

GAN を用いた被写体ブレの補正

Correction of object motion blur using GAN

2020Y01 板津 敬佑(Keisuke Itazu)

担当教員 出口利憲(Toshinori Deguchi)・安田真(Makoto Yasuda)

1. 序論

本研究では被写体ブレした画像を機械学習を用いて超解像処理を行う。これにより、被写体ブレの入った画像からブレを取り除くことで高解像度の画像を生成することを目的とする。

2. 研究の背景

本研究の背景として、近年、画像の分野で機械学習を取り入れる動きが盛んになっている。手ブレ等のノイズに対しては機械学習を用いることで一定の効果が得られる[1]。しかし、撮影機本体に無関係に動く被写体のブレの補正に機械学習の方面からアプローチした例は少ない。また現状ハードウェア面から被写体ブレを防ぐにはシャッタースピードを速めるしか方法は無く、この方法も被写体ブレを完全に無くすことは出来ない。そこで、現在手ブレにおいては機械学習が一定の成果を出しているため、被写体ブレに対しても機械学習が有効な対策になりうると感じたためこのテーマに至った。

3. 研究手法

本研究では被写体ブレした画像を機械学習を用いて超解像処理を行う。これにより、被写体ブレした画像からブレを取り除いた画像を生成することを目的とする。ここで、超解像処理とは低解像度の画像を高解像度画像に変換することを指す。特に最近ではGANを用いることで現実の写真に近い自然な画像を生成出来るようになっている。

本研究では ESRGAN と呼ばれる手法を用いる。ESRGAN の生成器は画像のピクセルを用いる Pixel loss、特徴抽出した結果を用いる Perceptual loss、生成器と識別機の出力結果を用いる Adversarial loss の3つの損失関数を最小化するように学習する。生成器のネットワークは、畳み込み層、23個の Residual-in-Residual Dense Block (RRDB)、畳み込み層、アップサンプリングレイヤー、そして二つの畳み込み層から構成される。基本的な

ESRGAN の構成を Fig. 1 に示す[2]。

なお、本研究では画像のブレに対してのみ補正を行うため、アップサンプリングは行わずに学習を行う。ESRGAN は SRGAN の Batch Normalization 層 (BN 層) と呼ばれる訓練データの平均と分散から特徴量の正規化を行う部分を削除しており、これにより SRGAN の課題であった訓練データとテストデータが大きく異なる時に生じるノイズを克服している。その構成を Fig. 2 に示す[2]。

RRDB は、層数や接続数を増やすことで性能が向上するという観測に基づいて、SRGAN のオリジナルの残差ブロックよりも深く複雑な Residual-in-residual の構造を採用しており、Dense Block を弱い残差で結合している。この RRDB を用いたより深いモデルは、例えば画像中の規則的な構造の意味情報をとらえる表現力が高く、生成画像のテクスチャをさらに向上させることが出来る。モデルの図を Fig. 3 に示す[2]。

GAN の学習は2段階で構成されており、事前学習では、本物画像と低解像度画像との間の L1 ノ

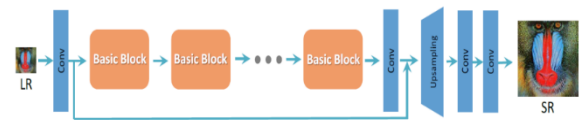


Fig. 1 Network structure of the generator.

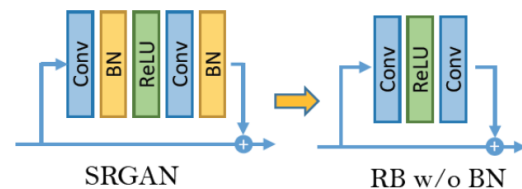


Fig. 2 Structure of residual block.

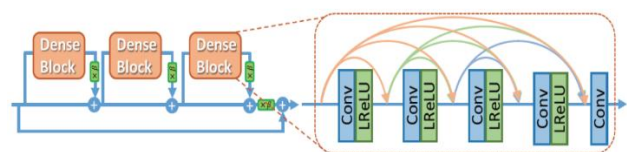


Fig. 3 Structure of RRDB.

ルムの距離を評価する損失関数 Pixel loss を用いる。G を生成器、 x_i を低解像度の本物画像、 $G(x_i)$ を生成画像、 y を本物画像とした時、

$$L_1 = E_{x_i} \|G(x_i) - y\|_1$$

となる。この時 $E_x[\cdot]$ はミニバッチ内のすべてのデータの平均を取る演算をする。画素単位の損失を用いた事前学習を行うことで、より自然な超解像画像を生成することが出来る。この事前学習を終えた後、ほかの損失関数も加えて学習を行う。特徴量抽出器 F によって得られた特徴量を用いる Perceptual loss を用いて、L1 ノルム距離が最小するように学習する [3]。

$$L_{percep} = E_{x_i} \|F(G(x_i)) - F(y)\|_1$$

次に、Adversarial loss は以下の通りになる。

$$L_G^{Ra} = -E_{x_r} \left[\log \left(1 - D_{Ra}(x_r, x_f) \right) \right] \\ - E_{x_f} \left[\log \left(D_{Ra}(x_f, x_r) \right) \right]$$

この時 D_{Ra} は実際の画像 x_r が偽の画像 x_f よりも比較的現実的である確率を予測する。生成器の Adversarial loss を用いることでより詳細なテクスチャの学習ができる。以上3つの損失関数を用いて学習し、異なる損失項のバランスをとるための重み係数 λ 、 η をかけることで算出した値を生成器の Loss として、生成器の学習を行う [3]。

$$L_G = L_{percep} + \lambda L_G^{Ra} + \eta L_1$$

識別機の損失関数においては Relativistic Average Discriminator (RAD) が用いられ、以下で定義される [2]。

$$L_D^{Ra} = -E_{x_r} \left[\log \left(D_{Ra}(x_r, x_f) \right) \right] \\ - E_{x_f} \left[\log \left(1 - D_{Ra}(x_f, x_r) \right) \right]$$

生成器および識別機の損失関数には RAD を用いており、これにより従来の SRGAN と異なり生成画像と本物画像の両方を用いて学習する事が出来る。

4. 実験方法

一般的には ESRGAN を用いた超解像を行う場合、本物画像を縮小したものを低解像度画像として扱うが、前述の通り画像のブレの補正に ESRGAN を応用するため、画像サイズは同じで本物画像にブレの入った画像を低解像度画像として実験を行う。また、一つの画像のサイズを 128×128 とし、それぞれ 60 枚ずつ用いて学習を行った。

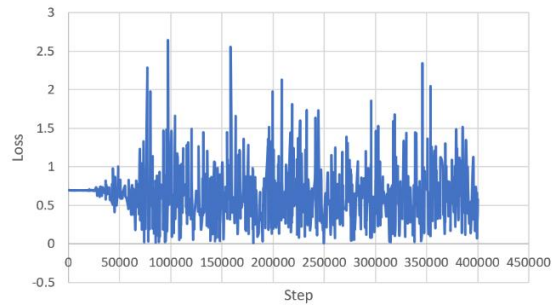


Fig. 5 Discriminator loss.

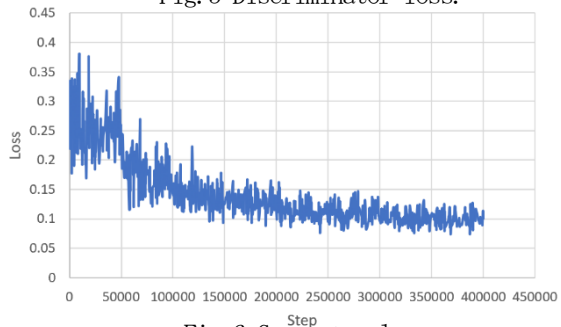


Fig. 6 Generator loss.

5. 結論

識別機と生成機の損失をそれぞれ Fig. 5、Fig. 6 に示す。Fig. 5 から識別機の損失は 50000 ステップ目から増減を繰り返していることが分かる。Fig. 6 から生成機の損失は徐々に小さくなっていることが分かる。

ESRGAN では生成機が生成した画像を識別機が本物かどうか見分けることが出来無い状態を目指す。つまり、生成機の損失が最小、識別機の損失が最大になれば学習は成功であり、結果から学習は出来ていると考えられる。しかし、モデルのテスト結果では微かに輪郭が浮き出る程度に留まっている。理由として、学習用のデータの人工的なブレが大き過ぎる事や、データのサイズや総数が小さかったことが考えられるため、今後はその部分を考慮して学習させる必要がある。

参考文献

- [1] S. Maeda, Unpaired Image Super-Resolution using Pseudo-Supervision, Conference on Computer Vision and Pattern Recognition. (CVPR), pp. 291-300, 2020.
- [2] T. Henighan et al, Scaling Laws for Auto-regressive Generative Modeling. arXiv:2010.14701, 2020.
- [3] C. Ledig et al, Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, Conference on Computer Vision and Pattern Recognition (CVPR), pp. 105-114, 2016.