CIFAR-10: Image Dataset

Throughout this course, we will teach you all basic skills and how to use all neccessary tools that you need to implement deep neural networks, which is the main focus of this class. However, you should also be proficient with handling data and know how to prepare it for your specific task. In fact, most of the jobs that involve deep learning in industry are very data related so this is an important skill that you have to pick up.

Therefore, we will take a deep dive into data preparation this week by implementing our own datasets and dataloader. In this notebook, we will focus on the image dataset CIFAR-10. The CIFAR-10 dataset consists of 50000 32x32 colour images in 10 classes, which are *plane*, *car*, *bird*, *cat*, *deer*, *dog*, *frog*, *horse*, *ship*, *truck*.

Let's start by importing some libraries that you will need along the way, as well as some code files that you will work on throughout this notebook.

In [1]:

```
import os
import pickle

import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
from tqdm import tqdm

from exercise_code.data import (
    ImageFolderDataset,
    RescaleTransform,
    NormalizeTransform,
    ComposeTransform,
    compute_image_mean_and_std,
)
from exercise_code.tests import (
    test_image_folder_dataset,
    test_rescale_transform,
    test_compute_image_mean_and_std,
    test_len_dataset,
    test_item_dataset,
    test_transform_dataset,
    save_pickle
)

%load_ext autoreload
%autoreload 2
%matplotlib inline

plt.rcParams['figure.figsize'] = (10.0, 8.0) # set default size of plots
plt.rcParams['image.interpolation'] = 'nearest'
plt.rcParams['image.cmap'] = 'gray'
```


1. Dataset Download

Let us get started by downloading the data. In `exercise_code/data/image_folder_dataset.py` you can find a class `ImageFolderDataset` , which you will have to complete throughout this notebook.

This class automatically downloads the raw data for you. To do so, simply initialize the class as below:

In [2]:

```
# Set up the output dataset folder
i2dl_exercises_path = os.path.dirname(os.path.abspath(os.getcwd()))
cifar_root = os.path.join(i2dl_exercises_path, "datasets", "cifar10")

# Init the dataset and display downloading information this one time
dataset = ImageFolderDataset(
    root=cifar_root,
    force_download=True,
    verbose=True
)
```

```
  0%|          | 0/120720579 [00:00<?, ?it/s]

Downloading cifar10.zip
Downloading https://cdn3.vision.in.tum.de/~dl4cv/cifar10.zip to /Users/meiqiliu/Downloads/datasets/cifar10/cifar10.zip

 99%|██████████| 119775232/120720579 [00:12<00:00, 9667932.92it/s]

Extracting cifar10.zip
Dataset successfully downloaded! Stored under: /Users/meiqiliu/Downloads/datasets/cifar10
```

You should now be able to see the images in `i2dl_exercises/datasets/cifar10` in your file browser, which should contain one subfolder per class, each containing the respective images labeled `0001.png` , `0002.png` , ...

By default, the dataset will only be downloaded the first time you initialize a dataset class. If, for some reason, your version of the dataset gets corrupted and you wish to re-download it, simply initialize the class with `force_download=True` in the download cell above.

2. Data Visualization

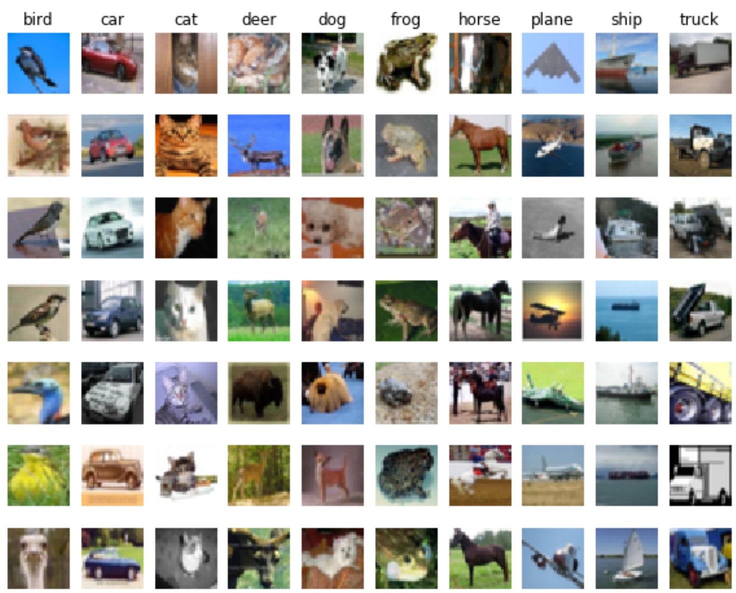
Before training any model you should *always* take a look at some samples of your dataset. In this way, you can make sure that the data input has worked as intended and also get a feeling for the dataset.

Let's load the CIFAR-10 data and visualize a subset of the images. To do so, `PIL.Image.open()` is used to open an image, and then `numpy.asarray()` to cast the image to a numpy array, which will have shape 32x32x3. In this way 7 images will be loaded per class, and then use `matplotlib.pyplot` to visualize those images in a grid.

In [3]:
```python
def load_image_as_numpy(image_path):
    return np.asarray(Image.open(image_path), dtype=float)

classes = [
    'plane', 'car', 'bird', 'cat', 'deer',
    'dog', 'frog', 'horse', 'ship', 'truck',
]
num_classes = len(classes)
samples_per_class = 7
for label, cls in enumerate(sorted(classes)):
    for i in range(samples_per_class):
        image_path = os.path.join(
            cifar_root,
            cls,
            str(i+1).zfill(4) + ".png"
        )  # e.g. cifar10/plane/0001.png
        image = np.asarray(Image.open(image_path))  # open image as nump
y array
        plt_idx = i * num_classes + label + 1  # calculate plot location
in the grid
        plt.subplot(samples_per_class, num_classes, plt_idx)
        plt.imshow(image.astype('uint8'))
        plt.axis('off')
        if i == 0:
            plt.title(cls)  # plot class names above columns
plt.show()
```

```
120725504it [00:30, 9667932.92it/s]
```



3. ImageFolderDataset Implementation

Loading images following steps above is a bit cumbersome. Therefore, the next step is to write a custom **Dataset** class, which takes care of the data loading. This is always the first thing you have to implement when starting a new deep learning project.

3.1 Dataset Class

The **Dataset** class is a wrapper that loads the data from a given file path and returns a dictionary containing already prepared data, as you have done above. Datasets always need to have the following two methods implemented:

- `__len__(self)` is a method that should simply calculate and return the number of images in the dataset. After it is implemented, you can simply call it with `len(dataset)`.
- `__getitem__(self, index)` should return the image with the given index from your dataset. Implementing this will allow you to access your dataset like a list, i.e. you can then simply call `dataset[9]` to access the 10th image in the dataset.

Generally, you will have to implement a different dataset for every project. However, base dataset classes for future projects will be provided for you in future projects.

3.2 ImageFolderDataset Implementation

Now it is your turn to implement such a dataset class for CIFAR-10. To do so, open `exercise_code/data/image_folder_dataset.py` and check the following three methods of `ImageFolderDataset`:

- `make_dataset(directory, class_to_idx)` should load the prepared data from a given directory root (`directory`) into two lists (`images` and `labels`). `class_to_idx` is a dict mapping class (e.g. 'cat') to label (e.g. 1).
- `__len__(self)` should calculate and return the number of images in your dataset.
- `__getitem__(self, index)` should return the image with the given index from your dataset.

Task: Check Code

Please read `make_dataset(directory, class_to_idx)` and make sure to familiarize with its output as you will need to interact with it for the following tasks. Additionally, it would be a wise decision to get familiar with python's os library which will be of utmost importance for most datasets you will write in future projects. As it is not beginner friendly, we removed it for this exercise but it is an important skill for a DL practicer.

Task: Implement

Implement the `__len__(self)` method in `exercise_code/data/image_folder_dataset.py` and test your implementation by running the following cell.

In [4]:
```
dataset = ImageFolderDataset(
    root=cifar_root,
)

_ = test_len_dataset(dataset)
```

```
LenTestInt passed.
LenTestCorrect passed.
Method __len__() correctly implemented. Tests passed: 2/2
Score: 100/100
```

Task: Implement

Implement the `__getitem__(self, index)` method in `exercise_code/data/image_folder_dataset.py` and test your implementation by running the following cell.

Hint: You may want to reuse parts of the '2. Data Visualization' code above in your implementation of `__getitem__()` .

In [5]:
```
dataset = ImageFolderDataset(
    root=cifar_root,
)

_ = test_item_dataset(dataset)
```

```
GetItemTestType passed.
GetItemTestImageShape passed.
Method __getitem__() correctly implemented. Tests passed: 2/2
Score: 100/100
```

3.3 Dataset Usage

Now that you have implemented all required parts of the ImageFolderDataset, using the `__getitem__()` method, you can now access our dataset as conveniently as you would access a list:

In [6]:
```
sample_item = dataset[0]
sample_image = sample_item["image"]
sample_label = sample_item["label"]

print('Sample image shape:', sample_image.shape)
print('Sample label:', sample_label)
print('Sample image first values:', sample_image[0][0])
```

```
Sample image shape: (32, 32, 3)
Sample label: 0
Sample image first values: [  3. 125. 233.]
```

As you can see, the images are represented as uint8 values for each of the three RGB color channels. The data type and scale will be important later.

As you have implemented both `__len__()` and `__getitem__()` , you can now even iterate over the dataset with a simple for loop!

In [7]:

```python
num_samples = 0
for sample in tqdm(dataset):
    num_samples += 1

print("Number of samples:", num_samples)
```


6/2/2021 1_cifar10-image-dataset

```
  0%|          | 0/50000 [00:00<?, ?it/s]
  1%|          | 429/50000 [00:00<00:11, 4288.83it/s]
  2%|          | 868/50000 [00:00<00:11, 4318.34it/s]
  3%|          | 1320/50000 [00:00<00:11, 4376.44it/s]
  4%|          | 1759/50000 [00:00<00:11, 4378.02it/s]
  4%|          | 2120/50000 [00:00<00:11, 4114.10it/s]
  5%|          | 2502/50000 [00:00<00:11, 4021.13it/s]
  6%|          | 2939/50000 [00:00<00:11, 4117.90it/s]
  7%|          | 3377/50000 [00:00<00:11, 4190.05it/s]
  8%|          | 3833/50000 [00:00<00:10, 4292.57it/s]
  8%|          | 4248/50000 [00:01<00:10, 4242.25it/s]
  9%|          | 4673/50000 [00:01<00:10, 4243.03it/s]
 10%|          | 5090/50000 [00:01<00:10, 4218.69it/s]
 11%|          | 5524/50000 [00:01<00:10, 4252.82it/s]
 12%|          | 5959/50000 [00:01<00:10, 4280.13it/s]
 13%|          | 6385/50000 [00:01<00:10, 4232.67it/s]
 14%|          | 6821/50000 [00:01<00:10, 4269.94it/s]
 15%|          | 7259/50000 [00:01<00:09, 4301.33it/s]
 15%|          | 7692/50000 [00:01<00:09, 4308.15it/s]
 16%|          | 8132/50000 [00:01<00:09, 4333.58it/s]
 17%|          | 8566/50000 [00:02<00:09, 4301.76it/s]
 18%|          | 8997/50000 [00:02<00:09, 4183.18it/s]
 19%|          | 9416/50000 [00:02<00:10, 4036.29it/s]
 20%|          | 9822/50000 [00:02<00:10, 3839.90it/s]
 20%|          | 10210/50000 [00:02<00:11, 3442.87it/s]
 21%|          | 10565/50000 [00:02<00:11, 3375.47it/s]
 22%|          | 10918/50000 [00:02<00:11, 3417.89it/s]
 23%|          | 11265/50000 [00:02<00:13, 2976.71it/s]
 23%|          | 11577/50000 [00:02<00:13, 2783.82it/s]
 24%|          | 11945/50000 [00:03<00:12, 3002.32it/s]
 25%|          | 12294/50000 [00:03<00:12, 3127.87it/s]
 25%|          | 12639/50000 [00:03<00:11, 3216.37it/s]
 26%|          | 12969/50000 [00:03<00:11, 3238.78it/s]
 27%|          | 13340/50000 [00:03<00:10, 3366.38it/s]
 27%|          | 13697/50000 [00:03<00:10, 3424.47it/s]
 28%|          | 14059/50000 [00:03<00:10, 3480.18it/s]
 29%|          | 14411/50000 [00:03<00:10, 3468.73it/s]
 30%|          | 14836/50000 [00:03<00:09, 3670.95it/s]
 31%|          | 15287/50000 [00:03<00:08, 3886.14it/s]
 31%|          | 15741/50000 [00:04<00:08, 4061.07it/s]
 32%|          | 16156/50000 [00:04<00:08, 4087.00it/s]
 33%|          | 16593/50000 [00:04<00:08, 4166.53it/s]
 34%|          | 17014/50000 [00:04<00:08, 3844.13it/s]
 35%|          | 17406/50000 [00:04<00:08, 3657.65it/s]
 36%|          | 17779/50000 [00:04<00:09, 3561.81it/s]
 36%|          | 18141/50000 [00:04<00:09, 3444.38it/s]
 37%|          | 18491/50000 [00:04<00:09, 3453.11it/s]
 38%|          | 18840/50000 [00:04<00:09, 3440.93it/s]
 38%|          | 19187/50000 [00:05<00:09, 3411.12it/s]
 39%|          | 19530/50000 [00:05<00:08, 3406.07it/s]
 40%|          | 19887/50000 [00:05<00:08, 3452.92it/s]
 40%|          | 20234/50000 [00:05<00:08, 3423.36it/s]
 41%|          | 20578/50000 [00:05<00:08, 3427.67it/s]
 42%|          | 20922/50000 [00:05<00:08, 3423.16it/s]
 43%|          | 21283/50000 [00:05<00:08, 3474.68it/s]
 43%|          | 21631/50000 [00:05<00:08, 3468.71it/s]
 44%|          | 22026/50000 [00:05<00:07, 3599.54it/s]
```

```
 45%|████      | 22461/50000 [00:05<00:07, 3794.07it/s]
 46%|████      | 22845/50000 [00:06<00:07, 3742.76it/s]
 47%|████      | 23268/50000 [00:06<00:06, 3876.69it/s]
 47%|████      | 23712/50000 [00:06<00:06, 4029.34it/s]
 48%|████      | 24119/50000 [00:06<00:06, 3951.54it/s]
 49%|████      | 24518/50000 [00:06<00:06, 3816.67it/s]
 50%|█████     | 24903/50000 [00:06<00:07, 3555.71it/s]
 51%|█████     | 25265/50000 [00:06<00:07, 3476.67it/s]
 51%|█████     | 25618/50000 [00:06<00:06, 3485.03it/s]
 52%|█████     | 26052/50000 [00:06<00:06, 3703.41it/s]
 53%|█████     | 26471/50000 [00:07<00:06, 3836.42it/s]
 54%|█████     | 26863/50000 [00:07<00:05, 3859.02it/s]
 55%|█████     | 27253/50000 [00:07<00:06, 3691.98it/s]
 55%|█████     | 27629/50000 [00:07<00:06, 3711.31it/s]
 56%|█████     | 28004/50000 [00:07<00:06, 3643.91it/s]
 57%|█████     | 28371/50000 [00:07<00:06, 3397.79it/s]
 57%|█████     | 28716/50000 [00:07<00:06, 3321.02it/s]
 58%|█████     | 29102/50000 [00:07<00:06, 3465.81it/s]
 59%|█████     | 29465/50000 [00:07<00:05, 3511.95it/s]
 60%|██████    | 29849/50000 [00:07<00:05, 3602.96it/s]
 61%|██████    | 30272/50000 [00:08<00:05, 3769.65it/s]
 61%|██████    | 30706/50000 [00:08<00:04, 3923.03it/s]
 62%|██████    | 31155/50000 [00:08<00:04, 4076.69it/s]
 63%|██████    | 31586/50000 [00:08<00:04, 4142.30it/s]
 64%|██████    | 32016/50000 [00:08<00:04, 4187.85it/s]
 65%|██████    | 32438/50000 [00:08<00:04, 4117.46it/s]
 66%|██████    | 32852/50000 [00:08<00:04, 3922.02it/s]
 66%|██████    | 33248/50000 [00:08<00:04, 3806.86it/s]
 67%|██████    | 33632/50000 [00:08<00:04, 3761.75it/s]
 68%|██████    | 34011/50000 [00:09<00:04, 3650.80it/s]
 69%|██████    | 34379/50000 [00:09<00:04, 3301.32it/s]
 69%|██████    | 34718/50000 [00:09<00:05, 2646.72it/s]
 70%|███████   | 35017/50000 [00:09<00:05, 2740.53it/s]
 71%|███████   | 35352/50000 [00:09<00:05, 2898.67it/s]
 71%|███████   | 35702/50000 [00:09<00:04, 3053.82it/s]
 72%|███████   | 36056/50000 [00:09<00:04, 3182.73it/s]
 73%|███████   | 36387/50000 [00:09<00:04, 3217.64it/s]
 73%|███████   | 36717/50000 [00:09<00:04, 3118.27it/s]
 74%|███████   | 37035/50000 [00:10<00:04, 2678.82it/s]
 75%|███████   | 37320/50000 [00:10<00:04, 2727.95it/s]
 75%|███████   | 37604/50000 [00:10<00:04, 2724.10it/s]
 76%|███████   | 37946/50000 [00:10<00:04, 2899.89it/s]
 76%|███████   | 38245/50000 [00:10<00:04, 2645.60it/s]
 77%|███████   | 38556/50000 [00:10<00:04, 2768.57it/s]
 78%|███████   | 38910/50000 [00:10<00:03, 2960.93it/s]
 79%|███████   | 39272/50000 [00:10<00:03, 3131.47it/s]
 79%|███████   | 39647/50000 [00:10<00:03, 3294.05it/s]
 80%|████████  | 40023/50000 [00:11<00:02, 3421.05it/s]
 81%|████████  | 40385/50000 [00:11<00:02, 3477.09it/s]
 81%|████████  | 40748/50000 [00:11<00:02, 3521.14it/s]
 82%|████████  | 41105/50000 [00:11<00:02, 3428.78it/s]
 83%|████████  | 41454/50000 [00:11<00:02, 3445.68it/s]
 84%|████████  | 41812/50000 [00:11<00:02, 3483.54it/s]
 84%|████████  | 42167/50000 [00:11<00:02, 3502.01it/s]
 85%|████████  | 42530/50000 [00:11<00:02, 3539.17it/s]
 86%|████████  | 42885/50000 [00:11<00:02, 3539.31it/s]
 86%|████████  | 43248/50000 [00:11<00:01, 3565.60it/s]
```



```
 87%|████████  | 43622/50000 [00:12<00:01, 3615.08it/s]
 88%|████████  | 43992/50000 [00:12<00:01, 3638.58it/s]
 89%|████████  | 44357/50000 [00:12<00:01, 3611.64it/s]
 89%|████████  | 44739/50000 [00:12<00:01, 3670.61it/s]
 90%|█████████ | 45110/50000 [00:12<00:01, 3681.58it/s]
 91%|█████████ | 45479/50000 [00:12<00:01, 3663.94it/s]
 92%|█████████ | 45846/50000 [00:12<00:01, 3539.11it/s]
 92%|█████████ | 46202/50000 [00:12<00:01, 3090.12it/s]
 93%|█████████ | 46523/50000 [00:12<00:01, 3084.79it/s]
 94%|█████████ | 46840/50000 [00:13<00:01, 3036.05it/s]
 94%|█████████ | 47153/50000 [00:13<00:00, 3062.56it/s]
 95%|█████████ | 47486/50000 [00:13<00:00, 3134.94it/s]
 96%|█████████ | 47803/50000 [00:13<00:00, 2938.41it/s]
 96%|█████████ | 48102/50000 [00:13<00:00, 2781.46it/s]
 97%|█████████ | 48421/50000 [00:13<00:00, 2892.08it/s]
 98%|█████████ | 48764/50000 [00:13<00:00, 3031.94it/s]
 98%|█████████ | 49076/50000 [00:13<00:00, 3055.42it/s]
 99%|█████████ | 49386/50000 [00:13<00:00, 2929.65it/s]
 99%|█████████ | 49683/50000 [00:14<00:00, 2877.20it/s]
100%|██████████| 50000/50000 [00:14<00:00, 3534.39it/s]

Number of samples: 50000
```

4. Transforms and Image Preprocessing

Before training machine learning models, you often need to pre-process the data. For image datasets, two commonly applied techniques are:

1. Normalize all images so that each value is either in [-1, 1] or [0, 1]. By doing so the image are also converted to floating point numbers.
2. Compute the mean over all images and subtract this mean from all images in the dataset

These transform classes are callables, meaning that you will be able to simply use them as follows:

```
transform = Transform()
images_transformed = transform(images)
```

This will be realized in the pipeline by defining so called transforms. Instead of applying them globally to the input data, you will apply those seperatly to each sample after loading it in the `__getitem__` call of the dataset.

Task: Implement

Modify the `__getitem__(self, index)` method in `exercise_code/data/image_folder_dataset.py` such that it applies `self.transform` . With this change you can simply define the transforms during dataset creation and apply those automatically for each `__getitem__` call. Make sure not to break it though ;).

In [8]:
```
dataset = ImageFolderDataset(
    root=cifar_root,
)

_ = test_transform_dataset(dataset)
```

```
GetItemTestType passed.
GetItemTestImageShape passed.
GetItemTestTransformApplied passed.
Method __getitem__() correctly implemented. Tests passed: 3/3
Score: 100/100
```

Equipped with this change, you can now easily add the two preprocessing techniques above for CIFAR-10. You will do so in the following steps by implementing the classes `RescaleTransform` and `NormalizeTransform` in `exercise_code/data/transforms.py` .

4.1 Rescaling Images using RescaleTransform

Let's start by implementing `RescaleTransform` . If you look at the `__init__()` method, you will notice it has four arguments:

- **out_range** is the range you wish to rescale your images to. E.g. if you want to scale your images to [-1, 1], you would use `range=(-1, 1)` . By default, they will be scaled to [0, 1].
- **in_range** is the value range of the data prior to rescaling. For uint8 images, this will always be (0, 255).

Task: Implement

Implement the `__call__()` method of `RescaleTransform` in `exercise_code/data/transforms.py` and test your implementation by running the following cell.

In [9]:
```
rescale_transform = RescaleTransform()
dataset_rescaled = ImageFolderDataset(
    root=cifar_root,
    transform=rescale_transform
)

_ = test_rescale_transform(dataset_rescaled)
```

```
  0%|          | 0/50000 [00:00<?, ?it/s]
  1%|          | 299/50000 [00:00<00:16, 2989.25it/s]
  1%|          | 599/50000 [00:00<00:16, 2991.20it/s]
  2%|          | 907/50000 [00:00<00:16, 3015.53it/s]
  2%|          | 1245/50000 [00:00<00:15, 3115.71it/s]
  3%|          | 1530/50000 [00:00<00:16, 3029.21it/s]
  4%|          | 1784/50000 [00:00<00:17, 2719.36it/s]
  4%|          | 2047/50000 [00:00<00:17, 2691.47it/s]
  5%|          | 2344/50000 [00:00<00:17, 2766.89it/s]
  5%|          | 2645/50000 [00:00<00:16, 2832.48it/s]
  6%|          | 2925/50000 [00:01<00:16, 2819.95it/s]
  6%|          | 3212/50000 [00:01<00:16, 2831.68it/s]
  7%|          | 3491/50000 [00:01<00:17, 2674.70it/s]
  8%|          | 3758/50000 [00:01<00:17, 2653.44it/s]
  8%|          | 4023/50000 [00:01<00:17, 2576.68it/s]
  9%|          | 4320/50000 [00:01<00:17, 2681.83it/s]
  9%|          | 4630/50000 [00:01<00:16, 2794.58it/s]
 10%|          | 4937/50000 [00:01<00:15, 2871.73it/s]
 10%|          | 5246/50000 [00:01<00:15, 2932.01it/s]
 11%|          | 5551/50000 [00:01<00:14, 2965.73it/s]
 12%|          | 5849/50000 [00:02<00:15, 2882.44it/s]
 12%|          | 6139/50000 [00:02<00:15, 2848.17it/s]
 13%|          | 6425/50000 [00:02<00:15, 2808.33it/s]
 13%|          | 6707/50000 [00:02<00:15, 2755.36it/s]
 14%|          | 6984/50000 [00:02<00:15, 2728.14it/s]
 15%|          | 7275/50000 [00:02<00:15, 2777.30it/s]
 15%|          | 7568/50000 [00:02<00:15, 2819.36it/s]
 16%|          | 7864/50000 [00:02<00:14, 2859.58it/s]
 16%|          | 8155/50000 [00:02<00:14, 2873.77it/s]
 17%|          | 8443/50000 [00:02<00:14, 2871.88it/s]
 17%|          | 8747/50000 [00:03<00:14, 2919.78it/s]
 18%|          | 9056/50000 [00:03<00:13, 2968.18it/s]
 19%|          | 9354/50000 [00:03<00:13, 2956.19it/s]
 19%|          | 9650/50000 [00:03<00:13, 2932.28it/s]
 20%|          | 9944/50000 [00:03<00:15, 2532.97it/s]
 20%|          | 10245/50000 [00:03<00:14, 2658.29it/s]
 21%|          | 10546/50000 [00:03<00:14, 2754.42it/s]
 22%|          | 10829/50000 [00:03<00:14, 2704.30it/s]
 22%|          | 11110/50000 [00:03<00:14, 2734.41it/s]
 23%|          | 11388/50000 [00:04<00:14, 2711.13it/s]
 23%|          | 11662/50000 [00:04<00:14, 2623.79it/s]
 24%|          | 11927/50000 [00:04<00:14, 2611.81it/s]
 24%|          | 12198/50000 [00:04<00:14, 2639.72it/s]
 25%|          | 12464/50000 [00:04<00:14, 2587.29it/s]
 25%|          | 12726/50000 [00:04<00:14, 2592.61it/s]
 26%|          | 13014/50000 [00:04<00:13, 2669.72it/s]
 27%|          | 13286/50000 [00:04<00:13, 2684.06it/s]
 27%|          | 13622/50000 [00:04<00:12, 2853.10it/s]
 28%|          | 13937/50000 [00:04<00:12, 2933.58it/s]
 29%|          | 14253/50000 [00:05<00:11, 2996.87it/s]
 29%|          | 14591/50000 [00:05<00:11, 3101.77it/s]
 30%|          | 14904/50000 [00:05<00:11, 3107.21it/s]
 30%|          | 15217/50000 [00:05<00:11, 3000.15it/s]
 31%|          | 15520/50000 [00:05<00:11, 2975.81it/s]
 32%|          | 15820/50000 [00:05<00:11, 2956.19it/s]
 32%|          | 16117/50000 [00:05<00:11, 2915.43it/s]
 33%|          | 16430/50000 [00:05<00:11, 2975.91it/s]
```



```
 33%|          | 16742/50000 [00:05<00:11, 3017.05it/s]
 34%|          | 17045/50000 [00:06<00:10, 3014.54it/s]
 35%|          | 17348/50000 [00:06<00:11, 2884.03it/s]
 35%|          | 17638/50000 [00:06<00:11, 2776.03it/s]
 36%|          | 17918/50000 [00:06<00:11, 2745.78it/s]
 36%|          | 18195/50000 [00:06<00:11, 2677.08it/s]
 37%|          | 18515/50000 [00:06<00:11, 2813.70it/s]
 38%|          | 18823/50000 [00:06<00:10, 2888.51it/s]
 38%|          | 19136/50000 [00:06<00:10, 2955.28it/s]
 39%|          | 19452/50000 [00:06<00:10, 3012.65it/s]
 40%|          | 19759/50000 [00:06<00:09, 3028.11it/s]
 40%|          | 20070/50000 [00:07<00:09, 3051.38it/s]
 41%|          | 20390/50000 [00:07<00:09, 3094.23it/s]
 41%|          | 20717/50000 [00:07<00:09, 3141.58it/s]
 42%|          | 21035/50000 [00:07<00:09, 3151.44it/s]
 43%|          | 21351/50000 [00:07<00:09, 3146.63it/s]
 43%|          | 21667/50000 [00:07<00:09, 3094.15it/s]
 44%|          | 21977/50000 [00:07<00:09, 3034.16it/s]
 45%|          | 22283/50000 [00:07<00:09, 3039.71it/s]
 45%|          | 22588/50000 [00:07<00:09, 3035.52it/s]
 46%|          | 22906/50000 [00:07<00:08, 3076.57it/s]
 46%|          | 23234/50000 [00:08<00:08, 3134.23it/s]
 47%|          | 23551/50000 [00:08<00:08, 3144.04it/s]
 48%|          | 23866/50000 [00:08<00:08, 3045.51it/s]
 48%|          | 24175/50000 [00:08<00:08, 3056.07it/s]
 49%|          | 24482/50000 [00:08<00:08, 3031.63it/s]
 50%|          | 24798/50000 [00:08<00:08, 3067.22it/s]
 50%|          | 25115/50000 [00:08<00:08, 3095.37it/s]
 51%|          | 25454/50000 [00:08<00:07, 3177.64it/s]
 52%|          | 25773/50000 [00:08<00:07, 3159.15it/s]
 52%|          | 26091/50000 [00:08<00:07, 3164.97it/s]
 53%|          | 26408/50000 [00:09<00:07, 3135.33it/s]
 53%|          | 26722/50000 [00:09<00:07, 3101.21it/s]
 54%|          | 27034/50000 [00:09<00:07, 3106.60it/s]
 55%|          | 27345/50000 [00:09<00:07, 3088.84it/s]
 55%|          | 27655/50000 [00:09<00:07, 3075.51it/s]
 56%|          | 27963/50000 [00:09<00:07, 3009.06it/s]
 57%|          | 28265/50000 [00:09<00:07, 2736.18it/s]
 57%|          | 28544/50000 [00:09<00:08, 2452.35it/s]
 58%|          | 28847/50000 [00:09<00:08, 2600.27it/s]
 58%|          | 29155/50000 [00:10<00:07, 2727.37it/s]
 59%|          | 29455/50000 [00:10<00:07, 2801.43it/s]
 59%|          | 29742/50000 [00:10<00:07, 2683.19it/s]
 60%|          | 30045/50000 [00:10<00:07, 2777.08it/s]
 61%|          | 30356/50000 [00:10<00:06, 2867.57it/s]
 61%|          | 30648/50000 [00:10<00:06, 2877.53it/s]
 62%|          | 30939/50000 [00:10<00:08, 2223.31it/s]
 62%|          | 31186/50000 [00:10<00:09, 1889.94it/s]
 63%|          | 31401/50000 [00:11<00:11, 1603.76it/s]
 63%|          | 31587/50000 [00:11<00:12, 1453.28it/s]
 64%|          | 31753/50000 [00:11<00:12, 1404.50it/s]
 64%|          | 31909/50000 [00:11<00:13, 1367.31it/s]
 64%|          | 32057/50000 [00:11<00:13, 1311.64it/s]
 64%|          | 32197/50000 [00:11<00:13, 1331.04it/s]
 65%|          | 32336/50000 [00:11<00:14, 1257.85it/s]
 65%|          | 32467/50000 [00:12<00:14, 1235.36it/s]
 65%|          | 32594/50000 [00:12<00:14, 1211.45it/s]
```

```
 65%|| 32718/50000 [00:12<00:14, 1204.40it/s]
 66%|| 32841/50000 [00:12<00:14, 1189.51it/s]
 66%|| 32963/50000 [00:12<00:14, 1195.40it/s]
 66%|| 33084/50000 [00:12<00:14, 1197.68it/s]
 66%|| 33205/50000 [00:12<00:14, 1174.49it/s]
 67%|| 33323/50000 [00:12<00:14, 1174.95it/s]
 67%|| 33441/50000 [00:12<00:14, 1147.52it/s]
 67%|| 33557/50000 [00:12<00:14, 1143.90it/s]
 67%|| 33682/50000 [00:13<00:13, 1173.09it/s]
 68%|| 33809/50000 [00:13<00:13, 1198.26it/s]
 68%|| 33943/50000 [00:13<00:12, 1236.24it/s]
 68%|| 34090/50000 [00:13<00:12, 1297.29it/s]
 68%|| 34229/50000 [00:13<00:11, 1321.80it/s]
 69%|| 34363/50000 [00:13<00:12, 1290.71it/s]
 69%|| 34493/50000 [00:13<00:11, 1292.73it/s]
 69%|| 34623/50000 [00:13<00:12, 1268.60it/s]
 70%|| 34751/50000 [00:13<00:12, 1226.58it/s]
 70%|| 34875/50000 [00:13<00:12, 1213.58it/s]
 70%|| 34997/50000 [00:14<00:12, 1209.15it/s]
 70%|| 35119/50000 [00:14<00:12, 1205.80it/s]
 70%|| 35241/50000 [00:14<00:12, 1209.29it/s]
 71%|| 35381/50000 [00:14<00:11, 1258.77it/s]
 71%|| 35508/50000 [00:14<00:11, 1211.48it/s]
 71%|| 35630/50000 [00:14<00:12, 1178.90it/s]
 71%|| 35749/50000 [00:14<00:12, 1156.19it/s]
 72%|| 35866/50000 [00:14<00:12, 1152.82it/s]
 72%|| 35983/50000 [00:14<00:12, 1154.70it/s]
 72%|| 36099/50000 [00:15<00:12, 1133.69it/s]
 72%|| 36213/50000 [00:15<00:12, 1112.25it/s]
 73%|| 36325/50000 [00:15<00:12, 1094.36it/s]
 73%|| 36435/50000 [00:15<00:12, 1095.36it/s]
 73%|| 36545/50000 [00:15<00:12, 1086.56it/s]
 73%|| 36655/50000 [00:15<00:12, 1087.94it/s]
 74%|| 36764/50000 [00:15<00:12, 1082.94it/s]
 74%|| 36873/50000 [00:15<00:12, 1077.15it/s]
 74%|| 36981/50000 [00:15<00:12, 1066.14it/s]
 74%|| 37088/50000 [00:15<00:12, 1007.84it/s]
 74%|| 37201/50000 [00:16<00:12, 1041.40it/s]
 75%|| 37325/50000 [00:16<00:11, 1093.87it/s]
 75%|| 37458/50000 [00:16<00:10, 1153.91it/s]
 75%|| 37582/50000 [00:16<00:10, 1177.07it/s]
 75%|| 37702/50000 [00:16<00:10, 1180.79it/s]
 76%|| 37829/50000 [00:16<00:10, 1203.53it/s]
 76%|| 37953/50000 [00:16<00:09, 1211.65it/s]
 76%|| 38076/50000 [00:16<00:09, 1214.15it/s]
 76%|| 38202/50000 [00:16<00:09, 1226.47it/s]
 77%|| 38329/50000 [00:16<00:09, 1238.33it/s]
 77%|| 38457/50000 [00:17<00:09, 1250.42it/s]
 77%|| 38598/50000 [00:17<00:08, 1293.83it/s]
 77%|| 38728/50000 [00:17<00:09, 1208.49it/s]
 78%|| 38851/50000 [00:17<00:09, 1167.31it/s]
 78%|| 38970/50000 [00:17<00:09, 1159.68it/s]
 78%|| 39087/50000 [00:17<00:09, 1148.88it/s]
 78%|| 39203/50000 [00:17<00:09, 1120.88it/s]
 79%|| 39316/50000 [00:17<00:09, 1117.98it/s]
 79%|| 39432/50000 [00:17<00:09, 1127.43it/s]
 79%|| 39546/50000 [00:18<00:09, 1122.02it/s]
```



```
 79%|| 39659/50000 [00:18<00:09, 1119.49it/s]
 80%|| 39772/50000 [00:18<00:09, 1100.74it/s]
 80%|| 39883/50000 [00:18<00:09, 1100.85it/s]
 80%|| 39994/50000 [00:18<00:09, 1102.38it/s]
 80%|| 40105/50000 [00:18<00:09, 1088.60it/s]
 80%|| 40221/50000 [00:18<00:08, 1107.78it/s]
 81%|| 40358/50000 [00:18<00:08, 1174.22it/s]
 81%|| 40493/50000 [00:18<00:07, 1218.68it/s]
 81%|| 40632/50000 [00:18<00:07, 1262.69it/s]
 82%|| 40760/50000 [00:19<00:07, 1246.82it/s]
 82%|| 40889/50000 [00:19<00:07, 1256.59it/s]
 82%|| 41021/50000 [00:19<00:07, 1274.29it/s]
 82%|| 41158/50000 [00:19<00:06, 1300.39it/s]
 83%|| 41289/50000 [00:19<00:06, 1267.32it/s]
 83%|| 41417/50000 [00:19<00:06, 1262.90it/s]
 83%|| 41544/50000 [00:19<00:06, 1246.68it/s]
 83%|| 41669/50000 [00:19<00:06, 1245.78it/s]
 84%|| 41794/50000 [00:19<00:06, 1236.04it/s]
 84%|| 41918/50000 [00:19<00:06, 1231.56it/s]
 84%|| 42044/50000 [00:20<00:06, 1238.71it/s]
 84%|| 42168/50000 [00:20<00:06, 1224.42it/s]
 85%|| 42291/50000 [00:20<00:06, 1223.00it/s]
 85%|| 42414/50000 [00:20<00:06, 1218.83it/s]
 85%|| 42550/50000 [00:20<00:05, 1256.67it/s]
 85%|| 42677/50000 [00:20<00:06, 1162.21it/s]
 86%|| 42795/50000 [00:20<00:06, 1145.77it/s]
 86%|| 42911/50000 [00:20<00:06, 1097.61it/s]
 86%|| 43023/50000 [00:20<00:06, 1063.27it/s]
 86%|| 43131/50000 [00:21<00:06, 1061.93it/s]
 86%|| 43238/50000 [00:21<00:06, 1053.08it/s]
 87%|| 43356/50000 [00:21<00:06, 1086.87it/s]
 87%|| 43481/50000 [00:21<00:05, 1130.05it/s]
 87%|| 43603/50000 [00:21<00:05, 1153.69it/s]
 87%|| 43728/50000 [00:21<00:05, 1178.55it/s]
 88%|| 43851/50000 [00:21<00:05, 1192.01it/s]
 88%|| 44025/50000 [00:21<00:04, 1316.02it/s]
 89%|| 44343/50000 [00:21<00:03, 1596.37it/s]
 89%|| 44694/50000 [00:21<00:02, 1908.14it/s]
 90%|| 45040/50000 [00:22<00:02, 2204.05it/s]
 91%|| 45402/50000 [00:22<00:01, 2497.00it/s]
 91%|| 45730/50000 [00:22<00:01, 2689.54it/s]
 92%|| 46068/50000 [00:22<00:01, 2864.22it/s]
 93%|| 46433/50000 [00:22<00:01, 3060.56it/s]
 94%|| 46798/50000 [00:22<00:00, 3215.94it/s]
 94%|| 47175/50000 [00:22<00:00, 3362.46it/s]
 95%|| 47529/50000 [00:22<00:00, 3410.63it/s]
 96%|| 47881/50000 [00:22<00:00, 3340.56it/s]
 96%|| 48223/50000 [00:22<00:00, 3286.02it/s]
 97%|| 48558/50000 [00:23<00:00, 3236.05it/s]
 98%|| 48886/50000 [00:23<00:00, 3195.26it/s]
 98%|| 49209/50000 [00:23<00:00, 3187.42it/s]
 99%|| 49530/50000 [00:23<00:00, 3167.61it/s]
100%|| 50000/50000 [00:23<00:00, 2122.04it/s]

  0%|          | 0/50000 [00:00<?, ?it/s]
  1%|          | 324/50000 [00:00<00:15, 3237.48it/s]
```

```
RescaleTransformTestMin passed.
```



```
  1%|| | 662/50000 [00:00<00:15, 3278.20it/s]
  2%|| | 1040/50000 [00:00<00:14, 3412.97it/s]
  3%|| | 1414/50000 [00:00<00:13, 3504.17it/s]
  4%|| | 1788/50000 [00:00<00:13, 3571.04it/s]
  4%|| | 2133/50000 [00:00<00:13, 3531.84it/s]
  5%|| | 2487/50000 [00:00<00:13, 3533.62it/s]
  6%|| | 2815/50000 [00:00<00:13, 3423.65it/s]
  6%|| | 3144/50000 [00:00<00:13, 3380.56it/s]
  7%|| | 3495/50000 [00:01<00:13, 3416.85it/s]
  8%|| | 3853/50000 [00:01<00:13, 3462.05it/s]
  8%|| | 4194/50000 [00:01<00:13, 3387.10it/s]
  9%|| | 4550/50000 [00:01<00:13, 3435.14it/s]
 10%|| | 4913/50000 [00:01<00:12, 3488.38it/s]
 11%|| | 5272/50000 [00:01<00:12, 3517.39it/s]
 11%|| | 5635/50000 [00:01<00:12, 3548.37it/s]
 12%|| | 5990/50000 [00:01<00:12, 3488.23it/s]
 13%|| | 6339/50000 [00:01<00:12, 3421.36it/s]
 13%|| | 6682/50000 [00:01<00:12, 3338.75it/s]
 14%|| | 7017/50000 [00:02<00:13, 3261.20it/s]
 15%|| | 7346/50000 [00:02<00:13, 3269.36it/s]
 15%|| | 7683/50000 [00:02<00:12, 3296.84it/s]
 16%|| | 8034/50000 [00:02<00:12, 3354.57it/s]
 17%|| | 8371/50000 [00:02<00:12, 3325.69it/s]
 17%|| | 8705/50000 [00:02<00:13, 3023.82it/s]
 18%|| | 9021/50000 [00:02<00:13, 3062.08it/s]
 19%|| | 9332/50000 [00:02<00:13, 3074.28it/s]
 19%|| | 9653/50000 [00:02<00:12, 3111.78it/s]
 20%|| | 9977/50000 [00:02<00:12, 3147.65it/s]
 21%|| | 10294/50000 [00:03<00:12, 3122.01it/s]
 21%|| | 10608/50000 [00:03<00:12, 3109.36it/s]
 22%|| | 10921/50000 [00:03<00:12, 3113.72it/s]
 22%|| | 11235/50000 [00:03<00:12, 3120.17it/s]
 23%|| | 11548/50000 [00:03<00:12, 3112.57it/s]
 24%|| | 11860/50000 [00:03<00:12, 3101.72it/s]
 24%|| | 12179/50000 [00:03<00:12, 3126.64it/s]
 25%|| | 12504/50000 [00:03<00:11, 3160.68it/s]
 26%|| | 12823/50000 [00:03<00:11, 3169.40it/s]
 26%|| | 13141/50000 [00:03<00:11, 3162.60it/s]
 27%|| | 13465/50000 [00:04<00:11, 3184.95it/s]
 28%|| | 13784/50000 [00:04<00:11, 3076.91it/s]
 28%|| | 14093/50000 [00:04<00:11, 3010.11it/s]
 29%|| | 14395/50000 [00:04<00:12, 2942.64it/s]
 29%|| | 14706/50000 [00:04<00:11, 2989.68it/s]
 30%|| | 15008/50000 [00:04<00:11, 2995.72it/s]
 31%|| | 15309/50000 [00:04<00:11, 2990.30it/s]
 31%|| | 15622/50000 [00:04<00:11, 3029.46it/s]
 32%|| | 15945/50000 [00:04<00:11, 3085.70it/s]
 33%|| | 16271/50000 [00:05<00:10, 3135.27it/s]
 33%|| | 16586/50000 [00:05<00:10, 3136.32it/s]
 34%|| | 16903/50000 [00:05<00:10, 3144.28it/s]
 34%|| | 17218/50000 [00:05<00:10, 3124.47it/s]
 35%|| | 17531/50000 [00:05<00:10, 3101.42it/s]
 36%|| | 17842/50000 [00:05<00:10, 3051.65it/s]
 36%|| | 18148/50000 [00:05<00:10, 3006.33it/s]
 37%|| | 18494/50000 [00:05<00:10, 3128.33it/s]
 38%|| | 18821/50000 [00:05<00:09, 3168.64it/s]
 38%|| | 19177/50000 [00:05<00:09, 3275.94it/s]
```

```
 39%|███▉      | 19507/50000 [00:06<00:09, 3192.01it/s]
 40%|███▉      | 19828/50000 [00:06<00:09, 3178.86it/s]
 40%|████      | 20148/50000 [00:06<00:09, 3007.12it/s]
 41%|████      | 20456/50000 [00:06<00:09, 3028.34it/s]
 42%|████▏     | 20769/50000 [00:06<00:09, 3057.53it/s]
 42%|████▏     | 21085/50000 [00:06<00:09, 3085.23it/s]
 43%|████▎     | 21395/50000 [00:06<00:09, 3087.40it/s]
 43%|████▎     | 21710/50000 [00:06<00:09, 3103.91it/s]
 44%|████▍     | 22021/50000 [00:06<00:09, 3102.90it/s]
 45%|████▍     | 22332/50000 [00:06<00:08, 3086.42it/s]
 45%|████▌     | 22641/50000 [00:07<00:08, 3070.62it/s]
 46%|████▌     | 22949/50000 [00:07<00:08, 3012.22it/s]
 47%|████▋     | 23271/50000 [00:07<00:08, 3069.90it/s]
 47%|████▋     | 23598/50000 [00:07<00:08, 3126.86it/s]
 48%|████▊     | 23915/50000 [00:07<00:08, 3138.30it/s]
 48%|████▊     | 24230/50000 [00:07<00:08, 3133.43it/s]
 49%|████▉     | 24544/50000 [00:07<00:08, 3131.60it/s]
 50%|████▉     | 24858/50000 [00:07<00:08, 3097.32it/s]
 50%|█████     | 25168/50000 [00:07<00:08, 3094.14it/s]
 51%|█████     | 25478/50000 [00:07<00:08, 3023.49it/s]
 52%|█████▏    | 25781/50000 [00:08<00:08, 3013.42it/s]
 52%|█████▏    | 26114/50000 [00:08<00:07, 3099.92it/s]
 53%|█████▎    | 26447/50000 [00:08<00:07, 3162.60it/s]
 54%|█████▎    | 26774/50000 [00:08<00:07, 3193.56it/s]
 54%|█████▍    | 27095/50000 [00:08<00:07, 3177.01it/s]
 55%|█████▍    | 27414/50000 [00:08<00:07, 2990.61it/s]
 55%|█████▌    | 27720/50000 [00:08<00:07, 3009.25it/s]
 56%|█████▌    | 28036/50000 [00:08<00:07, 3051.67it/s]
 57%|█████▋    | 28355/50000 [00:08<00:07, 3090.89it/s]
 57%|█████▋    | 28666/50000 [00:09<00:06, 3063.38it/s]
 58%|█████▊    | 28974/50000 [00:09<00:06, 3068.07it/s]
 59%|█████▊    | 29288/50000 [00:09<00:06, 3088.19it/s]
 59%|█████▉    | 29598/50000 [00:09<00:06, 3031.39it/s]
 60%|█████▉    | 29918/50000 [00:09<00:06, 3078.71it/s]
 60%|██████    | 30236/50000 [00:09<00:06, 3106.39it/s]
 61%|██████    | 30556/50000 [00:09<00:06, 3132.17it/s]
 62%|██████▏   | 30872/50000 [00:09<00:06, 3139.97it/s]
 62%|██████▏   | 31187/50000 [00:09<00:05, 3140.81it/s]
 63%|██████▎   | 31502/50000 [00:09<00:06, 3059.86it/s]
 64%|██████▎   | 31819/50000 [00:10<00:05, 3090.32it/s]
 64%|██████▍   | 32144/50000 [00:10<00:05, 3136.52it/s]
 65%|██████▍   | 32466/50000 [00:10<00:05, 3161.03it/s]
 66%|██████▌   | 32788/50000 [00:10<00:05, 3176.92it/s]
 66%|██████▌   | 33106/50000 [00:10<00:05, 3176.23it/s]
 67%|██████▋   | 33424/50000 [00:10<00:05, 3034.19it/s]
 67%|██████▋   | 33732/50000 [00:10<00:05, 3046.88it/s]
 68%|██████▊   | 34054/50000 [00:10<00:05, 3095.51it/s]
 69%|██████▊   | 34365/50000 [00:10<00:05, 3057.88it/s]
 69%|██████▉   | 34672/50000 [00:10<00:05, 3013.76it/s]
 70%|██████▉   | 34987/50000 [00:11<00:04, 3053.34it/s]
 71%|███████   | 35303/50000 [00:11<00:04, 3082.80it/s]
 71%|███████   | 35612/50000 [00:11<00:04, 3044.69it/s]
 72%|███████▏  | 35917/50000 [00:11<00:04, 2944.55it/s]
 72%|███████▏  | 36221/50000 [00:11<00:04, 2971.14it/s]
 73%|███████▎  | 36540/50000 [00:11<00:04, 3032.80it/s]
 74%|███████▎  | 36865/50000 [00:11<00:04, 3094.83it/s]
 74%|███████▍  | 37190/50000 [00:11<00:04, 3138.87it/s]
```



```
 75%|███████▌  | 37513/50000 [00:11<00:03, 3164.12it/s]
 76%|███████▌  | 37831/50000 [00:11<00:03, 3167.88it/s]
 76%|███████▋  | 38149/50000 [00:12<00:03, 3145.83it/s]
 77%|███████▋  | 38464/50000 [00:12<00:03, 3122.95it/s]
 78%|███████▊  | 38777/50000 [00:12<00:03, 3094.38it/s]
 78%|███████▊  | 39094/50000 [00:12<00:03, 3116.17it/s]
 79%|███████▉  | 39433/50000 [00:12<00:03, 3192.16it/s]
 80%|███████▉  | 39762/50000 [00:12<00:03, 3220.04it/s]
 80%|████████  | 40085/50000 [00:12<00:03, 3191.18it/s]
 81%|████████  | 40405/50000 [00:12<00:03, 2990.90it/s]
 81%|████████▏ | 40715/50000 [00:12<00:03, 3021.07it/s]
 82%|████████▏ | 41031/50000 [00:13<00:02, 3059.70it/s]
 83%|████████▎ | 41348/50000 [00:13<00:02, 3091.92it/s]
 83%|████████▎ | 41679/50000 [00:13<00:02, 3152.36it/s]
 84%|████████▍ | 42009/50000 [00:13<00:02, 3194.75it/s]
 85%|████████▍ | 42330/50000 [00:13<00:02, 3070.30it/s]
 85%|████████▌ | 42639/50000 [00:13<00:02, 3012.35it/s]
 86%|████████▌ | 42956/50000 [00:13<00:02, 3056.18it/s]
 87%|████████▋ | 43286/50000 [00:13<00:02, 3124.19it/s]
 87%|████████▋ | 43607/50000 [00:13<00:02, 3149.27it/s]
 88%|████████▊ | 43937/50000 [00:13<00:01, 3192.65it/s]
 89%|████████▊ | 44258/50000 [00:14<00:01, 3189.17it/s]
 89%|████████▉ | 44578/50000 [00:14<00:01, 3153.89it/s]
 90%|████████▉ | 44894/50000 [00:14<00:01, 3061.20it/s]
 90%|█████████ | 45202/50000 [00:14<00:01, 3033.75it/s]
 91%|█████████ | 45528/50000 [00:14<00:01, 3095.90it/s]
 92%|█████████▏| 45848/50000 [00:14<00:01, 3122.58it/s]
 92%|█████████▏| 46172/50000 [00:14<00:01, 3154.47it/s]
 93%|█████████▎| 46497/50000 [00:14<00:01, 3180.13it/s]
 94%|█████████▎| 46820/50000 [00:14<00:00, 3193.22it/s]
 94%|█████████▍| 47163/50000 [00:14<00:00, 3258.93it/s]
 95%|█████████▍| 47496/50000 [00:15<00:00, 3277.77it/s]
 96%|█████████▌| 47825/50000 [00:15<00:00, 3192.58it/s]
 96%|█████████▋| 48146/50000 [00:15<00:00, 3184.79it/s]
 97%|█████████▋| 48466/50000 [00:15<00:00, 3112.65it/s]
 98%|█████████▊| 48779/50000 [00:15<00:00, 3099.80it/s]
 98%|█████████▊| 49115/50000 [00:15<00:00, 3172.39it/s]
 99%|█████████▉| 49503/50000 [00:15<00:00, 3355.43it/s]
100%|██████████| 50000/50000 [00:15<00:00, 3161.45it/s]

RescaleTransformTestMax passed.
Class RescaleTransform correctly implemented. Tests passed: 2/2
Score: 100/100
```

If you look at the first image, you should now see that all values are between 0 and 1.

In [10]:
```
sample_item = dataset_rescaled[0]
sample_label = sample_item["label"]
sample_image = sample_item["image"]

print("Max value:", np.max(sample_image))
print("Min value:", np.min(sample_image))
print('Sample rescaled image first values:', sample_image[0][0])
```

```
Max value: 1.0
Min value: 0.00392156862745098
Sample rescaled image first values: [0.01176471 0.49019608 0.91372549]
```

4.2 Normalize Images to Standard Gaussian using NormalizeTransform

Let us now move on to the `NormalizeTransform` class. The `NormalizeTransform` class normalizes images channel-wise and its `__init__` method has two arguments:

- **mean** is the normalization mean, which will be subtracted from the dataset.
- **std** is the normalization standard deviation. By scaling the data with a factor of `1/std` the standard deviation will be normazlied accordingly.

Have a look at the code in `exercise_code/data/transforms.py` .

The next step is to normalize the CIFAR-10 **images channel-wise** to standard normal. To do so, you need to calculate the **per-channel image mean and standard deviation** first, which you can then provide to `NormalizeTransform` to normalize the data accordingly.

6/2/2021 1_cifar10-image-dataset

In [11]:
```
# You first have to load all rescaled images
rescaled_images = []
for sample in tqdm(dataset_rescaled):
    rescaled_images.append(sample["image"])
rescaled_images = np.array(rescaled_images)
```

```
  0%|          | 0/50000 [00:00<?, ?it/s]
  1%|          | 305/50000 [00:00<00:16, 3049.85it/s]
  1%|▏         | 649/50000 [00:00<00:15, 3156.30it/s]
  2%|▏         | 986/50000 [00:00<00:15, 3215.38it/s]
  3%|▎         | 1339/50000 [00:00<00:14, 3303.45it/s]
  3%|▎         | 1711/50000 [00:00<00:14, 3417.86it/s]
  4%|▍         | 2062/50000 [00:00<00:13, 3442.77it/s]
  5%|▍         | 2413/50000 [00:00<00:13, 3461.75it/s]
  5%|▌         | 2736/50000 [00:00<00:14, 3361.14it/s]
  6%|▌         | 3056/50000 [00:00<00:15, 3057.97it/s]
  7%|▋         | 3356/50000 [00:01<00:22, 2113.48it/s]
  7%|▋         | 3603/50000 [00:01<00:25, 1800.18it/s]
  8%|▊         | 3816/50000 [00:01<00:30, 1512.44it/s]
  8%|▊         | 3998/50000 [00:01<00:34, 1340.65it/s]
  8%|▊         | 4158/50000 [00:01<00:33, 1368.37it/s]
  9%|▊         | 4313/50000 [00:01<00:33, 1367.53it/s]
  9%|▉         | 4463/50000 [00:02<00:35, 1284.24it/s]
  9%|▉         | 4602/50000 [00:02<00:36, 1258.37it/s]
  9%|▉         | 4736/50000 [00:02<00:35, 1281.21it/s]
 10%|▉         | 4871/50000 [00:02<00:34, 1298.32it/s]
 10%|█         | 5009/50000 [00:02<00:34, 1320.42it/s]
 10%|█         | 5144/50000 [00:02<00:34, 1314.61it/s]
 11%|█         | 5278/50000 [00:02<00:34, 1289.43it/s]
 11%|█         | 5421/50000 [00:02<00:33, 1326.72it/s]
 11%|█         | 5555/50000 [00:02<00:33, 1327.40it/s]
 11%|█▏        | 5691/50000 [00:03<00:33, 1336.21it/s]
 12%|█▏        | 5836/50000 [00:03<00:32, 1367.02it/s]
 12%|█▏        | 5974/50000 [00:03<00:32, 1369.19it/s]
 12%|█▏        | 6112/50000 [00:03<00:32, 1349.74it/s]
 12%|█▎        | 6250/50000 [00:03<00:32, 1357.37it/s]
 13%|█▎        | 6386/50000 [00:03<00:33, 1289.05it/s]
 13%|█▎        | 6516/50000 [00:03<00:34, 1262.56it/s]
 13%|█▎        | 6644/50000 [00:03<00:34, 1242.12it/s]
 14%|█▎        | 6778/50000 [00:03<00:34, 1267.74it/s]
 14%|█▍        | 6909/50000 [00:03<00:33, 1278.24it/s]
 14%|█▍        | 7041/50000 [00:04<00:33, 1289.18it/s]
 14%|█▍        | 7171/50000 [00:04<00:34, 1248.77it/s]
 15%|█▍        | 7298/50000 [00:04<00:34, 1254.65it/s]
 15%|█▍        | 7432/50000 [00:04<00:33, 1277.77it/s]
 15%|█▌        | 7565/50000 [00:04<00:32, 1290.79it/s]
 15%|█▌        | 7704/50000 [00:04<00:32, 1317.95it/s]
 16%|█▌        | 7851/50000 [00:04<00:31, 1358.44it/s]
 16%|█▌        | 7991/50000 [00:04<00:30, 1370.26it/s]
 16%|█▋        | 8129/50000 [00:04<00:31, 1331.87it/s]
 17%|█▋        | 8263/50000 [00:04<00:31, 1318.34it/s]
 17%|█▋        | 8400/50000 [00:05<00:31, 1331.47it/s]
 17%|█▋        | 8534/50000 [00:05<00:31, 1331.01it/s]
 17%|█▋        | 8678/50000 [00:05<00:30, 1360.86it/s]
 18%|█▊        | 8815/50000 [00:05<00:31, 1305.68it/s]
 18%|█▊        | 8947/50000 [00:05<00:31, 1306.52it/s]
 18%|█▊        | 9081/50000 [00:05<00:31, 1316.13it/s]
 18%|█▊        | 9213/50000 [00:05<00:31, 1284.04it/s]
 19%|█▊        | 9342/50000 [00:05<00:31, 1275.58it/s]
 19%|█▉        | 9477/50000 [00:05<00:31, 1296.04it/s]
 19%|█▉        | 9621/50000 [00:06<00:30, 1333.63it/s]
 20%|█▉        | 9760/50000 [00:06<00:29, 1346.69it/s]
 20%|█▉        | 9904/50000 [00:06<00:29, 1373.18it/s]
```



```
 20%|██        | 10042/50000 [00:06<00:29, 1348.39it/s]
 20%|██        | 10178/50000 [00:06<00:31, 1266.69it/s]
 21%|██        | 10324/50000 [00:06<00:30, 1318.03it/s]
 21%|██        | 10461/50000 [00:06<00:29, 1332.33it/s]
 21%|██        | 10596/50000 [00:06<00:29, 1321.95it/s]
 21%|██▏       | 10735/50000 [00:06<00:29, 1341.61it/s]
 22%|██▏       | 10870/50000 [00:06<00:29, 1326.34it/s]
 22%|██▏       | 11013/50000 [00:07<00:28, 1355.45it/s]
 22%|██▏       | 11153/50000 [00:07<00:28, 1368.50it/s]
 23%|██▎       | 11294/50000 [00:07<00:28, 1379.35it/s]
 23%|██▎       | 11443/50000 [00:07<00:27, 1409.27it/s]
 23%|██▎       | 11585/50000 [00:07<00:27, 1386.62it/s]
 23%|██▎       | 11724/50000 [00:07<00:28, 1356.25it/s]
 24%|██▎       | 11861/50000 [00:07<00:28, 1354.77it/s]
 24%|██▍       | 11997/50000 [00:07<00:29, 1304.43it/s]
 24%|██▍       | 12129/50000 [00:07<00:29, 1293.62it/s]
 25%|██▍       | 12273/50000 [00:07<00:28, 1334.30it/s]
 25%|██▍       | 12412/50000 [00:08<00:27, 1350.44it/s]
 25%|██▌       | 12552/50000 [00:08<00:27, 1364.71it/s]
 25%|██▌       | 12689/50000 [00:08<00:27, 1365.37it/s]
 26%|██▌       | 12831/50000 [00:08<00:26, 1380.78it/s]
 26%|██▌       | 12973/50000 [00:08<00:26, 1392.31it/s]
 26%|██▌       | 13113/50000 [00:08<00:27, 1362.70it/s]
 27%|██▋       | 13258/50000 [00:08<00:26, 1385.80it/s]
 27%|██▋       | 13397/50000 [00:08<00:26, 1371.88it/s]
 27%|██▋       | 13535/50000 [00:08<00:27, 1344.49it/s]
 27%|██▋       | 13670/50000 [00:09<00:27, 1325.38it/s]
 28%|██▊       | 13803/50000 [00:09<00:28, 1280.37it/s]
 28%|██▊       | 13932/50000 [00:09<00:28, 1276.67it/s]
 28%|██▊       | 14061/50000 [00:09<00:28, 1274.38it/s]
 28%|██▊       | 14189/50000 [00:09<00:28, 1270.86it/s]
 29%|██▊       | 14324/50000 [00:09<00:27, 1292.24it/s]
 29%|██▉       | 14469/50000 [00:09<00:26, 1333.61it/s]
 29%|██▉       | 14607/50000 [00:09<00:26, 1346.55it/s]
 29%|██▉       | 14743/50000 [00:09<00:26, 1308.70it/s]
 30%|██▉       | 14885/50000 [00:09<00:26, 1337.93it/s]
 30%|███       | 15020/50000 [00:10<00:26, 1339.14it/s]
 30%|███       | 15155/50000 [00:10<00:26, 1320.69it/s]
 31%|███       | 15296/50000 [00:10<00:25, 1345.65it/s]
 31%|███       | 15437/50000 [00:10<00:25, 1362.77it/s]
 31%|███       | 15582/50000 [00:10<00:24, 1386.36it/s]
 31%|███▏      | 15721/50000 [00:10<00:25, 1366.76it/s]
 32%|███▏      | 15858/50000 [00:10<00:25, 1339.59it/s]
 32%|███▏      | 15993/50000 [00:10<00:25, 1338.29it/s]
 32%|███▏      | 16132/50000 [00:10<00:25, 1353.08it/s]
 33%|███▎      | 16275/50000 [00:10<00:24, 1373.22it/s]
 33%|███▎      | 16413/50000 [00:11<00:24, 1349.26it/s]
 33%|███▎      | 16549/50000 [00:11<00:26, 1254.16it/s]
 33%|███▎      | 16676/50000 [00:11<00:26, 1249.59it/s]
 34%|███▎      | 16803/50000 [00:11<00:29, 1141.62it/s]
 34%|███▍      | 16920/50000 [00:11<00:28, 1141.94it/s]
 34%|███▍      | 17040/50000 [00:11<00:28, 1156.51it/s]
 34%|███▍      | 17164/50000 [00:11<00:27, 1178.34it/s]
 35%|███▍      | 17286/50000 [00:11<00:27, 1188.30it/s]
 35%|███▍      | 17412/50000 [00:11<00:26, 1207.97it/s]
 35%|███▌      | 17546/50000 [00:12<00:26, 1243.07it/s]
 35%|███▌      | 17682/50000 [00:12<00:25, 1273.52it/s]
```

```
 36%|███▌      | 17823/50000 [00:12<00:24, 1309.47it/s]
 36%|███▌      | 17964/50000 [00:12<00:23, 1335.99it/s]
 36%|███▌      | 18099/50000 [00:12<00:24, 1323.83it/s]
 36%|███▋      | 18237/50000 [00:12<00:23, 1339.80it/s]
 37%|███▋      | 18382/50000 [00:12<00:23, 1369.19it/s]
 37%|███▋      | 18530/50000 [00:12<00:22, 1398.19it/s]
 37%|███▋      | 18671/50000 [00:12<00:23, 1323.47it/s]
 38%|███▊      | 18805/50000 [00:12<00:23, 1306.74it/s]
 38%|███▊      | 18937/50000 [00:13<00:24, 1292.01it/s]
 38%|███▊      | 19074/50000 [00:13<00:23, 1312.14it/s]
 38%|███▊      | 19212/50000 [00:13<00:23, 1330.34it/s]
 39%|███▊      | 19357/50000 [00:13<00:22, 1363.06it/s]
 39%|███▉      | 19505/50000 [00:13<00:21, 1394.55it/s]
 39%|███▉      | 19649/50000 [00:13<00:21, 1407.66it/s]
 40%|███▉      | 19791/50000 [00:13<00:21, 1406.63it/s]
 40%|███▉      | 19934/50000 [00:13<00:21, 1412.20it/s]
 40%|████      | 20076/50000 [00:13<00:21, 1363.66it/s]
 40%|████      | 20215/50000 [00:13<00:21, 1368.85it/s]
 41%|████      | 20353/50000 [00:14<00:22, 1323.83it/s]
 41%|████      | 20486/50000 [00:14<00:23, 1262.13it/s]
 41%|████      | 20614/50000 [00:14<00:23, 1240.44it/s]
 41%|████▏     | 20739/50000 [00:14<00:24, 1210.53it/s]
 42%|████▏     | 20869/50000 [00:14<00:23, 1233.66it/s]
 42%|████▏     | 21003/50000 [00:14<00:22, 1263.66it/s]
 42%|████▏     | 21134/50000 [00:14<00:22, 1276.63it/s]
 43%|████▎     | 21274/50000 [00:14<00:21, 1309.38it/s]
 43%|████▎     | 21409/50000 [00:14<00:21, 1321.09it/s]
 43%|████▎     | 21542/50000 [00:15<00:21, 1306.62it/s]
 43%|████▎     | 21673/50000 [00:15<00:21, 1300.64it/s]
 44%|████▎     | 21824/50000 [00:15<00:20, 1354.42it/s]
 44%|████▍     | 21961/50000 [00:15<00:22, 1265.20it/s]
 44%|████▍     | 22090/50000 [00:15<00:22, 1233.35it/s]
 44%|████▍     | 22215/50000 [00:15<00:22, 1231.41it/s]
 45%|████▍     | 22341/50000 [00:15<00:22, 1239.39it/s]
 45%|████▍     | 22473/50000 [00:15<00:21, 1261.91it/s]
 45%|████▌     | 22619/50000 [00:15<00:20, 1313.67it/s]
 46%|████▌     | 22761/50000 [00:15<00:20, 1342.68it/s]
 46%|████▌     | 22897/50000 [00:16<00:20, 1338.32it/s]
 46%|████▌     | 23032/50000 [00:16<00:20, 1341.05it/s]
 46%|████▋     | 23169/50000 [00:16<00:19, 1348.81it/s]
 47%|████▋     | 23314/50000 [00:16<00:19, 1376.56it/s]
 47%|████▋     | 23453/50000 [00:16<00:19, 1334.77it/s]
 47%|████▋     | 23587/50000 [00:16<00:19, 1331.95it/s]
 47%|████▋     | 23726/50000 [00:16<00:19, 1348.37it/s]
 48%|████▊     | 23862/50000 [00:16<00:20, 1305.23it/s]
 48%|████▊     | 23996/50000 [00:16<00:19, 1313.98it/s]
 48%|████▊     | 24128/50000 [00:17<00:21, 1197.78it/s]
 49%|████▊     | 24251/50000 [00:17<00:21, 1201.58it/s]
 49%|████▊     | 24373/50000 [00:17<00:21, 1180.43it/s]
 49%|████▉     | 24493/50000 [00:17<00:22, 1150.09it/s]
 49%|████▉     | 24617/50000 [00:17<00:21, 1173.12it/s]
 49%|████▉     | 24739/50000 [00:17<00:21, 1184.12it/s]
 50%|████▉     | 24868/50000 [00:17<00:20, 1213.36it/s]
 50%|█████     | 25005/50000 [00:17<00:19, 1255.43it/s]
 50%|█████     | 25138/50000 [00:17<00:19, 1275.45it/s]
 51%|█████     | 25267/50000 [00:17<00:19, 1261.84it/s]
 51%|█████     | 25398/50000 [00:18<00:19, 1275.23it/s]
```



```
 51%|█████     | 25526/50000 [00:18<00:19, 1231.05it/s]
 51%|█████▏    | 25653/50000 [00:18<00:19, 1241.68it/s]
 52%|█████▏    | 25787/50000 [00:18<00:19, 1267.59it/s]
 52%|█████▏    | 25928/50000 [00:18<00:18, 1306.48it/s]
 52%|█████▏    | 26060/50000 [00:18<00:18, 1292.69it/s]
 52%|█████▏    | 26190/50000 [00:18<00:19, 1203.81it/s]
 53%|█████▎    | 26312/50000 [00:18<00:19, 1192.36it/s]
 53%|█████▎    | 26441/50000 [00:18<00:19, 1219.09it/s]
 53%|█████▎    | 26574/50000 [00:19<00:18, 1248.40it/s]
 53%|█████▎    | 26707/50000 [00:19<00:18, 1269.32it/s]
 54%|█████▎    | 26839/50000 [00:19<00:18, 1283.20it/s]
 54%|█████▍    | 26977/50000 [00:19<00:17, 1309.22it/s]
 54%|█████▍    | 27122/50000 [00:19<00:16, 1346.23it/s]
 55%|█████▍    | 27258/50000 [00:19<00:17, 1275.48it/s]
 55%|█████▍    | 27387/50000 [00:19<00:18, 1250.42it/s]
 55%|█████▌    | 27513/50000 [00:19<00:18, 1242.28it/s]
 55%|█████▌    | 27654/50000 [00:19<00:17, 1287.87it/s]
 56%|█████▌    | 27792/50000 [00:19<00:16, 1313.27it/s]
 56%|█████▌    | 27928/50000 [00:20<00:16, 1324.60it/s]
 56%|█████▌    | 28062/50000 [00:20<00:16, 1298.02it/s]
 56%|█████▋    | 28193/50000 [00:20<00:16, 1290.75it/s]
 57%|█████▋    | 28324/50000 [00:20<00:16, 1294.40it/s]
 57%|█████▋    | 28454/50000 [00:20<00:17, 1235.56it/s]
 57%|█████▋    | 28579/50000 [00:20<00:17, 1227.96it/s]
 57%|█████▋    | 28703/50000 [00:20<00:17, 1215.94it/s]
 58%|█████▊    | 28833/50000 [00:20<00:17, 1239.09it/s]
 58%|█████▊    | 28968/50000 [00:20<00:16, 1270.11it/s]
 58%|█████▊    | 29110/50000 [00:20<00:15, 1310.35it/s]
 58%|█████▊    | 29242/50000 [00:21<00:17, 1194.96it/s]
 59%|█████▊    | 29364/50000 [00:21<00:17, 1187.54it/s]
 59%|█████▉    | 29501/50000 [00:21<00:16, 1234.97it/s]
 59%|█████▉    | 29627/50000 [00:21<00:16, 1216.28it/s]
 60%|█████▉    | 29755/50000 [00:21<00:16, 1234.31it/s]
 60%|█████▉    | 29886/50000 [00:21<00:16, 1252.42it/s]
 60%|██████    | 30012/50000 [00:21<00:15, 1251.08it/s]
 60%|██████    | 30138/50000 [00:21<00:15, 1251.85it/s]
 61%|██████    | 30264/50000 [00:21<00:16, 1227.55it/s]
 61%|██████    | 30394/50000 [00:22<00:15, 1247.42it/s]
 61%|██████    | 30525/50000 [00:22<00:15, 1264.84it/s]
 61%|██████▏   | 30655/50000 [00:22<00:15, 1272.67it/s]
 62%|██████▏   | 30783/50000 [00:22<00:15, 1272.09it/s]
 62%|██████▏   | 30915/50000 [00:22<00:14, 1285.95it/s]
 62%|██████▏   | 31044/50000 [00:22<00:14, 1284.16it/s]
 62%|██████▏   | 31182/50000 [00:22<00:14, 1309.79it/s]
 63%|██████▎   | 31323/50000 [00:22<00:13, 1336.24it/s]
 63%|██████▎   | 31457/50000 [00:22<00:14, 1289.81it/s]
 63%|██████▎   | 31601/50000 [00:22<00:13, 1330.03it/s]
 63%|██████▎   | 31735/50000 [00:23<00:14, 1268.99it/s]
 64%|██████▎   | 31863/50000 [00:23<00:14, 1241.46it/s]
 64%|██████▍   | 32008/50000 [00:23<00:13, 1296.17it/s]
 64%|██████▍   | 32147/50000 [00:23<00:13, 1321.79it/s]
 65%|██████▍   | 32290/50000 [00:23<00:13, 1350.69it/s]
 65%|██████▍   | 32426/50000 [00:23<00:13, 1277.13it/s]
 65%|██████▌   | 32556/50000 [00:23<00:14, 1223.71it/s]
 65%|██████▌   | 32680/50000 [00:23<00:14, 1215.40it/s]
 66%|██████▌   | 32812/50000 [00:23<00:13, 1243.83it/s]
 66%|██████▌   | 32957/50000 [00:24<00:13, 1297.34it/s]
```

```
 66%|██████    | 33088/50000 [00:24<00:13, 1298.19it/s]
 66%|██████    | 33226/50000 [00:24<00:12, 1321.57it/s]
 67%|██████    | 33367/50000 [00:24<00:12, 1346.55it/s]
 67%|██████    | 33509/50000 [00:24<00:12, 1367.66it/s]
 67%|██████    | 33649/50000 [00:24<00:11, 1375.98it/s]
 68%|██████    | 33787/50000 [00:24<00:11, 1373.10it/s]
 68%|██████    | 33925/50000 [00:24<00:11, 1361.19it/s]
 68%|██████    | 34062/50000 [00:24<00:11, 1349.94it/s]
 68%|██████    | 34198/50000 [00:24<00:11, 1344.04it/s]
 69%|██████    | 34333/50000 [00:25<00:11, 1327.59it/s]
 69%|██████    | 34466/50000 [00:25<00:12, 1283.37it/s]
 69%|██████    | 34604/50000 [00:25<00:11, 1307.97it/s]
 69%|██████    | 34736/50000 [00:25<00:11, 1289.60it/s]
 70%|██████    | 34876/50000 [00:25<00:11, 1319.46it/s]
 70%|███████   | 35013/50000 [00:25<00:11, 1334.21it/s]
 70%|███████   | 35147/50000 [00:25<00:11, 1307.39it/s]
 71%|███████   | 35284/50000 [00:25<00:11, 1324.57it/s]
 71%|███████   | 35422/50000 [00:25<00:10, 1338.78it/s]
 71%|███████   | 35557/50000 [00:25<00:10, 1337.14it/s]
 71%|███████   | 35691/50000 [00:26<00:10, 1304.37it/s]
 72%|███████   | 35822/50000 [00:26<00:10, 1303.85it/s]
 72%|███████   | 35969/50000 [00:26<00:10, 1348.73it/s]
 72%|███████   | 36105/50000 [00:26<00:10, 1321.92it/s]
 73%|███████   | 36254/50000 [00:26<00:10, 1366.97it/s]
 73%|███████   | 36392/50000 [00:26<00:11, 1215.72it/s]
 73%|███████   | 36529/50000 [00:26<00:10, 1256.61it/s]
 73%|███████   | 36661/50000 [00:26<00:10, 1273.83it/s]
 74%|███████   | 36799/50000 [00:26<00:10, 1301.64it/s]
 74%|███████   | 36931/50000 [00:27<00:10, 1296.67it/s]
 74%|███████   | 37068/50000 [00:27<00:09, 1316.03it/s]
 74%|███████   | 37204/50000 [00:27<00:09, 1327.55it/s]
 75%|███████   | 37342/50000 [00:27<00:09, 1341.39it/s]
 75%|███████   | 37490/50000 [00:27<00:09, 1377.93it/s]
 75%|███████   | 37629/50000 [00:27<00:09, 1321.44it/s]
 76%|███████   | 37763/50000 [00:27<00:09, 1284.49it/s]
 76%|███████   | 37908/50000 [00:27<00:09, 1328.79it/s]
 76%|███████   | 38056/50000 [00:27<00:08, 1368.50it/s]
 76%|███████   | 38195/50000 [00:27<00:08, 1371.94it/s]
 77%|███████   | 38333/50000 [00:28<00:08, 1360.82it/s]
 77%|███████   | 38470/50000 [00:28<00:08, 1332.10it/s]
 77%|███████   | 38604/50000 [00:28<00:08, 1333.46it/s]
 77%|███████   | 38738/50000 [00:28<00:08, 1302.42it/s]
 78%|███████   | 38881/50000 [00:28<00:08, 1337.22it/s]
 78%|███████   | 39022/50000 [00:28<00:08, 1355.58it/s]
 78%|███████   | 39158/50000 [00:28<00:09, 1157.51it/s]
 79%|███████   | 39292/50000 [00:28<00:08, 1206.49it/s]
 79%|███████   | 39440/50000 [00:28<00:08, 1277.01it/s]
 79%|███████   | 39572/50000 [00:29<00:09, 1109.62it/s]
 79%|███████   | 39703/50000 [00:29<00:08, 1162.86it/s]
 80%|███████   | 39832/50000 [00:29<00:08, 1197.72it/s]
 80%|███████   | 39959/50000 [00:29<00:08, 1217.61it/s]
 80%|████████  | 40102/50000 [00:29<00:07, 1273.05it/s]
 80%|████████  | 40243/50000 [00:29<00:07, 1310.10it/s]
 81%|████████  | 40377/50000 [00:29<00:07, 1306.80it/s]
 81%|████████  | 40510/50000 [00:29<00:07, 1313.24it/s]
 81%|████████  | 40650/50000 [00:29<00:06, 1338.03it/s]
 82%|████████  | 40787/50000 [00:30<00:06, 1346.13it/s]
```



```
 82%|████████  | 40923/50000 [00:30<00:06, 1319.98it/s]
 82%|████████  | 41056/50000 [00:30<00:06, 1308.61it/s]
 82%|████████  | 41195/50000 [00:30<00:06, 1331.44it/s]
 83%|████████  | 41329/50000 [00:30<00:06, 1305.47it/s]
 83%|████████  | 41471/50000 [00:30<00:06, 1336.95it/s]
 83%|████████  | 41606/50000 [00:30<00:06, 1305.45it/s]
 83%|████████  | 41738/50000 [00:30<00:06, 1245.00it/s]
 84%|████████  | 41864/50000 [00:30<00:06, 1203.36it/s]
 84%|████████  | 41986/50000 [00:30<00:06, 1185.10it/s]
 84%|████████  | 42106/50000 [00:31<00:06, 1170.00it/s]
 84%|████████  | 42226/50000 [00:31<00:06, 1176.67it/s]
 85%|████████  | 42352/50000 [00:31<00:06, 1191.68it/s]
 85%|████████  | 42477/50000 [00:31<00:06, 1207.78it/s]
 85%|████████  | 42599/50000 [00:31<00:06, 1205.32it/s]
 85%|████████  | 42725/50000 [00:31<00:05, 1220.87it/s]
 86%|████████  | 42853/50000 [00:31<00:05, 1236.23it/s]
 86%|████████  | 42977/50000 [00:31<00:05, 1232.88it/s]
 86%|████████  | 43101/50000 [00:31<00:05, 1221.97it/s]
 86%|████████  | 43240/50000 [00:31<00:05, 1265.09it/s]
 87%|████████  | 43367/50000 [00:32<00:05, 1244.03it/s]
 87%|████████  | 43504/50000 [00:32<00:05, 1277.78it/s]
 87%|████████  | 43633/50000 [00:32<00:05, 1228.07it/s]
 88%|████████  | 43767/50000 [00:32<00:04, 1258.87it/s]
 88%|████████  | 43905/50000 [00:32<00:04, 1290.61it/s]
 88%|████████  | 44046/50000 [00:32<00:04, 1322.97it/s]
 88%|████████  | 44179/50000 [00:32<00:04, 1256.16it/s]
 89%|████████  | 44319/50000 [00:32<00:04, 1295.31it/s]
 89%|████████  | 44450/50000 [00:32<00:04, 1294.55it/s]
 89%|████████  | 44586/50000 [00:33<00:04, 1311.17it/s]
 89%|████████  | 44718/50000 [00:33<00:04, 1289.52it/s]
 90%|████████  | 44848/50000 [00:33<00:04, 1281.28it/s]
 90%|████████  | 44977/50000 [00:33<00:03, 1278.95it/s]
 90%|█████████ | 45116/50000 [00:33<00:03, 1308.74it/s]
 90%|█████████ | 45248/50000 [00:33<00:03, 1258.70it/s]
 91%|█████████ | 45375/50000 [00:33<00:03, 1171.21it/s]
 91%|█████████ | 45516/50000 [00:33<00:03, 1233.31it/s]
 91%|█████████ | 45642/50000 [00:33<00:03, 1197.42it/s]
 92%|█████████ | 45764/50000 [00:34<00:03, 1156.20it/s]
 92%|█████████ | 45910/50000 [00:34<00:03, 1231.77it/s]
 92%|█████████ | 46056/50000 [00:34<00:03, 1291.37it/s]
 92%|█████████ | 46192/50000 [00:34<00:02, 1309.18it/s]
 93%|█████████ | 46333/50000 [00:34<00:02, 1335.91it/s]
 93%|█████████ | 46468/50000 [00:34<00:02, 1307.68it/s]
 93%|█████████ | 46611/50000 [00:34<00:02, 1340.03it/s]
 93%|█████████ | 46746/50000 [00:34<00:02, 1286.33it/s]
 94%|█████████ | 46876/50000 [00:34<00:02, 1257.10it/s]
 94%|█████████ | 47022/50000 [00:34<00:02, 1308.75it/s]
 94%|█████████ | 47167/50000 [00:35<00:02, 1346.44it/s]
 95%|█████████ | 47303/50000 [00:35<00:02, 1329.68it/s]
 95%|█████████ | 47437/50000 [00:35<00:01, 1283.28it/s]
 95%|█████████ | 47581/50000 [00:35<00:01, 1324.77it/s]
 95%|█████████ | 47715/50000 [00:35<00:01, 1291.78it/s]
 96%|█████████ | 47846/50000 [00:35<00:01, 1275.23it/s]
 96%|█████████ | 47978/50000 [00:35<00:01, 1287.47it/s]
 96%|█████████ | 48111/50000 [00:35<00:01, 1299.56it/s]
 97%|█████████ | 48253/50000 [00:35<00:01, 1328.30it/s]
 97%|█████████ | 48387/50000 [00:35<00:01, 1327.07it/s]
```

```
 97%|█████████▋| 48520/50000 [00:36<00:01, 1303.72it/s]
 97%|█████████▋| 48651/50000 [00:36<00:01, 1303.23it/s]
 98%|█████████▊| 48782/50000 [00:36<00:00, 1298.20it/s]
 98%|█████████▊| 48912/50000 [00:36<00:00, 1201.52it/s]
 98%|█████████▊| 49048/50000 [00:36<00:00, 1243.39it/s]
 98%|█████████▊| 49188/50000 [00:36<00:00, 1286.05it/s]
 99%|█████████▊| 49319/50000 [00:36<00:00, 1292.06it/s]
 99%|█████████▉| 49464/50000 [00:36<00:00, 1335.49it/s]
 99%|█████████▉| 49599/50000 [00:36<00:00, 1250.70it/s]
 99%|█████████▉| 49737/50000 [00:37<00:00, 1285.74it/s]
100%|█████████▉| 49868/50000 [00:37<00:00, 1270.10it/s]
100%|██████████| 50000/50000 [00:37<00:00, 1342.12it/s]
```

Task: Implement

Implement the `compute_image_mean_and_std()` method and the `__call__()` method of `NormalizeTransform` in `exercise_code/data/transforms.py`. Compute the rescaled dataset's mean and variance by running the following cell.

In [12]:
```
cifar_mean, cifar_std = compute_image_mean_and_std(rescaled_images)
print("Mean:\t", cifar_mean, "\nStd:\t", cifar_std)
```

```
Mean:    [0.49191375 0.48235852 0.44673872]
Std:     [0.24706447 0.24346213 0.26147554]
```

To test your implementation, run the following code:

In [13]:
```
_ = test_compute_image_mean_and_std(cifar_mean, cifar_std)
```

```
CIFARImageMeanTest passed.
CIFARImageStdTest passed.
Method compute_image_mean_and_std() correctly implemented. Tests passe
d: 2/2
Score: 100/100
```

In [14]:
```
# The rescaled images will be deleted now from your ram as they are no l
onger needed
try:
    del rescaled_images
except NameError:
    pass
```

Now you can use the mean and standard deviation you computed to normalize the loaded data. This can be done by simply adding the `NormalizeTransform` to the list of transformations our dataset applies in `__getitem__()`.

Task: Check Code

Please check out the `ComposeTransform` in `transforms.py`. Later on, we will most often use multiple transforms and chain them together. Remember that the order is of importance here!

In [15]:
```
# Set up both transforms using the parameters computed above
rescale_transform = RescaleTransform()
normalize_transform = NormalizeTransform(
    mean=cifar_mean,
    std=cifar_std
)

final_dataset = ImageFolderDataset(
    root=cifar_root,
    transform=ComposeTransform([rescale_transform, normalize_transform])
)
```

You can now check out the results of the transformed samples:

In [16]:
```
sample_item = final_dataset[0]
sample_label = sample_item["label"]
sample_image = sample_item["image"]

print('Sample normalized image shape:', sample_image.shape)
print('Sample normalized image first values:', sample_image[0][0])
```

```
Sample normalized image shape: (32, 32, 3)
Sample normalized image first values: [-1.94341602  0.03219212  1.78596
73 ]
```

5. Save your Dataset

Now save your dataset and transforms using the following cell. This will save it to a pickle file `models/cifar_dataset.p`. We will use this dataset for the next notebook and this will count for the submission.

Note

Each time you make changes in `dataset`, you need to rerun the following code to make your changes saved, but **this is NOT the file which you should submit**. You will find the final file for submission in the second notebook.

In [17]:
```
save_pickle(
    data_dict={
        "dataset": final_dataset,
        "cifar_mean": cifar_mean,
        "cifar_std": cifar_std,
    },
    file_name="cifar_dataset.p"
)
```

Key Takeaways

1. Always have a look at your data before you start training any models on it.
2. Datasets should be organized in corresponding **Dataset** classes that support `__len__` and `__getitem__` methods, which allow us to call `len(dataset)` and `dataset[index]` .
3. Data often needs to be preprocessed. Such preprocessing can be implemented in **Transform** classes, which are callables that can be simply applied via `data_transformed = transform(data)` . However, we will rarely do that and apply transforms on the fly using a dataloader which we will introduce in the next notebook.