	0	3	0	0	0	3	0	0	0
2	16	6	0	13	6	0	7	13	8	1
3	1	2	6	0	1	19	1	15	8	4
4	0	11	2	2	0	2	6	4	3	20
5	5	0	3	15	8	0	10	2	4	7
6	9	4	1	0	4	8	0	0	2	0
7	0	22	11	3	6	2	0	0	1	23
8	14	5	6	22	9	30	3	7	0	4
9	7	5	5	10	23	7	0	40	3	0

** k = 5:

-1	0	1	2	3	4	5	6	7	8	9
0	0	0	1	3	1	5	6	1	0	0
1	0	0	3	0	0	0	3	0	0	0
2	14	6	0	20	5	0	8	10	10	1
3	0	3	5	0	3	13	1	11	10	4
4	1	10	5	3	0	3	6	4	2	26
5	5	1	2	21	8	0	14	2	6	7
6	10	4	2	0	3	6	0	0	3	0
7	0	21	9	4	8	2	0	0	1	31
8	13	3	6	17	8	31	3	6	0	7
9	6	5	5	10	18	6	0	31	3	0