

# SR3(Image Super-Resolution via Iterative Refinement)

## ソース

- [Image Super-Resolution via Iterative Refinement](#)
- ソースコード
  - <https://github.com/Janspiry/Image-Super-Resolution-via-Iterative-Refinement/tree/master>

## 概要

- Conditional Diffusion ModelによるSuper Resolution
- 条件が低画質画像
- ノイズを予測するNNの入力が3つ(ノイズ付与画像、ステップ数、低画質画像)
- NNはUNet
- UNetでは、低画質画像をノイズ付与画像と画質をそろえるために、条件の低画質画像をbicubic interpolationで拡大したあとにconcatenateした
- ノイズスケジュールについては、WaveGradにしたがった
  - WaveGrad: Estimating Gradients for Waveform Generation. In ICLR, 2021
  - 一様分布からステップ数をサンプリング
  - そのあとにノイズを得るためにさらに一様分布からサンプリングする
  - 詳細は以下にある

## 提案手法

- Conditional Diffusion ModelでSuper Resolutionするというだけである
- 以下のゼロ作の数式と異なる点は2点(左がゼロ作、右が論文)
  - <https://github.com/msdsm/oreilly-deeplearning-5/blob/main/09/09.ipynb>
  - $\beta \rightarrow \alpha$
  - $\bar{\alpha} \rightarrow \gamma$
- 低画質画像が $x$ で高画質画像が $y$

## 学習時

- 以下を繰り返す
  - i.  $(\mathbf{x}, \mathbf{y}_0) \sim p(\mathbf{x}, \mathbf{y})$  (データセットからサンプリング)
  - ii.  $\gamma \sim p(\gamma)$
  - iii.  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  - iv.  $\nabla_{\boldsymbol{\theta}} \|f_{\boldsymbol{\theta}}(\mathbf{x}, \sqrt{\gamma}\mathbf{y}_0 + \sqrt{1-\gamma}\boldsymbol{\epsilon}, \gamma) - \boldsymbol{\epsilon}\|_p^p$  (損失計算)
- $f_{\boldsymbol{\theta}}$ はU-Net
  - 入力が順に低画質画像、ノイズ付与画像、ステップ数
  - 2つ目の入力のノイズ付与画像は $q(\mathbf{y}_t | \mathbf{y}_0)$ の式
  - ステップ数 $t$ ではなく $\gamma$ を入力している
  - 低画質画像(条件) $\mathbf{x}$ をbicubic interpolationで拡大して $\mathbf{y}_t$ と画質を一致させた後にconcatenation
- ノイズスケジュール  $p(\gamma)$  について
  - $p(\gamma) = \sum_{t=1}^T \frac{1}{T} U(\gamma_{t-1}, \gamma_t)$
  - 学習時の $\gamma \sim p(\gamma)$ は以下の2段階の一様分布サンプリングで得る
    - $t \sim U(0, T)$
    - $\gamma \sim U(\gamma_{t-1}, \gamma_t)$

## 生成時

1.  $\mathbf{y}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2.  $t = T, \dots, 1$ まで順に以下を繰り返す
  - i.  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  ( $t = 1$ の時は $\mathbf{z} = \mathbf{0}$ )
  - ii.  $\mathbf{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}_t, \gamma_t) \right) + \sqrt{1-\alpha_t} \mathbf{z}$
3.  $\mathbf{y}_0$ を出力