

CSC2561/24

เอกสารคู่มือการใช้งาน

(User Manual)

การตรวจจับการจอดรถในพื้นที่ห้ามจอดโดยวิธีการแตกเชิงความหมาย
Prohibited Car Parking Area Detection Using Semantic Segmentation
Approach

โดย

รหัสนักศึกษา 583020719-3 นายธีรไนย แซ่ลิ้ม

รหัสนักศึกษา 583020734-7 นายรัชชานนท์ เคนชมภู

อาจารย์ที่ปรึกษา : ผศ.ดร.ปัญญาพล หอระตะ

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 322 499
โครงการวิจัยทางวิทยาการคอมพิวเตอร์ ระดับปริญญาตรี 2
สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยขอนแก่น
(เดือน พฤษภาคม พ.ศ. 2562)

คำนำ

เอกสารนี้เป็นส่วนหนึ่งของรายวิชา 322 499 โครงการวิจัยทางวิทยาการคอมพิวเตอร์ ระดับปริญญาตรี 2 สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น โดยรายวิชา 322 499 โครงการคอมพิวเตอร์ 2 ได้มอบหมายให้นักศึกษาทำโครงการ โดยให้นักศึกษาศึกษาและนำความรู้ความสามารถที่ได้ศึกษานำมาใช้ให้เกิดประโยชน์ ก่อนที่จะเริ่มจัดทำบทความวิจัยเรื่องการตรวจจับการจอดรถในพื้นที่ห้ามจอดโดยวิธีการตรวจจับความหมายต้องปรึกษากับอาจารย์ที่ปรึกษาเพื่อให้ได้ความถูกต้องได้มากที่สุด จากนั้นนำความรู้ทางด้าน MATLAB มาใช้ให้เกิดประโยชน์และนำความรู้ทักษะมาต่อยอดให้มีประสิทธิภาพยิ่งขึ้น

ในการจัดทำโครงการครั้งนี้ ผู้จัดทำหวังเป็นอย่างยิ่งว่าโครงการชิ้นนี้จะเป็นประโยชน์และแนวทางที่ดี ให้แก่ผู้ที่สนใจสามารถนำไปศึกษาหรือพัฒนาต่อไปเป็นองค์ความรู้เพื่อเพิ่มประสิทธิภาพการตรวจจับความหมายให้ดียิ่งขึ้น

คณะผู้จัดทำ

นายรัชชานนท์ เคนชมภู

นายธีรไนย แซ่ลิ้ม

สารบัญ

เรื่อง	หน้า
คำนำ	ก
สารบัญภาพ	ค
สารบัญภาพ (ต่อ)	ง
1. การฝึกสอน (Training)	1
1.1 ผังขั้นตอนการฝึกสอน	1
1.2 ขั้นตอนการฝึกสอน	2
2.) การทำงานของระบบ	10
2.1) ผังขั้นตอนการทำงานของระบบ	10
2.2) การทำงานตัวโปรแกรม	10

สารบัญภาพ

เรื่อง	หน้า
ภาพที่ 1 ภาพตัวอย่างผังขั้นตอนการฝึกสอนตัวแบบ	1
ภาพที่ 2 ภาพตัวอย่าง Add-on VGG-16 Network.....	2
ภาพที่ 3 ภาพตัวอย่างการรันรหัสโปรแกรม vgg16()	2
ภาพที่ 4 ภาพตัวอย่างรหัสโปรแกรมการกำหนดที่อยู่ไฟล์ข้อมูลนำเข้า.....	2
ภาพที่ 5 ภาพตัวอย่างข้อมูลนำเข้ารูปภาพ.....	3
ภาพที่ 6 ภาพตัวอย่างข้อมูลนำเข้าภาพกำกับป้าย.....	3
ภาพที่ 7 ภาพตัวอย่างรหัสโปรแกรมการบันทึกข้อมูลภาพ.....	4
ภาพที่ 8 ภาพตัวอย่างข้อมูลที่ถูกเก็บโดย ฟังก์ชัน imageDatastore	4
ภาพที่ 9 ภาพตัวอย่างรหัสโปรแกรมกำหนดคลาส	4
ภาพที่ 10 ภาพตัวอย่างผลลัพธ์การกำหนดคลาส	5
ภาพที่ 11 ภาพตัวอย่างรหัสสี RGB ที่กำหนดให้คลาส	5
ภาพที่ 12 ภาพตัวอย่างรหัสโปรแกรมกำหนดค่าป้ายกำกับ	5
ภาพที่ 13 ภาพตัวอย่างรหัสโปรแกรมการบันทึกข้อมูลภาพ	6
ภาพที่ 14 ภาพตัวอย่างรหัสโปรแกรมปรับขนาดรูปภาพ	6
ภาพที่ 15 ภาพตัวอย่างเตรียมข้อมูลฝึกสอนและชุดข้อมูลทดสอบ.....	7
ภาพที่ 16 ภาพตัวอย่างผลการนับจำนวนภาพฝึกสอนและทดสอบ.....	7
ภาพที่ 17 ภาพตัวอย่างรหัสโปรแกรมการสร้างเครือข่าย.....	7
ภาพที่ 18 ภาพตัวอย่างรหัสโปรแกรมการปรับค่าน้ำหนักคลาส.....	7
ภาพที่ 19 ภาพตัวอย่างผลการปรับค่าน้ำหนักแต่ละคลาส	7
ภาพที่ 20 ภาพตัวอย่างรหัสโปรแกรมการเพิ่มข้อมูลการฝึกสอน	7
ภาพที่ 21 ภาพตัวอย่าง code การตั้งค่าการฝึกสอน.....	8
ภาพที่ 22 ภาพตัวอย่างรหัสโปรแกรมการฝึกสอนตัวแบบ	8
ภาพที่ 23 ภาพตัวอย่างการฝึกสอน.....	9
ภาพที่ 24 ภาพตัวอย่างรหัสโปรแกรม semantic segmentation	9
ภาพที่ 25 ภาพตัวอย่างผลลัพธ์การสกัดวัตถุในรูปภาพ จากการฝึกสอนจากตัวแบบ	9
ภาพที่ 26 ภาพตัวอย่างผังขั้นตอนการทำงานของระบบตรวจจับการจอดรถในพื้นที่ห้ามจอด	10
ภาพที่ 27 ภาพตัวอย่างแพ็คเกจ IP Cameras และ ANDROID MOBILE CAMERA CONNECTOR	10
ภาพที่ 28 ภาพตัวอย่างแอปพลิเคชัน IP Webcam.....	11
ภาพที่ 29 ภาพตัวอย่างหน้าจอเมนูแอปพลิเคชัน IP Webcam.....	11
ภาพที่ 30 ภาพตัวอย่างการสตรีมวิดีโอที่ส่งจากแอปพลิเคชัน IP Webcam	12

สารบัญภาพ (ต่อ)

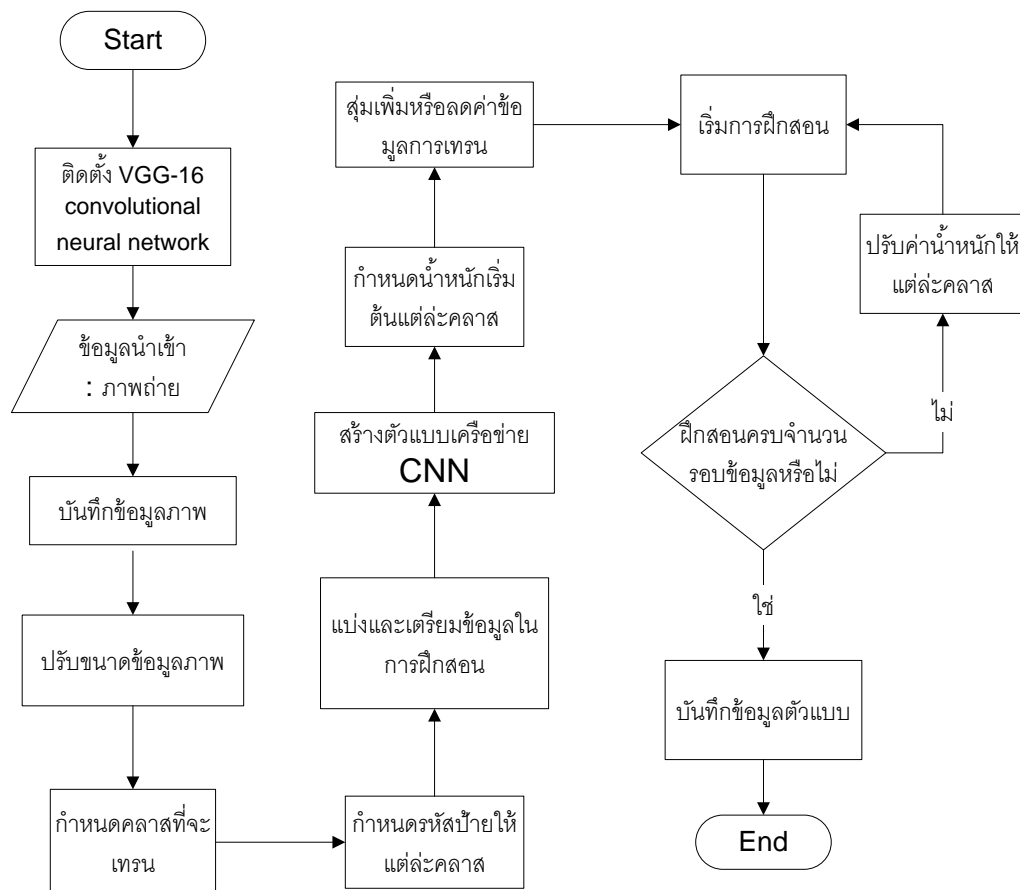
เรื่อง	หน้า
ภาพที่ 31 ภาพตัวอย่างรหัสโปรแกรมตัวแปร url ในไฟล์ FinalDemo.m	12
ภาพที่ 32 ภาพตัวอย่างรหัสโปรแกรมการกำหนดขอบหน้าต่างการคำนวณในไฟล์ FinalDemo.m.....	13
ภาพที่ 33 ภาพตัวอย่างผลการทำงานของโปรแกรม	13

จากการศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง สามารถวิเคราะห์และออกแบบระบบตรวจจับการจอดรถในพื้นที่ห้ามจอดได้ข้อส่วนใหญ่ๆ ดังนี้

1. การฝึกสอน (Training)

ในโครงการนี้ใช้ตัวแบบ VGG-16 CNN และ โปรแกรม Matlab R2018a ในการฝึกสอน

1.1 ผังขั้นตอนการฝึกสอน

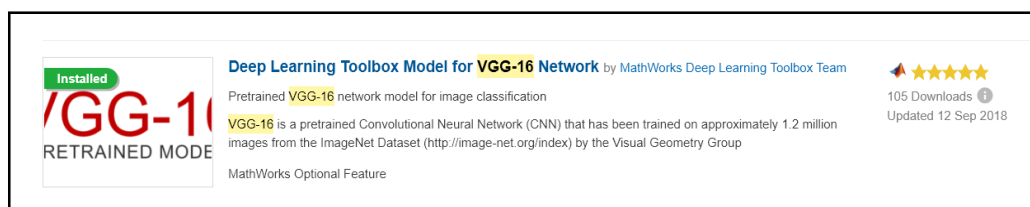


ภาพที่ 1 ภาพตัวอย่างผังขั้นตอนการฝึกสอนตัวแบบ

1.2 ขั้นตอนการฝึกสอน

1. ติดตั้ง VGG-16 convolutional neural network

1.1 ดาวน์โหลด Add-on VGG-16 Network ในโปรแกรม matlab



ภาพที่ 2 ภาพตัวอย่าง Add-on VGG-16 Network

1.2 พิมพ์ vgg16() ใน Command Window เพื่อเริ่มการติดตั้ง

```
>> vgg16()

ans =

SeriesNetwork with properties:

Layers: [41x1 nnet.cnn.layer.Layer]
```

ภาพที่ 3 ภาพตัวอย่างการรันรหัสโปรแกรม vgg16()

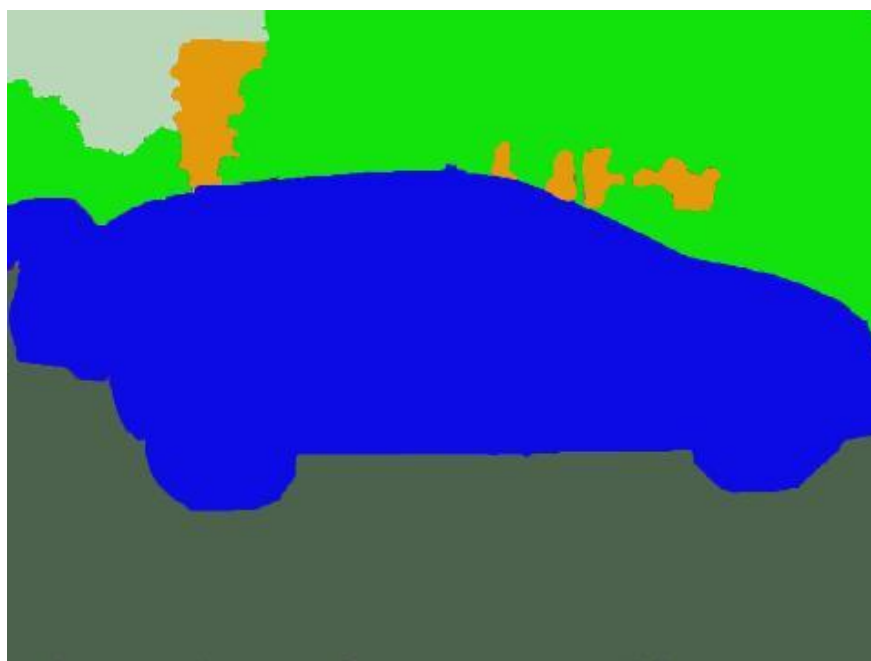
2. กำหนดที่อยู่ไฟล์ข้อมูลนำเข้า

```
outputFolder = fullfile('D:\MATLAB\R2017a\CamVid'); %กำหนดที่อยู่ไฟล์รูปภาพทั้งหมด
imgDir = fullfile('D:\MATLAB\R2017a\CamVid\imraw'); %กำหนดที่อยู่รูปภาพ
labelDir = fullfile('D:\MATLAB\R2017a\CamVid\imlabel'); %กำหนดที่อยู่ภาพ label
labelFolder = fullfile('D:\MATLAB\R2017a\CamVid\imlabel'); %กำหนดที่อยู่ Folder ของ ภาพ label
```

ภาพที่ 4 ภาพตัวอย่างรหัสโปรแกรมการกำหนดที่อยู่ไฟล์ข้อมูลนำเข้า



ภาพที่ 5 ภาพตัวอย่างข้อมูลนำเข้ารูปภาพ



ภาพที่ 6 ภาพตัวอย่างข้อมูลนำเข้าภาพกำกับป้าย

3. บันทึกข้อมูลภาพ

```
imds = imageDatastore(imgDir); %เก็บข้อมูลรูปภาพเป็นชุดข้อมูล
```

ภาพที่ 7 ภาพตัวอย่างรหัสโปรแกรมการบันทึกข้อมูลภาพ

imds.Files	
	1
1	D:\MATLAB\R2017a\CamVid\imraw\R1.jpg
2	D:\MATLAB\R2017a\CamVid\imraw\R10.jpg
3	D:\MATLAB\R2017a\CamVid\imraw\R11.jpg
4	D:\MATLAB\R2017a\CamVid\imraw\R12.jpg
5	D:\MATLAB\R2017a\CamVid\imraw\R13.jpg
6	D:\MATLAB\R2017a\CamVid\imraw\R14.jpg
7	D:\MATLAB\R2017a\CamVid\imraw\R15.jpg
8	D:\MATLAB\R2017a\CamVid\imraw\R2.jpg
9	D:\MATLAB\R2017a\CamVid\imraw\R3.jpg
10	D:\MATLAB\R2017a\CamVid\imraw\R4.jpg
11	D:\MATLAB\R2017a\CamVid\imraw\R5.jpg
12	D:\MATLAB\R2017a\CamVid\imraw\R6.jpg
13	D:\MATLAB\R2017a\CamVid\imraw\R7.jpg
14	D:\MATLAB\R2017a\CamVid\imraw\R8.jpg
15	D:\MATLAB\R2017a\CamVid\imraw\R9.jpg

ภาพที่ 8 ภาพตัวอย่างข้อมูลที่ถูกรวบรวมโดย ฟังก์ชัน imageDatastore

4. กำหนดคลาส และ ใส่รหัส RGB แต่ละคลาส

4.1 กำหนดคลาส

```
%ประกาศคลาส
classes = [
    "Sky"
    "Building"
    "Tree"
    "Car"
    "Road"
    "Black-White"
];
```

ภาพที่ 9 ภาพตัวอย่างรหัสโปรแกรมกำหนดคลาส

ใน prototype นี้ ผู้ทำโครงงานได้กำหนดคลาสตามภาพตัวอย่างรหัสโปรแกรมดังนี้

1.ท้องฟ้า, 2.ตึก, 3.ต้นไม้, 4.รถ, 5.ถนน, 6.เส้นขาว-ดำ

	1	
1	Sky	
2	Building	
3	Tree	
4	Car	
5	Road	
6	Black-White	
7		

ภาพที่ 10 ภาพตัวอย่างผลลัพธ์การกำหนดคลาส

4.2 กำหนดรหัสสี RGB ให้คลาส โดยใช้ฟังก์ชัน `cmap = HelperFunctions.camvidColorMap()`

ผู้ทำโครงการกำหนด รหัสสี RGB ให้คลาสในฟังก์ชันดังนี้

```
cmap = [
    185 216 185    % ท้องฟ้า
    229 154 11     % ดึก
    17 228 11      % ต้นไม้
    11 11 229      % รถ
    76 98 77       % ถนน
    0 0 0          % ขาว-ดำ
];
```

ภาพที่ 11 ภาพตัวอย่างรหัสสี RGB ที่กำหนดให้คลาส

5. กำหนด label ids

```
labelIDs = HelperFunctions.camvidPixelLabelIDs(); %กำหนดค่า id ของ label
```

ภาพที่ 12 ภาพตัวอย่างรหัสโปรแกรมกำหนดค่าป้ายกำกับ

ทำงานของฟังก์ชันคล้ายคลึงกับ `cmap = HelperFunctions.camvidColorMap()` เพียงแต่เปลี่ยนตัวแปรในการเก็บข้อมูล

6. บันทึกข้อมูลภาพกำกับ

```
pxds = pixelLabelDatastore(labelDir, classes, labelIDs);
```

ภาพที่ 13 ภาพตัวอย่างรหัสโปรแกรมการบันทึกข้อมูลภาพ

การทำงานของฟังก์ชันนี้คล้ายคลึงกับ imageDatastore

7. ปรับขนาดภาพ

```
HelperFunctions.prepareData(imgDir, labelDir); %ปรับขนาดรูปภาพ กับ รูปภาพ label
```

ภาพที่ 14 ภาพตัวอย่างรหัสโปรแกรมปรับขนาดรูปภาพ

ฟังก์ชันนี้จะปรับขนาดภาพ เป็น 360 x 480

8. เตรียมข้อมูลฝึกสอนและชุดข้อมูลการทดสอบ

```
[imdsTrain, imdsTest, pxdsTrain, pxdsTest] = HelperFunctions.partitionCamVidData(imds, pxds, labelIDs);
```

ภาพที่ 15 ภาพตัวอย่างเตรียมข้อมูลฝึกสอนและชุดข้อมูลทดสอบ

ฟังก์ชันนี้จะแบ่งชุดข้อมูลไว้สำหรับการฝึกสอน 60% และ แบ่งไว้สำหรับการ Test 40%

ในตัวอย่างนี้มีภาพทั้งหมด 15 ภาพ จึงสามารถแบ่งภาพได้เป็น ภาพที่ใช้ฝึกสอน 9 ภาพ ภาพที่ใช้ test 6 ภาพ

```
numTrainingImages =  
9  
  
numTestingImages =  
6
```

ภาพที่ 16 ภาพตัวอย่างผลการนับจำนวนภาพฝึกสอนและทดสอบ

9. สร้างตัวแบบเครือข่าย

```
imageSize = [360 480 3]; %กำหนดขนาดรูปภาพ  
numClasses = numel(classes); %นับจำนวนคลาส  
% segnetLayers returns SegNet network layers, lgraph, 1  
lgraph = segnetLayers(imageSize, numClasses, 'vgg16');
```

ภาพที่ 17 ภาพตัวอย่างรหัสโปรแกรมการสร้างเครือข่าย

segnetLayers เป็นฟังก์ชันสร้างเครือข่าย โดยใช้ น้ำหนักเริ่มต้น ของตัวแบบ vgg16 และทำการแปลงค่า น้ำหนักอัตโนมัติ และ เพิ่มเลเยอร์ที่จำเป็นในการสร้างตัวแบบ

10. ปรับค่าน้ำหนักคลาส

```
imageFreq = tbl.PixelCount ./ tbl.ImagePixelCount;
classWeights = median(imageFreq) ./ imageFreq
```

ภาพที่ 18 ภาพตัวอย่างรหัสโปรแกรมการปรับค่าน้ำหนักคลาส

```
classWeights =

    12.1241
     0.8027
     1.3258
     1.5077
     0.0032
     0.6807
```

ภาพที่ 19 ภาพตัวอย่างผลการปรับค่าน้ำหนักแต่ละคลาส

11. เพิ่มข้อมูลการฝึกสอน

```
augmenter = imageDataAugmenter('RandXReflection',true,...
    'RandXTranslation', [-10 10], 'RandYTranslation', [-10 10]); %
datasource = pixelLabelImageSource(imdsTrain,pxdsTrain,...
    'DataAugmentation',augmenter); %ข้อมูลและส่วนเสริมในการเทรนเก็บรวมกัน
```

ภาพที่ 20 ภาพตัวอย่างรหัสโปรแกรมการเพิ่มข้อมูลการฝึกสอน

ฟังก์ชันจะทำการทำการสุ่มค่า ตั้งแต่ -10 ถึง 10 พิกเซล สุ่มแกน x แกน y และ สุ่มค่าทางซ้ายหรือทางขวา เพื่อเพิ่มข้อมูลในการฝึกสอน ทำให้ตัวแบบมีความแม่นยำมากขึ้น

12. ตั้งค่าการฝึกสอน

```
|
options = trainingOptions('sgdm', ...
    'Momentum', 0.9, ...
    'InitialLearnRate', 1e-2, ...
    'L2Regularization', 0.0005, ...
    'MaxEpochs', 1000, ...
    'MiniBatchSize', 1, ...
    'Shuffle', 'every-epoch', ...
    'Verbose', false, ...
    'Plots', 'training-progress'); %กา
```

ภาพที่ 21 ภาพตัวอย่าง code การตั้งค่าการฝึกสอน

อัลกอริธึมที่ใช้ในการฝึกสอน คือ SGDM อัลกอริธึมนี้จะสุ่มการไล่ระดับสีแบบโมเมนตัม

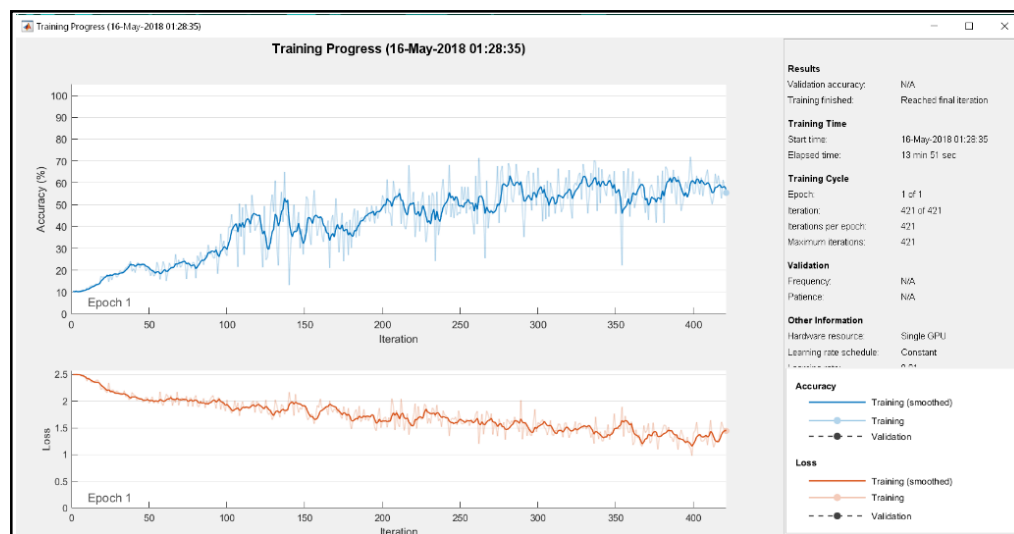
อัตราการเรียนรู้อยู่ที่ 0.01

กำหนดการฝึกสอน 1000 รอบ

13. เริ่มฝึกสอน

```
%เริ่มเทรน
tic
[net, info] = trainNetwork(datasource,lgraph,options);
toc
save('TestMyCNN3.mat','net','info','options'); %save โมเดลเทรน
disp('NN trained');
```

ภาพที่ 22 ภาพตัวอย่างรหัสโปรแกรมการฝึกสอนตัวแบบ



ภาพที่ 23 ภาพตัวอย่างการฝึกสอน

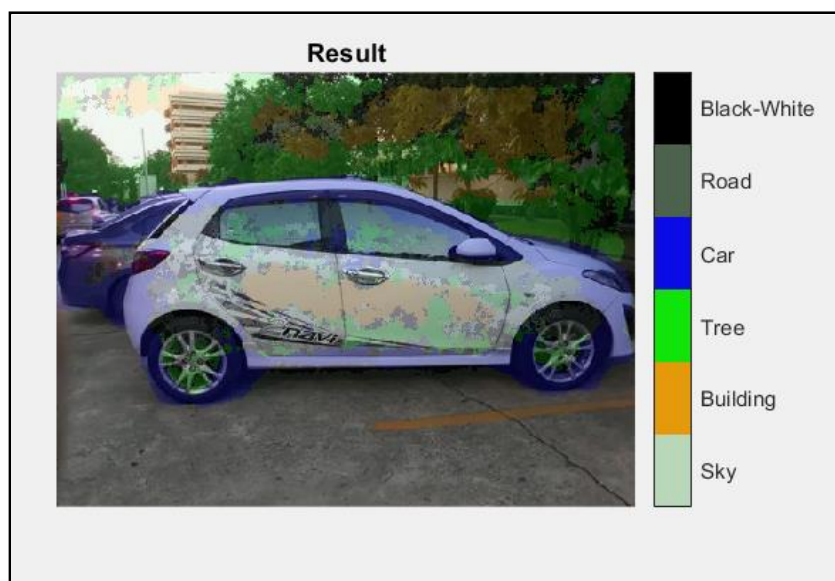
14. บันทึกการฝึกสอนตัวแบบ

```

Model = load('TestMyCNN3.mat'); %โหลดโมเดล
pic_num = 1; %กำหนดรูป
I = readimage(imds, pic_num); %อ่านรูป
% Show the results of the semantic segmentation
C = semanticseg(I, Model.net); %นำรูปที่อ่านมา semantic segmentation กับ โมเดล
CB = labeloverlay(I, C, 'Colormap', cmap, 'Transparency',0.8); %ใส่สีให้ผลลัพธ์ตาม class
figure
imshow(CB)
HelperFunctions.pixelLabelColorbar(cmap, classes); %แสดง Colorbar
title('Result')

```

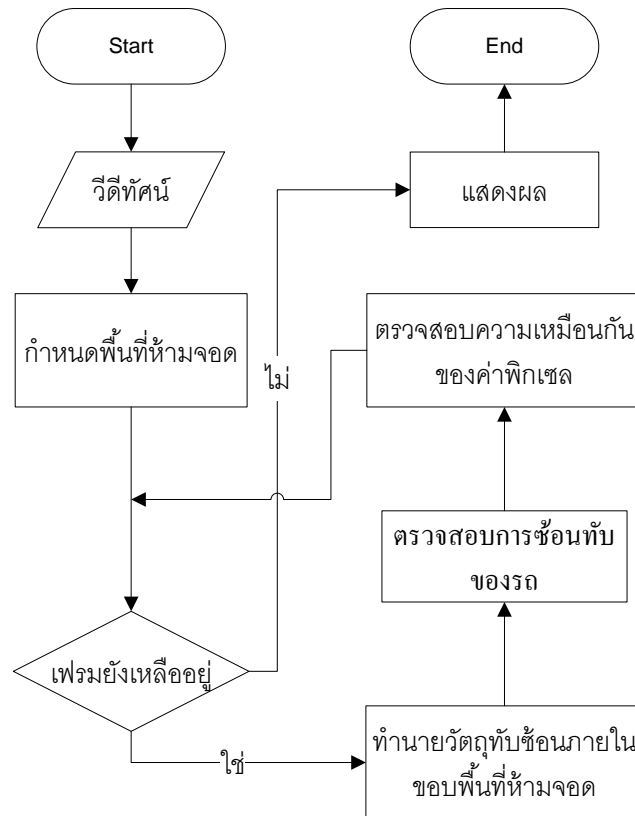
ภาพที่ 24 ภาพตัวอย่างรหัสโปรแกรม semantic segmentation



ภาพที่ 25 ภาพตัวอย่างผลลัพธ์การสกัดวัตถุในรูปภาพ จากการฝึกสอนจากตัวแบบ

2.) การทำงานของระบบ

2.1) ฟังก์ชันตอนการทำงานของระบบ





ภาพที่ 26 ภาพตัวอย่างฟังก์ชันตอนการทำงานของระบบตรวจจับการจอดรถในพื้นที่ห้ามจอด

2.2) การทำงานตัวโปรแกรม

2.2.1) การติดตั้งเครื่องมือก่อนการทำงาน

1. ติดตั้งแพ็คเกจ MATLAB Support Package for IP Cameras และ ANDROID MOBILE CAMERA CONNECTOR

Name	Type	Author	Install...
 MATLAB Support Package for IP Cameras version 18.1.0	Hardware Support Package		20 May 2019
 ANDROID MOBILE CAMERA CONNECTOR version 1.0	Toolbox	Bala murugan	20 May 2019

ภาพที่ 27 ภาพตัวอย่างแพ็คเกจ IP Cameras และ ANDROID MOBILE CAMERA CONNECTOR

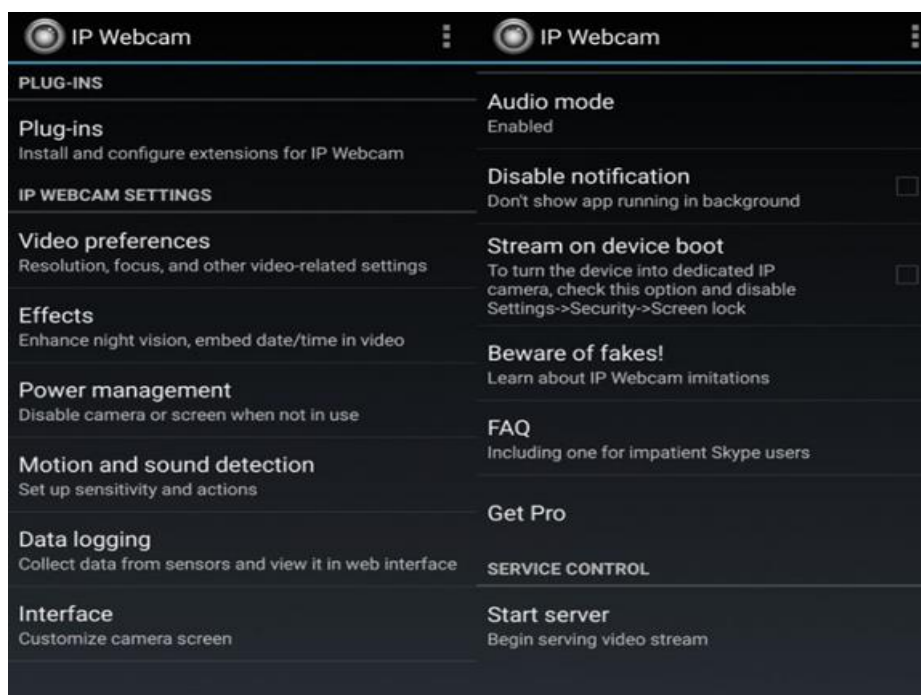
2. ติดตั้งแอปพลิเคชัน IP Webcam บนโทรศัพท์มือถือ



ภาพที่ 28 ภาพตัวอย่างแอปพลิเคชัน IP Webcam

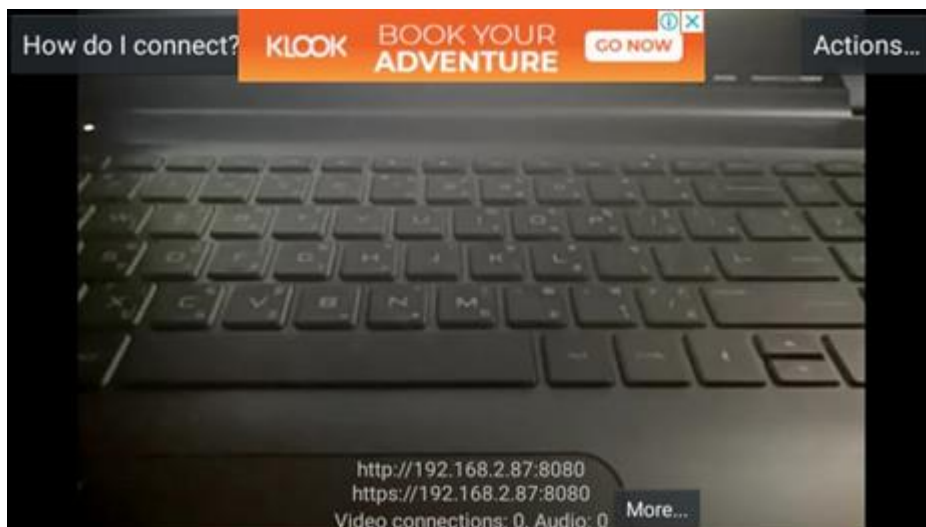
3. ตั้งค่า URL จาก IP Webcam

เริ่มแรกทำการเปิดแอปพลิเคชัน IP Webcam และทำการกดเมนู Start server เพื่อทำการสตรีม (การถ่ายทอดสด) วิดีทัศน์



ภาพที่ 29 ภาพตัวอย่างหน้าจอเมนูแอปพลิเคชัน IP Webcam

นำหมายเลข ip ที่อยู่ด้านล่างของหน้าจอโทรศัพท์ นำมาใส่ในตัวแปร url ในไฟล์ FinalDemo.m บรรทัดที่ 20



ภาพที่ 30 ภาพตัวอย่างการสตรีมวิดีโอจากแอปพลิเคชัน IP Webcam

```
url = ('http://192.168.2.87:8080/shot.jpg?rnd=350264');
```

ภาพที่ 31 ภาพตัวอย่างรหัสโปรแกรมตัวแปร url ในไฟล์ FinalDemo.m

4.การกำหนดขอบหน้าต่างการคำนวณหารถในพื้นที่ห้ามจอด

กำหนดขอบหน้าต่างการคำนวณหารถในพื้นที่ห้ามจอดโดยมีตัวแปรดังนี้

LMAX ค่าเริ่มต้นของแกน x

RMAX ค่าสิ้นสุดของแกน x

BMAX ค่าเริ่มต้นของแกน y

HMAX ค่าสิ้นสุดของแกน y

```

LMax = 50;
RMax = 400;
HMax = 150;
BMax = 225;

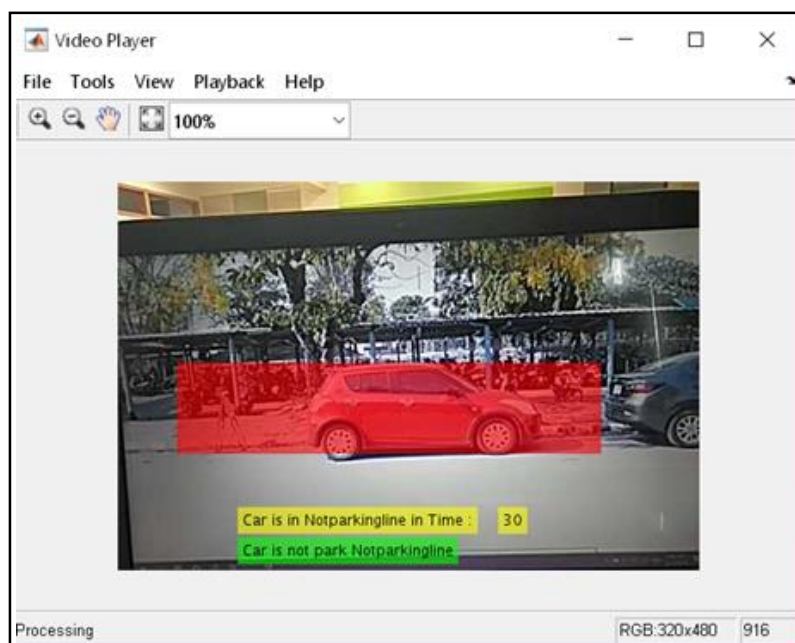
boxA = LMax;
boxB = HMax;
boxC = RMax - LMax;
boxD = BMax - HMax;
bbox = [boxA,boxB,boxC,boxD];

```

ภาพที่ 32 ภาพตัวอย่างรหัสโปรแกรมการกำหนดขอบหน้าต่างการคำนวณในไฟล์ FinalDemo.m

2.2.2) เริ่มการทำงาน

1. เปิดเปิดแอปพลิเคชัน IP Webcam และทำการสตรีมวิดีโอ
2. ทำการรันรหัสโปรแกรม FinalDemo.m ด้วยโปรแกรม matlab



ภาพที่ 33 ภาพตัวอย่างผลการทำงานของโปรแกรม