

ภาคผนวก

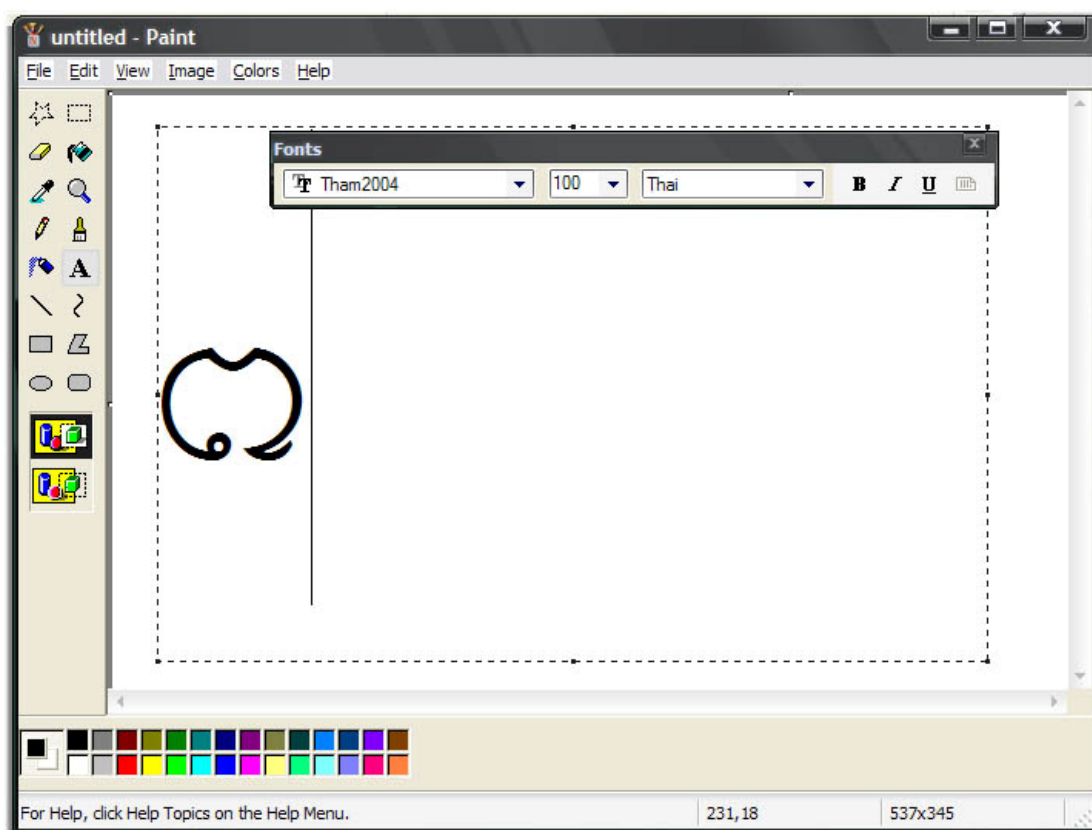
ผนวก ก

การเตรียมข้อมูลภาพอักขรธรรมอีสาน

1. พิมพ์ข้อมูลตัวอักษร

1.1 เปิดโปรแกรม Paint ขึ้นมา

1.2 พิมพ์ตัวอักขระอักษรธรรม



ภาพที่ 1 การทำภาพอักขระอักษรธรรม

1.3 จากนั้น เลือกเอาเฉพาะรูปภาพแล้วคัดลอกมาวางในโปรแกรม Photoshop ที่มีขนาดพื้นที่ 150 x 200 pixel (ดูภาพที่ 2)

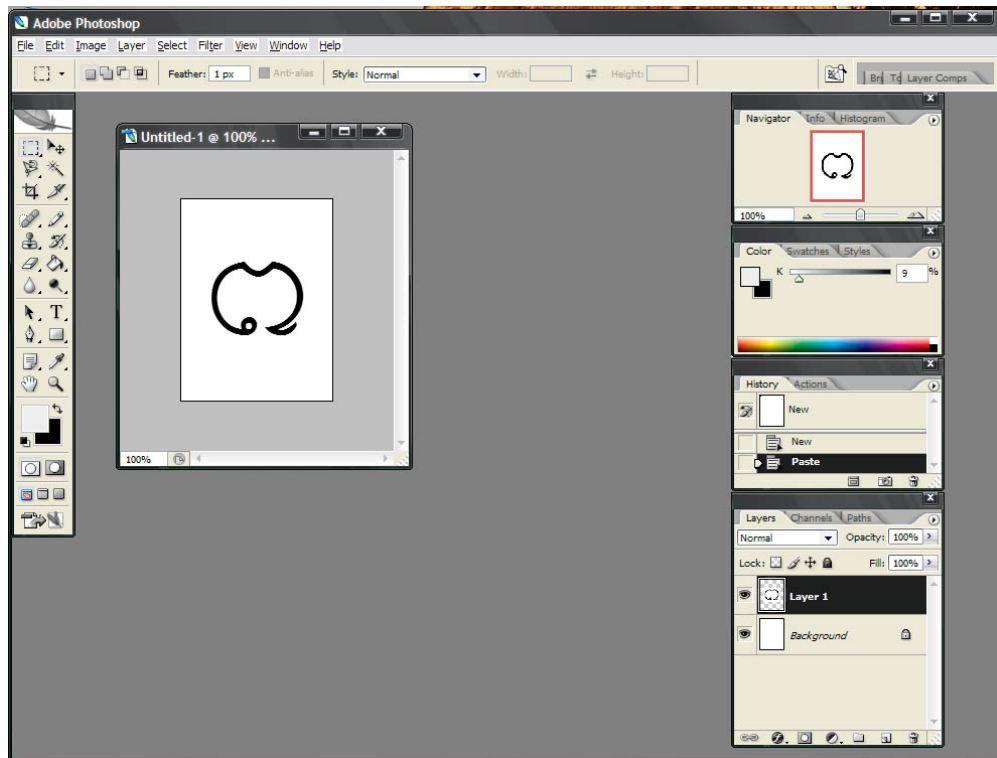
1.4 แล้วทำการบันทึกเป็นแฟ้มรูปภาพ (Bitmap Image)

2. ปรับข้อมูลภาพเป็น Black & White

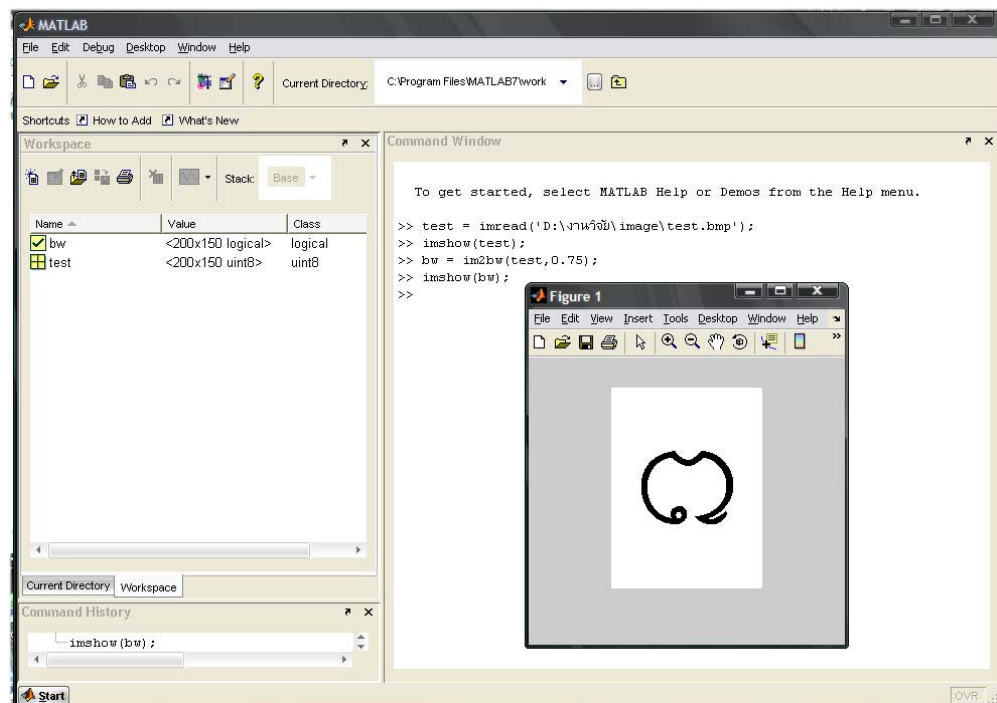
2.1 การนำข้อมูลภาพอักขระอักษรธรรมอีสานเข้าสู่กระบวนการ เตรียมข้อมูล (pre-processing) นั้น ต้องปรับข้อมูลภาพให้เป็นภาพขาวดำ (Black & White) ก่อน โดยใช้โปรแกรม MATLAB

2.2 เปิดโปรแกรม MATLAB ขึ้นมา แล้วเปิดแฟ้มรูปภาพอักขระอักษรธรรม จากนั้นสั่งให้แปลงข้อมูลภาพเป็นแบบขาวและดำ โดยกำหนดค่า Threshold เป็น 0.75 จะได้ดังภาพที่ 3

2.3 เมื่อได้ภาพอักขระเป็นภาพขาวและดำแล้ว ก็นำกลับไปบันทึกเป็นแฟ้มรูปภาพอีกต่อไป



ภาพที่ 2 การทำข้อความอักขระให้เป็นแฟ้มรูปภาพ



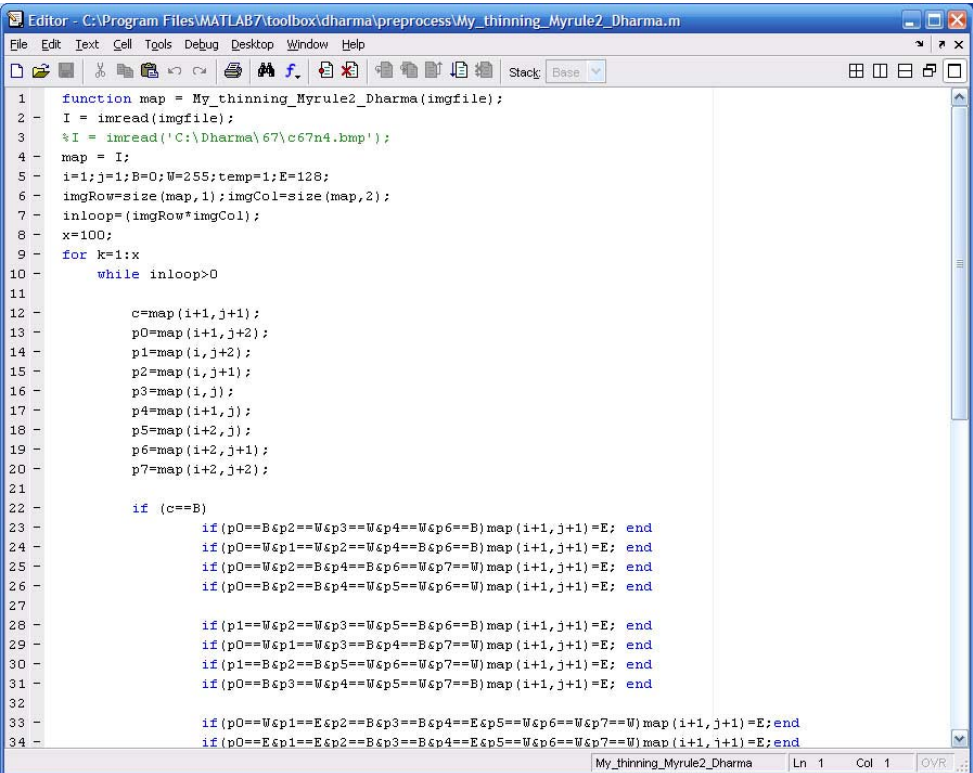
ภาพที่ 3 การปรับข้อมูลภาพเป็นขาวและดำ

การประมวลผลเบื้องต้น (Pre-Processing)

1. การลดความหนาของตัวอักษร

1.1 เขียนโปรแกรม จากขั้นตอนวิธีการประมวลผล ด้วยเครื่องมือของโปรแกรม MATLAB

1.2 จะได้โค้ดของโปรแกรมสำหรับลดความหนาของตัวอักษร ดังภาพที่ 4



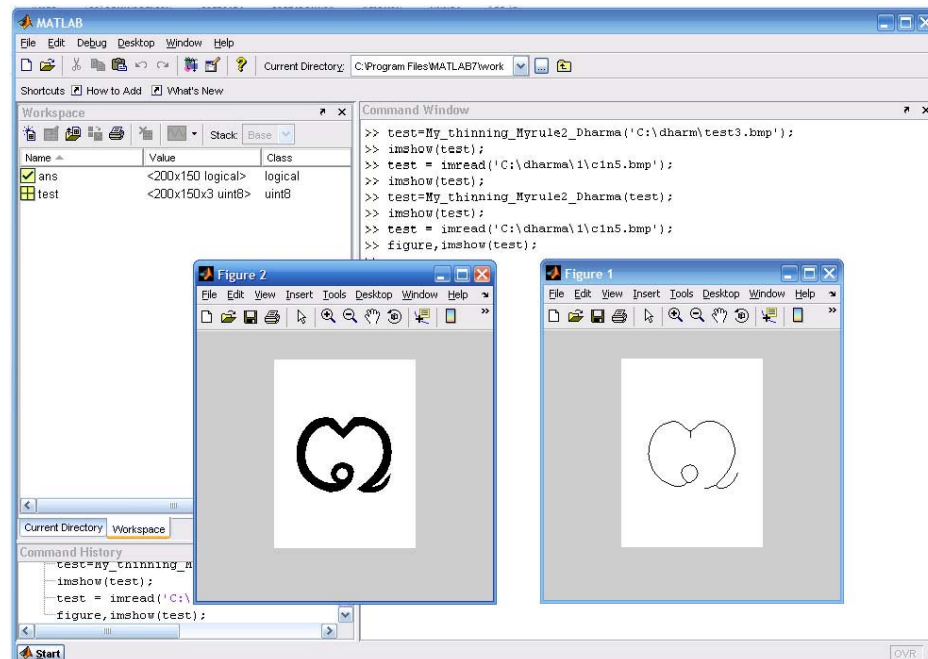
```

1 function map = My_thinning_Myrule2_Dharma(imgfile);
2 I = imread(imgfile);
3 %I = imread('C:\Dharma\67\c67n4.bmp');
4 map = I;
5 i=1;j=1;B=0;W=255;temp=1;E=128;
6 imgRow=size(map,1);imgCol=size(map,2);
7 inloop=(imgRow*imgCol);
8 x=100;
9 for k=1:x
10 while inloop>0
11
12 c=map(i+1,j+1);
13 p0=map(i+1,j+2);
14 p1=map(i,j+2);
15 p2=map(i,j+1);
16 p3=map(i,j);
17 p4=map(i+1,j);
18 p5=map(i+2,j);
19 p6=map(i+2,j+1);
20 p7=map(i+2,j+2);
21
22 if (c==B)
23 if (p0==B & p2==W & p3==W & p4==W & p6==B) map(i+1,j+1)=E; end
24 if (p0==W & p1==W & p2==W & p4==B & p6==B) map(i+1,j+1)=E; end
25 if (p0==W & p2==B & p4==B & p6==W & p7==W) map(i+1,j+1)=E; end
26 if (p0==B & p2==B & p4==W & p5==W & p6==W) map(i+1,j+1)=E; end
27
28 if (p1==W & p2==W & p3==W & p5==B & p6==B) map(i+1,j+1)=E; end
29 if (p0==W & p1==W & p3==B & p4==B & p7==W) map(i+1,j+1)=E; end
30 if (p1==B & p2==B & p5==W & p6==W & p7==W) map(i+1,j+1)=E; end
31 if (p0==B & p3==W & p4==W & p5==W & p7==B) map(i+1,j+1)=E; end
32
33 if (p0==W & p1==E & p2==B & p3==B & p4==E & p5==W & p6==W & p7==W) map(i+1,j+1)=E; end
34 if (p0==E & p1==E & p2==B & p3==B & p4==E & p5==W & p6==W & p7==W) map(i+1,j+1)=E; end
  
```

ภาพที่ 4 โค้ดของโปรแกรมสำหรับการลดความหนาของตัวอักษร

1.3 ป้อนข้อมูลอักขระอักขรธรรม เป็นข้อมูลนำเข้า ในที่นี้ทดสอบตัว ก

1.4 สั่ง run โปรแกรม จะได้ผลดังภาพที่ 5

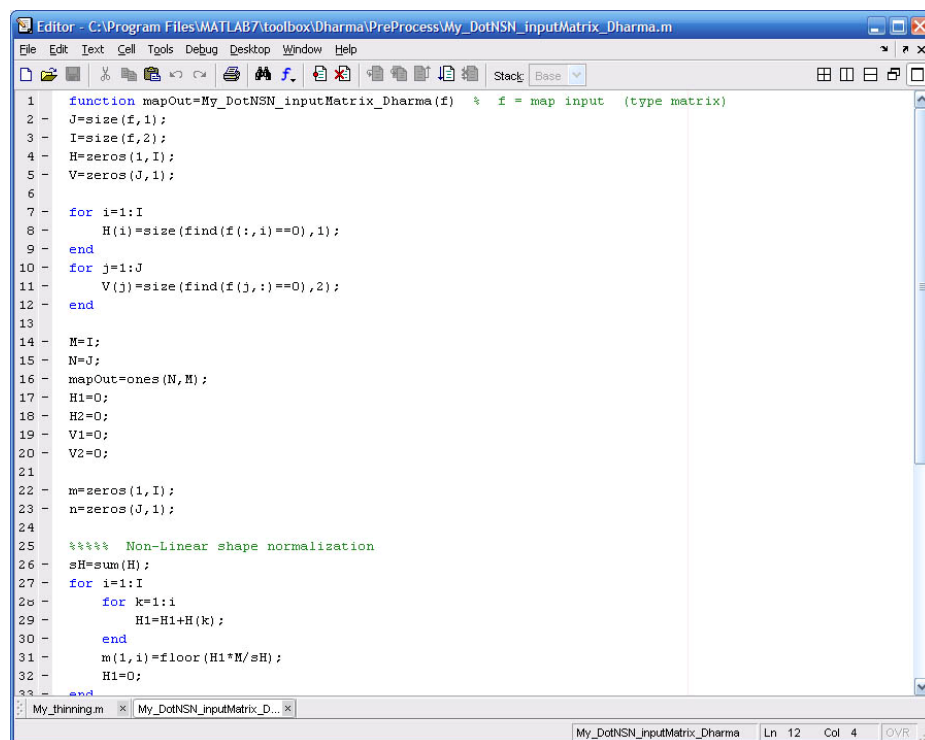


ภาพที่ 5 ผลการทำงานของโปรแกรมลดความหนาของตัวอักษร

2. การปรับการกระจายความหนาแน่นของจุด

2.1 เขียนโปรแกรม จากขั้นตอนวิธีการประมวลผล ด้วยเครื่องมือของโปรแกรม MATLAB

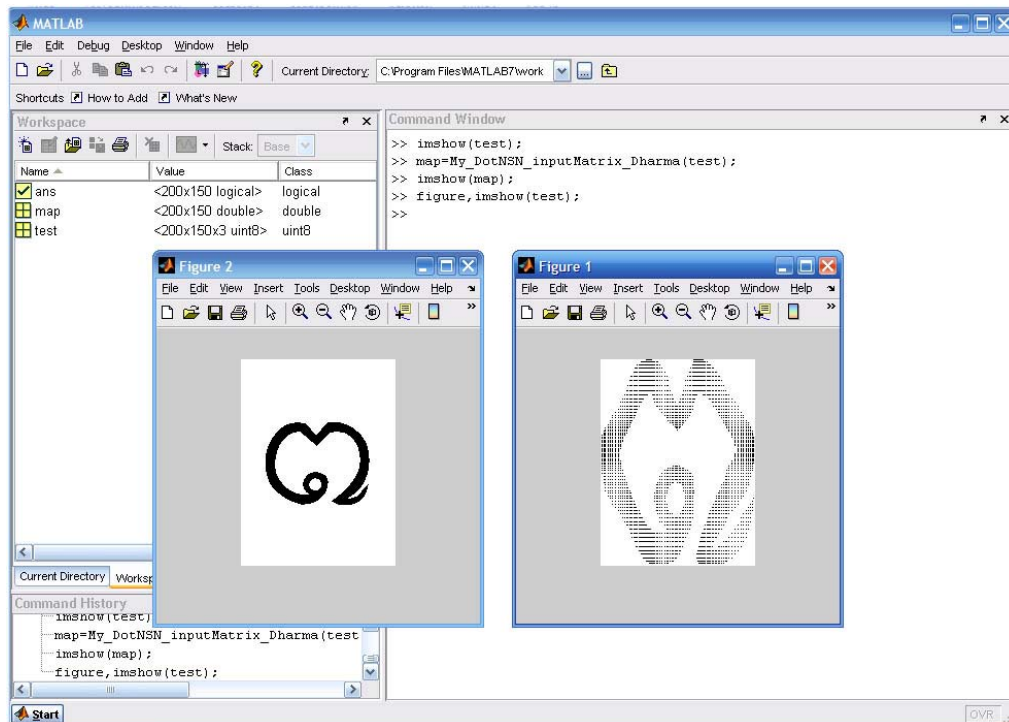
2.2 จะได้โค้ดของโปรแกรมสำหรับลดความหนาของตัวอักษร ดังภาพที่ 6



ภาพที่ 6 โค้ดของโปรแกรมสำหรับปรับการกระจายความหนาแน่นของจุด

2.3 ป้อนข้อมูลอักขระอักษรธรรม เป็นข้อมูลนำเข้า ในที่นี้ทดสอบตัว ก

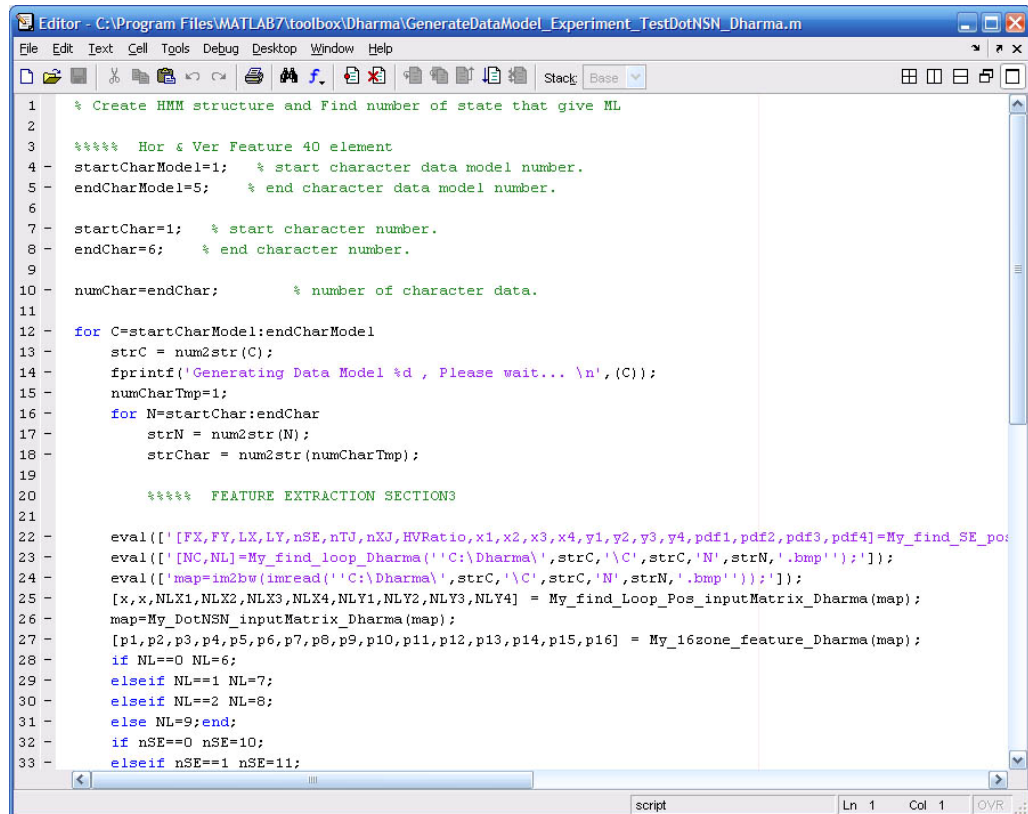
2.4 ตั้ง run โปรแกรม จะได้ผลดังภาพที่ 7



ภาพที่ 7 ผลการทำงานของโปรแกรมปรับการกระจาย

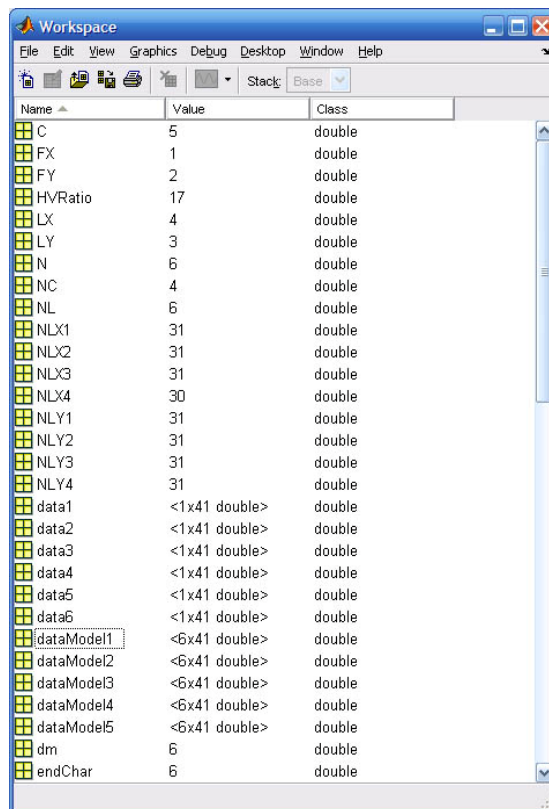
การสกัดลักษณะเฉพาะ (Feature Extraction)

1. เขียน โปรแกรมเพื่อสกัดเอาคุณลักษณะเฉพาะของอักขรกรรมอีสานออกมา ดังตัวอย่างในภาพที่ 8



ภาพที่ 8 โค้ดของโปรแกรมการสกัดคุณลักษณะเฉพาะของอักขรกรรมอีสาน

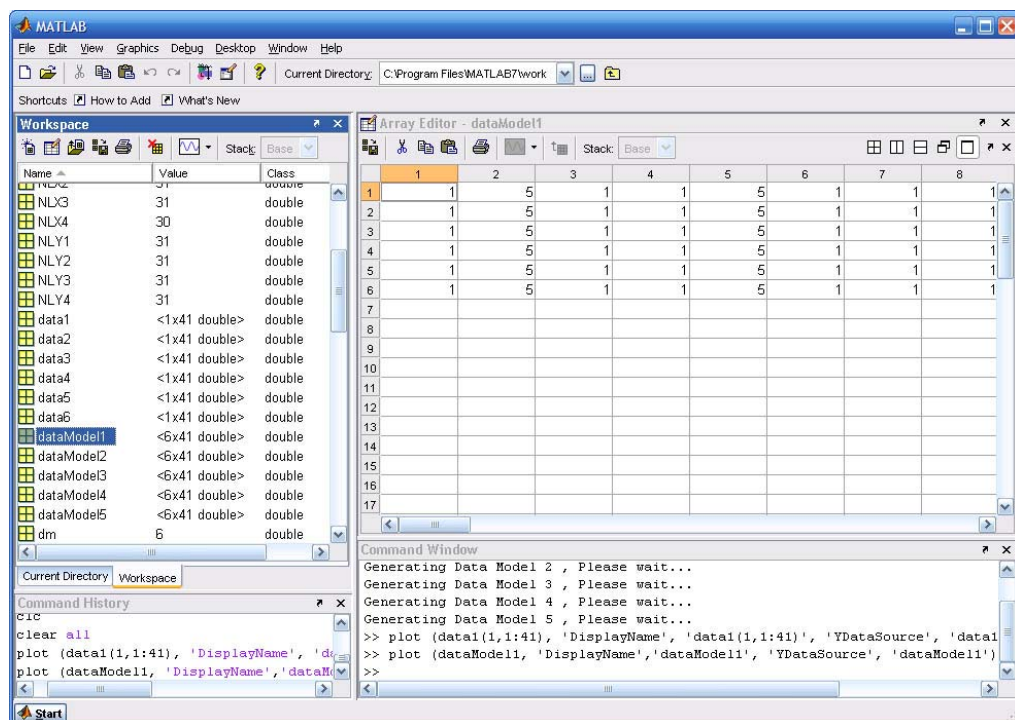
2. โปรแกรมนี้ จะทำการอ่านข้อมูลภาพอักขระอักขรกรรมอีสาน ทุกแบบทุกตัว แล้วเข้าสู่กระบวนการประมวลผลเบื้องต้น จนสุดท้ายจะได้ผลลัพธ์ออกมาเป็นข้อมูลก้อนหนึ่ง เรียกว่า Feature Vectors ซึ่งเป็นอาร์เรย์ขนาด 1x41 ดังนี้ [x1, x2, x3, x4, y1, y2, y3, y4, NL, nSE, HVRatio, nTJ, nXJ, pdf1, pdf2, pdf3, pdf4, NLX1, NLX2, NLX3, NLX4, NLY1, NLY2, NLY3, NLY4, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15, p16] (ดูภาพที่ 9)
3. นอกจากจะได้ Feature Vectors แล้ว ยังได้ข้อมูลตัวแบบ (Data Model) จำนวน 91 ตัว (ดูภาพที่ 10) สำหรับนำไปเข้ากระบวนการรู้จำอีกต่อไป



The image shows the MATLAB Workspace window. It contains a table with three columns: Name, Value, and Class. The variables listed are C, FX, FY, HVRatio, LX, LY, N, NC, NL, NLX1, NLX2, NLX3, NLX4, NLY1, NLY2, NLY3, NLY4, data1, data2, data3, data4, data5, data6, dataModel1, dataModel2, dataModel3, dataModel4, dataModel5, dm, and endChar. Most variables are of type 'double'.

Name	Value	Class
C	5	double
FX	1	double
FY	2	double
HVRatio	17	double
LX	4	double
LY	3	double
N	6	double
NC	4	double
NL	6	double
NLX1	31	double
NLX2	31	double
NLX3	31	double
NLX4	30	double
NLY1	31	double
NLY2	31	double
NLY3	31	double
NLY4	31	double
data1	<1x41 double>	double
data2	<1x41 double>	double
data3	<1x41 double>	double
data4	<1x41 double>	double
data5	<1x41 double>	double
data6	<1x41 double>	double
dataModel1	<6x41 double>	double
dataModel2	<6x41 double>	double
dataModel3	<6x41 double>	double
dataModel4	<6x41 double>	double
dataModel5	<6x41 double>	double
dm	6	double
endChar	6	double

ภาพที่ 9 แสดง Feature Vectors



The image shows the MATLAB interface. The top part displays the Workspace window with a list of variables. The middle part shows the Array Editor for 'dataModel1', which is a 17x8 matrix. The bottom part shows the Command Window with the following commands and output:

```

clear all
plot (data1(1,1:41), 'DisplayName', 'data1')
plot (dataModel1, 'DisplayName', 'dataModel1')

```

The Command Window output shows the following messages:

```

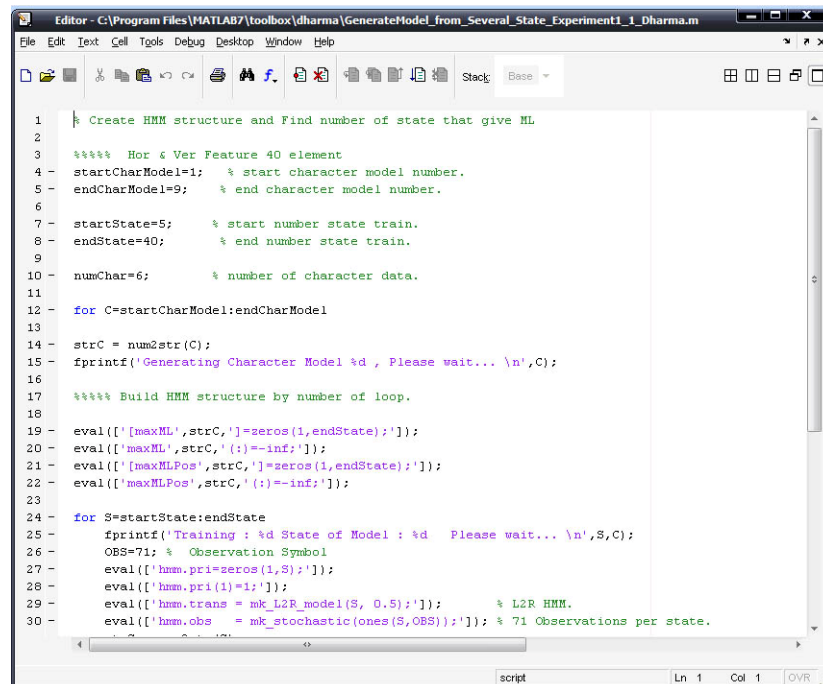
Generating Data Model 2 , Please wait...
Generating Data Model 3 , Please wait...
Generating Data Model 4 , Please wait...
Generating Data Model 5 , Please wait...
>> plot (data1(1,1:41), 'DisplayName', 'data1(1,1:41)', 'YDataSource', 'data1')
>> plot (dataModel1, 'DisplayName', 'dataModel1', 'YDataSource', 'dataModel1')
>>

```

ภาพที่ 10 แสดงผลลัพธ์ที่ได้ จะเป็น Data Model ตัวที่ 1 ถึง 91

การสอนตัวแบบเพื่อการเรียนรู้จำตัวอักษร

1. สร้างตัวแบบฮิดเดนมาร์คอฟ จะได้โปรแกรมดังภาพที่ 11
2. หลังจากนั้นสั่ง run โปรแกรม จะได้ผลดังภาพที่ 12

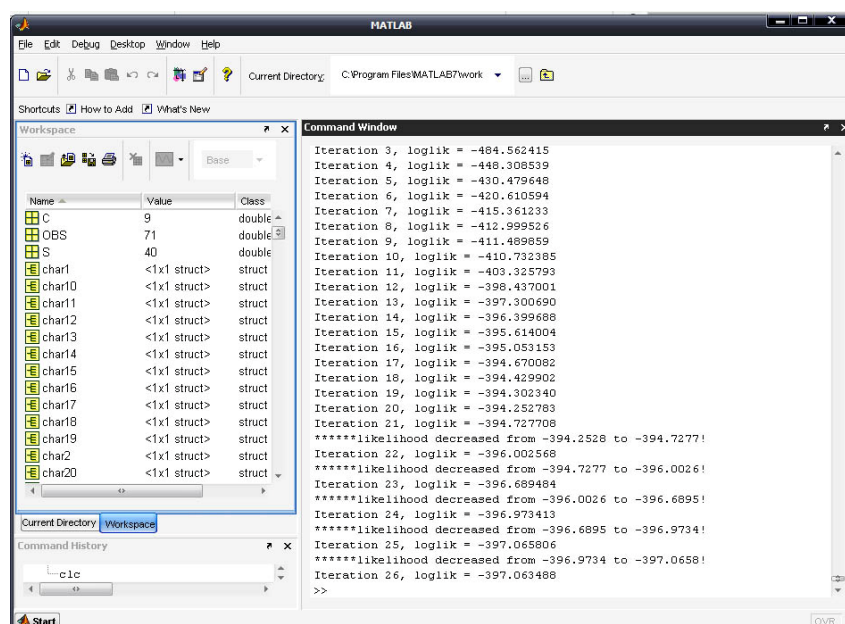


```

1  % Create HMM structure and Find number of state that give ML
2
3  % Hor & Ver Feature 40 element
4  startCharModel=1; % start character model number.
5  endCharModel=9; % end character model number.
6
7  startState=5; % start number state train.
8  endState=40; % end number state train.
9
10 numChar=6; % number of character data.
11
12 for C=startCharModel:endCharModel
13
14     strC = num2str(C);
15     fprintf('Generating Character Model %d , Please wait... \n',C);
16
17     % Build HMM structure by number of loop.
18
19     eval(['maxML',strC,'']=zeros(1,endState));
20     eval(['maxML',strC,'{:-inf}']);
21     eval(['maxMLPos',strC,'']=zeros(1,endState));
22     eval(['maxMLPos',strC,'{:-inf}']);
23
24     for S=startState:endState
25         fprintf('Training : %d State of Model : %d Please wait... \n',S,C);
26         OBS=71; % Observation Symbol
27         eval(['hmm.pri=zeros(1,S)']);
28         eval(['hmm.pri(1)=1']);
29         eval(['hmm.trans = mk_L2R_model(S, 0.5)']); % L2R HMM.
30         eval(['hmm.obs = mk_stochastic(ones(S,OBS))']); % 71 Observations per state.

```

ภาพที่ 11 แสดงโค้ดของโปรแกรมสำหรับการสร้างตัวแบบฮิดเดนมาร์คอฟ



```

Iteration 3, loglik = -484.562415
Iteration 4, loglik = -448.308539
Iteration 5, loglik = -430.479648
Iteration 6, loglik = -420.610594
Iteration 7, loglik = -415.361233
Iteration 8, loglik = -412.989526
Iteration 9, loglik = -411.489859
Iteration 10, loglik = -410.732385
Iteration 11, loglik = -403.325793
Iteration 12, loglik = -398.437001
Iteration 13, loglik = -397.300690
Iteration 14, loglik = -396.399688
Iteration 15, loglik = -395.614004
Iteration 16, loglik = -395.053153
Iteration 17, loglik = -394.670082
Iteration 18, loglik = -394.429902
Iteration 19, loglik = -394.302340
Iteration 20, loglik = -394.252783
Iteration 21, loglik = -394.727708
*****likelihood decreased from -394.2528 to -394.7277!
Iteration 22, loglik = -396.002568
*****likelihood decreased from -394.7277 to -396.0026!
Iteration 23, loglik = -396.689484
*****likelihood decreased from -396.0026 to -396.6895!
Iteration 24, loglik = -396.973413
*****likelihood decreased from -396.6895 to -396.9734!
Iteration 25, loglik = -397.065806
*****likelihood decreased from -396.9734 to -397.0658!
Iteration 26, loglik = -397.063488
>>

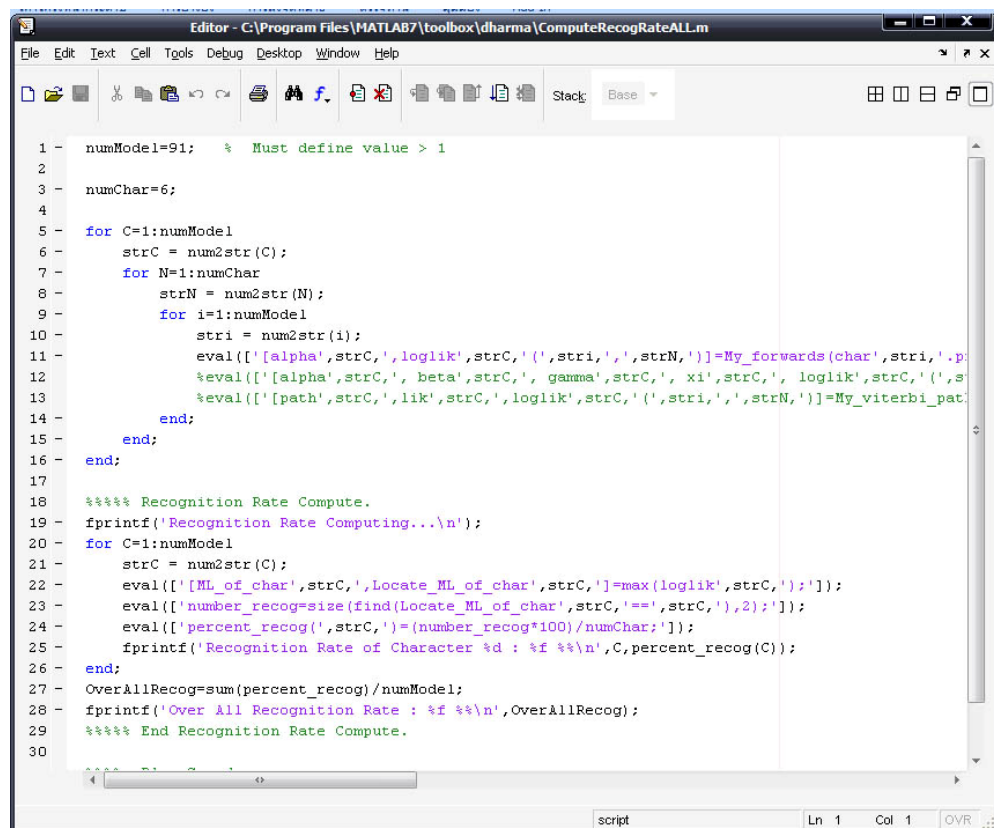
```

ภาพที่ 12 แสดงผลลัพธ์ของการสร้างตัวแบบฮิดเดนมาร์คอฟ

คำนวณหาผลการรู้จำ

1. หลังจากทำการสร้างตัวแบบฮิดเดนมาร์คอฟแล้ว ระบบจะทำการจดจำลักษณะของอักขระอักษรธรรมอีสาน แล้วจึงนำไปคำนวณหาว่า ประสิทธิภาพของการรู้จำเป็นเท่าใด ดังตัวอย่างโปรแกรมในภาพที่

13



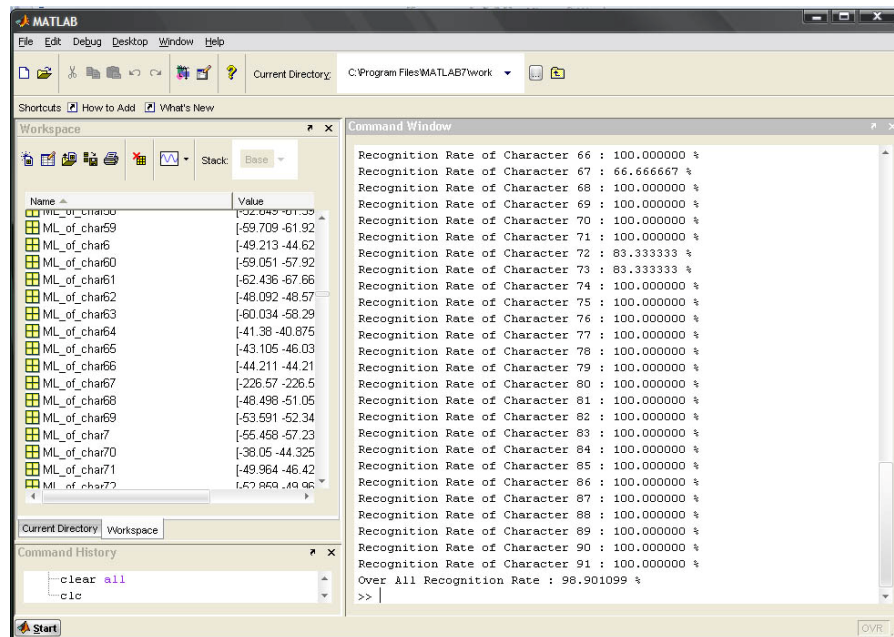
```

1 - numModel=91; % Must define value > 1
2 -
3 - numChar=6;
4 -
5 - for C=1:numModel
6 -     strC = num2str(C);
7 -     for N=1:numChar
8 -         strN = num2str(N);
9 -         for i=1:numModel
10 -            stri = num2str(i);
11 -            eval(['[alpha',strC,',loglik',strC,',' ,stri,',',strN,']]=My_forwards(char',stri,',' ,strC,',' ,strN,']');
12 -            %eval(['[alpha',strC,', beta',strC,', gamma',strC,', xi',strC,', loglik',strC,',' ,stri,',',strN,']]=My_viterbi_pat
13 -            %eval(['[path',strC,',lik',strC,',loglik',strC,',' ,stri,',',strN,']]=My_viterbi_pat
14 -        end;
15 -    end;
16 - end;
17 -
18 - %*** Recognition Rate Compute.
19 - fprintf('Recognition Rate Computing...\n');
20 - for C=1:numModel
21 -     strC = num2str(C);
22 -     eval(['[ML_of_char',strC,',Locate_ML_of_char',strC,']]=max(loglik',strC,')');
23 -     eval(['number_recog=size(find(Locate_ML_of_char',strC,')==,strC,')/2;']);
24 -     eval(['percent_recog(' ,strC,')=(number_recog*100)/numChar;']);
25 -     fprintf('Recognition Rate of Character %d : %f %%\n',C,percent_recog(C));
26 - end;
27 - OverAllRecog=sum(percent_recog)/numModel;
28 - fprintf('Over All Recognition Rate : %f %%\n',OverAllRecog);
29 - %*** End Recognition Rate Compute.
30 -

```

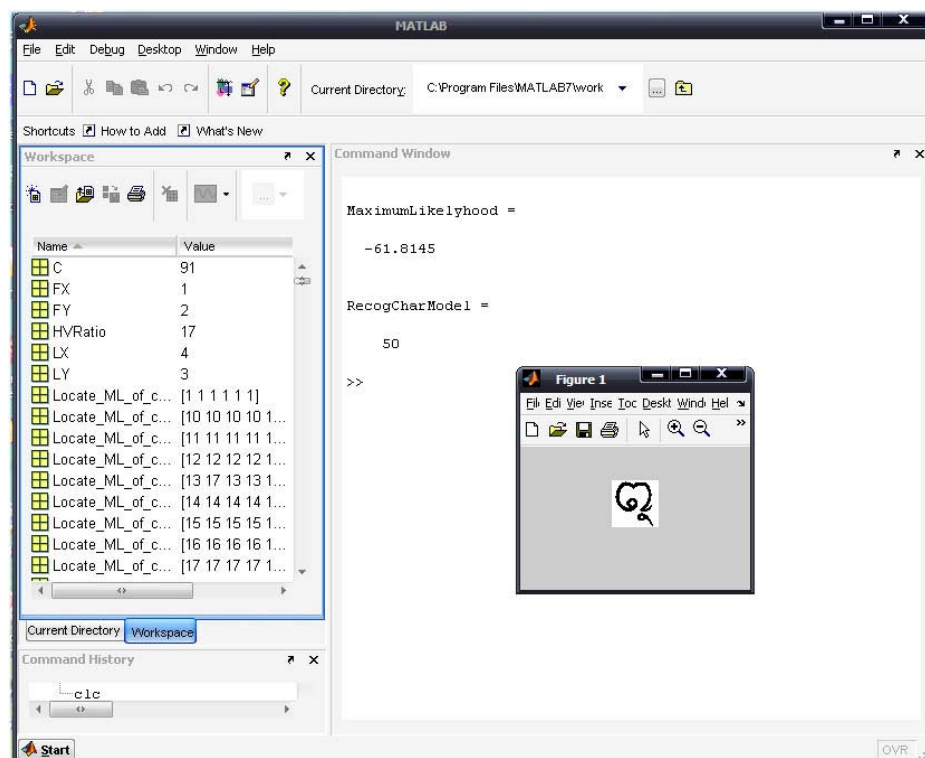
ภาพที่ 13 แสดงโค้ดโปรแกรมการคำนวณหาประสิทธิภาพการรู้จำ

2. หลังจากนั้น run โปรแกรม เครื่องจะทำการคำนวณอัตราการรู้จำเป็นร้อยละ ดังตัวอย่างในภาพที่ 14



ภาพที่ 14 แสดงผลลัพธ์ที่ได้จากการหาประสิทธิภาพการรู้จำ

3. สุดท้ายทดสอบการรู้จำอักขระอักษรธรรม โดยป้อนข้อมูลนำเข้าเป็นอักขระอักษรธรรมตัวใดๆ โปรแกรมจะนำตัวอักษรนั้น ไปคำนวณหาค่า Feature แล้วพิจารณาค่าความน่าจะเป็นที่มากที่สุด แล้วแสดงผลพ้อออกมา ดังตัวอย่างในภาพที่ 15



ภาพที่ 15 แสดงผลการทดสอบการรู้จำ เมื่อนำข้อมูลอักขระลำดับที่ 50 มาทดสอบ