

Tiki's implementation of a custom transformation

class MyTransform

I implemented a custom transformation class `MyTransform` to modify each input image tensor used by a PyTorch `DataLoader`. The `DataLoader` will apply my transform through a provided `collate` function, so my implementation must define how each image is transformed before training. The goal is to learn to integrate PyTorch transformations directly into data pipelines.

My implementation:

```
import torch # We import PyTorch so we can do fast tensor math on images.
```

```
class MyTransform:
    def __call__(self, x: torch.Tensor) -> torch.tensor:
        #How bright is the image
        mean = x.mean()

        #How much contrast
        std = x.std()

        #Stabilize math
        eps = x.new_tensor(1e-6)
        std = torch.clamp(std, min=eps)

        #normalize per image
        x_norm = (x - mean) / std

        #Prevent outliers from exploding
        x_norm = x_norm.clamp(min=-3.0, max = 3.0)

        #keep output in valid image range
        x_out = (x_norm + 3.0) / 6.0

        #return output
        x_out
```