기초부터 쌓아가는 딥러닝

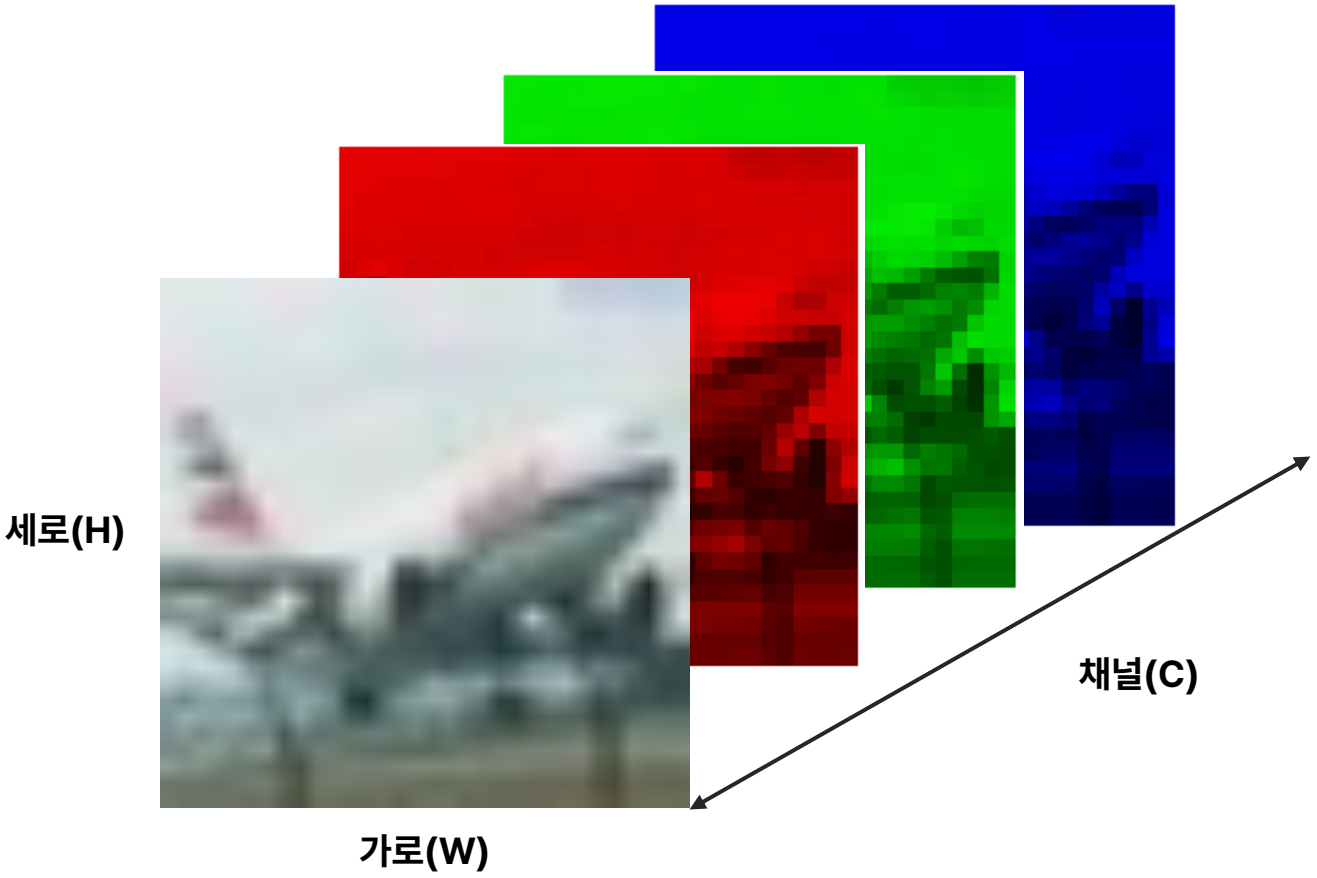
# 01. 이미지 데이터 및 Pytorch 네트워크 이해하기

# 오늘의 목표

1. 이미지 데이터의 3차원 텐서 구조에 대해 이해한다.
2. 이를 통해 "**비정형 데이터가 어떻게 “다차원 텐서”로 표현될 수 있는지** 인지한다.
3. 다차원 텐서가 어떻게 신경망에 입력되고 처리돼 결과로 출력되는지 확인한다.

# 01. 이미지 데이터 이해하기

- 이미지 데이터는 너비(W),높이(H), 채널(C)인 숫자로 표현 할 수 있다.
- 채널 : 이미지는 RGB 3가지 색깔의 조합으로 다양한 색상을 표현할 수 있음

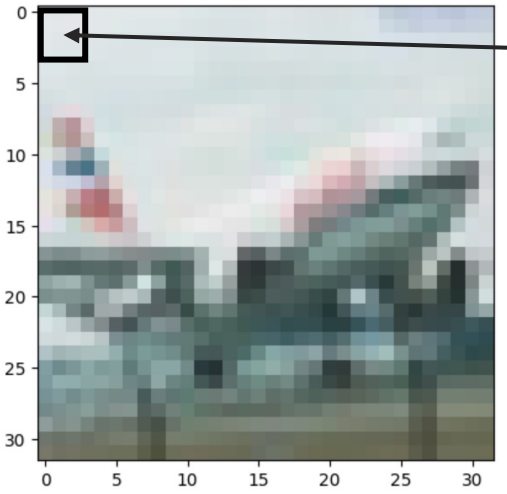



# 01. 이미지 데이터 이해하기

이미지는 32 X 32 형태로 총 32픽셀x32픽셀 크기로 총 1024의 픽셀로 구성되어 있음

**가로(W)**
**32**

**세로(H)**
**32**



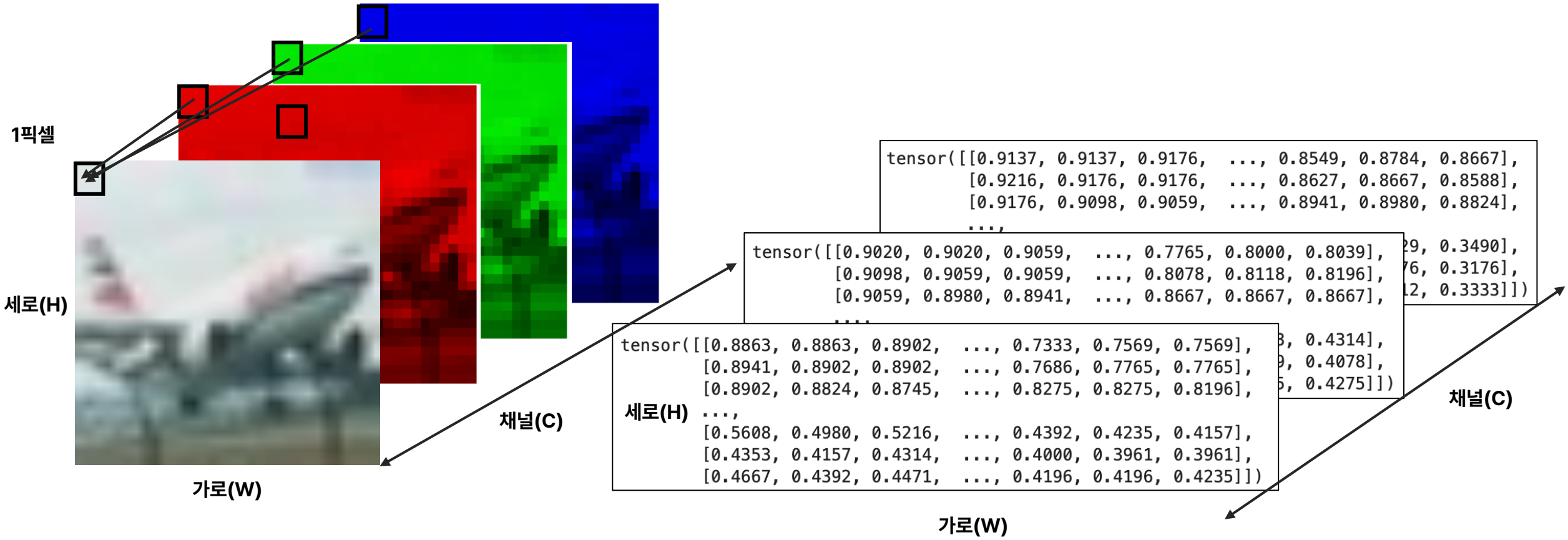

**가로(W)**
**32열**

**세로(H)**
**32줄**

```
tensor([[0.8863, 0.8863, 0.8902, 0.8902,  ..., 0.7412, 0.7333, 0.7569, 0.7569],
        [0.8941, 0.8902, 0.8902, 0.8902,  ..., 0.7843, 0.7686, 0.7765, 0.7765],
        [0.8902, 0.8824, 0.8745, 0.8745,  ..., 0.8235, 0.8275, 0.8275, 0.8196],
        [0.8784, 0.8706, 0.8706, 0.8667,  ..., 0.8353, 0.8314, 0.8314, 0.8235],
        ...,
        [0.4549, 0.4784, 0.5020, 0.5333,  ..., 0.5137, 0.5137, 0.5137, 0.5098],
        [0.5608, 0.4980, 0.5216, 0.4588,  ..., 0.4431, 0.4392, 0.4235, 0.4157],
        [0.4353, 0.4157, 0.4314, 0.4157,  ..., 0.3961, 0.4000, 0.3961, 0.3961],
        [0.4667, 0.4392, 0.4471, 0.4353,  ..., 0.4235, 0.4196, 0.4196, 0.4235]])
```

# 01. 이미지 데이터 이해하기

32x32 이미지가 채널별로 3개가 존재함. 우리가 보는 이미지는 사실은 3겹의 데이터로 표현이 가능함.



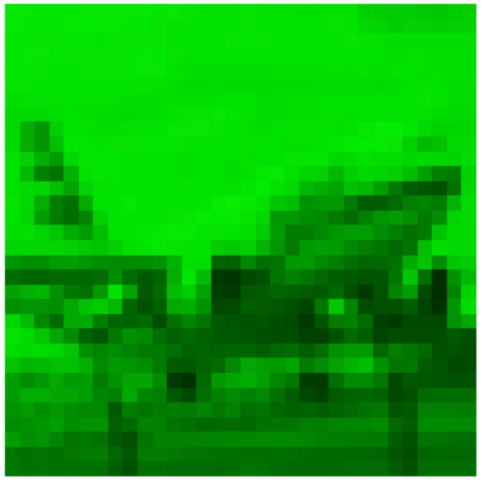
# 02. 이미지를 데이터로 표현한다면?

- 따라서 R채널을 제외하고 모두 0으로 만든다면 아래와 같이 붉은색만 존재하는 그림으로 표현됨.

```
tensor([[0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
tensor([[0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
tensor([[0.8863, 0.8863, 0.8902,  ..., 0.7333, 0.7569, 0.7569],
        [0.8941, 0.8902, 0.8902,  ..., 0.7686, 0.7765, 0.7765],
        [0.8902, 0.8824, 0.8745,  ..., 0.8275, 0.8275, 0.8196],
        ...,
        [0.5608, 0.4980, 0.5216,  ..., 0.4392, 0.4235, 0.4157],
        [0.4353, 0.4157, 0.4314,  ..., 0.4000, 0.3961, 0.3961],
        [0.4667, 0.4392, 0.4471,  ..., 0.4196, 0.4196, 0.4235]])
```

# 02. 이미지를 데이터로 표현한다면?

- 마찬가지로 G채널을 제외하고 모두 0으로 만든다면 아래와 같이 초록색만 존재하는 그림으로 표현됨.



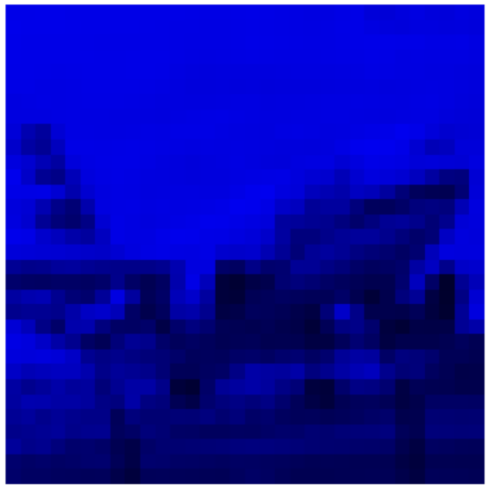
```
tensor([[0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
```

```
tensor([[0.9020, 0.9020, 0.9059,  ..., 0.7765, 0.8000, 0.8039],
        [0.9098, 0.9059, 0.9059,  ..., 0.8078, 0.8118, 0.8196],
                                       0.8667, 0.8667, 0.8667],
                                                               ])
                                       0.4510, 0.4353, 0.4314],
                                       0.4078, 0.4039, 0.4078],
                                       0.4275, 0.4275, 0.4275]])
```

```
tensor([[0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
        ...,
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.]])
```

# 02. 이미지를 데이터로 표현한다면?

- 마찬가지로 B채널을 제외하고 모두 0으로 만든다면 아래와 같이 파랑색만 존재하는 그림으로 표현됨.

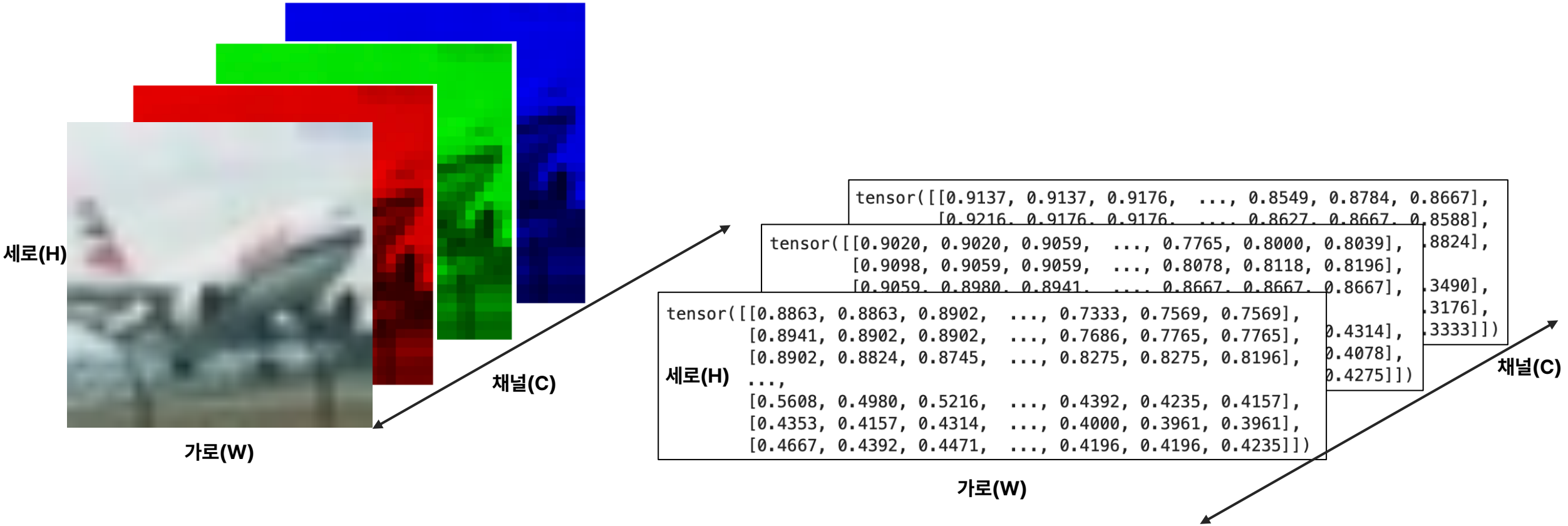


```
tensor([[0.9137, 0.9137, 0.9176,  ..., 0.8549, 0.8784, 0.8667],
        [0.9216, 0.9176, 0.9176,  ..., 0.8627, 0.8667, 0.8588],
                                                0.8941, 0.8980, 0.8824],

tensor([[0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],                0.3686, 0.3529, 0.3490],
                                                .],     0.3216, 0.3176, 0.3176],
tensor([[0., 0., 0.,  ..., 0., 0., 0.],                0.3451, 0.3412, 0.3333]])
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.],                .],
        ...,                                            .],
        [0., 0., 0.,  ..., 0., 0., 0.],                .]])
        [0., 0., 0.,  ..., 0., 0., 0.],
        [0., 0., 0.,  ..., 0., 0., 0.]])
```

# 02. 이미지를 데이터로 표현한다면?

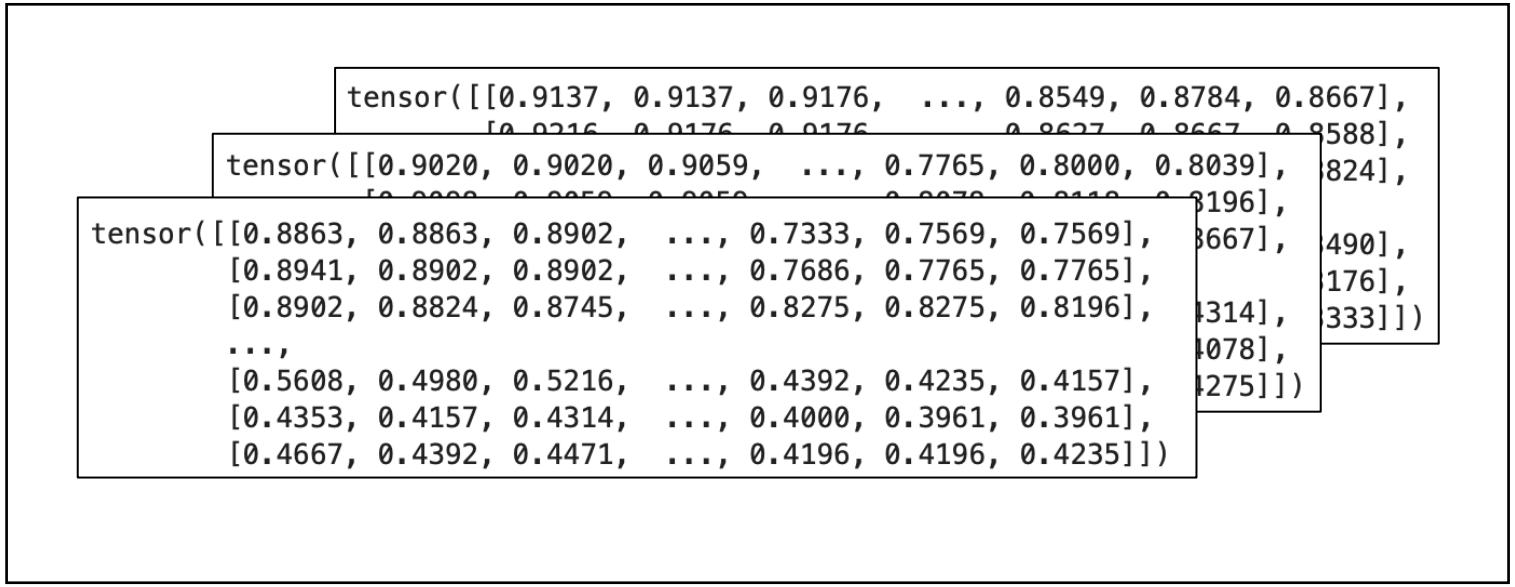
- 즉 비정형데이터는 일정형식이 존재하는 숫자로 표현이 가능하며,
- 그 중 이미지는 가로(W), 높이(H), 채널(C)이라는 3차원의 어레이(Array)로 숫자표현이 가능



# 03. 이미지의 데이터표현 텐서(Tensor)

- 이미지는 가로(W), 높이(H), 채널(C)이라는 3차원의 어레이(Array)로 숫자표현이 가능
- PyTorch,Tensorflow와 같은 딥러닝 프레임워크는 텐서(Tensor)라는 다차원 Array 자료형으로 처리하고 계산함.

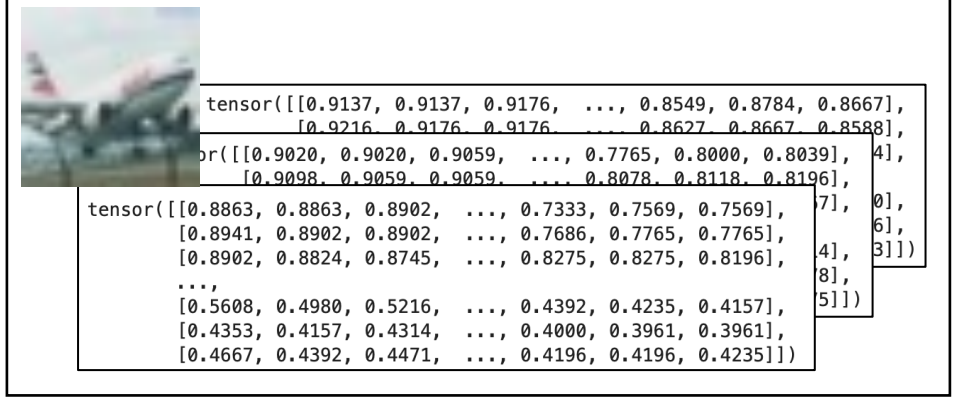
**텐서(Tensor)**



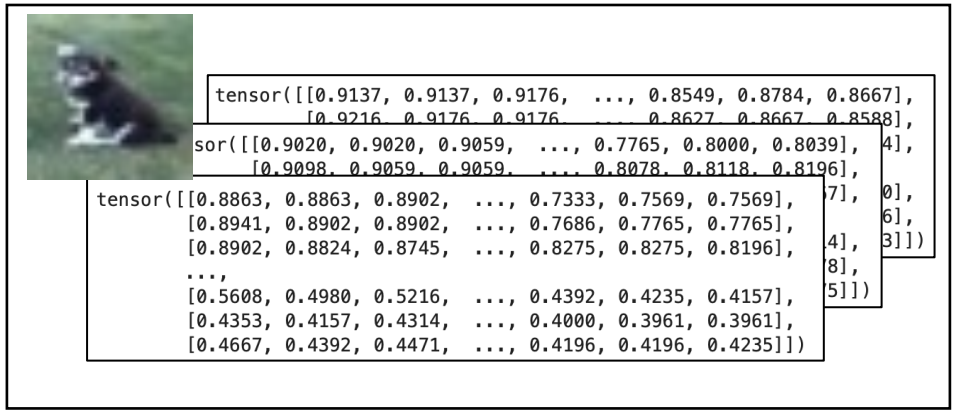
# 03. 이미지의 데이터표현 텐서(Tensor)

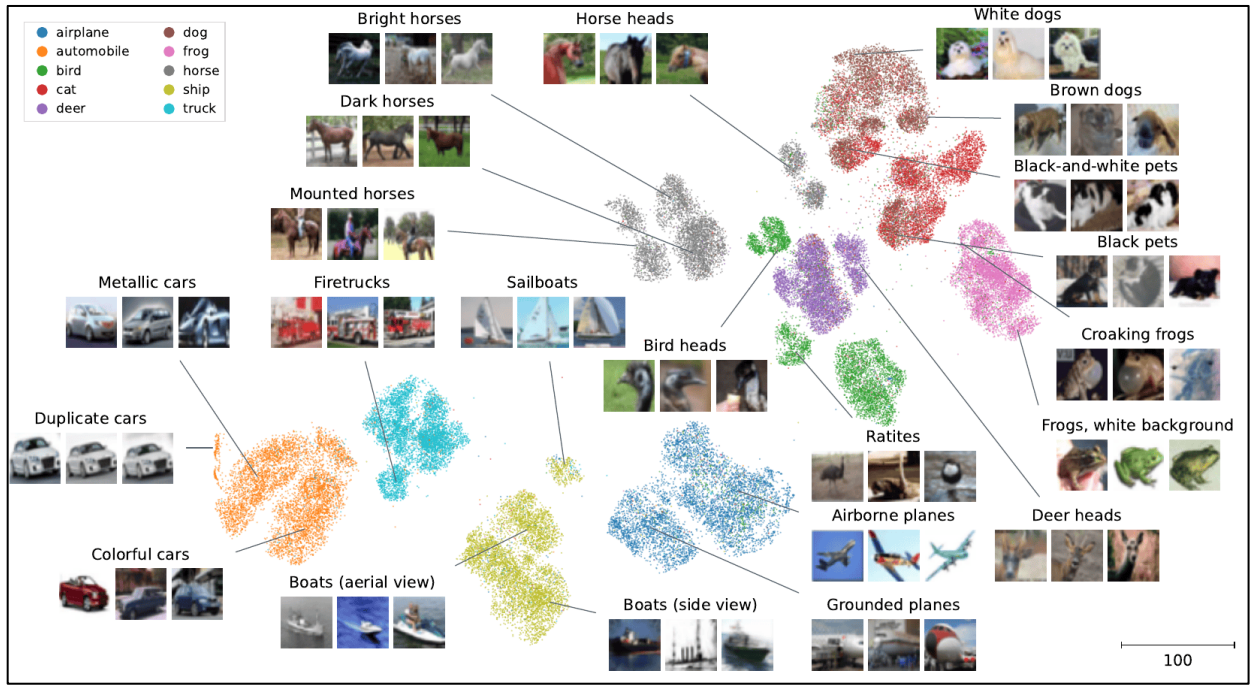
- 이미지를 텐서라는 다차원 Array로 표현이 가능하다는 것은 이미지를 좌표 평면 상에 한 점으로 표시가 가능하다는 것
- 좌표 평면 상에 한 점으로 표시가 가능한 것은 계산이 가능하다는 것

텐서(Tensor)




텐서(Tensor)




https://www.arxiv-vanity.com/papers/2210.09879/