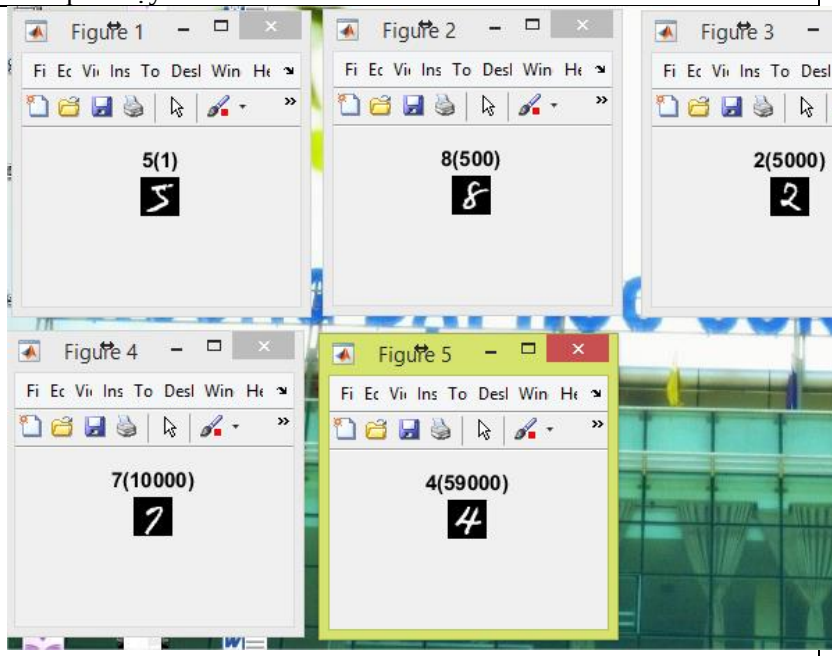
Họ tên: Lý Trọng Nhân

MSHV: CH1601015

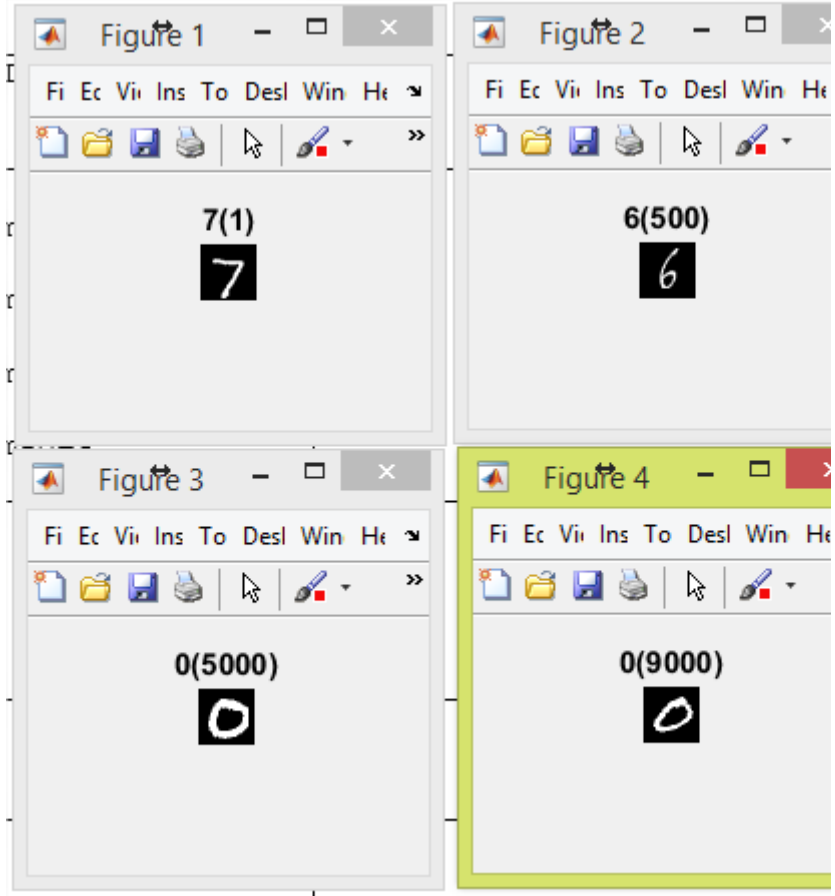
Link GitHub: https://github.com/vra-nhanlt/vra-nhanlt

BÀI TẬP THỰC HÀNH 3

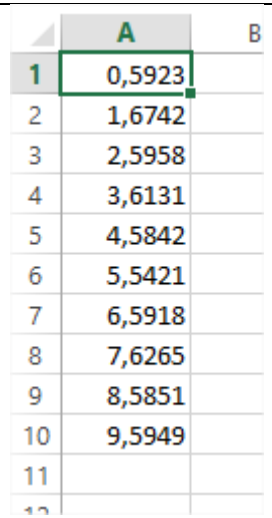
	Yêu cầu	Code	Kết quả chạy

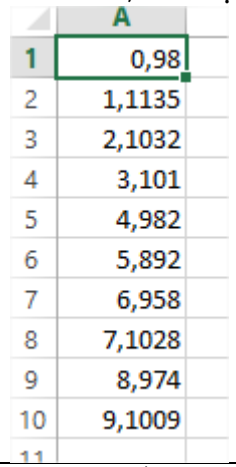
Q1	Function hiển thị ảnh và label training thứ n	`function buoi02_showanimage( images, labels, number )`
		`  figure;`
		`  tmpImage = images(:, number);`
		`  tmp2DImage = reshape(tmpImage, 28, 28);`
		`  tmpLabel = num2str(labels(number));`
		`  tmpLabel = [tmpLabel, '(', num2str(number), ')'];`
		`  imshow(tmp2DImage);`
		`  title(tmpLabel);`
		`end`
		
		% Lệnh gọi và test
		`% Load data`
		`allTrainingImages = loadMNISTImages('./train-images.idx3-ubyte');`
		`allTrainingLabels = loadMNISTLabels('./train-labels.idx1-ubyte');`
		`allTestingImages = loadMNISTImages('./t10k-images.idx3-ubyte');`
		`allTestingLabels = loadMNISTLabels('./t10k-labels.idx1-ubyte');`
		
		`% Show images & labels`
		`buoi02_showanimage(allTrainingImages, allTrainingLabels, 1);`
		`buoi02_showanimage(allTrainingImages, allTrainingLabels, 500);`
		`buoi02_showanimage(allTrainingImages, allTrainingLabels, 5000);`
		`buoi02_showanimage(allTrainingImages, allTrainingLabels, 10000);`
			

		buoi02_showanimage(allTrainingImages, allTrainingLabels, 59000);	

Q2	Function hiển thị ảnh và label trong tập test thứ n	Tương tự Q1
		buoi02_showanimage(allTestingImages, allTestingLabels, 1);
		buoi02_showanimage(allTestingImages, allTestingLabels, 500);
		buoi02_showanimage(allTestingImages, allTestingLabels, 5000);
		buoi02_showanimage(allTestingImages, allTestingLabels, 9000);
			

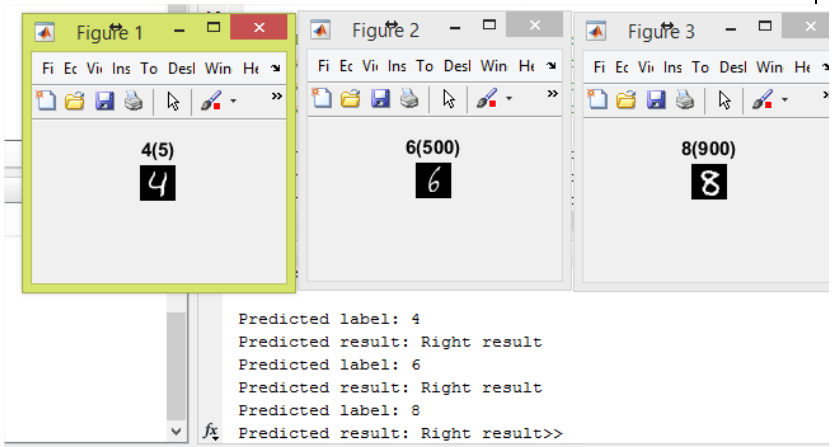
Q3	Function thống kê số lượng ảnh theo label tập training	function result = btth3_countbylabel(labels, filename)
		result = zeros([10 2]);
		% add label 0->9
		i = 0;
		while (i < 10)
		result(i+1,1) = i;
		i = i + 1;
		end
			Cấu trúc output CSV mỗi dòng gồm:
			<label>,<sốlượng>

		<pre>% count label
		i = 1;
		while (i <= size(labels, 1))
		    label = labels(i);
		    result(label+1,2) = result(label+1,2) + 1;
		    i = i + 1;
		end
		
		% export to csv file
		csvwrite(filename,result);
		end
		
		% Lệnh gọi và test
		result = btth3_countbylabel(allTrainingLabels,
		"q3.csv");</pre>
			
			A
			1 0,5923
			2 1,6742
			3 2,5958
			4 3,6131
			5 4,5842
			6 5,5421
			7 6,5918
			8 7,6265
			9 8,5851
			10 9,5949
			11

Q4	Function thống kê số lượng ảnh theo label tập test	Tương tự Q3 % Lệnh gọi và test result = btth3_countbylabel(allTrainingLabels, "q4.csv");	**Cấu trúc output CSV mỗi dòng gồm: <label>,<sốlượng>**  A 1 0,98 2 1,1135 3 2,1032 4 3,101 5 4,982 6 5,892 7 6,958 8 7,1028 9 8,974 10 9,1009

Q5	Function trả về kết quả nhận dạng của ảnh trong tập	function result = recognize(trainingImages,
		trainingLabels, testingImages, n)
		md = fitcknn(trainingImages', trainingLabels);
		testingImage = testingImages(:, n);
		result = predict(md, testingImage');
		end
			Result lần lượt là 4, 6, 8

	test có thứ tự n	% lệnh test result = recognize(allTrainingImages, allTrainingLabels, allTestingImages, 5) result = recognize(allTrainingImages, allTrainingLabels, allTestingImages, 500) result = recognize(allTrainingImages, allTrainingLabels, allTestingImages, 900)	

Q6	Lệnh lấy vecto dòng 10 của ma trận A: q6 = A(10, :);	% Hàm con show image
		function buoi02_showanimage(images, labels, number)
		figure;
		tmpImage = images(:, number);
		tmp2DImage = reshape(tmpImage, 28, 28);
		tmpLabel = num2str(labels(number));
		tmpLabel = [tmpLabel, '(', num2str(number), ')'];
		imshow(tmp2DImage);
		title(tmpLabel);
		end
		
		% Hàm con nhận dạng image
		function result = recognize(trainingImages, trainingLabels, testingImages, n)
		md = fitcknn(trainingImages', trainingLabels);
		testingImage = testingImages(:, n);
		result = predict(md, testingImage');
		end
		
		% Hàm con hiển thị kết quả nhận dạng
		function result = check(predictedLabel, testingLabels, n)
		result = "Wrong result";
		
		label = testingLabels(n);
		if (label == predictedLabel)
		result = "Right result";
		end
		end
		
		% Hàm cha gọi các hàm con thực hiện yêu cầu
			N = 5
			 

<table>
<tr>
<td></td>
<td></td>
<td>

```
function check_recognize_result(trainingImages,
trainingLabels, testingImages, testingLabels, n)
% show image
buoi02_showanimage(testingImages, testingLabels, n)
% show predicted label
predictedLabel = recognize(trainingImages,
trainingLabels, testingImages, n);
fprintf("\n Predicted label: %d", predictedLabel);
% show result
result = check(predictedLabel, testingLabels, n);
fprintf("\n Predicted result: %s", result);
end
```

</td>
<td></td>
</tr>
<tr>
<td>Q7</td>
<td>Function trả về số lượng ảnh có label n bị nhận sai bằng knn</td>
<td>

```
% Hàm build model
function md = build_model(trainingImages,
trainingLabels)
md = fitcknn(trainingImages', trainingLabels);
end

% Hàm nhận dạng ảnh thứ n trong testingImages
function result = recognite_from_md(md,
testingImages, n)
testingImage = testingImages(:, n);
result = predict(md, testingImage');
end

% Hàm đếm số ảnh có label n bị nhận dạng sai (sử
dụng model)
function result =
wrong_recognition_from_md_count(md, testingImages,
testingLabels, number)
count = size(testingLabels, 1);
i = 1;
wrong_count = 0;
while (i <= count)
    if (testingLabels(i) == number &&
recognite_from_md(md, testingImages, i) ~= number)
```

</td>
<td>

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

</td>
</tr>
</table>

```
            wrong_count = wrong_count + 1;
    end
    i = i + 1;
end
result = wrong_count;
end

% Hàm đếm số lượng ảnh có label n bị nhận dạng sai
function result =
wrong_recognition_count(trainingImages,
trainingLabels, testingImages, testingLabels,
number)
md = build_model(trainingImages, trainingLabels);
result = wrong_recognition_from_md_count(md,
testingImages, testingLabels, number);
end

% Load data và lệnh test
% Load data
allTrainingImages = loadMNISTImages('./train-
images.idx3-ubyte');
allTrainingLabels = loadMNISTLabels('./train-
labels.idx1-ubyte');
allTestingImages = loadMNISTImages('./t10k-
images.idx3-ubyte');
allTestingLabels = loadMNISTLabels('./t10k-
labels.idx1-ubyte');

wrong_recognition_count(allTrainingImages,
allTrainingLabels, allTestingImages,
allTestingLabels, 0)
wrong_recognition_count(allTrainingImages,
allTrainingLabels, allTestingImages,
allTestingLabels, 1)
wrong_recognition_count(allTrainingImages,
allTrainingLabels, allTestingImages,
allTestingLabels, 2)
wrong_recognition_count(allTrainingImages,
allTrainingLabels, allTestingImages,
allTestingLabels, 3)
```

		wrong_recognition_count(allTrainingImages, allTrainingLabels, allTestingImages, allTestingLabels, 4) wrong_recognition_count(allTrainingImages, allTrainingLabels, allTestingImages, allTestingLabels, 5) wrong_recognition_count(allTrainingImages, allTrainingLabels, allTestingImages, allTestingLabels, 6) wrong_recognition_count(allTrainingImages, allTrainingLabels, allTestingImages, allTestingLabels, 7) wrong_recognition_count(allTrainingImages, allTrainingLabels, allTestingImages, allTestingLabels, 8) wrong_recognition_count(allTrainingImages, allTrainingLabels, allTestingImages, allTestingLabels, 9)	

Q7*		function result = confusion_matrix(trainingImages, trainingLabels, testingImages, testingLabels)
		md = fitcknn(trainingImages', trainingLabels);
		
		% Init matrix
		result = zeros(11,11);
		for i = 0:9
		    result((i + 2), 1) = i;
		    result(1, (i + 2)) = i;
		end
		
		count = size(testingLabels, 1);
		i = 1;
		
		while (i <= count)
		    row = testingLabels(i) + 2;
		    column = recognite_from_md(md, testingImages, i) + 2;
		    result(row, column) = result(row, column) + 1;
		    i = i + 1;
		end
		end
			0 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

		% Load data và lệnh test
		% Load data
		allTrainingImages = loadMNISTImages('./train-images.idx3-ubyte');
		allTrainingLabels = loadMNISTLabels('./train-labels.idx1-ubyte');
		allTestingImages = loadMNISTImages('./t10k-images.idx3-ubyte');
		allTestingLabels = loadMNISTLabels('./t10k-labels.idx1-ubyte');
		
		confusion_matrix(allTrainingImages, allTrainingLabels, allTestingImages, allTestingLabels)	

Q8**		Tham khảo custom knn model
		https://www.mathworks.com/help/stats/classification knn-class.html
		
		% Hàm tính confusion matrix với custom model (distance, k)
		
		function result = confusion_matrix_custom_model(trainingImages, trainingLabels, testingImages, testingLabels, distance, k)
		md = fitcknn(trainingImages', trainingLabels,'NumNeighbors', k, 'Distance', distance);
		
		% Init matrix
		result = zeros(11,11);
		for i = 0:9
		  result((i + 2), 1) = i;
		  result(1, (i + 2)) = i;
		end
		
		%count = size(testingLabels, 1);
		count = 500;
		i = 1;
		
		while (i <= count)
			Do thời gian chạy khá lâu với bộ ảnh testing 10000 ảnh nên em chỉ chạy với 500 ảnh để kiểm tra tính đúng của các functions
			0 42 0 0 0 0 0 0 0 0 0
			1 0 67 0 0 0 0 0 0 0 0
			2 0 0 54 0 0 0 0 1 0 0
			3 0 0 0 44 0 0 0 1 0 0
			4 0 0 0 0 53 0 1 0 0 1
			5 0 0 0 0 0 50 0 0 0 0
			6 1 0 0 0 0 0 42 0 0 0
			7 0 0 0 0 1 0 0 48 0 0
			8 2 1 0 0 0 0 0 0 37 0
			9 0 1 0 0 0 0 0 0 2 51
			
			average_recall =
			0.9749
			
			average_precision =
			0.9752

```
    row = testingLabels(i) + 2;
    column = recognite_from_md(md, testingImages,
i) + 2;
    result(row, column) = result(row, column) + 1;
    i = i + 1;
end
end

% Hàm tính average recall và everage precision
function [average_recall, average_precision] =
evaluate(trainingImages, trainingLabels,
testingImages, testingLabels, distance, k)
cufusion_matrix =
confusion_matrix_custom_model(trainingImages,
trainingLabels, testingImages, testingLabels,
distance, k)

% average recall; average precision
sum_recall = 0;
sum_precision = 0;
label_max = 9;
for (i = 0: label_max)
    all_correct_result_count = 0;
    all_returned_result_count = 0;
    for (j = 0: label_max)
        all_correct_result_count =
all_correct_result_count + cufusion_matrix((i + 2),
(j + 2));
        all_returned_result_count =
all_returned_result_count + cufusion_matrix((j +
2),(i + 2));
    end
    returned_correct_result_count =
cufusion_matrix((i + 2), (i + 2));

    current_recall = returned_correct_result_count
/ all_correct_result_count;
    sum_recall = sum_recall + current_recall;
```

```
    current_precision =
returned_correct_result_count /
all_returned_result_count;
    sum_precision = sum_precision +
current_precision;
end

average_recall = sum_recall/(label_max + 1)
average_precision = sum_precision/(label_max + 1)
end

% Lệnh load data và test
% Load data
allTrainingImages = loadMNISTImages('./train-
images.idx3-ubyte');
allTrainingLabels = loadMNISTLabels('./train-
labels.idx1-ubyte');
allTestingImages = loadMNISTImages('./t10k-
images.idx3-ubyte');
allTestingLabels = loadMNISTLabels('./t10k-
labels.idx1-ubyte');

evaluate(allTrainingImages, allTrainingLabels,
allTestingImages, allTestingLabels, 'cosine', 3)
```