

# Computer Vision 보고서 13주차

임형찬, 202102559, 인공지능학과

황재성, 202102571, 인공지능학과

## [과제]

### - 구현코드 및 코드설명

```
class IntensityTransform(nn.Module):
    def __init__(self, intensities, channels, **kwargs):
        super(IntensityTransform, self).__init__(**kwargs)
        self.channels = channels
        self.max_value = intensities - 1

    def forward(self, inputs):
        """
        인자 정보
        inputs: two inputs
        inputs[0]: 3 channel image normalized [-1, 1], bs x 3 x H x W
        inputs[1]: bs x 3 x 256 transform feature vectors per R, G, B normalized [-1, 1]

        return
        outputs: bs x 3 x H x W
        """

        #####
        # TODO Intensity Transform 구현
        # 참고 - 사용된 함수들 (반드시 아래 함수 안써도 될 자유롭게 구현)
        # torch.split
        # torch.squeeze
        # F.one_hot
        # torch.bmm
        #####
        images, transforms = inputs

        bs, c, h, w = images.shape

        # images [-1, 1] -> [0, 1]
        images = 0.5 * images + 0.5
        # images [0, 1] -> [0, self.scale] integer type
        denormalized_images = torch.round(self.max_value * images).type(torch.int64).to(DEVICE)

        transforms = transforms.unsqueeze(3).to(DEVICE)
        w_ = denormalized_images.size(3)
        i, cnt = 0, 1
        while w_ > cnt:
            cnt *= 2
            i += 1

        for _ in range(i):
            transforms = torch.cat([transforms, transforms], dim=3)

        images, transforms = torch.split(denormalized_images, 1, dim=1), torch.split(transforms, 1, dim=1)

        c = torch.gather(input=transforms[0], dim=2, index=images[0])
        h = torch.gather(input=transforms[1], dim=2, index=images[1])
        w = torch.gather(input=transforms[2], dim=2, index=images[2])
        out = torch.cat([c, h, w], dim=1)

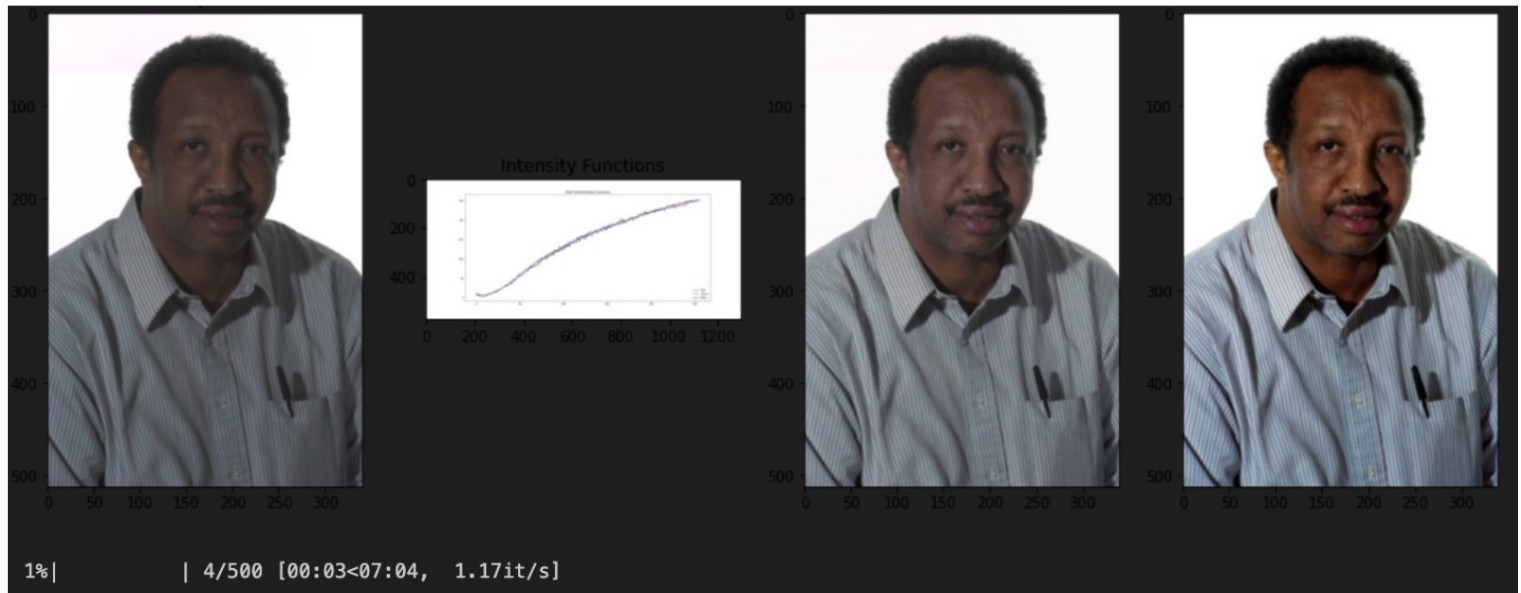
        return out
```

4번째차원(=width)을 확장(by. unsqueeze(3))

Denormalized image의 크기를 확인, 이를 기준으로 반복한 후 transforms의 4번째 차원 즉, 너비를 이를 기준으로 확장.

이후 R,G,B채널로 분할하고 각 채널별 반환값을 gathe함수를 이용해 채널별 선택해 c,h,w에 저장.  
이들을 합쳐서 out텐서에 반환한다.

이때, 중요포인트는 torch.gather함수로 인덱스에 따라 값을 선택하는데 사용하고, 이때 R,G,B채널을 기준으로 변환된 값을 선택하고 이를 c,h,w에 저장해 다시 결합한다.



Train & Test results

Epoch	Time	Loss Type	Value
1	[1129_14:05]	Total loss	0.2953
		L1 loss	0.1498
1	[Test]	PSNR / SSIM	23.3191 / 0.8620
		Perceptual loss	0.1455
2	[1129_14:07]	Total loss	0.2552
		L1 loss	0.1363
2	[Test]	PSNR / SSIM	23.4597 / 0.8751
		Perceptual loss	0.1189
3	[1129_14:10]	Total loss	0.2444
		L1 loss	0.1313
3	[Test]	PSNR / SSIM	23.4018 / 0.8779
		Perceptual loss	0.1131