Traffic Light Classification using Convolution Neural Network

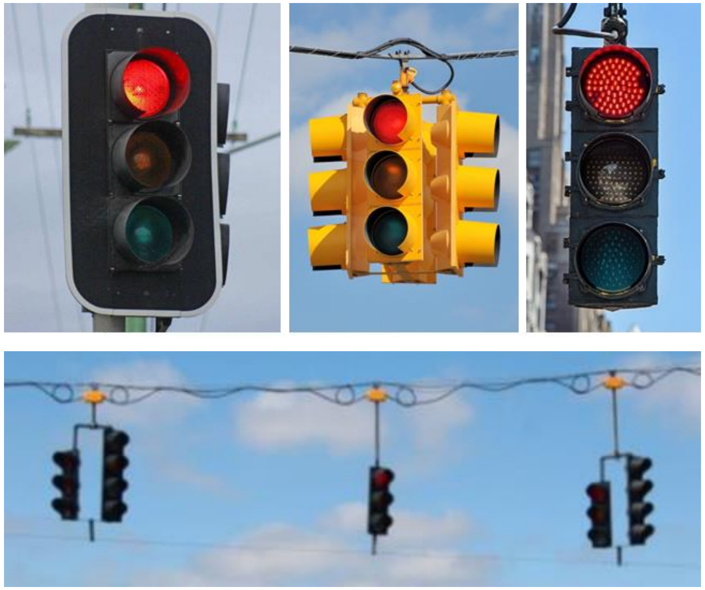
A2017020. Park, yusang
A2018024. Son, weonil

Intelligent Vehicle Design Lab. (Advisor : Prof. Kihong Park)
Graduate School of Automotive Engineering, Kookmin University
+82-2-943-7630

Intelligent Vehicle Design Lab.

Contents

1. 프로젝트 목적, 동기



Generate dataset
based on each signal

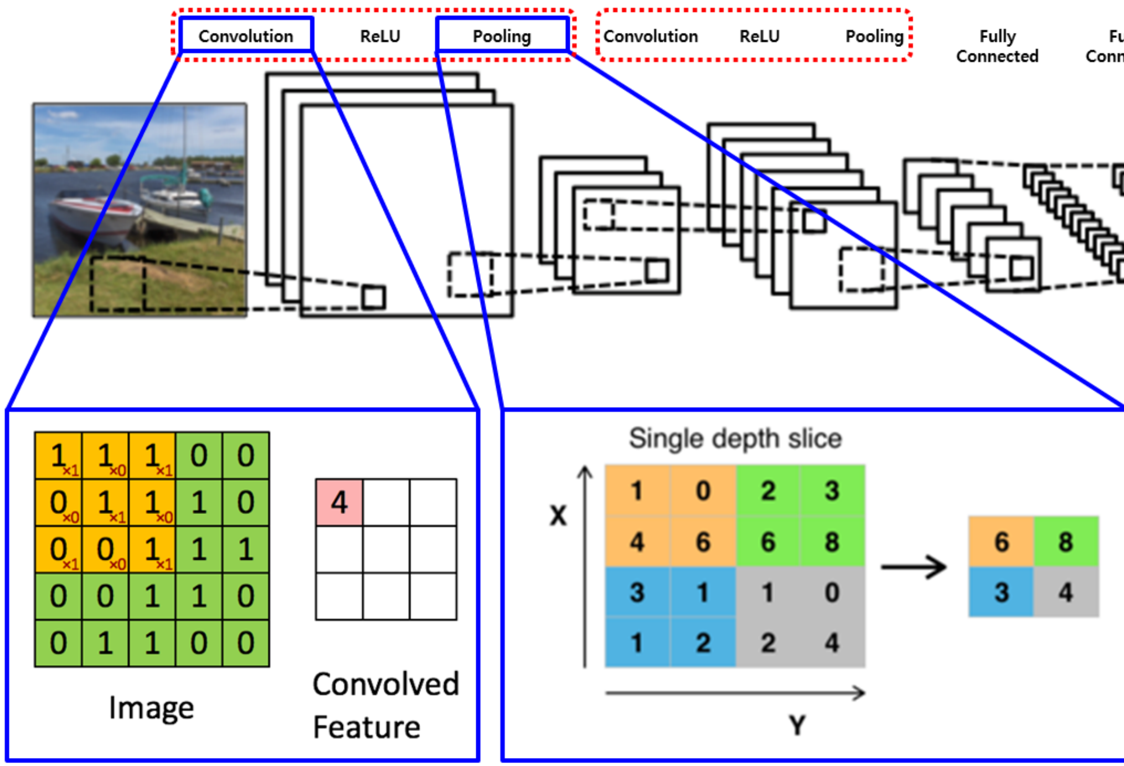
- The shapes of the traffic lights vary, and the existing data sets are traffic light data used in foreign countries.
- **We need data that matches the environment in Korea.**

- The traffic light data will be collected through the image of the black box attached to the vehicle.

1) Convolutional Neural Network(CNN)

In machine learning, a CNN is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex

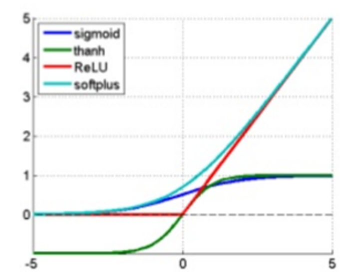
The convolution layer is mainly composed of three layers (Convolution layer, Pooling layer, Fully connected layer)



Convolution layer

Pooling layer (ex. Max pooling)

- ReLU Activation function



$$f(x) = \max(0, x)$$

$$f'(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases}$$

2. 모델 구성

```
def build_net(self, in_dim, out_dim) :
    L1_ch = 16
    L2_ch = 32
    L3_ch = 64


    self.input_img = tf.placeholder("float", [None, 64, 64, 3])
    self.Y = tf.placeholder("float", [None, out_dim])


    W1 = tf.get_variable("W1", shape = [3, 3, 3, L1_ch],
            initializer = tf.contrib.layers.xavier_initializer())

    W2 = tf.get_variable("W2", shape = [3, 3, L1_ch, L2_ch],
            initializer = tf.contrib.layers.xavier_initializer())

    W3 = tf.get_variable("W3", shape = [3, 3, L2_ch, L3_ch],
            initializer = tf.contrib.layers.xavier_initializer())


    W4 = tf.get_variable("W4", shape = [8 * 8 * L3_ch, 625],
            initializer = tf.contrib.layers.xavier_initializer())

    B4 = tf.get_variable("B4", shape = [625],
            initializer = tf.contrib.layers.xavier_initializer())

    W5 = tf.get_variable("W5", shape = [625, out_dim],
            initializer = tf.contrib.layers.xavier_initializer())

    B5 = tf.get_variable("B5", shape = [out_dim],
            initializer = tf.contrib.layers.xavier_initializer())


    L1 = tf.nn.conv2d(self.input_img, W1, strides = [1, 1, 1, 1], padding = 'SAME')
    L1 = tf.nn.relu(L1)
    L1 = tf.nn.max_pool(L1, ksize=[1, 2, 2, 1], strides = [1, 2, 2, 1], padding = 'SAME')
```

Neural Net Model 구성 및 학습

→ 박유상

Generate Training/Test data set

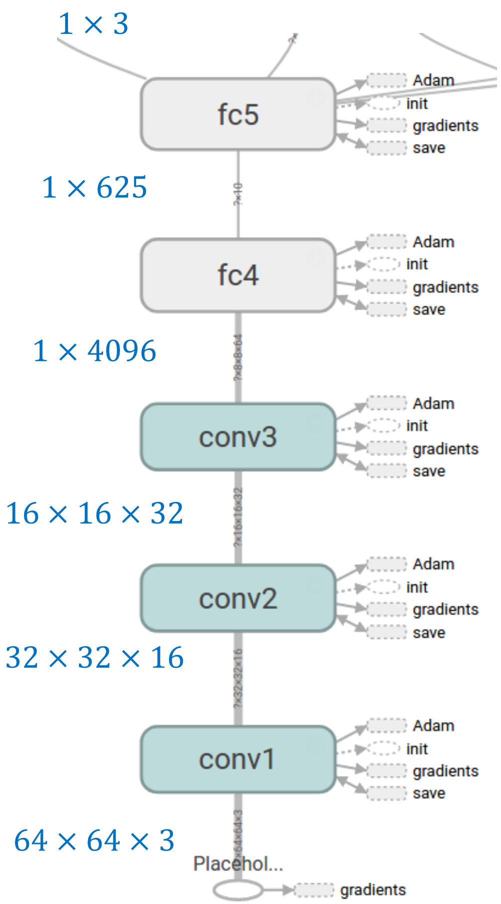
→ 손원일

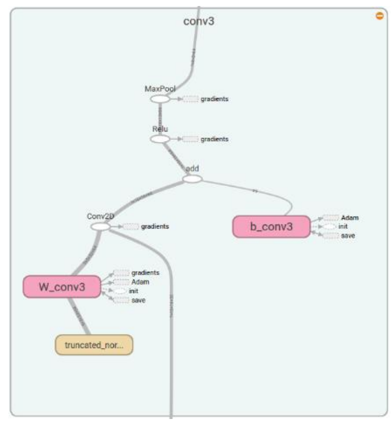
https://github.com/parkys7175/KMUML_TermProject

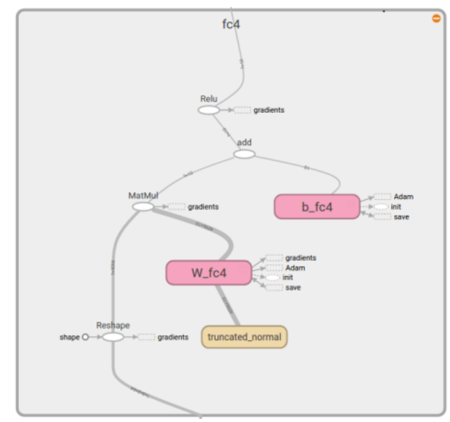
2. 모델 구성

Hyper parameter

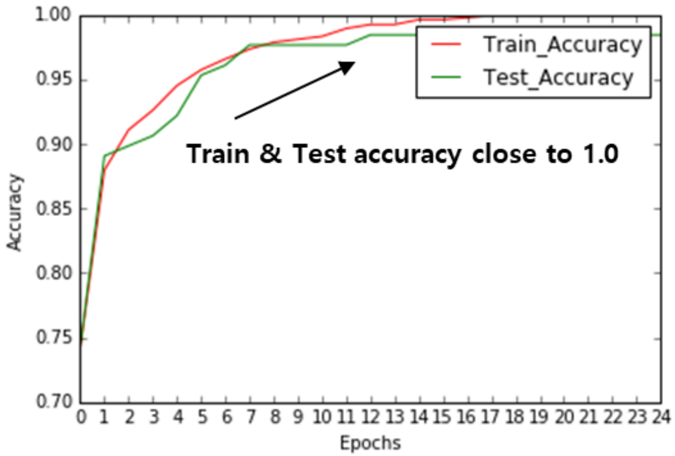
Training Function : AdamOptimizer(0.001, 0.9) , Batch Size : 32
Weight initial Value : xavier distribution (mean : 0, stddev : 0.1)
Convolution Layer Filter Size : 3 by 3 (stride : 1, padding : “SAME”)

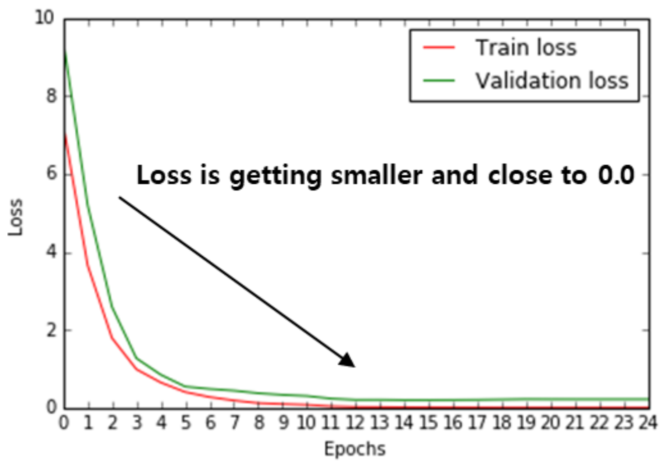






- Set the padding size of the Convolution Layer to "SAME" and set the output height and width to be ½ of the input Padding
- Set the padding size of the Pooling Layer to "SAME" and set the output and input size to be equal to padding



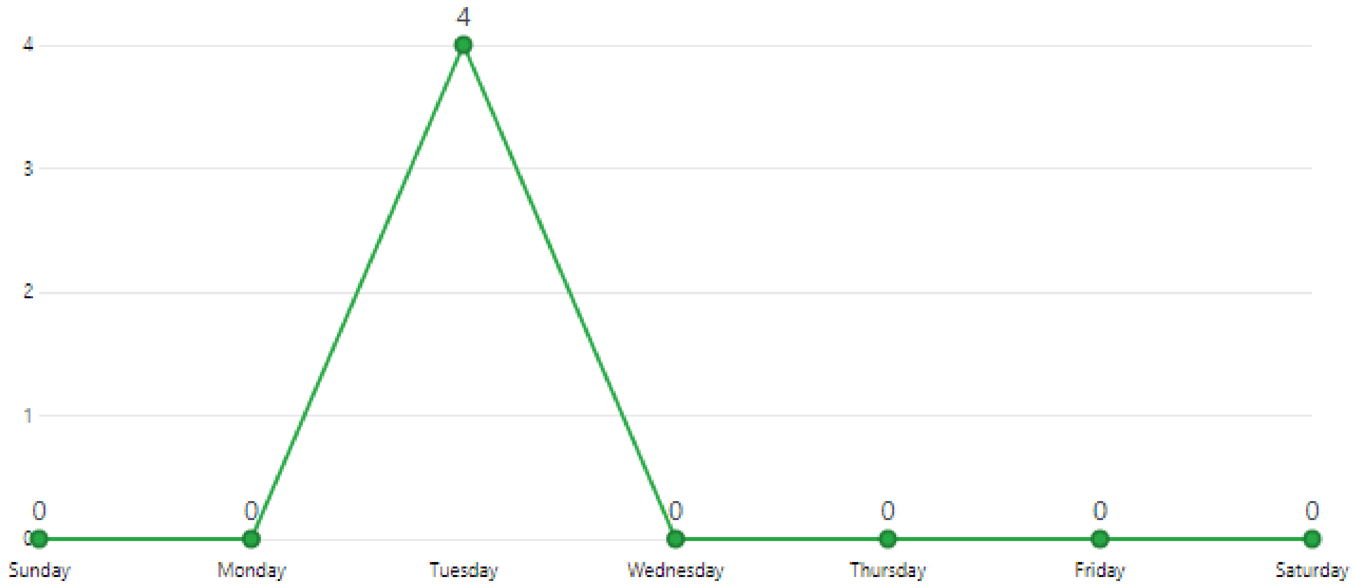



```
Instructions for updating:
Use `tf.global_variables_initializer` instead.
Starting training... [1100 training examples]
Epoch     1: loss:    50.8976402283, val. loss:    67.4993057251
Epoch     2: loss:    35.0620765686, val. loss:    49.9294662476
Epoch     3: loss:    25.5480136871, val. loss:    39.0094337463
Epoch     4: loss:    19.4923515320, val. loss:    31.0736103058
Epoch     5: loss:    14.9140558243, val. loss:    24.7881374359
Epoch     6: loss:    12.2071199417, val. loss:    20.7929191589
Epoch     7: loss:    10.4887542725, val. loss:    18.2450370789
Epoch     8: loss:     9.2224893570, val. loss:    15.3319034576
Epoch     9: loss:     8.1565217972, val. loss:    14.2656202316
Epoch    10: loss:     7.3590760231, val. loss:    12.1426315308
Epoch    11: loss:     6.5919122696, val. loss:    11.2685194016
Epoch    12: loss:     5.8827600479, val. loss:     9.5084695816
Epoch    13: loss:     5.2598943710, val. loss:     8.8494844437
Epoch    14: loss:     4.7556905746, val. loss:     7.8124408722
Epoch    15: loss:     4.3264470100, val. loss:     7.6585688591
Epoch    16: loss:     4.0145287514, val. loss:     6.3987312317
Epoch    17: loss:     3.7380197048, val. loss:     6.8411378860
Epoch    18: loss:     3.3752672672, val. loss:     5.9478483200
Epoch    19: loss:     3.1076576710, val. loss:     5.0701255798
Epoch    20: loss:     2.9255490303, val. loss:     4.3520798683
Epoch    21: loss:     2.7314138412, val. loss:     4.4579029083
Epoch    22: loss:     2.6059072018, val. loss:     3.4970731735
Epoch    23: loss:     2.5007209778, val. loss:     3.4413919449
Epoch    24: loss:     2.3178780079, val. loss:     2.9463334084
Epoch    25: loss:     2.2475883961, val. loss:     3.2216069698
Epoch    26: loss:     2.1821777821, val. loss:     2.5234611034
Epoch    27: loss:     1.9322553873, val. loss:     2.4208142757
Epoch    28: loss:     1.8584405184, val. loss:     2.3332271576
Epoch    29: loss:     1.7501019239, val. loss:     2.1924605370
Epoch    30: loss:     1.6678992510, val. loss:     1.9591627121
test_accuracy: 98.041
---61.64 seconds---
Actual: red, predicted: green
<IPython.core.display.Image object>
Actual: yellow, predicted: red
<IPython.core.display.Image object>
Actual: green, predicted: red
<IPython.core.display.Image object>
Actual: yellow, predicted: green
<IPython.core.display.Image object>
```

Accuracy : 98.041%

4. 커밋 그래프

4
3
2
1
0
06/18
07/09
07/30
08/20
09/10
10/01
10/22
11/12
12/03
12/24
01/14
02/04
02/25
03/18
04/08
04/29
05/20
06/10



4
3
2
1
0
4
0
0
0
0
0
0
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday

5. 향후 계획

신호등에 해당하는 바운딩 박스를 실시간으로 검출하여, 자율주행 시 사용가능 하도록 딥러닝 알고리즘 개발

경청해 주셔서 감사드립니다

KMU GSAEK
Graduate School of Automotive Engineering,
Kookmin University

iVDL
Intelligent Vehicle Design Lab.