

コンパクトなGAN

Compact GAN

峯澤 彰*
Akira Minezawa
吉田 浩*
Hiroshi Yoshida
杉本和夫*
Kazuo Sugimoto

要旨

近年、画像処理技術はディープラーニングによって飛躍的な進歩を遂げており、画像合成分野では特に敵対的生成ネットワーク(Generative Adversarial Network⁽¹⁾: GAN)と呼ばれる手法によって、実物と見分けがつかないほどの写実的な画像を合成することが可能になりつつある。一方、この手法は演算負荷や消費メモリが膨大であり、利用可能な環境が限定的であることが課題であった。三菱電機はこの課題に対し、演算量やメモリ量を削減し、高速かつ省メモリで画像合成が可能な方式を開発した。

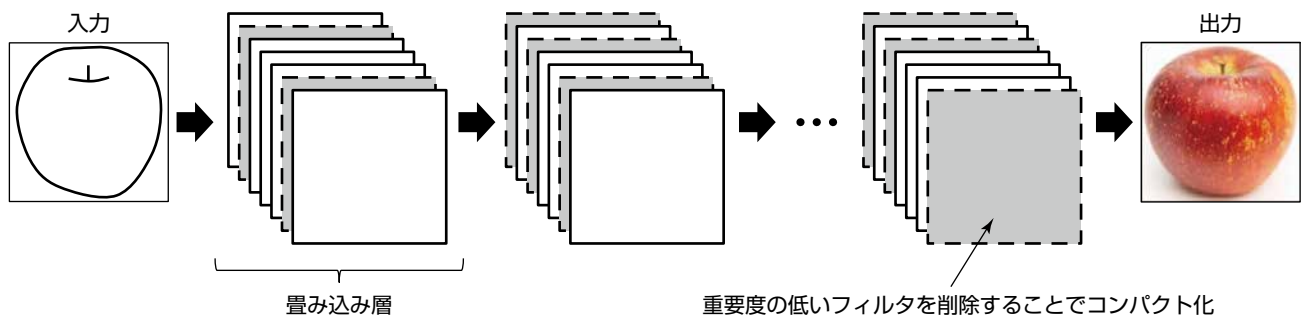
開発方式は、通常のGANと同様の学習を実施後、学習

したニューラルネットワークの畳み込み層について、畳み込み層の各フィルタに対して画像合成処理の性能寄与度の低いフィルタを削除することでコンパクトな画像合成ネットワークを実現している。このとき、性能寄与度として真贋(しんがん)判定AI(Artificial Intelligence)による判断結果が反映される点が特徴である。

シミュレーションの結果、開発方式は通常のGANによって学習されたモデルに対して、同程度の画像合成精度を保ちつつ、画像合成処理時間及び使用メモリ量を10分の1に削減できることを確認した。

- ・新アルゴリズムによって画像合成AIの各層のフィルタの重要度(合成画像の品質にどれだけ寄与するか)を評価
- ・重要なフィルタだけを残し、演算量とメモリ量を10分の1にする

画像合成AIを構成する多層ニューラルネットワーク



- (1) ニューラルネットワークの畳み込み層の各フィルタの重要度を評価する
- (2) 重要度の低いフィルタを削除し、重要なフィルタだけを残す

GANによる画像合成AIのコンパクト化

GANは、画像合成AIであるGeneratorと真贋判定AIであるDiscriminatorという二つのニューラルネットワークが互いに敵対して学習することで高品質な画像を合成可能なGeneratorを得ることができる。しかしながら、Generatorは高い表現力を獲得するために非常に多層の畳み込みニューラルネットワークで構築されることが多いため、その演算負荷が課題となる。これを各畳み込み層のフィルタの重要度に基づいて効率的に削除することで、画像合成品質を保ったまま高速処理・省メモリを実現した。

1. ま え が き

近年のディープラーニングの発展は目覚ましく、画像認識・画像合成技術でのブレークスルーを実現している。画像合成分野では特に敵対的生成ネットワーク(GAN)という技術の誕生によって、ニューラルネットワークを用いて実物と見分けがつかないほどの写実的な画像を合成することが可能になりつつあり⁽²⁾、リアルな合成画像を必要とする様々な用途への応用の可能性が広がる一方で、その合成画像の画質の高さから、特定の人物をターゲットにした実物と見分けのつかない“偽動画”の制作・拡散によるプライバシー侵害が世界的な社会問題化するなど、ディープラーニングによる画像合成技術の日常生活への影響は拡大している。一方、この技術を利用するには潤沢な計算資源が要求される点が技術課題であった。今回、この課題に対して演算量やメモリ量を削減し、高速かつ省メモリで画像合成が可能な方式を開発した。

2. GAN

2.1 ニューラルネットワークによる画像合成

図1は、畳み込みニューラルネットワークの概念図である。畳み込みニューラルネットワークは、各層を畳み込みフィルタで層を構成し、さらにその層を積み重ねたものであり、入力画像に対してフィルタ処理を繰り返しながら特徴を抽出し、抽出された特徴を処理することで所望の出力を得る。学習段階では、画像と期待される出力結果とのペアが入力され、期待される出力が得られるように各フィルタの係数が自動的に調整される。

これまで、画像を得るにはカメラを用いたり、又はコンピュータグラフィックスによって描画したりするなどの手段が取られてきたが、ニューラルネットワークの層を多数積み重ねたネットワークであるディープラーニングの発展によって、多層の畳み込みニューラルネットワークによる写実的な画像合成が可能になってきている。こうした画像合成技術によって、例えば機械学習に用いる学習データの量を増やして性能を向上させることなどが可能になる。

2.2 新しい画像合成ネットワークGAN

写実的な画像合成が可能な技術として、

GANが知られている。この技術では、画像合成AIであるGeneratorと真贋判定AIであるDiscriminatorという二つのニューラルネットワークを用いた学習手法を特徴としている。GANの概念図を図2に示す。ここで、Generatorは画像を合成するニューラルネットワークである。一方、Discriminatorは画像を入力として受け取り、その画像が本物の画像なのか、又はGeneratorが合成した偽の画像なのかを判別する。一般に、両者はどちらも畳み込みニューラルネットワークで構成される。

GANの学習では、Generatorは、例えば参考文献(1)に示すような乱数ベクトルや図2に示すようなスケッチ等の出力に対する補助情報を入力して、Discriminatorが本物なのか偽者なのかを判別できないような画像を合成するように学習し、Discriminatorは、Generatorが合成した画像と実際の画像を格納したデータベースからサンプリングされた画像を受け取り、受け取った画像が合成画像なのか、実画像なのかを識別するように学習する。このように両者が相互に敵対しながら学習することで、最終的にGeneratorはDiscriminatorが真贋を判定できないような写実的な画像を合成できるようになる。

しかし、一般的にGANでは、特にGeneratorに大規模な多層ニューラルネットワークが用いられることが多く、

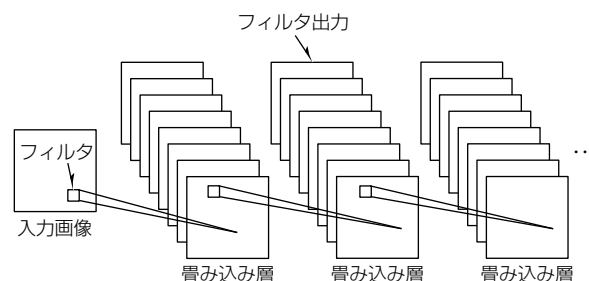


図1. 畳み込みニューラルネットワーク

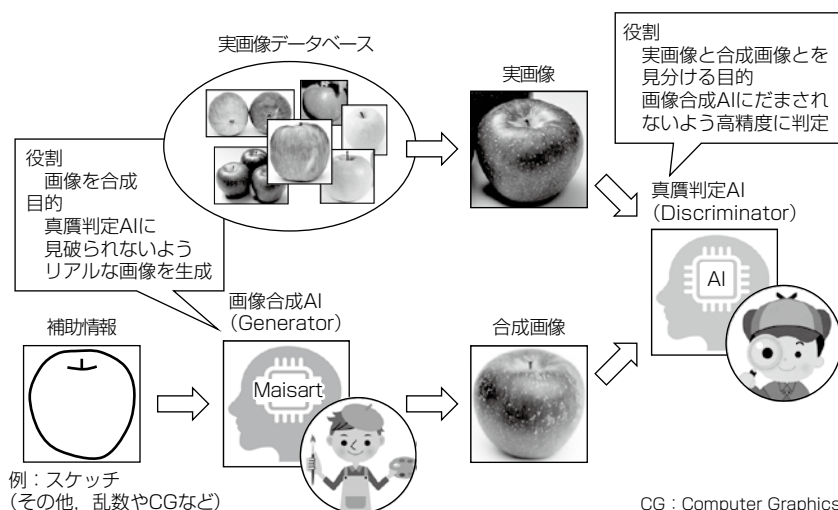


図2. GANの概念図

CG : Computer Graphics

演算量が大きいことが課題になっていた。GANの学習完了後は、Discriminatorは不要になりGeneratorだけが画像生成のために利用されるが、Generatorの演算量が多い場合、例えば高速な画像合成が必要とされる応用や、組み込み機器などの計算資源が限られた環境での動作が必要な応用には適用できない。

本稿では、上記課題を解決するために、GANで用いられるGeneratorを小規模化することで演算量やメモリ量を削減し、高速かつ省メモリで画像合成を可能にする方式を提案する。

3. GANのコンパクト化

3.1 開発方式

開発方式では次の手順を反復することによってGeneratorの小規模化を実現する。

- (1) GANによるGeneratorの学習
- (2) 重要度に基づく重みパラメータの削減

図3に開発方式の処理手順のフローチャートを示す。まずGANの学習を行う。次に、Generatorとして用いているニューラルネットワークのサイズを小さくし、Generatorだけ再学習する。ここで、あらかじめGeneratorのサイズに対して目標値を設定しておき、Generatorのサイズが目標値を下回るならば処理を終了する。一方、Generatorのサイズが目標値よりも大きいならば、再びGeneratorのサイズを小さくして目標値を達成するまで同様の処理を繰り返す。

3.2 GANによるGeneratorの学習

この節ではGeneratorの学習方法について述べる。ここで、GeneratorをG、DiscriminatorをDと表し、GANの学習での目的関数は式(1)で表現される。

$$\min_G \max_D V(D, G) \dots\dots\dots (1)$$

$V(D, G)$ は、実画像は実画像であると、生成画像は生成画像であるとDiscriminatorが正答する期待値(正答：1、誤答：0)を示しており、Generatorは $V(D, G)$ が小さくなるように、Discriminatorは $V(D, G)$ が大きくなるように相反して学習を行うことを表している。図3に示すGANの学習が完了した後は、GANでのGeneratorとDiscriminatorのうち、Generatorとして用いられるニューラルネットワークの規模を小さくして、画像合成時のメモリ使用量や演算時間を削減することを目的とする。

3.3 重要度に基づく重みパラメータの削減

Generatorを小規模化する手法について述べる。一般に

Generatorは、畳み込みニューラルネットワークによって構成される。畳み込みニューラルネットワークは、畳み込みフィルタを多層に重ねたものであり、学習を通して各フィルタの係数が最適化される。各畳み込み層には図1に示すように複数のフィルタが存在し、このフィルタの数が多ければ多いほどニューラルネットワークの規模が大きくなり大量のメモリと演算量が必要になる。そこで、開発方式では、各畳み込み層でのフィルタの数を削減するため、図4に示すように各フィルタに対してスケーリングファクタを導入し、各フィルタからの出力に対してこのスケーリングファクタを乗ずる。このスケーリングファクタは、各フィルタの重要度を示すパラメータとなる。開発方式では、このスケーリングファクタをGANの学習時に自動的に最適化するため、GANの目的関数に式(2)のように追加する。

$$\min_G \max_D V(D, G) + \lambda \sum F(\alpha) \dots\dots\dots (2)$$

ここで、Generatorのコンパクト化を実現するために、関数Fはスケーリングファクタの疎性を促すL1ノルムを用いる。 λ は目的関数の第2項の影響度を調整する変数である。GANの学習完了後、重要度の低いフィルタについてはスケーリングファクタが小さくなるため、あらかじめ設定したしきい値よりも小さなスケーリングファクタ

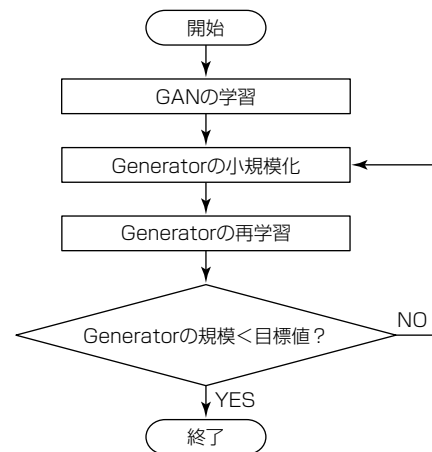


図3. 開発方式の処理手順

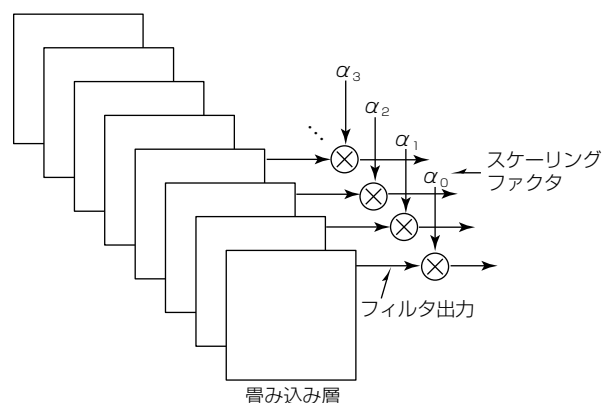


図4. スケーリングファクタの導入

タを持つフィルタを削除することで、性能を落とさずにGeneratorの規模を小さくすることが可能になる。このとき、フィルタの削除によって上記目的関数の第1項の値が増加しない場合、フィルタを削除してもDiscriminatorに対して見破られやすくならず削除前と同等の画像合成品質を持つことを示している。さらに、開発方式では、小規模化したGeneratorを再学習させることで小規模化による性能低下を最小限に抑える。ここで、最初のGANの学習でDiscriminatorは十分な性能を獲得しているため、Discriminatorの学習は行わず、Generatorだけ再学習させることで性能向上を図る。

一般に、GeneratorとDiscriminatorの性能を拮抗(きっこう)させながら学習を行うことは困難であることが知られている。開発方式で、GANの学習が完了した段階でGeneratorの規模を小さくしすぎると、Discriminatorの性能が大幅にGeneratorを上回り、続くGeneratorの再学習が不安定になる。そこで開発方式では、Generatorを段階的に小規模化していくことで学習を安定化させる。つまり、フィルタ数削減のためにスケールングファクタに設定するしきい値を小さくしておき、一回の小規模化で多数のフィルタが削除されることを避ける。しかしこの場合は小規模化の効果も小さくなるため、Generatorのサイズが目標値を下回るまで小規模化と再学習を繰り返すようにすることで、Generatorの十分な小規模化を実現する。

3.4 性能評価

開発方式の性能を評価するため、路上のシミュレート画像を用いて、セグメンテーション画像からシミュレート画像を合成するタスクを評価した。ここで、セグメンテーション画像とは物体ごとに色分けして塗りつぶされた画像のことを示す。このタスクでは、写実的な画像を人工的に合成したい場合に、セグメンテーション画像のような、目的の画像に対応する補助情報を用意するだけで実現できるようにすることが目的となる。このようなタスクは、例えば画像認識向けAIのための学習用画像を人工的に自動合成するといった用途に応用可能と考える。

この実験では、従来のGANによって学習したモデルと、開発方式によって学習したモデルに対して、表1に示すノートパソコンの計算機環境で評価した。その結果、従来方式では、Generatorによる画像合成に画像当たり約3.40秒を要するところ、開発方式では画像当たり約0.34秒で実行

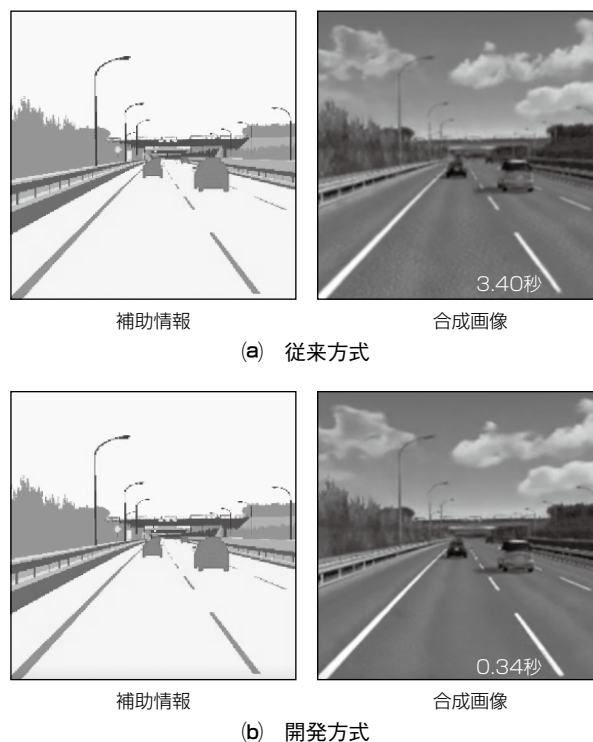


図5. 画像合成の性能評価結果の例

可能であることを確認した。図5に両方式の出力結果の例を示す。図から、開発方式によって、出力画像の画質を劣化させることなく画像合成の処理時間を10分の1にすることを実現していることが分かる。また、この実験でのGeneratorのパラメータ数も2,900万を超える従来方式に対して開発方式では10分の1以下の約250万となることを確認した。このことから画像合成処理時にGeneratorで用いるメモリ量も削減できることが分かる。

4. むすび

最新の画像合成技術であるGANでの、画像合成AIであるGeneratorを性能劣化なく高速処理・省メモリを実現する方式について提案した。

開発方式では、Generatorを構成するニューラルネットワークの各層の重要度を評価し、重要な層だけを残すことで画像生成品質を保ったままコンパクトなニューラルネットワークを合成することを実現した。その結果、画像合成処理の演算量、メモリ量を従来に比べて10分の1に削減することを達成した。

参考文献

- (1) Goodfellow, I.J., et al.: Generative adversarial nets, Proc. of The Twenty-eighth Conference on Neural Information Processing Systems, 2672~2680 (2014)
- (2) Radford, A., et al.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, Proc. of The International Conference on Learning Representations (2016)

表1. 計算機環境

項目	仕様
CPU	Intel Core i7-7500U
実行時の並列数	1
メモリ	16GB