

이름 : 강시온

학번 : 201802045

학과 : 컴퓨터공학과

구현코드

```
model = keras.Sequential([
    keras.Input(shape=(32,32,3)),
    keras.layers.Conv2D(
        filters=12, kernel_size=5, strides=(1, 1), padding='valid',
        data_format=None, dilation_rate=(1, 1), groups=1, activation=None,
        use_bias=True, kernel_initializer='glorot_uniform',
        bias_initializer='zeros', kernel_regularizer=None,
        bias_regularizer=None, activity_regularizer=None, kernel_constraint=None,
        bias_constraint=None
    ),
    keras.layers.MaxPool2D(),
    keras.layers.Conv2D(
        filters=8, kernel_size=5, strides=(1, 1), padding='valid',
        data_format=None, dilation_rate=(1, 1), groups=1, activation=None,
        use_bias=True, kernel_initializer='glorot_uniform',
        bias_initializer='zeros', kernel_regularizer=None,
        bias_regularizer=None, activity_regularizer=None, kernel_constraint=None,
        bias_constraint=None
    ),
    keras.layers.Flatten(),
    keras.layers.Dense(10, activation='softmax')
])
```

코드 설명

두번의 conv2D 레이어를 거치도록 했으며

32, 32, 3차원의 이미지를

28, 28, 12,

10, 10 8차원으로 만들어야 하므로

첫번째 conv2D의 filters는 12, kernel\_size는 5가된다.

그리고 maxpool레이어를 거치면

14, 14, 12가 되므로

두번째 conv2D의 filters는 8, kernel\_size는 5가된다.

그리고 마지막에 결과를 1차원의 10개로 분류하기 위해 flatten으로 펴주고 softmax를 이용해 전 체합이 1이 되도록 분류하였다.

## 결과 이미지

```
Epoch 1/20
/usr/local/lib/python3.7/dist-packages/tensorflow/python/util/dispatch.py:1096: UserWarning: ``sparse_categorical_crossentropy`` received `from_logits`
  return dispatch_target(*args, **kwargs)
6250/6250 [=====] - 38s 5ms/step - loss: 1.6998 - acc: 0.4054 - val_loss: 1.5108 - val_acc: 0.4688
Epoch 2/20
6250/6250 [=====] - 29s 5ms/step - loss: 1.4743 - acc: 0.4885 - val_loss: 1.4027 - val_acc: 0.5237
Epoch 3/20
6250/6250 [=====] - 29s 5ms/step - loss: 1.3949 - acc: 0.5204 - val_loss: 1.3764 - val_acc: 0.5225
Epoch 4/20
6250/6250 [=====] - 26s 4ms/step - loss: 1.3564 - acc: 0.5349 - val_loss: 1.3490 - val_acc: 0.5271
Epoch 5/20
6250/6250 [=====] - 26s 4ms/step - loss: 1.3345 - acc: 0.5432 - val_loss: 1.4017 - val_acc: 0.5242
Epoch 6/20
6250/6250 [=====] - 26s 4ms/step - loss: 1.3186 - acc: 0.5476 - val_loss: 1.3534 - val_acc: 0.5314
Epoch 7/20
6250/6250 [=====] - 28s 5ms/step - loss: 1.3114 - acc: 0.5525 - val_loss: 1.3464 - val_acc: 0.5377
Epoch 8/20
6250/6250 [=====] - 26s 4ms/step - loss: 1.2951 - acc: 0.5548 - val_loss: 1.3199 - val_acc: 0.5494
Epoch 9/20
6250/6250 [=====] - 26s 4ms/step - loss: 1.2889 - acc: 0.5575 - val_loss: 1.3658 - val_acc: 0.5311
Epoch 10/20
6250/6250 [=====] - 29s 5ms/step - loss: 1.2772 - acc: 0.5615 - val_loss: 1.3413 - val_acc: 0.5323
Epoch 11/20
6250/6250 [=====] - 26s 4ms/step - loss: 1.2679 - acc: 0.5644 - val_loss: 1.4269 - val_acc: 0.5116
Epoch 12/20
6250/6250 [=====] - 27s 4ms/step - loss: 1.2650 - acc: 0.5677 - val_loss: 1.3854 - val_acc: 0.5282
Epoch 13/20
6250/6250 [=====] - 26s 4ms/step - loss: 1.2628 - acc: 0.5702 - val_loss: 1.3483 - val_acc: 0.5288
Epoch 14/20
6250/6250 [=====] - 25s 4ms/step - loss: 1.2540 - acc: 0.5701 - val_loss: 1.3341 - val_acc: 0.5494
Epoch 15/20
6250/6250 [=====] - 26s 4ms/step - loss: 1.2532 - acc: 0.5697 - val_loss: 1.2984 - val_acc: 0.5524
Epoch 16/20
6250/6250 [=====] - 26s 4ms/step - loss: 1.2444 - acc: 0.5736 - val_loss: 1.3206 - val_acc: 0.5474
Epoch 17/20
6250/6250 [=====] - 26s 4ms/step - loss: 1.2392 - acc: 0.5744 - val_loss: 1.4261 - val_acc: 0.5258
Epoch 18/20
6250/6250 [=====] - 26s 4ms/step - loss: 1.2426 - acc: 0.5740 - val_loss: 1.5100 - val_acc: 0.5117
Epoch 19/20
6250/6250 [=====] - 26s 4ms/step - loss: 1.2366 - acc: 0.5767 - val_loss: 1.4495 - val_acc: 0.5201
Epoch 20/20
6250/6250 [=====] - 25s 4ms/step - loss: 1.2323 - acc: 0.5767 - val_loss: 1.4221 - val_acc: 0.5185
<keras.callbacks.History at 0x7f13b0327090>
```

## 난이도

모델을 구성하면서 재미있는 과제였다.