AlexNet

<ImageNet Classification with Deep Convolutional Neural Networks>

서울과학기술대학교 국방인공지능응용학과
이찬호

Part 1

AlexNet 구현

Part 1

AlexNet 구현 – Top-1 Error & Top-5 Error

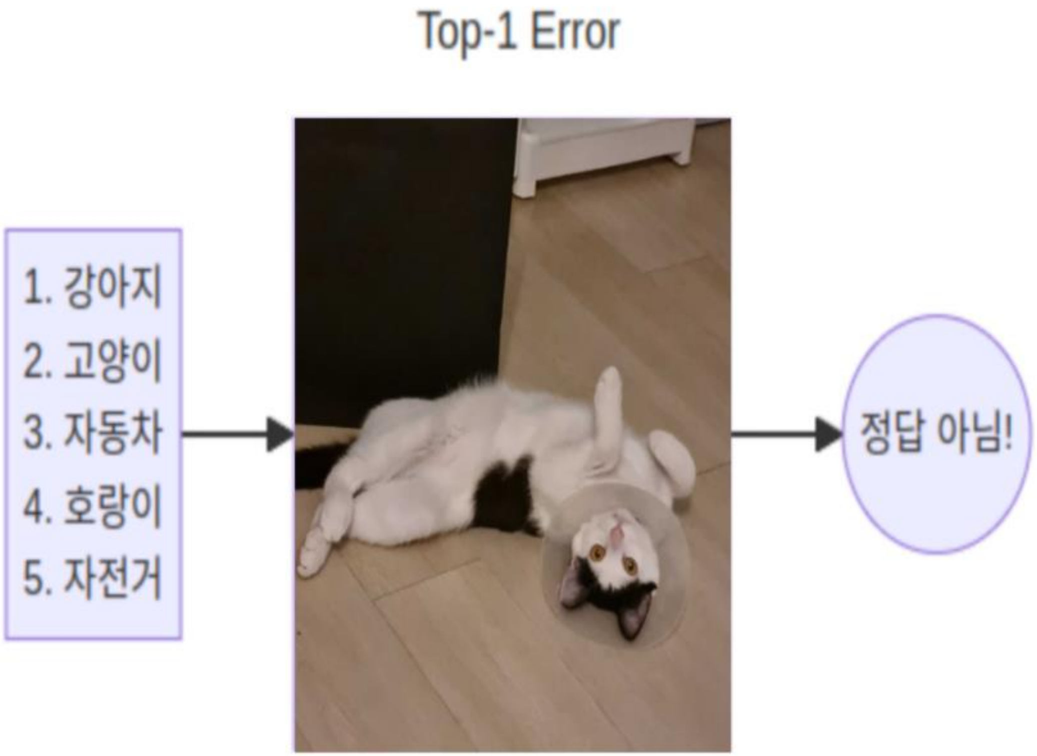


Fig 1. Top-1 Error

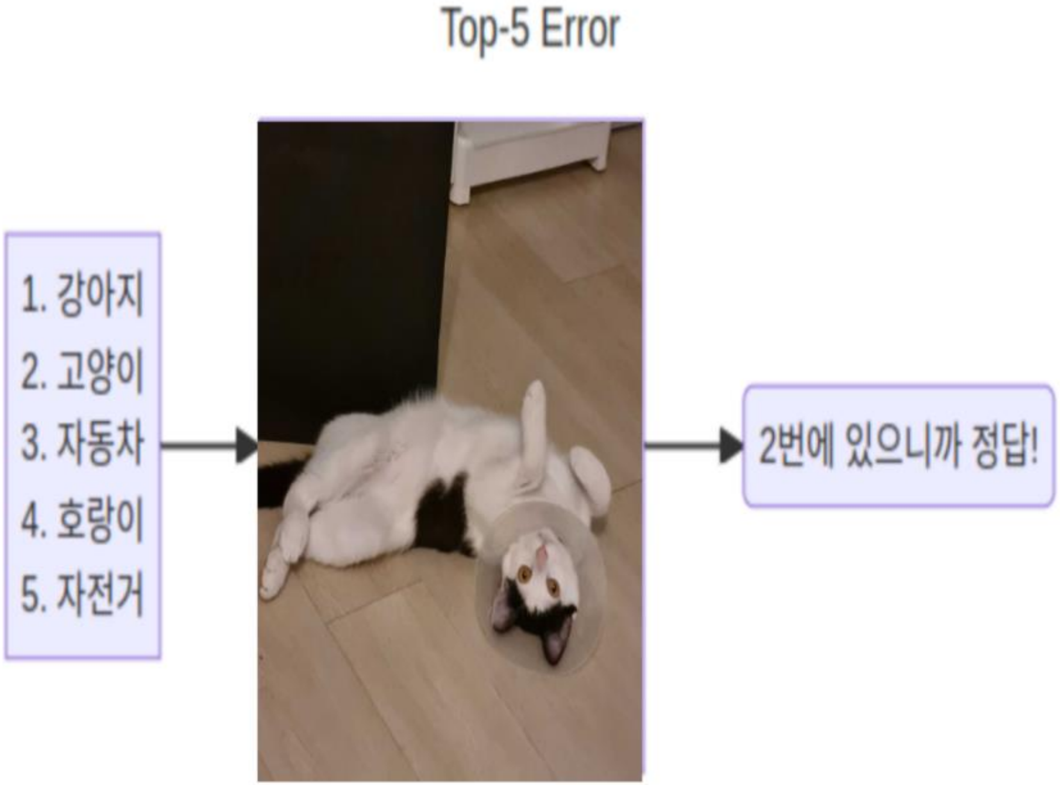


Fig 2. Top-5 Error

AlexNet 구현 –Architecture

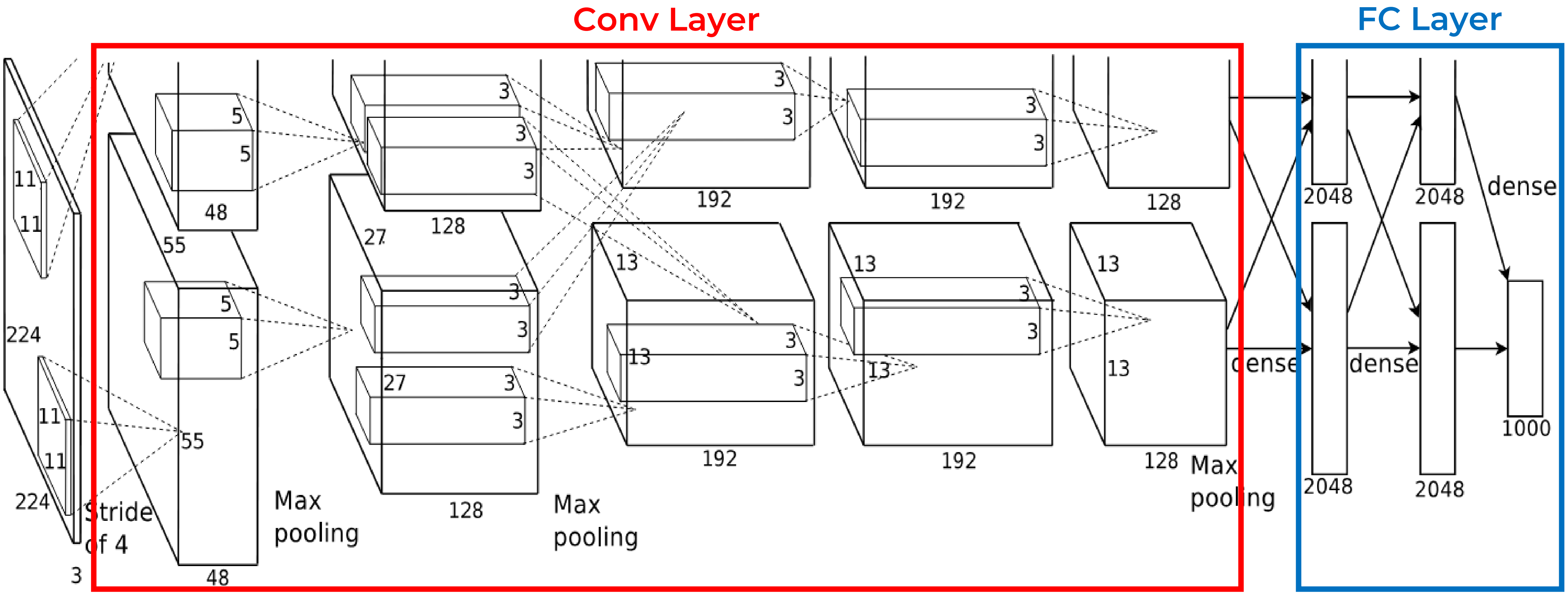


Fig 1. AlexNet Architecture

Part 1

AlexNet 구현 –Architecture

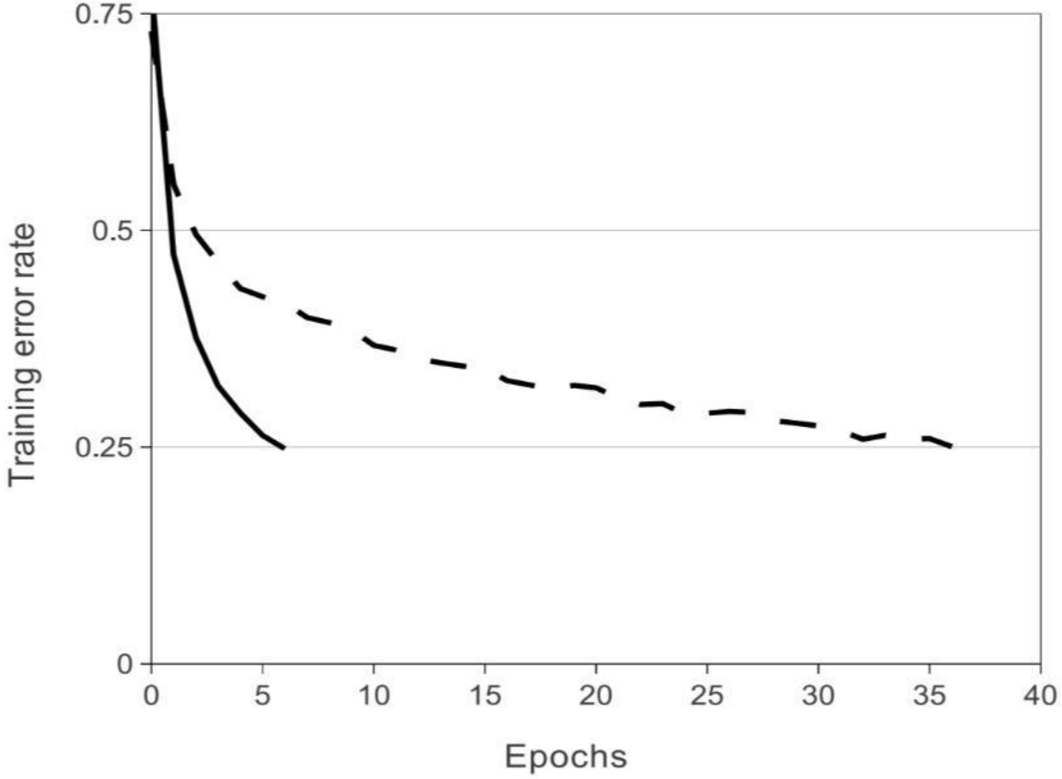
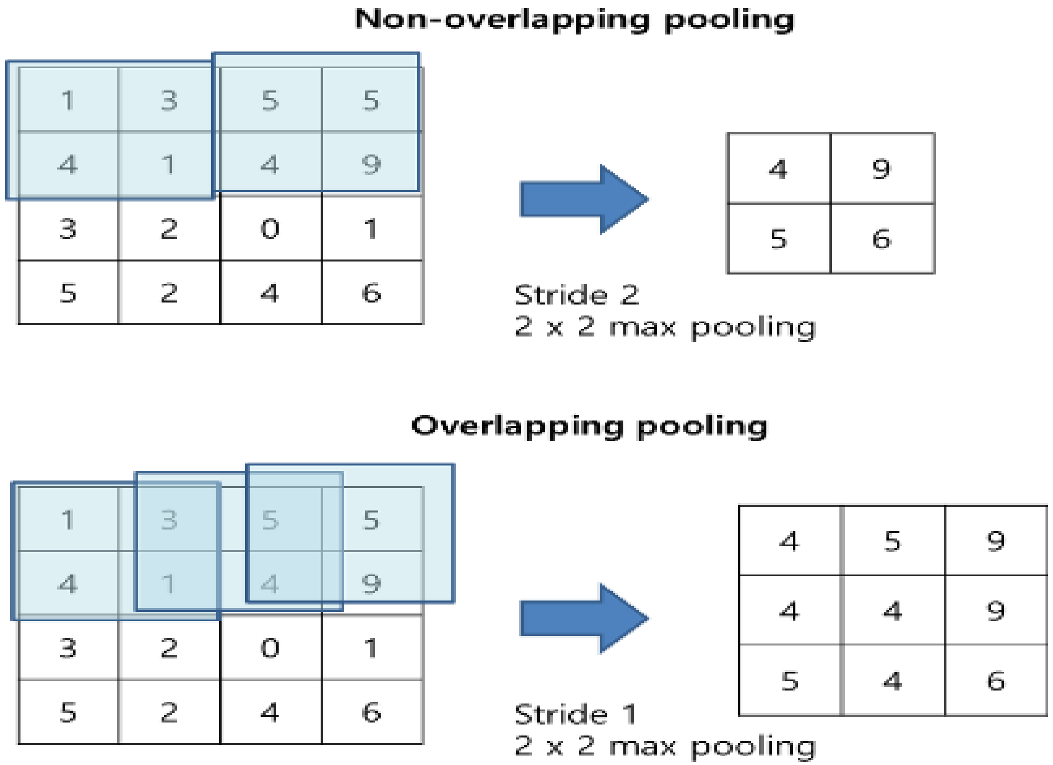


Fig 2. ReLU(실선) VS tanh(점선)



1	3	5	5
4	1	4	9
3	2	0	1
5	2	4	6

4	9
5	6

1	3	5	5
4	1	4	9
3	2	0	1
5	2	4	6

4	5	9
4	4	9
5	4	6

Fig 3. Overlapping Pooling

Part 1

AlexNet 구현 –Architecture

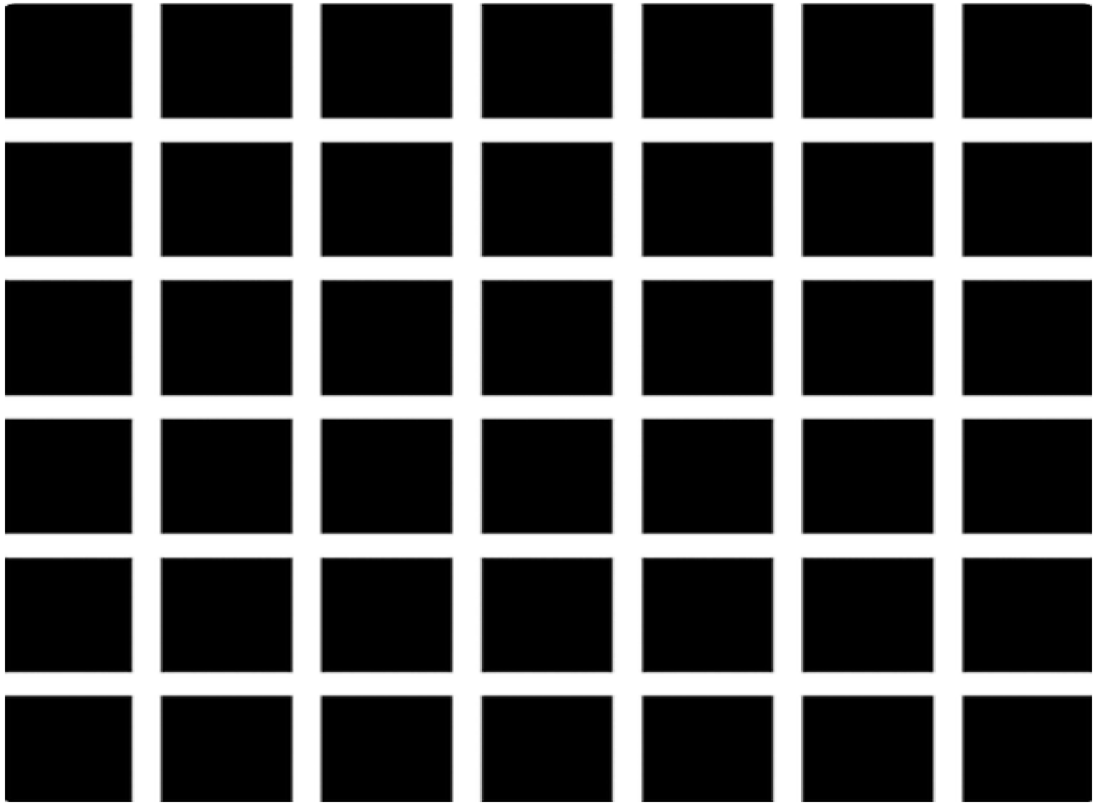


Fig 4. Local Response Normalization

Part 1

AlexNet 구현 – Reducing Overfitting

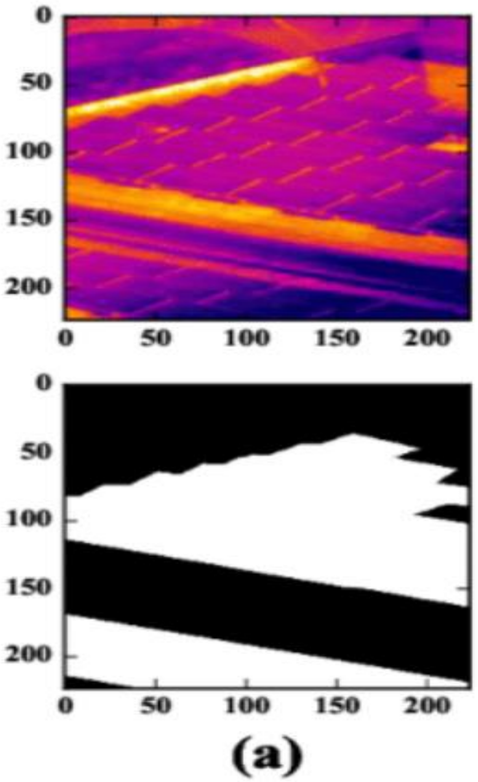


Fig 5. Random Crop

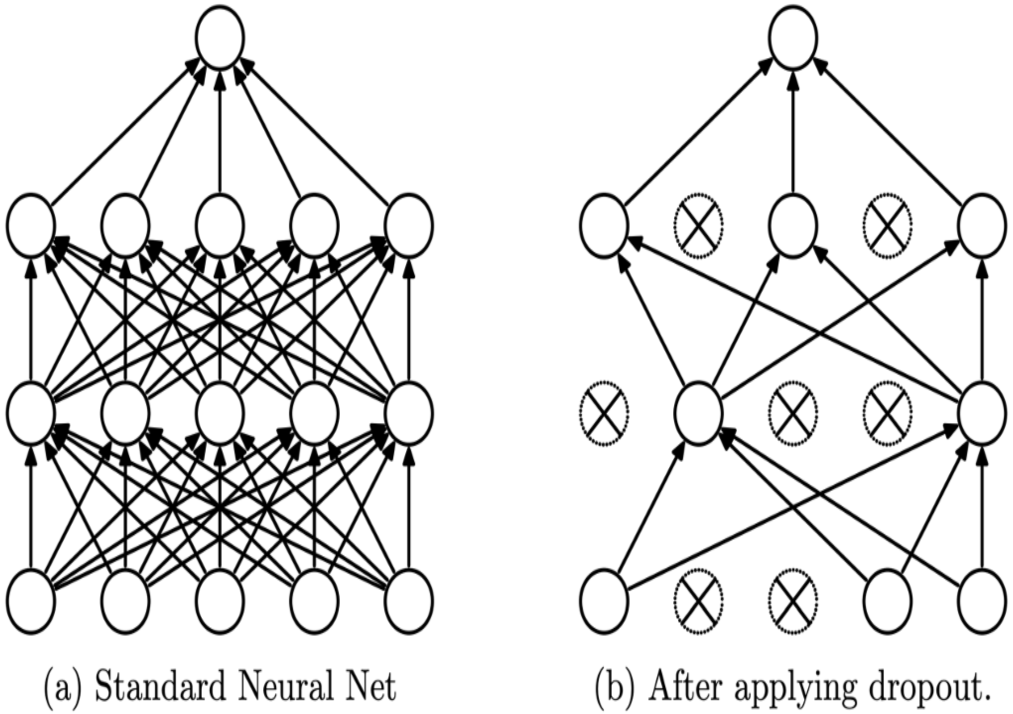


Fig 6. DropOut

Part 2

AlexNet 구현 –Data Set

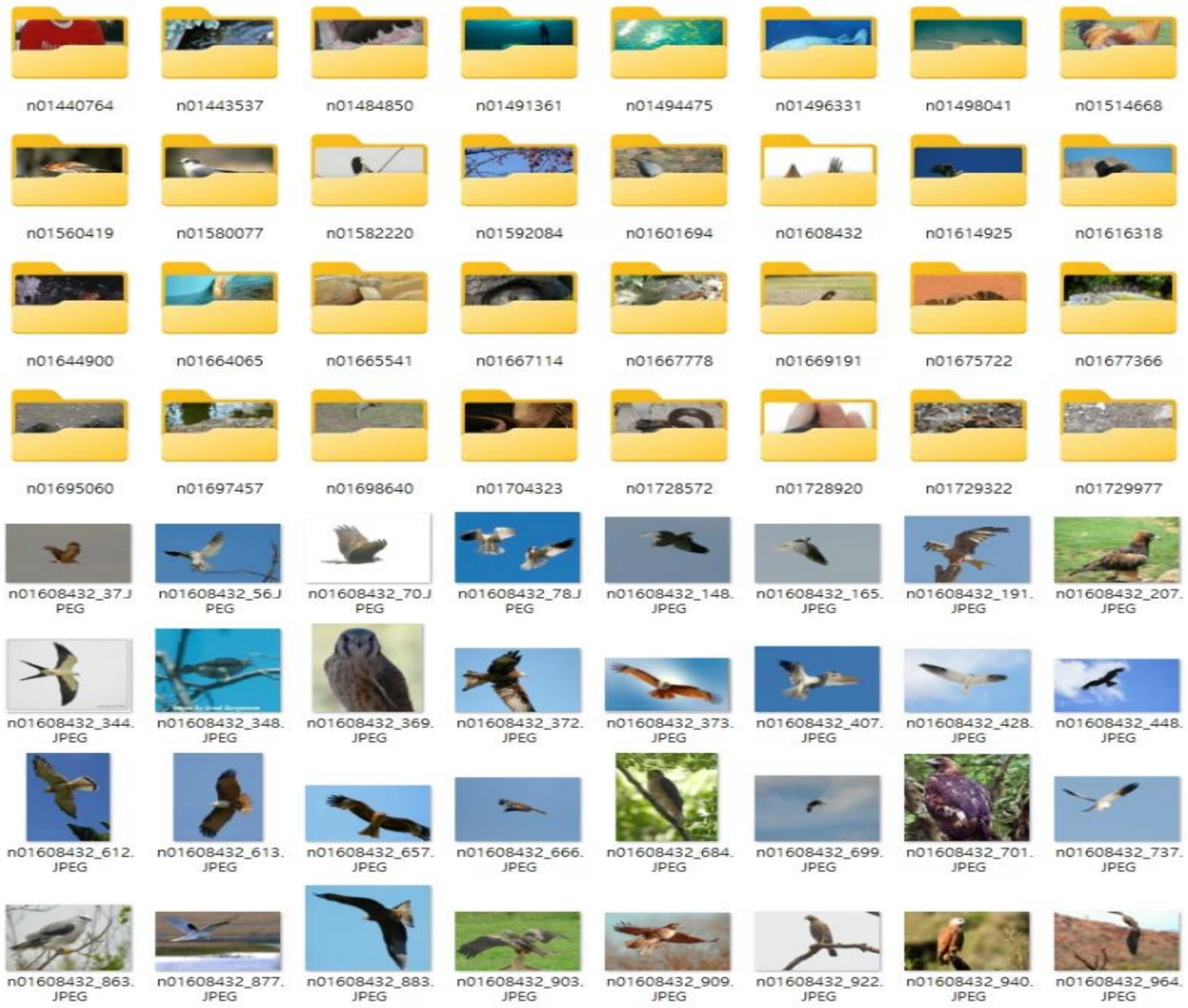
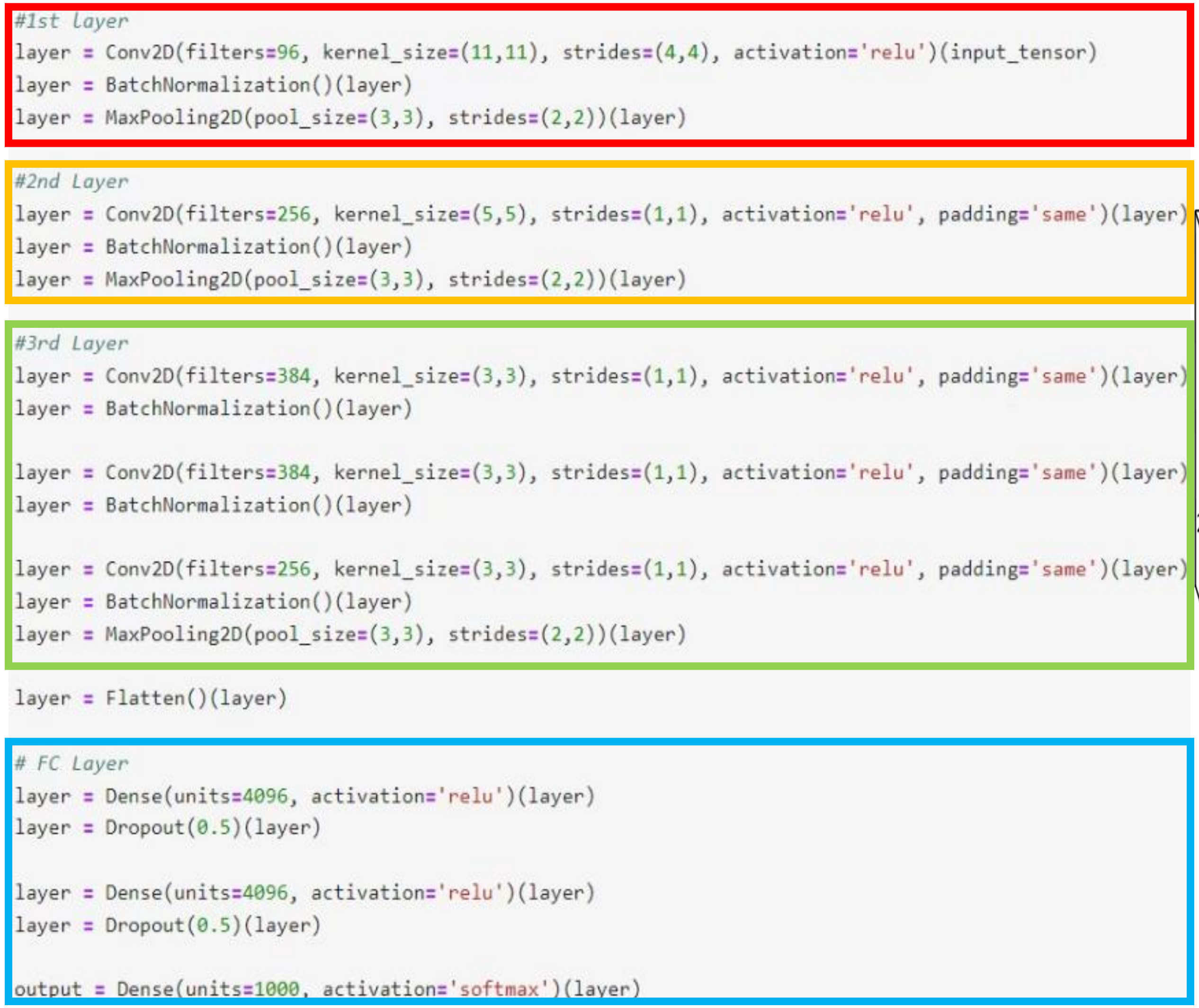


Fig 7. Train Class 일부 (위) & Img 중 일부 (아래)

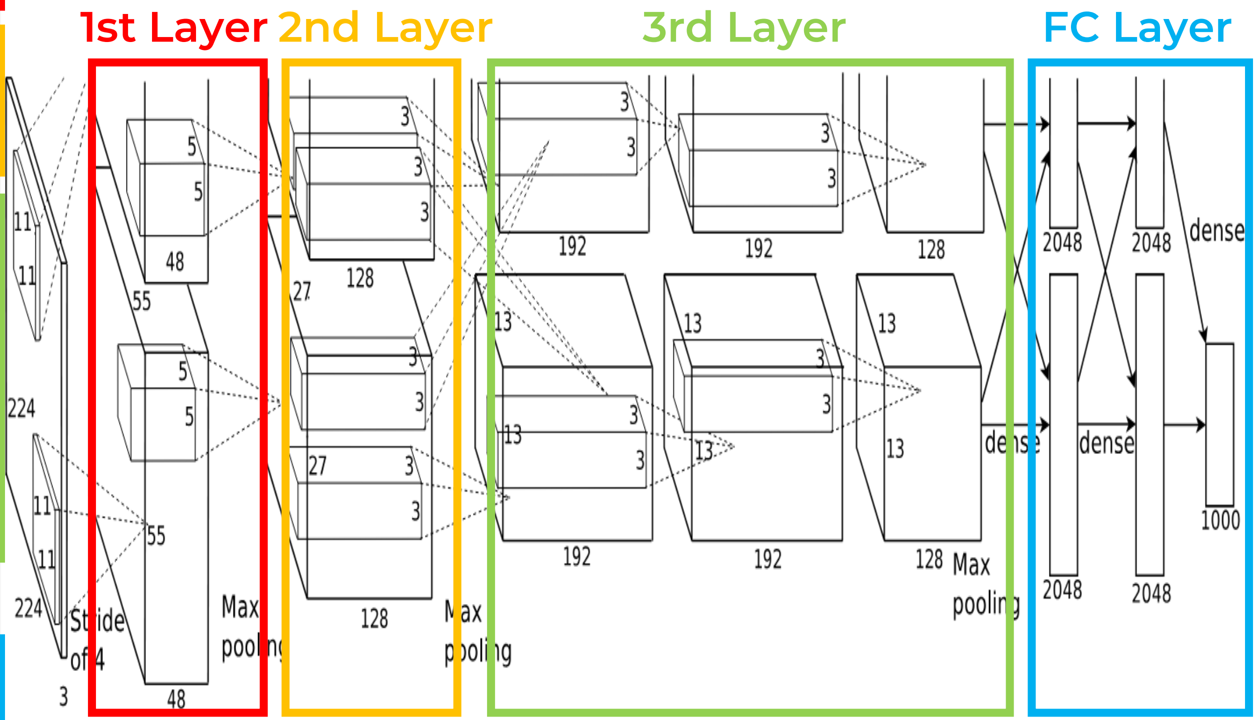
Part 2

AlexNet 구현 –Code



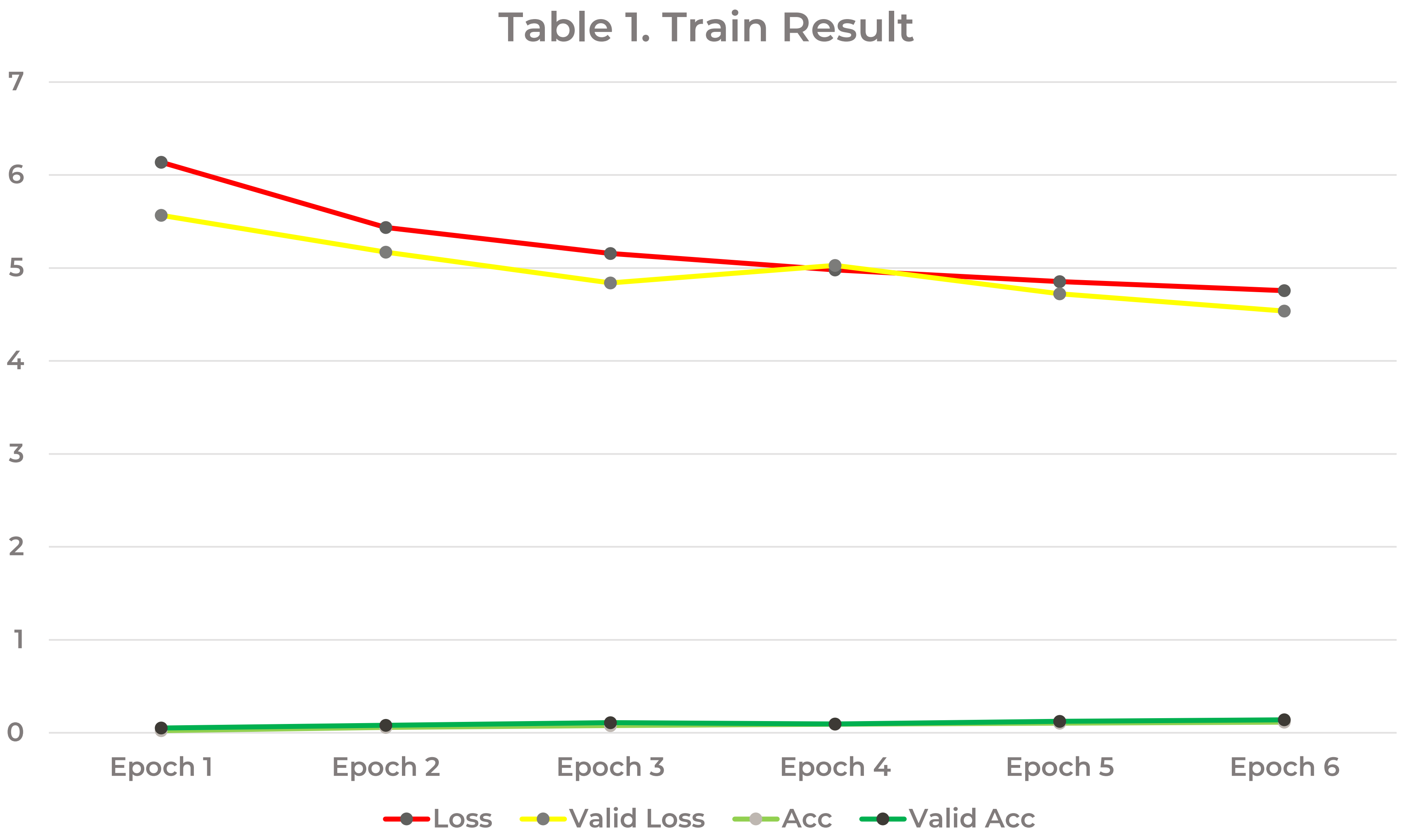
```
#1st layer
layer = Conv2D(filters=96, kernel_size=(11,11), strides=(4,4), activation='relu')(input_tensor)
layer = BatchNormalization()(layer)
layer = MaxPooling2D(pool_size=(3,3), strides=(2,2))(layer)

#2nd Layer
layer = Conv2D(filters=256, kernel_size=(5,5), strides=(1,1), activation='relu', padding='same')(layer)
layer = BatchNormalization()(layer)
layer = MaxPooling2D(pool_size=(3,3), strides=(2,2))(layer)

#3rd Layer
layer = Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), activation='relu', padding='same')(layer)
layer = BatchNormalization()(layer)

layer = Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), activation='relu', padding='same')(layer)
layer = BatchNormalization()(layer)

layer = Conv2D(filters=256, kernel_size=(3,3), strides=(1,1), activation='relu', padding='same')(layer)
layer = BatchNormalization()(layer)
layer = MaxPooling2D(pool_size=(3,3), strides=(2,2))(layer)

layer = Flatten()(layer)

# FC Layer
layer = Dense(units=4096, activation='relu')(layer)
layer = Dropout(0.5)(layer)

layer = Dense(units=4096, activation='relu')(layer)
layer = Dropout(0.5)(layer)

output = Dense(units=1000, activation='softmax')(layer)
```

Fig 8. AlexNet 구현 코드



Part 2

AlexNet 구현 –Result



THANK YOU FOR YOUR ATTENTION