# Supplementary Material
# GIF2Video: Color Dequantization and Temporal Interpolation of GIF images


## Abstract

*In our main paper, we have proposed GIF2Video, the first learning-based method for enhancing the visual quality of GIFs in the wild. It consists of two components that perform color dequantization and temporal interpolation of GIF images respectively. In this supplementary material, we provide more details and results regarding experiments on GIF-Faces and GIF-Moments datasets, which could not be included in the main paper due to page limitation.*


## 1. Experiment Details

### 1.1. GIF creating tools

We use MATLAB(2014a) to create GIF sequences. More specifically, we use the following function, $[X,\ map] = rgb2ind(RGB,\ n,\ dither\_option)$. It converts the $RGB$ image to an indexed image $X$ using minimum variance quantization and optional dithering. $map$ is a color palette of at most $n$ colors. In our experiments, we set $n = 32$. As discussed in the main paper, GIF images often contain undesirable visual artifacts such as flat color regions, false contours, color shift, and dotted patterns.

### 1.2. Architecture of U-Net in CCDNet

Figure 1 illustrate the architecture of the U-Net used in the proposed CCDNet. For convolutional feature transformation (blue arrow), two consecutive convolutional modules with kernel size of 3 are typically used. For feature downsampling (green arrow) and upsampling (yellow arrow), Average Pooling and Deconvolution layers are used respectively.

### 1.3. Hyper-parameters for training CCDNet

For data augmentation, we perform random image cropping and horizontal flipping. For data normalization, we linearly map the pixel values of input GIFs from the range $[0,\ 255]$ to the range $[-1,\ 1]$. For optimization, we use ADAM optimizer ($\beta_1 = 0.5, \beta_2 = 0.999$) and a weight decay of $0.0001$. For training without adversarial losses, the learning rate starts with 0.0001 and gradually decays (decay ratio $\gamma = 0.5$) every 10 epochs. The training stops after 60 epochs. For training with adversarial losses, the learning rate is set to 0.0002 with no decay. The training stops after 30 epochs.

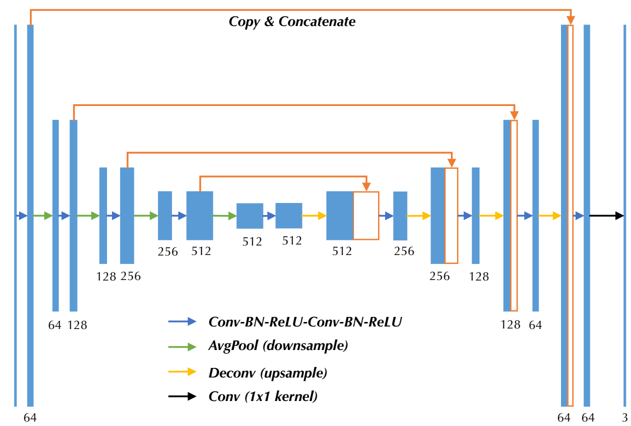


Figure 1. **Architecture of U-Net in CCDNet.**

### 1.4. Dynamic adversarial training of CCDNet

When adversarial loss is included for training the CCDNet, it is critical to use a robust training strategy that can ensure the conditional GAN framework trains stably. We dynamically control the learning of the generator $G$ and the discriminator $D$, depending on the prediction accuracy $Acc(D)$ of $D$ on the current input mini-batch. If $Acc(D) \leq 25\%$, that means $D$ is under-trained, so we only train $D$ using current mini-batch. If $Acc(D) \geq 75\%$, that means $D$ is over-trained, so we only train $G$ using current mini-batch. Otherwise when $25\% < Acc(D) < 75\%$, $G$ and $D$ are jointly trained in this case.

## 2. More Qualitative Results

In this supplementary material, we present more qualitative results. Please look at the HTML files attached.