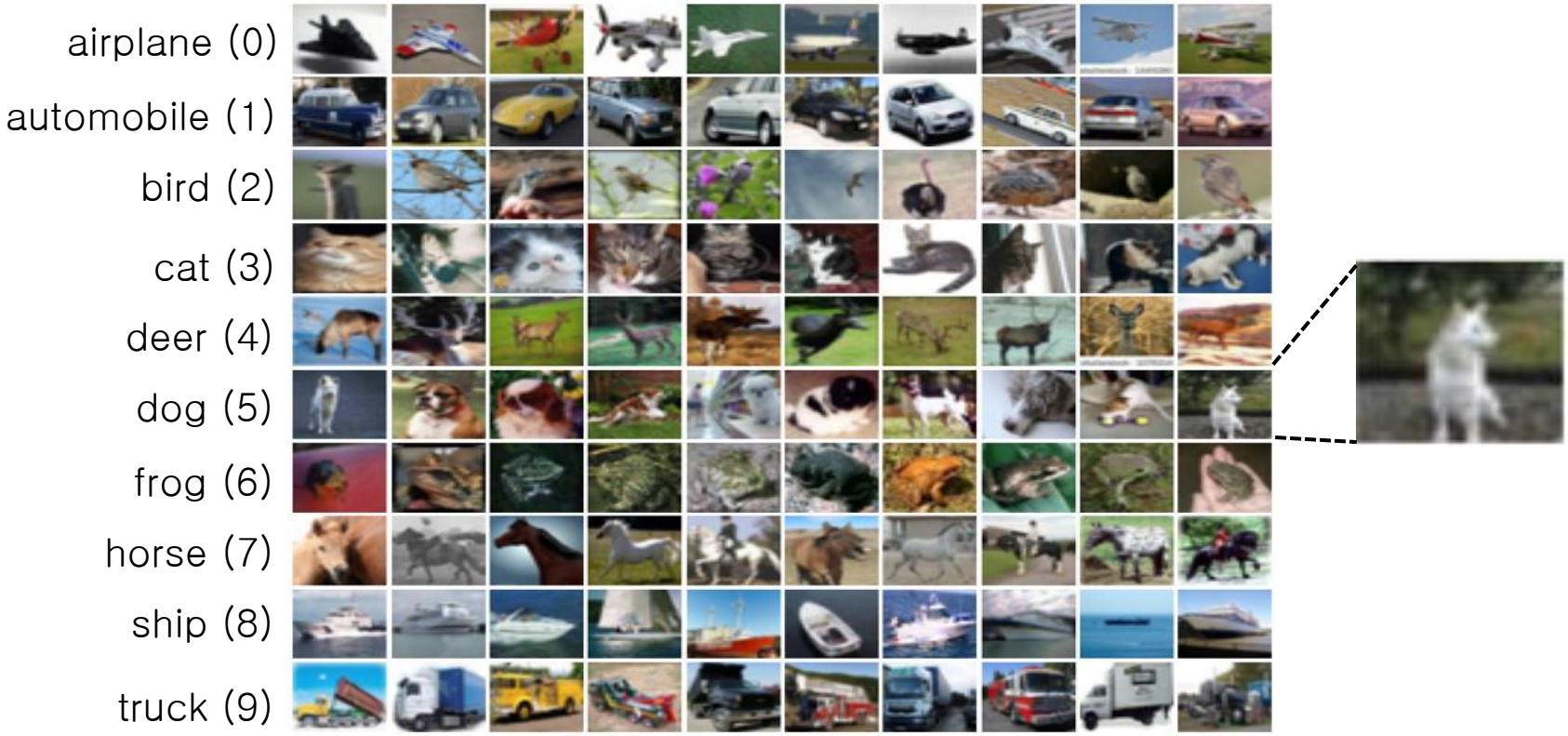
cifar10 dataset



- CIFAR 10 데이터는 airplane, automobile, bird 등의 10개의 정답으로 분류된 이미지이며, 딥러닝 학습을 위해 총 50,000개 학습데이터와 10,000개의 테스트 데이터로 이루어져 있음

- CIFAR 10 데이터에서 각각의 이미지는 32 x 32 크기의 작은 컬러 이미지, 즉 32 x 32 x 3 형상(shape)을 가지는 아주 작은 컬러 데이터들로 구성됨

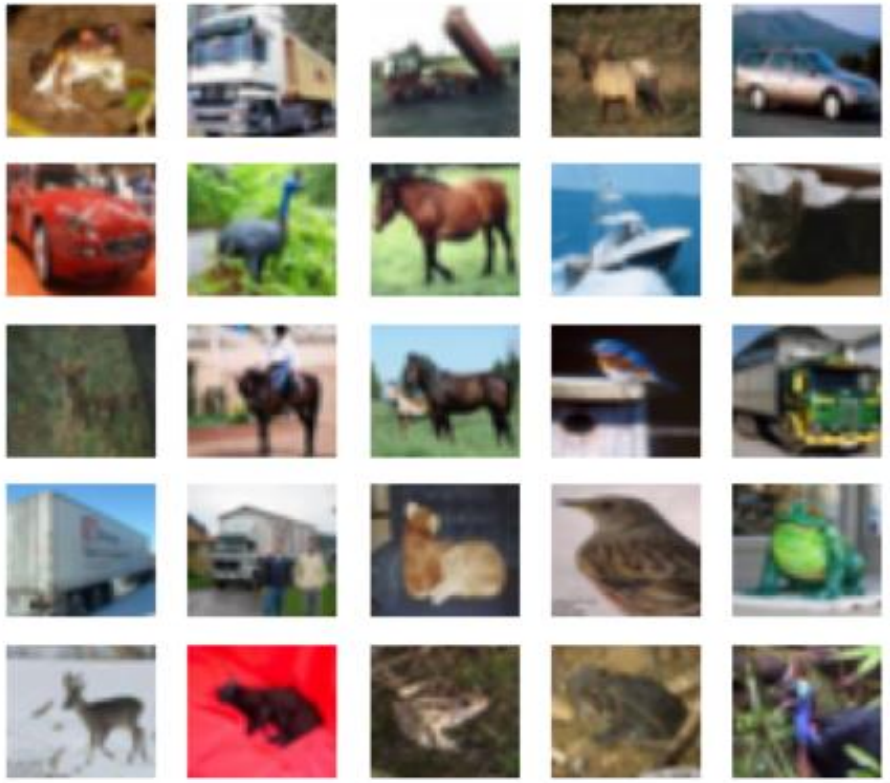
데이터 불러오기 및 확인

```
import numpy as np
import os
import tensorflow as tf
from tensorflow.keras.datasets import cifar10
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from tensorflow.keras.layers import Flatten, Dense, Dropout
from tensorflow.keras.optimizers import Adam
import matplotlib.pyplot as plt
```

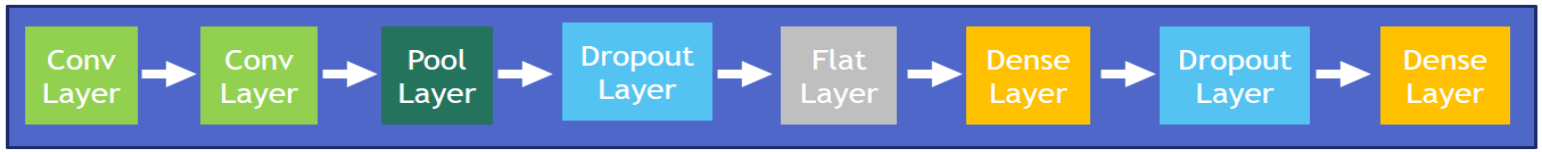
```
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

print(x_train.shape, x_test.shape)
print(y_train.shape, y_test.shape)

x_train = x_train.astype(np.float32) / 255.0
x_test = x_test.astype(np.float32) / 255.0
```

```
(50000, 32, 32, 3) (10000, 32, 32, 3)
(50000, 1) (10000, 1)
```

```
plt.figure(figsize=(6,6))

for index in range(25):

    plt.subplot(5, 5, index+1)
    plt.imshow(x_train[index])
    plt.axis('off')

plt.show()
```



모델 구축




```
cnn = Sequential()

cnn.add(Conv2D(input_shape=(32,32,3), kernel_size=(3,3),
               filters=32, activation='relu'))
cnn.add(Conv2D(kernel_size=(3,3),
               filters=64, activation='relu'))
cnn.add(MaxPool2D(pool_size=(2,2)))
cnn.add(Dropout(0.25))

cnn.add(Flatten())

cnn.add(Dense(128, activation='relu'))
cnn.add(Dropout(0.5))
cnn.add(Dense(10, activation='softmax'))
```

모델 컴파일, 학습

```
cnn.compile(loss='sparse_categorical_crossentropy', optimizer=tf.keras.optimizers.Adam(), metrics=['accuracy'])

cnn.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 32, 32, 32)        896

 conv2d_5 (Conv2D)           (None, 32, 32, 32)        9248

 max_pooling2d_2 (MaxPooling  (None, 16, 16, 32)       0
 2D)

 dropout_3 (Dropout)         (None, 16, 16, 32)        0

 flatten_1 (Flatten)         (None, 8192)              0

 dense_2 (Dense)             (None, 128)               1048704

 dropout_4 (Dropout)         (None, 128)               0

 dense_3 (Dense)             (None, 10)                1290

=================================================================
Total params: 1,060,138
Trainable params: 1,060,138
Non-trainable params: 0
```

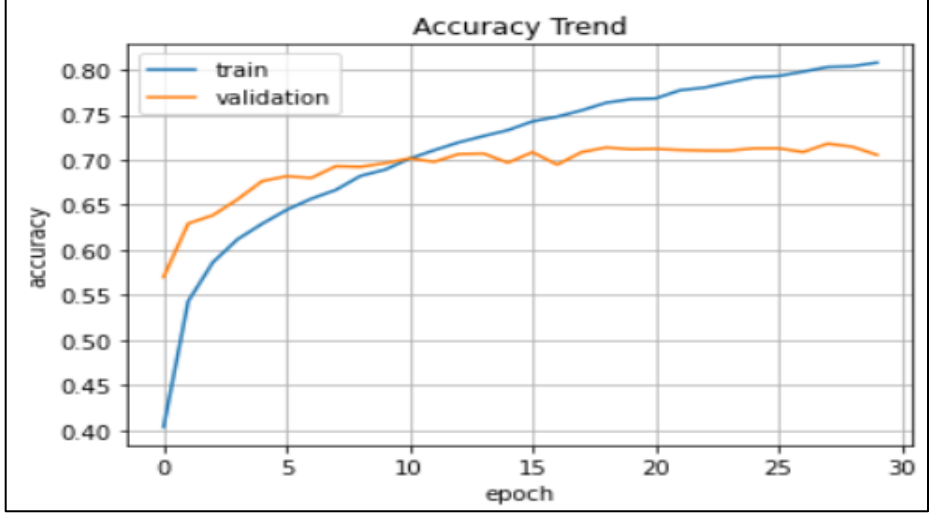
모델 컴파일, 학습

```
hist = cnn.fit(x_train, y_train, batch_size=128, epochs=30, validation_data=(x_test, y_test))
```

```
cnn.evaluate(x_test, y_test)
```

```
313/313 [==============================] - 1s 3ms/step - loss: 0.9591 - accuracy: 0.7056
[0.9591064453125, 0.7056000232696533]
```

```
import matplotlib.pyplot as plt

plt.plot(hist.history['accuracy'])
plt.plot(hist.history['val_accuracy'])
plt.title('Accuracy Trend')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train','validation'], loc='best')
plt.grid()
plt.show()
```




```
plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title('Loss Trend')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train','validation'], loc='best')
plt.grid()
plt.show()
```