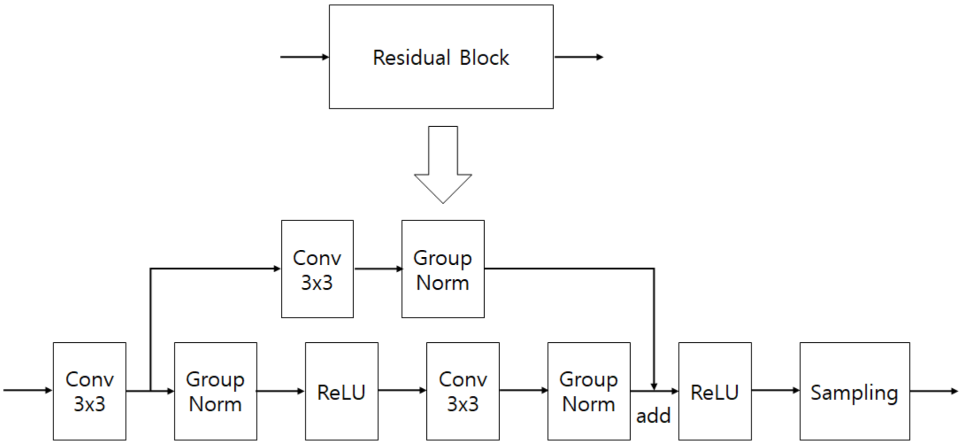
# Anime Sketch Colorizer - Automatic Sketch Colorization with reference image

tlwh1179@gmail.com

Note: This paper is not formal report. In this paper, only brief descriptions on model and examples are contained.

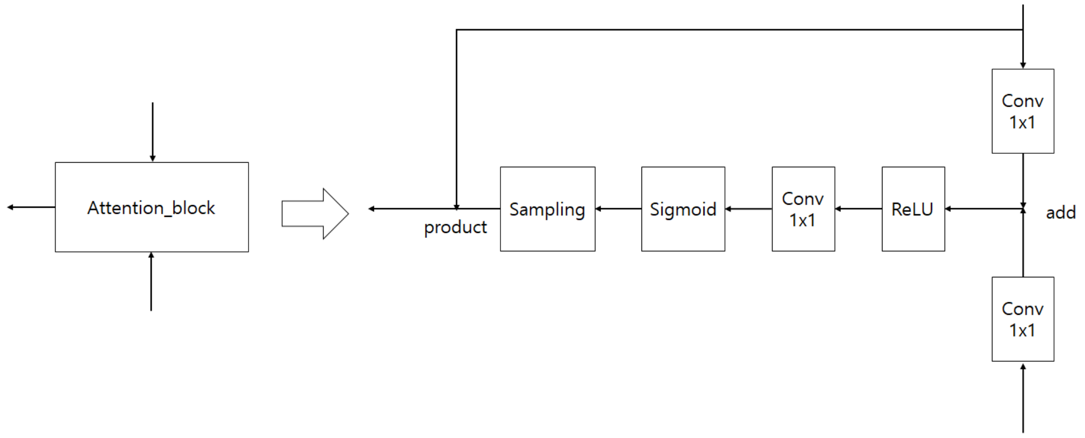
## 1. Model description

### 1) Residual Block



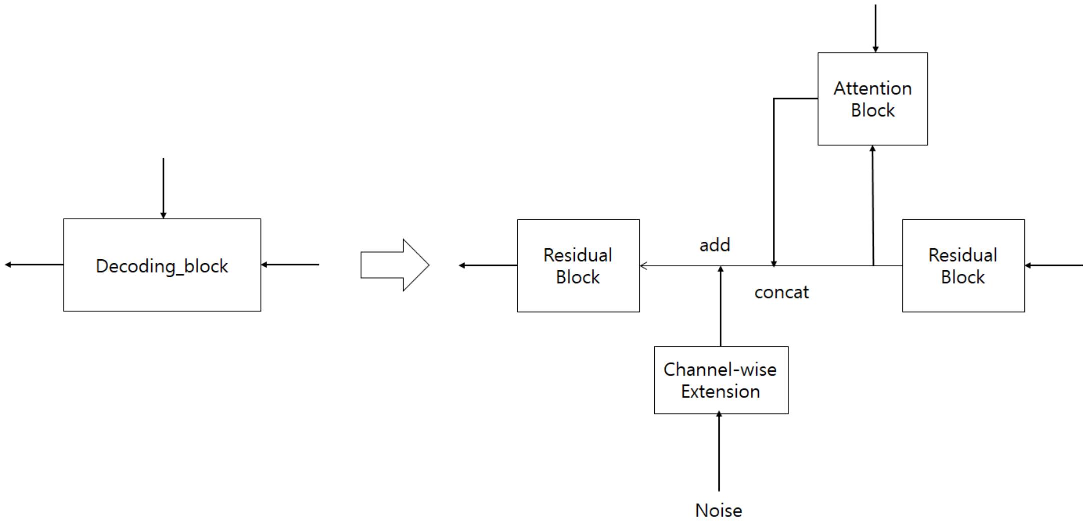
[Figure 1] shows a block diagram of Residual Block. As training batch size (2) is very small, the batch normalization (BN) would not be effective to help training as well. Instead, referring [1], BN is replaced by group normalization (GN) and weight standardization (WS) is applied to all convolutional layers. According to the [1], this change gives similar or better effect of using BN. The sampling layer has two modes; for 'down' mode, do max-pooling. for 'up' mode, do up-sampling.

## 2) Attention Block



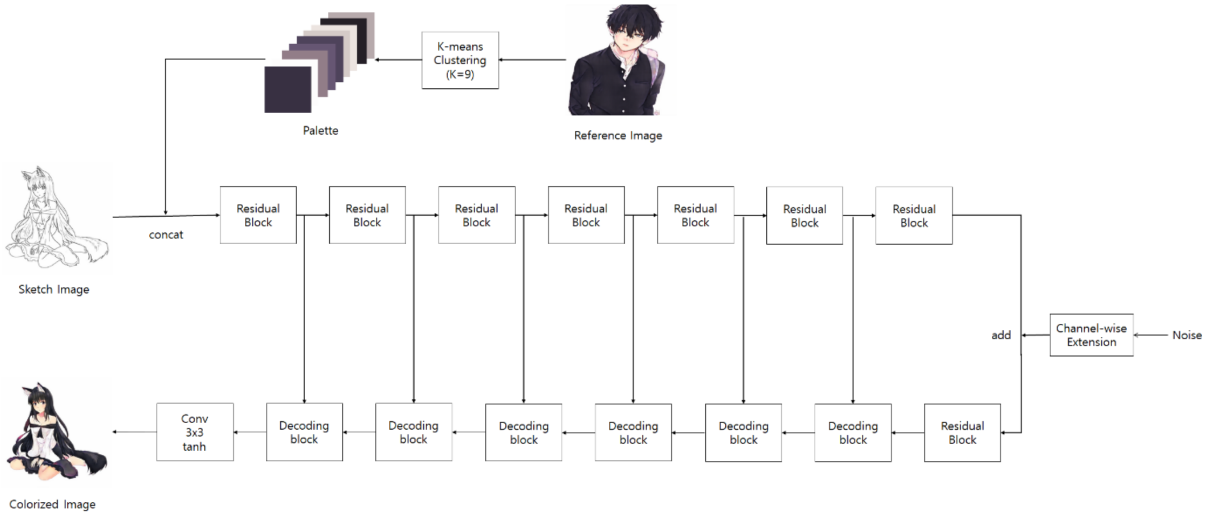
[Figure 2] shows a block diagram of Attention Block. This block learns to suppress irrelevant regions in an input image while highlighting salient features useful for colorization task. [2] For more details, please refer the paper [2].

## 3) Decoding Block



[Figure 3] shows a block diagram of Decoding Block. This block is placed in decoder and converts low-level feature to high-level feature. To improve the details, random noise is scaled per-channel and injected to feature. This task gives 'Stochastic variation' on result. [3] Note that the weight of noise scaling is trainable parameter.

### 4) Entire Generator



[Figure 4] shows a block diagram of the entire generator. Before forwarding sketch image on model, the extracted color palette is concatenated to sketch image. The color palette is tensor of 3*9 channels and generated by applying K-means clustering algorithm on the reference image. The number of clusters is 9. Finally, the input of colorization model has 30 channels. As there is attention block in all decoding block, the model is similar to that shown on [2].

### 5) Discriminator

Comparing with usual discriminator model, there is no significant change on discriminator except the spectral normalization. Referring [4], applying spectral normalization on the weight of discriminator encourages convergence.

## 2. Loss Function

There are four loss functions to be minimized;

(1) Adversarial Loss (LSGAN): L2 distance of real/fake label and result of discriminator
(2) Content Loss: L1 distance between fake and real
(3) Feature Loss: L2 distance between high level features of fake and real
(extracted from four top layers of VGG16 pretrained on ImageNet)
(4) TV Loss: Total variance of fake image.

And the total loss is $L_{Adversarial} + \lambda_1 L_{Content} + \lambda_2 L_{Feature} + \lambda_3 L_{TV}$. The values of $\lambda_1$=100, $\lambda_2$=1e-4, $\lambda_3$=1e-2 were used during training.

# 3. Reference

[1] Siyuan Qiao et al., "Weight Standardization", https://arxiv.org/abs/1903.10520, 2019. 3. 25., 2020.1.19.

[2] Ozan Oktay et al., "Attention U-Net: Learning Where to Look for the Pancreas", MIDL 2018, 2018.5.20.

[3] Tero Karras, Samuli Laine, Timo Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks", https://arxiv.org/abs/1812.04948, 2019.3.29., 2020.1.22.

[4] Takeru Miyato et al., "Spectral Normalization for Generative Adversarial Networks", ICLR 2018, 2018.2.18.