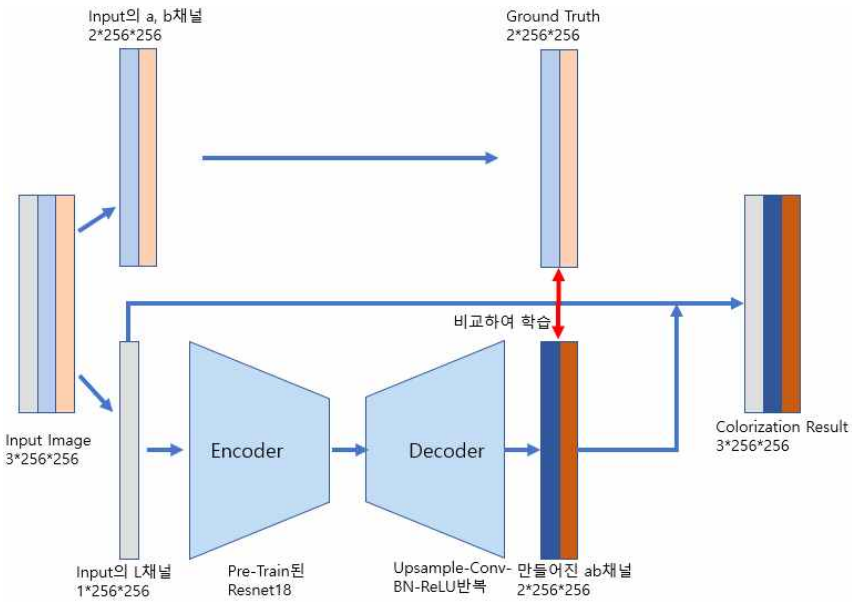
딥러닝의 기초 프로젝트 7조
Flower Colorization - Model #1 설명

1. Background - 칼라 이미지의 Lab채널

칼라 이미지는 RGB 채널을 이용해서도 나타낼 수 있지만, Lab 채널을 이용해서도 나타낼 수 있습니다. Lab 채널에서 L은 밝기(Lightness)를 의미하고 a와 b는 각각 적록과 황청을 의미한다는데 저도 잘은 모르겠습니다. 다만 우리가 Lab 채널을 사용하는 까닭은, Lab 채널로 표현된 칼라 이미지에서 L 채널만 빼내면 그것이 곧 Grayscale된 이미지가 되기 때문입니다. 따라서 우리는 남은 2개의 채널인 a, b 채널만 학습하면 되므로 3개의 채널을 학습시켜야 하는 RGB보다 유리합니다.

RGB와 Lab의 변환공식 또한 구글링하면 잘 나옵니다만 어차피 우리는 skimage.color 라이브러리의 lab2rgb, rgb2lab 함수를 쓸 것이기 때문에 몰라도 될 것 같습니다. 그냥 칼라 이미지는 Lab 채널로 표현되고, 우리는 그 중에서도 a 채널, b 채널만 학습한다는 사실만 아시면 될 것 같습니다.

2. 모델 설명



위 그림은 제가 구현한 모델을 보여줍니다. 구체적으로 아래와 같이 작동합니다.

1. Input image로부터 L 채널과 ab 채널을 분리합니다.
2. L 채널을 Encoder에 넣습니다. 이때 Encoder는 입력된 이미지로부터 Feature map을 생성해주는 역할을 합니다.
3. Encoder의 결과를 Decoder에 넣습니다. 이때 Decoder의 output은 채널이 2개가 되도록 합니다.
4. Decoder의 Output과 1에서 분리한 ab 채널을 비교하여 Loss를 계산한 후 Backpropagation을 합니다.
5. Decoder의 Output과 1에서 분리한 L 채널을 합쳐 Colorization Result를 Ouput으로 내보냅니다.

Encoder는 pytorch 라이브러리에서 기본적으로 제공해주는 resnet18을 이용했고 Decoder는 Upsampling layer - Convolution Layer - Batch Normalization - ReLU 의 반복입니다. 다만 마지막 활성화 함수로는 Sigmoid를 사용하였는데, 이는 이미지의 a, b값이 -1에서 1의 값만 가지기 때문에 Sigmoid를 사용한 것입니다.

★중요한 점★

여기서 위 모델을 통째로 학습시킬 경우 Encoder가 Feature를 제대로 추출하지 못해서 Colorization이 그냥 하나의 색으로 통일되어버립니다. 아래는 그 예시입니다.

그래서 저는 Encoder가 Feature를 제대로 추출할 수 있도록 아래 방법을 사용했습니다.

1. Encoder(여기서는 resnet18)을 먼저 Classifier로써 학습시킨다.
2. 충분히 학습되었으면 Encoder의 마지막 linear layer를 제외한 weight들을 본 model에 로드한다.

* 참고 : Encoder를 Classifier로써 학습시켰을 때 최종 Validation Accuracy는 91.076%였습니다.

제가 드린 첨부파일의 checkpoint/encoder 폴더에 들어가시면 'flower102_resnet18_91.076.pth'파일이 있습니다. 이 파일이 바로 Classifier로써 학습된 Encoder의 Weight입니다.

3. 파일 실행하기

(1) 로컬에서 작업

받으신 zip 파일을 적당한 곳에 압축을 푸신 후, jupyter notebook으로 'main.ipynb'파일을 실행하시면 됩니다. 이때 numpy, matplotlib, skimage, torch 라이브러리가 설치되어있어야 합니다.

(2) Google Colab에서 작업

받으신 zip 파일의 압축을 푸시고 구글 드라이브에 업로드한 후 'main.ipybn'파일을 Google Colab에서 여시면 됩니다. 구글 Colab에는 필요한 라이브러리가 기본적으로 설치되어있으므로 별도의 라이브러리를 설치할 필요가 없습니다.

4. 모델 트레이닝하기

(1) 코드 설명

코드에 대한 설명은 'main.ipynb' 파일에 같이 있으니 참고하시면 됩니다. 정 모르겠다 싶으시면 그냥 모든 셀에 대해서 한 번씩 실행(Shift 엔터)하셔도 작동합니다.

(2) Hyper parameter 튜닝

4. Train the Model

4.1 Set hyperparameters, optimizer, loss, etc.

하이퍼파라미터, Loss Function, 옵티마이저, epoch 등을 결정합니다.

In [27]:
```
epoch_num = 30 # 반복할 epoch 수
learning_rate = 0.01 # learning rate
criterion = nn.MSELoss() # Loss 함수
# MSE로하면 학습할때 회색조로 학습되는 문제가 있습니다. 아무래도 Loss를 다른걸로 써야할 것 같습니다.
optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate, momentum=0.9, weight_decay=1e-5, nesterov=True) # 옵티마이저
train_loader = torch.utils.data.DataLoader(train_imagefolder, batch_size=32, shuffle=True) # 배치사이즈
```

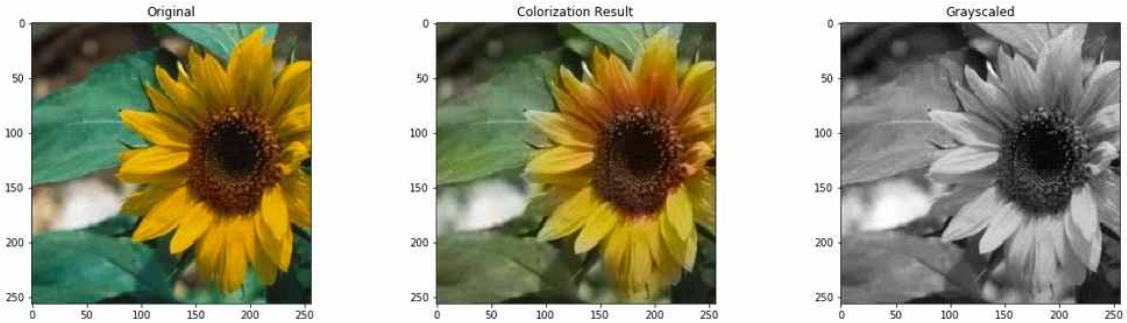
Hyper parameter 튜닝은 여기서 하시면 됩니다. 반복할 에폭 수, Learning rate, Loss, Optimizer, Batch Size를 설정할 수 있습니다.

5. 개선해야할 점

(1) 부족한 Detail

위 사진은 해바라기의 Colorization 결과의 일부를 확대한 것인데요. 잘 보시면 꽃잎 주위로 꽃잎이 아닌데도 노르스름하게 색칠되어 있는걸 볼 수 있습니다. 이렇게 디테일한 부분의 Colorization은 조금 부족 합니다.

(2) 칙칙한 색감



위 사진도 해바라기의 Colorization 결과인데, 원본(좌측)에 비하면 Colorization 결과(가운데)가 상당히 칙칙하게 느껴집니다. 비단 위 사진뿐만 아니라 전체적으로 색칠이 칙칙하게 되는 경향이 있습니다.

(3) 꽃만 색칠함

제가 아직 꽃에 대해서만 학습시켜서 꽃이 아닌 사물에 대해선 제대로 색칠을 못합니다.