

機械学習を用いた模擬マスクパターン生成手法の一検討 A Study on Pseudo Mask Pattern Generation Using Machine Learning

黒瀬 皓平[†] 花田 隆二[†] 稲木 雅人[†] 永山 忍[†]
Kohei Kurose Ryuji Hanada Masato Inagi Shinobu Nagayama

1. はじめに

今日、LSI は様々な機器に使われており、日常生活に欠かせないものとなっている。LSI の設計は多くの工程からなり、それぞれが多数の要素を扱う複雑な工程であるため、各工程において設計者を支援するための様々な自動設計手法が研究開発されている [1]。それらの手法の研究開発のためには手法の性能評価が不可欠である。

LSI 設計の最終段階では、LSI 中に形成する電子回路の各層の型であるマスクパターンを作成する。このマスクパターンに対して、製造可能性を向上させるための補正を行う工程や問題点を検出する工程があり、そのための手法が研究されている。しかし LSI を設計する企業にとってマスクパターンは企業秘密であり、原則公開しない。そのため、外部の研究者は各種手法の性能評価等が困難な場合があり、一部で公開されているマスクパターン [2] や自作した回路のマスクパターンでしか評価できない現状にある。よって、手法の研究開発のため、より多くの実回路のマスクパターンが必要とされている。多くのマスクパターンがあれば、機械学習に基づく設計支援手法の学習データとしての利用も期待できる。

ところで近年、ニューラルネットワーク (NN) [3] に基づく生成 AI が注目されている。特に画像生成 AI [4, 5] は自然画像に似た画像や画家の画風を真似た画像など様々な画像が生成でき、広く実用が始まっている。画像生成 AI のための基本的な NN として変分オートエンコーダ (VAE) と敵対的生成ネットワーク (GAN) が知られている。

VAE はオートエンコーダ (AE) と呼ばれる NN を拡張した NN である。AE は入力画像に対してエンコーダと呼ばれる NN で重要な特徴量だけを抽出した後、デコーダと呼ばれる NN で特徴量から再度元の画像を生成する。入力画像と出力画像の誤差を最小化するように学習を行うことにより、特徴量を抽出することができる。特徴量を指定して画像を生成できるようにしたものが VAE である。

GAN は生成器と識別器と呼ばれる二つの NN から構成される画像生成を行うモデルである。生成器は画像の生成を、識別器は実画像と偽画像 (生成画像) の判定を行う NN である。生成器はランダム変数から画像を生成し識別器を騙せるように学習する。識別器は実画像と生成画像を見分けられるよう、両者の違いを学習する。この二つの NN が敵対的に学習を進めることにより実画像と見分けの付き難い画像を生成することが可能となる。

そこで本研究では、生成モデルとしてこの VAE と GAN を用いて実回路のマスクパターンに近い疑似マスクパターン画像の生成を試みた。

まず、VAE に基づくマスクパターン画像の生成を行うため、マスクパターン画像に適した NN のパラメータを探索したところ、マスクパターンに類似した画像を生成する

ことができた。しかしながら、マスクパターンが二値画像であるのに対して生成画像は不鮮明な部分が多くなってしまった。

次に、GAN に基づくマスクパターン画像の生成を行うため、マスクパターン画像の生成に適したパラメータを探索した。このとき、生成された画像の多様性が限られたため、評価関数 (損失関数) に多様性を評価する項を加えた。その結果、前述の VAE と比較して多様で鮮明かつマスクパターンに類似していると思われる画像を生成することができた。

最後に、VAE と GAN による疑似マスクパターン画像の生成について、できるだけ条件を揃え、定量的な評価を試みた。ここでは本物のマスクパターン画像と生成画像との類似性を評価するため、GAN の識別器を用いて本物の画像と生成画像の判定を行った。その結果、本研究におけるパラメータの下では、GAN による生成画像の方が本物の画像に近いことを示す結果を得た。

本論文の構成は次の通りである。第 2 章では準備として LSI 設計とマスクパターン、VAE、GAN について説明する。第 3 章では、VAE によるマスクパターン画像生成、第 4 章では、GAN によるマスクパターン画像生成について説明する。第 5 章で両手法の定量的比較を試み、第 6 章で本論文のまとめとする。

2. 準備

2.1 LSI とマスクパターン

LSI は、多数のトランジスタと配線からなる電子回路を、一つの半導体チップに組み込んだ集積回路である [1]。LSI の設計工程は上流設計と下流設計の 2 つに大きく分けられる。上流設計において、まずシステム設計で要求される製品の動作や特性を実現できるように仕様をまとめ、機能や性能を検証してより詳細に設計する。次の論理設計では、システム設計で決められた仕様を基に論理回路を設計する。次に下流設計では物理的な設計を行う。まず論理回路の素子の配置設計と素子間の配線設計を行い、製造条件から決められる設計ルール通りに配置されているかを確認する。デザインルールチェック (DRC) を行った後、素子の位置や配線などの情報を LSI の層ごとにマスクパターンに変換する。その後マスクパターンに対して、製造性を向上する光学近接効果補正 (OPC) や、製造性を確認するリソグラフィシミュレーションが行われ、問題が発見されれば前の工程に戻って設計を修正する。

2.2 変分オートエンコーダ (VAE)

本節では、画像生成 AI として用いられる変分オートエンコーダと、その元となったオートエンコーダについて説明する。

2.2.1 AE と VAE の構造

オートエンコーダ (Autoencoder, AE) [4, 5] とはニューラルネットワーク (NN) の一つで、入力されたデータを一度

[†] 広島市立大学 Hiroshima City University

圧縮し重要な特徴量だけを残した後、元の次元に復元処理をするアルゴリズムである。AE は生成モデルとしても用いることができる。

AE の構造を図 1 に示す。円は NN のノード、矢印はエッジを表す。ノードに入力された数値は各エッジにおいて重み付けられた後、次層のノードに入力される。各ノードには複数のエッジから入力があり、それらの和がそのノードへの最終的な入力値となる。AE の入力側にはエンコーダ、出力側にはデコーダと呼ばれる NN がある。エンコーダとデコーダの間は少数の特徴量を表すノードで接続されている。AE は、AE への入力と AE からの出力の値が同じになるように学習する。これによりデータの中から復元のために必要となる重要な特徴量だけが抽出できる。

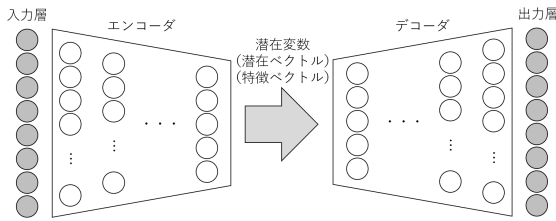


図 1 AE の構造

変分オートエンコーダ (Variational Auto Encoder, VAE) [4, 5] は AE の一種であり、生成モデルとして広く使われている。通常の AE との違いは、入力データを圧縮して得られる潜在ベクトル (特徴ベクトル, 潜在変数) の各次元 (特徴量) を正規分布に従う確率変数として表す点である。一般的な AE では特徴量の分布が規定されていないため、データの特徴量で表したとき、特徴量の分布が偏っている可能性があり、ランダムに特徴量を指定したときの生成画像が安定しないという欠点がある。しかし VAE では、特徴量が正規分布に従うという制約を設けることでデータの特徴量の分布に連続性が生じ、似た特徴ベクトルからは似たデータが安定して生成されるようになる。

VAE の構造を図 2 に示す。エンコーダの出力から平均 μ と標準偏差 σ が計算される。それらで表される正規分布に従うランダムサンプリングによりデコーダに入力する潜在ベクトルを決める。

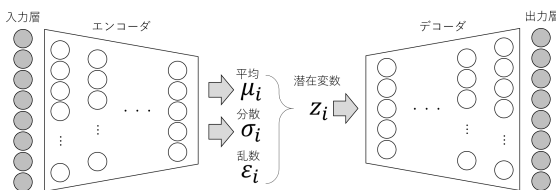


図 2 VAE の構造

2.2.2 KL 情報量

AE の損失関数は入力と出力の差である再構成誤差である。しかし、VAE の損失関数は再構成誤差に加えて潜在ベクトルの各次元 (特徴量) が真の正規分布 (平均 0, 標準偏差 1) に従うよう、正則化の誤差も考慮する必要がある。このとき考慮する正則化の誤差のことを、KL 情報量 [5] と呼ぶ。VAE の損失関数 J は再構成誤差 J_{REC} と、KL 情報量 J_{REG} の合計となり、 $J = J_{REC} + J_{REG}$ と表すことができる。また、KL 情報量は潜在次元数 n , そして潜在ベ

クトルの i 番目要素の分布の平均 μ_i , 標準偏差 σ_i を使用し、次の式 1 で表される。

$$J_{REG} = -\frac{1}{2} \sum_{i=1}^n (1 + \log \sigma_i^2 - \mu_i^2 - \sigma_i^2) \quad (1)$$

KL 情報量は $\sigma = 1$ かつ $\mu = 0$ のときに最小化するため、潜在ベクトルの各次元について平均と分散が真の正規分布から外れると損失が発生するようになっている。

2.3 敵対的生成ネットワーク (GAN)

本節では敵対的生成ネットワーク (Generative Adversarial Network, GAN) [4, 5] について説明する。

GAN は生成器と識別器と呼ばれる二つのニューラルネットワーク (NN) で構成されており、画像を生成する学習モデルである。GAN は実画像を学習データとして用い、実画像に似た画像を生成する。一般的には画像生成モデルの中でも鮮明でリアルな画像を生成できると言われている。図 3 に GAN の構造を示す。

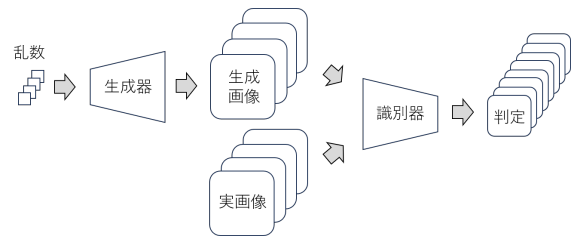


図 3 GAN の構造

2.3.1 識別器

識別器は GAN の構成要素の一つで、NN である。識別器の入力は画像 (ベクトル) で出力は 0 から 1 の確率値である。GAN での識別器の役割は実画像と偽画像 (生成画像) を見分けることである。識別器は入力画像に対し、実画像だと判定したときは 1 (に近い値) を、偽画像だと判定したときは 0 (に近い値) を出力する。識別器は教師あり学習を行いながら判定精度を上げるよう学習する。

2.3.2 生成器

生成器は GAN の構成要素の一つで識別器と同じく NN である。生成器の入力はランダムノイズで出力は画像である。ランダムノイズは正規分布からサンプリングされ、出力は実画像に似た画像となる。出力画像を実画像に似せるために、生成器は識別器の判定値と正解を用い、識別器が実画像だと誤認するような画像を生成できるよう学習を進める。

2.3.3 GAN の学習方法

本項では GAN 全体の構造について図 3 を用いて説明する。ランダムノイズと学習データの数を共に 4 として説明する。はじめに 4 つのランダムノイズを生成器に与え、画像 (図 3 内では生成画像) を 4 枚生成する。次に識別器に生成画像及び実画像をそれぞれ 4 枚ずつ入力し、8 つの判定を得る。識別器は判定結果と正解値を元に生成器の生成画像を偽画像だと見抜けるよう学習を行う。その後、生成器は識別器に生成画像を実画像と誤認させるよう学習する。以上を繰り返してそれぞれが敵対し合いながら学習し、最終的に生成器が実画像のような画像を生成できるようになる。

2.3.4 関連研究

マスクパターンの特徴量を抽出した研究および類似マスクパターン生成の研究として [6] が挙げられる。[6] では主成分分析により既存の公開マスクパターンデータ [2] を分析し、データの偏りを指摘している。またデザインルールを満たすランダムパターンの生成により [2] のデータを拡張したマスクパターンデータを公開している。AE および GAN に基づく回路パターンライブラリ生成の研究として [7] が挙げられる。

3. VAE: マスクパターン画像の学習と生成

本章では、VAE によるマスクパターン画像の学習と疑似マスクパターン画像の生成について述べる。

3.1 マスクパターン画像の学習

本節では、VAE にマスクパターン画像を学習させる。VAE を構成する様々なパラメータを変化させて評価を行い、より元の画像に近い復元画像が得られるよう調整する。

3.1.1 学習データ

本研究で用いるマスクパターン画像として、ホットスポット・非ホットスポット画像で構成されている ICCAD2012 benchmark dataset [2] を使用した。パターンの例を図 4 および 5 に示す。このデータは、LSI のマスクパターンの中からホットスポットと呼ばれる製造時に高い確率で欠陥を生じる可能性があるパターンを発見するための手法の評価のために提供されている。

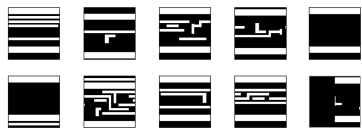


図 4 ICCAD2012: array_benchmark1 の例



図 5 ICCAD2012: array_benchmark3 の例

3.1.2 実験内容

VAE を構成するパラメータのうち、潜在ベクトルの次元数、エポック数、学習データ数、中間層の数を変更して評価を行った。エポック数とは NN の学習過程において、学習データセット全体を処理する回数のことである。

初期パラメータとして、VAE のエンコーダの構成は、入力層から順に 784(28x28)、256(16x16)、64(8x8) ノード (ピクセル) とした。層間は全結合とした。デコーダの構成はエンコーダと対称である。学習データには 521 枚の画像を持つデータセット array_benchmark1 を使用した。ただし、左右反転した画像をデータセットに追加することで、データ数を 2 倍にかさまししている。

まず、エポック数を 400 とし、潜在次元数を 8, 16, 32 とした場合について評価した。次に、エポック数を 800 と

し、潜在次元数を 16 とした場合について評価した。

さらに 5349 枚の画像を持つデータセット array_benchmark3 に対し VAE を適用したところ、画像の復元度が不十分であることが明らかであったため、エンコーダの入力層 (入力画像の解像度) および中間層の構成を変更するパラメータ調整を行った。

VAE は Python3.8 および Keras ライブラリを用いて実装した。

3.1.3 潜在次元数とエポック数の調整

エポック数を 400 とし、潜在次元数を 8, 16, 32 とした場合について、テスト画像の復元画像の評価を行った。結果を図 6 に示す。n は潜在次元数を表す。

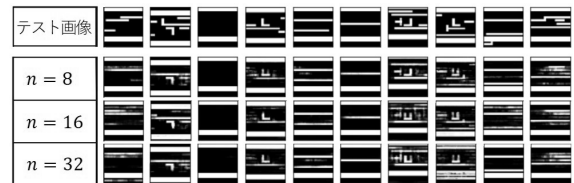


図 6 VAE: テスト画像と次元数ごとの復元画像

この結果から、明確な差があるとはまて言えないが、潜在次元数 $n=8$ のときに配線の折れ曲り部に欠損が見られることがあること、 $n=32$ のときに太い水平な配線が復元できていない場合があることから、間をとって潜在次元数を 16 に決定した。

次はエポック数を 400 から 800 に増加させ、どの程度精度向上が図れるかを分析した。図 7 はそのときの復元画像である (上段がテスト画像、下段が復元画像)。目視では明確な差は分らなかったが、損失値はやや改善した。エポック数に対する損失値の変化を図 8 に示す。

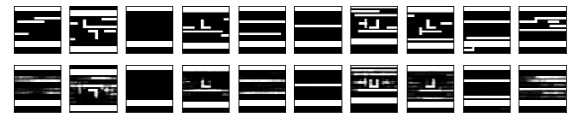


図 7 VAE: テスト画像と復元画像 (エポック数 800)

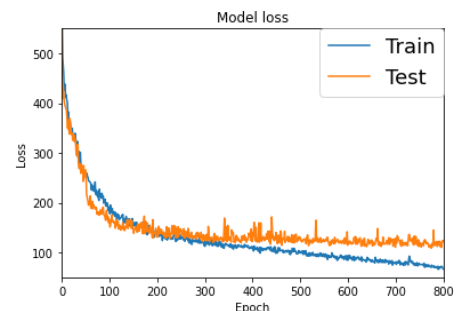


図 8 VAE: エポック数に対する損失値

3.1.4 NN アーキテクチャの検討

次に 5349 枚の画像を持つデータセット array_benchmark3 に対し VAE を適用したところ、復元画像が図 9 のようになり、画像の復元度が不十分であることが明らかであったため、エンコーダの入力層 (入力画像の解像度) および中間層の構成を変更するパラメータ調整を行った。中間層の

数, サイズを調整し, 精度の良い復元画像に近づける. 潜在次元数は 16, エポック数は 500 とし, 中間層の構成は次の 2 つの条件下で評価を行った. 層間は全結合とした. 数字は各層のノード数 (ピクセル数) である.

- (1) 3136(入力層)-784-196-64-16(潜在次元)
- (2) 3136(入力層)-1936-784-196-64-16(潜在次元)

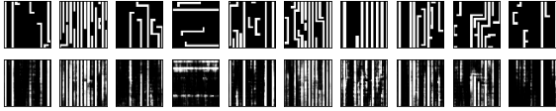


図 9 VAE: テスト画像と復元画像 (array3)

array_benchmark3 は array_benchmark1 より複雑な構造であるため, 入力画像のサイズが 28×28 のままではうまく学習が出来ないと考えた. そのため入力サイズを 56×56 (= 3136) に変更し, 高精細な画像に変更した.

テスト画像の復元結果をの図 10 に示す.



図 10 VAE: テスト画像と復元画像 (層構成 (1)(2))

図より, 層構成 (1) の時点でも復元度が向上している事が分かる. 更に 44×44 (= 1936) の中間層を追加した層構成 (2) では, テスト画像により近い画像を復元する事が出来た.

3.2 マスクパターン画像の生成

本研究では, 既存のマスクパターン画像を学習して新たな画像を作りたい. そこで潜在ベクトル空間上の 20×20 の格子点から画像を生成した. データセットは array_benchmark3 を使用し, エポック数は 400 とした. エンコーダは入力層から順に, (56×56)-(44×44)-(28×28)-(16×16)-(8×8)-2(潜在次元数)とした. デコーダはエンコーダと対称である. 図 11 に生成結果を示す.

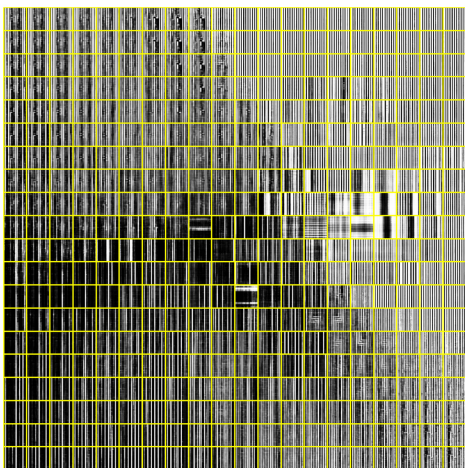


図 11 VAE: ベクトル分布からの生成画像

図から, 連続的に画像が変化していることが分かる. 前節で述べたように画像サイズおよび中間層を増加させるとより本物のマスクパターンに似た画像になることが期待される.

4. GAN: マスクパターン画像の学習と生成

本章では, VAE によるマスクパターン画像の学習と疑似マスクパターン画像の生成について説明する.

4.1 実験環境

本研究では各手法は Python3.8 で記述し, GAN の機械学習ライブラリとして TensorFlow, PyTorch, Scikit-learn, 数値計算ライブラリとして NumPy, グラフ描画ライブラリとして matplotlib を使用した.

4.2 NN アーキテクチャ

GAN の生成器には転置畳み込み構造を, 識別器には畳み込み構造を使用した. 生成器の層構成は, 入力層から順に (1×1,10ch)-(4×4,512ch)-(8×8,256ch)-(16×16,128ch)-(32×32,64ch)-(64×64,1ch)とした. 識別器の層構成は, 入力層から順に (64×64,1ch)-(32×32,64ch)-(16×16,128ch)-(8×8,256ch)-(4×4,512ch)-(1×1,1ch)とした. カーネルサイズは 4×4, スライドは 2 とした. 活性化関数には, 生成器では ReLU 関数を (出力層: Tanh 関数), 識別器では Leaky ReLU 関数 (出力層: Sigmoid 関数) を使用した.

4.3 学習データ

VAE の場合と同じ学習データを使用した.

4.4 GAN への多様性の導入

array_benchmark1 を学習データとして GAN による疑似マスクパターン生成を試行したところ, マスクパターンに似た画像は生成できたが, すべての生成画像がほぼ同じであり, 多様性がない状態であった. そこで損失関数において多様性を考慮することで生成画像に多様性を持たせることを考えた.

本研究では多様性を, 画像を表す二次元行列の各要素同士の差の絶対値の総和とした. この多様性の評価関数を式 2 として定義する.

$$\text{Diversity}(I) = \frac{1}{N-1} \sum_{i=0}^{N-2} |I_i - I_{i+1}| \quad (2)$$

式 2 において I は生成画像の列, I_i は i 番目の生成画像の二次元行列, N は生成画像の枚数を表す. また, $|A|$ は画像行列 A の各要素の絶対値の総和を表すものとする. この式 2 では隣接する添字の画像間において多様性を考慮している. 損失関数内の多様性評価関数の重みを α とする.

これにより多様性は大きく改善された. 予備実験により $\alpha = \frac{1}{12}$ と設定した.

損失関数の変更による画像生成の生成画像の多様化についての研究として [8] が挙げられる.

4.5 パラメータ調整

本節では GAN によるマスクパターン画像の学習のためのパラメータ調整について説明する.

各項で順にバッチサイズの調整, 学習率の調整, エポックの調整について説明し, 最後に GAN の生成結果を示す. 学習データは 545 枚のマスクパターン画像から構成される array_benchmark1 を用いた. 各生成例では 64 枚の画像を生

成した。

4.5.1 バッチサイズの調整

本項ではバッチサイズの調整について述べる。機械学習では大量のデータを学習する際、データをいくつかのグループに分割して学習を行う方法が存在する。このグループの事をバッチ、グループ数の事をバッチ数、グループのサイズの事をバッチサイズと呼ぶ。バッチサイズが大きいと一度に考慮する学習データ数が増えるため、精度が向上する一方で、消費メモリや CPU への負荷が大きくなる場合がある。バッチサイズが小さいと一度に考慮する学習データ数が減るため、消費メモリや CPU 負荷が少なくなる一方で、精度は低下する傾向にある。

バッチサイズを 1, 16, 32, 64, 96, 128, 160, 192, 256, 545 として画像を生成した結果、バッチサイズが大きいほど多様なパターンが生成できていることを確認した。バッチサイズが小さいとバッチ数が増加し、前のグループでの学習結果を後のグループでの学習が上書きしてしまった可能性がある。一方で、生成画像の鮮明さはバッチサイズが中間の値のときの方が良かった。しかし画像の鮮明さはエポック数の増加に伴って改善される傾向であることが判ったため、多様性が最も良かった 545 をバッチサイズとして選択した。このときの生成画像を図 12 として示す。図 12 の生成画像の鮮明さについての問題点として、垂直方向の線が揺らいでいる場合があることが挙げられる。

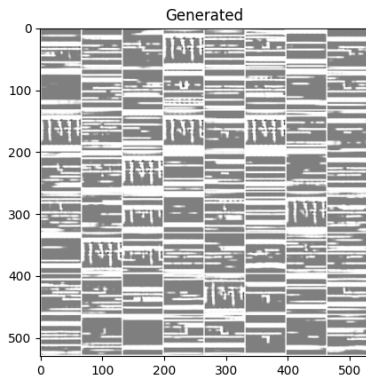


図 12 GAN: バッチサイズ 545 のときの生成結果

4.5.2 学習率の調整

本項では学習率の調整について述べる。学習率とは学習の際に NN の重みパラメータを一度にどの程度変化させるかを表すパラメータである。学習率を小さくすると学習の収束に時間がかかるが発散のリスクが減り、より安定して学習が収束できる。学習率を大きくすると学習の収束は早まるが発散のリスクが高まり、学習が収束しないことがある。

学習率を 0.0001, 0.0002, 0.0006, 0.002, 0.006, 0.01, 0.02, 0.1 として生成画像を評価した。損失関数での多様性評価関数の重み α は $\frac{1}{12}$ 、バッチサイズは 545、エポック数は 1000 とした。

その結果から、生成画像が程よく複雑であり多様性も保った学習率 0.006 を採用した。このときの生成画像を図 13 として示す。

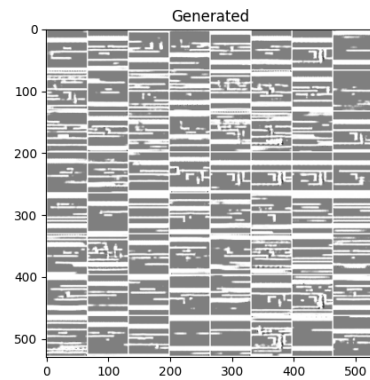


図 13 GAN: 学習率 0.006 のときの生成結果

4.5.3 エポック数の調整

本項ではエポック数の調整について述べる。損失関数での多様性評価関数の重み α は $\frac{1}{12}$ 、バッチサイズは 545、学習率は 0.006 として、良い生成画像が得られるエポック数を調べた。

その結果、エポック数が増加するにつれて生成画像は良くなり、エポック数 5800 を超えたところで学習が発散してしまうことが確認できたため、エポック数として 5800 を採用した。発散の原因の特定は今後の課題である。

4.6 生成結果

本項ではそれぞれのパラメータ調整後の生成結果を示す。採用されたパラメータの値はそれぞれ α は $\frac{1}{12}$ 、バッチサイズは 545 枚 (学習データと同数)、学習率は 0.006、エポック数は 5800 である。その生成結果 64 枚 (図 14) とその拡大図 7 枚 (図 15) を示す。

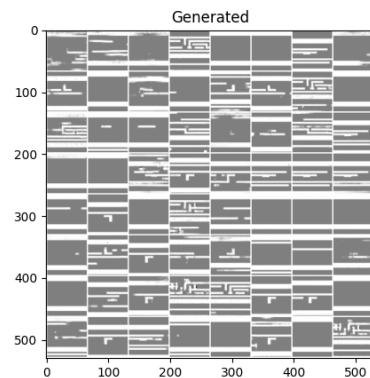


図 14 GAN: パラメータ調整後の生成結果



図 15 GAN: パラメータ調整後の生成結果 (拡大)

パラメータ調整の結果、本物のマスクパターン画像に近い疑似マスクパターン画像を生成することができた。

5. VAE と GAN による生成画像の比較

本章では、VAE と GAN による生成画像を比較し、疑似マスクパターン生成においてどちらの生成画像がより良いかを検討する。

5.1 VAE と GAN の NN アーキテクチャ

本章で用いた VAE および GAN の NN の構造について説明する。第 3 章で用いた VAE は層間が全結合となっている層構成であった。一方、第 4 章の GAN では生成器に転置畳み込み構造、識別器に畳み込み構造を採用している。よって公平な比較のため、本章では VAE と GAN の両方に転置畳み込み、畳み込み構造を用いる。

GAN の NN の構造は第 4 章と同様とした。VAE の構造も GAN と同様とした(潜在変数の層を除く)。VAE の活性化関数はエンコーダに ReLU 関数、デコーダに ReLU 関数と Sigmoid 関数を使用した。(エンコーダに対して Leaky ReLU 関数はより良い結果を得られなかった。)

VAE の潜在変数の次元数を調整した結果、復元画像については潜在次元数が 32 のときが最も良かったが、生成画像についてはうまく生成できず、潜在次元数が 2 のときが最も良かった。原因の特定は今後の課題である。潜在次元数が 2 のときの生成画像の拡大図を図 16 に示す。



図 16 VAE: 潜在次元数 2 の場合の生成画像 (拡大)

エポック数は検討の結果 3000 を用いた。

5.2 VAE の生成結果

図 17 に、GAN と同様の層構成を採用した場合の VAE による疑似マスクパターン画像の生成結果を示す。

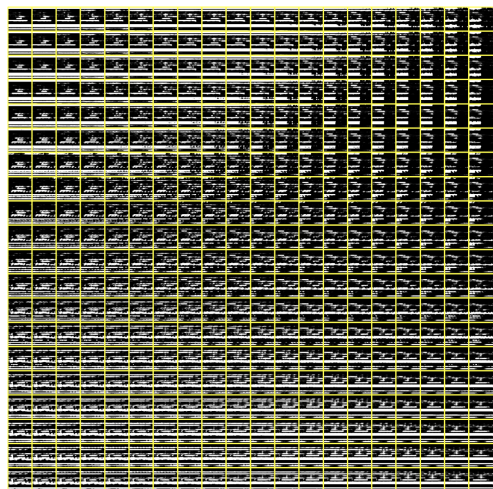


図 17 VAE: 潜在次元数 2 の場合の生成画像

5.3 GAN と VAE の定量的比較

本節では GAN と VAE の生成画像の比較を定量的に行うことを試みる。どちらの生成画像が本物のマスクパターン画像により近いかを比較するために、GAN の識別器を用いた定量評価を行う。

5.3.1 識別器の設定

生成画像の定量評価に用いる識別器の作成方法を説明する。まず GAN に `array_benchmark1` を学習させつつ、一定のエポックごとに識別器のエッジの重みを保存する。このときのパラメータの値はそれぞれ α を $\frac{1}{15}$ 、バッチサイズを 545 (学習データと同数)、生成器の学習率を 0.006、識別器の学習率を 0.0002 とした。その後、保存したエッジの重みを識別器に読み込み、マスクパターン画像と生成画像を判定する。エポック数の最大値である 5800 のときの識別器は判定精度が高すぎ、VAE と GAN の比較ができなかった。そこで本物の画像の判定結果の平均値が 0.9 となった、エポック数が 2000 のときの識別器を、VAE と GAN の定量評価のために採用した。

5.3.2 比較結果と考察

本項では識別器による GAN と VAE の比較結果を示す。GAN と VAE の生成画像それぞれ 200 枚に対し、識別器を用いて判定を行い、平均値を求めた。

その結果、GAN の生成画像の評価は 0.1749、VAE の生成画像の評価は 2.7234×10^{-9} となった。識別器の出力が 0 に近いほど、偽の画像の可能性が高いという判定を受けたことになる。よって GAN は VAE より本物に近いマスクパターン画像を生成できることがわかった。

ただし、用いた VAE は潜在次元数が 2 以外のときに良い生成画像を得られないという問題があった。これを解決することで VAE もより良い画像を生成できる可能性がある。

6. まとめと今後の課題

本論文では、GAN と VAE による疑似マスクパターン画像の生成を試行し、比較を行った。GAN の識別器を用いてマスクパターンらしさを定量評価した結果、GAN による生成画像の方がより本物に近いとの判定を得た。

今後の課題として、別のベンチマークデータに対する評価実験、グレースケール画像からの矩形からなる二値画像への変換法の検討、期待された結果が得られなかったケースの分析などが挙げられる。

参考文献

- [1] L.-T. Wang, Y.-W. Chang, K.-T. T. Cheng, *Electronic Design Automation: Synthesis, Verification, and Test*, Morgan Kaufmann (2009).
- [2] J.A. Torres, "ICCAD-2012 CAD contest in fuzzy pattern matching for physical verification and benchmark suite," in Proc. IEEE/ACM ICCAD 2012, pp.349-350 (2012).
- [3] 麻生英樹ら, 深層学習, 近代科学社 (2015).
- [4] J. Langr and V. Bok, *GANs in Action: Deep learning with Generative Adversarial Networks*, Manning Publication (2019).
- [5] 毛利ら, GAN ディープラーニング実装ハンドブック 秀和システム (2021).
- [6] G.R. Reddy, K. Madkour and Y. Makris, "Machine Learning-Based Hotspot Detection: Fallacies, Pitfalls and Marching Orders," in Proc. IEEE/ACM ICCAD2019 (2019).
- [7] H. Yang et al., "DeePattern: Layout Pattern Generation With Transforming Convolutional Auto-Encoder," in IEEE Trans. on Semiconductor Manufacturing, vol.35, no.1, pp.67-77 (2022).
- [8] 児玉涼次, 中村剛士, 加納政芳, 山田晃嗣, "GAN における多様な画像生成のための損失関数の提案," 第 32 回人工知能学会全国大会 予稿集, 1Z3-01 (2018).