

GAN 出力画像をフィルタリングするための 複数の CNN を用いた画像評価

相場 築[†] 荒澤 孔明[†] 服部 峻^{††}

^{†,††} 室蘭工業大学 ウェブ知能時空間研究室 〒050-8585 北海道室蘭市水元町 27-1

E-mail: [†]{17024001,18096001}@mmm.muroran-it.ac.jp, ^{††}hattori@csse.muroran-it.ac.jp

あらまし 近年、様々な場面で AI 技術の利用、研究がなされている。有名なものでは、ソフトバンクの「Pepper」や、掃除ロボットの「ルンバ」などが挙げられる。また、画像を分類、生成する研究も盛んに行われている。特に、画像を生成する技術は、絵が描けない人や、絵に時間をかけられない人にとってはとても需要が高い。そこで、本稿では画像生成に特化した AI である「GAN」について着目し、GAN 出力画像の問題点や課題について分析を行う。また、ユーザの求めるような画像が生成されるまで、GAN の出力を全て表示し、ユーザがその都度画像の評価を行う必要がある問題の解決策として、表示優先順位を設ける。さらに、GAN 出力特有の形が歪であるような中途半端な画像を排除するための画像評価について試行し、複数の CNN の画像認識結果を利用して除外対象を選定するなどの独自のフィルタリングを試作して、ユーザが求める画像を効率良くフィルタリングするためのシステム構築を目指す。

キーワード GAN, CNN, 画像フィルタリング, 画像評価, リクエスト解析

Image Evaluation Using Multiple CNNs for Filtering GAN-output Images

Kizuku AIBA[†], Komei ARASAWA[†], and Shun HATTORI^{††}

^{†,††} Web Intelligence Time-Space (WITS) Laboratory, Muroran Institute of Technology
27-1 Mizumoto-cho, Muroran, Hokkaido 050-8585, Japan

E-mail: [†]{17024001,18096001}@mmm.muroran-it.ac.jp, ^{††}hattori@csse.muroran-it.ac.jp

Abstract In recent years, AI technologies have been utilized and researched in various situations. The famous examples have Softbank's "Pepper" and the cleaning robot "Roomba". In addition, a lot of researches on Image Classification and Image Generation are being actively conducted. Especially, the technology to generate images is in great demand for those who cannot draw or who cannot spend time drawing. Therefore, this paper focuses on "GAN" that is an AI specialized in image generation, and analyzes the problems and issues of GAN-output images. Moreover, as a solution of the problem that it is necessary to display all outputs of GAN and evaluate them each time until the images that a user wants are output, this paper proposes a method to assign GAN-output images with "priority" for display. Furthermore, this paper aims to develop a system for efficiently filtering of the images that a user wants by making image evaluation for eliminating half-finished images that have a distorted shape unique to GAN-outputs and proposing unique filtering such as selecting exclusion targets using image recognition results of image recognition of multiple CNNs.

Key words GAN, CNN, Image Filtering, Image Evaluation, Request Analysis

1. ま え が き

昨今、様々な場所で AI の利用、研究がなされている。有名なものでは、ソフトバンクが開発した人間の感情を認識できるロボットである「Pepper」や、掃除ロボットの「ルンバ」、自動車の自動運転などが挙げられる。また、画像を認識、分類、生成する研究や、小説の冒頭部分を生成するシステムな

ど多種多様である。特に、本稿で取り上げる「GAN (Generative Adversarial Network)」[2] は画像を生成することに特化しており、絵が描けない人にとっては、欲しい絵の条件を入力すると機械が絵を出力してくれるため、フリー画像を探す手間や、絵を描いてもらう料金がかからずとても有用である。

GAN とはデータを新たに生成できる生成モデルの一種で、主な構造は 2 つに分かれている。1 つ目は「Generator」であ

り、与えられた画像集合から類似したデータを作る。2つ目は「Discriminator」と呼ばれ、Generatorで作られたデータの真偽を入力データと比較して判別する。これを繰り返すことで、Generatorは本物に近いデータを生成できるようになり、さらに、複数の入力データの部分部分を組み替えて、まったく新しいデータを作ることも可能である。つまり、入力データから特徴を学習し、特徴を含んだデータを新しく生成できる点がGANの特徴である。既存のデータをマイニングするのとは違い、生成するデータの特徴を指定することでユーザーの細かな要求にも対応ができ、有用性が高い。

しかし、GANの生成画像のクオリティはまばらである。生成されたデータは輪郭のはっきりしない画像から、現実に存在しないような画像でも実際にあると思ってしまう様なものまで生成できるため、画像が他コンテンツに利用できるクオリティであるか否か、定量的に評価するのが難しく、現状のGANの画像評価は、学習データの知識を有する者が、実際に目で見て判断することが多い。また、ユーザーの求める画像に細かな指定があったり、要求する画像の量が多い場合は、要求を完璧に満たす画像を得るために大量の出力を必要とする上、要求通りではない生成画像も大量に含まれることから、生成の度にユーザーの目で確認しなければならず、多大な手間が発生する。

そこで、本稿では、GANによって生成された出力画像を『ぼやけ具合』『生成画像からオブジェクト物体が認識できる度合い』『ユーザーリクエストの反映度』の3種類の観点から自動的に評価し、ユーザーが求めるクオリティの高い画像を効率良くフィルタリングする手法について提案する。

2. 提案手法

GANはテキストを入力することで画像を生成できる構造(text-to-image) [1]であると仮定し、学習データは画像とその画像の説明が記述されたものの組の集合とする。このようなGANから得られる大量の生成画像に対する画質評価や、生成対象評価などを自動的に行い、人の評価を介さず選別を行うことで手間を削減し、ユーザーのリクエスト(例えば、「黒い体の鳥」)に合った画像を効率的に収集するための手法について提案する。

まず、提案手法の概要について説明する。概観は図1の通りである。画像とその画像の説明が記述された学習データを用いて構築したGANから、生成画像を任意の数だけ収集し、生成画像に対して複数の処理を通して評価付けを行う。提案手法には大きく分けて4つの処理が存在する。

- (1) 画像のぼやけている度合いを計算する。
- (2) 対象物が認識できる度合いを計算する。
- (3) ユーザーのリクエストの反映度を計算する。
- (4) 全処理1~3の評価値を用いて総合評価を付ける。

処理1~3を行った後、それぞれの処理によって得られる評価値を用いて総合評価を付け、総合評価の順に画像をランキング表示することでユーザーのリクエストに合った画像を優先的に表示し効率化を図る。また、「req-CNN」「web-CNN」について、詳細は後述するが、この2つのCNNは入力されたリクエスト

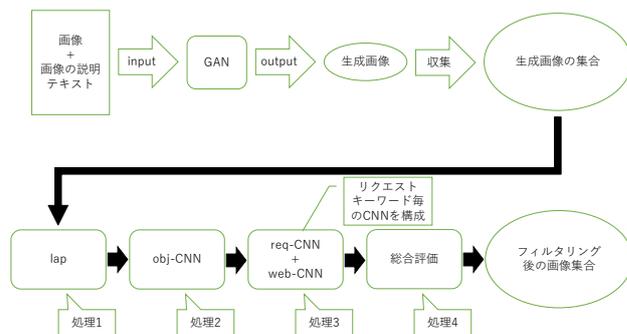


図1 提案手法の概観



図2 lap(img)
= 180.69



図3 lap(img)
= 1959.44



図4 lap(img)
= 327.04

のキーワード(例えば、「blue wings」)毎に用意するものであり、「blue wings」が入力された場合であれば、2つのCNNは「青い羽根の鳥」に該当する画像の評価を高く付けるタスクを担う。図1では、1つだけ書いてあるが、実際にはリクエストのキーワード毎に構築されるため無数に用意する必要がある。

2.1 処理1: ぼやけ具合の評価

処理1では、生成画像を『ぼやけ具合』の観点で評価付けする。GANから生成される画像はしばしば色が複雑に混ざり合ったようなぼやけている画像が出力される。このような画像は特殊な用途を除き、他コンテンツで利用できるようなクオリティではないため、取り除くことができるように指標を用意する。OpenCVライブラリ [3] のラプラシアンフィルタを用いることで、画像のエッジ(境界線部分)を検出し、評価対象の生成画像の分散を求め、分散の大きさに比例させて評価値を上げることによって『ぼやけ具合』を表現する。以降、画像imgを入力した時の本節の評価値をlap(img)と表記する。図2~4のキャプションの数値は、それぞれの画像に処理1を行った場合の評価値の例である。図2は精度が未熟なGANから出力された画像であり、「鳥のような何か」が写っている程度まで判断できるが、はっきりと断言することが難しいような画像である。図3は実際に撮影された画像で、誰が見ても「鳥が写っている」と断言できるような画像であり、「鳥と周辺の情景」も輪郭のはっきりと捉えられる。図4は精度が中程度のGANから出力された画像であるが、図2よりもlap(img)が高く、実際の画像である図3よりも大幅に低い。つまり、評価値lap(img)を用いることで、GANから生成されるぼやけた画像を除外するために活用できると考えられる。

2.2 処理2: オブジェクト評価

処理2では、GANに学習させているオブジェクト(例えば、「鳥」)が認識できる度合いについての観点から生成画像毎に評価

付けを行う。GANに限らず現存するAIの殆どが、実用的な段階までの精度にするために学習データや機能を限定的にする場合が多い。本稿で想定しているGANも生成できる画像の大きなオブジェクトは決まっているものとし、そのオブジェクトが捉えられていない画像の評価を落とすことで、取り除くための指標を用意する。オブジェクトが含まれているかを判断するためにGANと同じ学習データまたは、オブジェクトと同じカテゴリを持つデータセットを学習させたCNN(Convolutional Neural Network)を用意し、生成画像にオブジェクトが捉えられるかを数値的に出すことで評価値を出す。以降、他のCNNと区別するために本節のCNNを「obj-CNN」と呼称し、画像に対するobj-CNNから得られる評価値を $\text{obj-CNN}(\text{img})$ と表記する。obj-CNNに画像を入力すると、画像のオブジェクトが認識できる確率を $0.0 \leq \text{obj-CNN}(\text{img}) \leq 1.0$ で返す。本稿の評価実験では、GANは鳥の種類を複数学習させたものを想定しており、この場合であればobj-CNNに分析させる要素は『鳥らしいもの』が評価軸に該当する。例として挙げると、3.1節の図6、図8を入力したときの評価値は、図6が $\text{obj-CNN}(\text{img}) = 0.45$ となり、図7が $\text{obj-CNN}(\text{img}) = 0.90$ と出力される。オブジェクトが捉えられていないにも関わらず、ユーザの要求が反映されているとは考えにくいため、このような画像を取り除く目的でここで算出される評価値を2.4節での総合評価の数式(3)で利用する。

2.3 処理3: ユーザリクエストの反映度評価

処理3では、ユーザの要求(例えば、「青い羽根の鳥」)が反映されているかの観点から生成画像毎に評価付けを行う。この処理で使用するCNNも構築方法は2.2節と同様にGANに学習させるデータを用いて構築を行う。ユーザの要求はキーワード形式を想定しており、GANの学習に用いている画像の説明データからキーワードを検索し、ユーザ要求のキーワードが説明文中に含まれている画像だけを集めてCNN(以降、req-CNNと表記)を構築する。これにより、評価対象の生成画像にキーワード要素が含まれているかをreq-CNNで分析できるようになる。画像に対するreq-CNNから得られる画像の評価値 $\text{req-CNN}(\text{img})$ を2.4節での総合評価の数式(3)で用いることでユーザの要求に近い画像を優先的に表示する。また、本稿の評価実験ではキーワード要素を分析するreq-CNNについては、インターネットから得られる画像を集めて同じキーワード要素を分析できるCNN(以降、web-CNNと表記)を用意し組み合わせることで、自動的に構築したreq-CNNの精度に信憑性が無い場合でもロバスト性の向上を狙っている。

2.4 処理4: 総合評価の定式化

これまでに挙げた4種類の評価値を基に、本節では総合評価の計算手法を提案する。まず、2.1節の評価値 $\text{lap}(\text{img})$ と2.2節の評価値 $\text{obj-CNN}(\text{img})$ の値を閾値で区切って数式(1)と数式(2)のように2値化することで、評価の低い生成画像の表示を制限する。最適な閾値に関しては、4章で検証する。次に、2.3節のreq-CNNとweb-CNNについては、重みパラメータ $0.0 \leq \alpha \leq 1.0$ で線形結合する。最も良い α の値を求めため、変化させた場合の精度の変化を3章で観察する。総合評

価 $\text{eval}(\text{img})$ の算出方法を数式(3)に記述する。 $\text{lap}^b(\text{img})$ によって、3.1節の図5のような画像を排除し、 $\text{obj-CNN}^b(\text{img})$ によって、3.1節の図6のようなある程度輪郭を持ちつつも形が歪な画像を排除する。その後、req-CNNとweb-CNNによって、ユーザのリクエストに合っている画像の評価値を高く設定し、総合評価の高い順に並び替えて優先表示を行うというコンセプトで作られている。

$$\text{lap}^b(\text{img}) = \begin{cases} 1 & (\text{lap}(\text{img}) \geq 300) \\ 0 & (\text{lap}(\text{img}) < 300) \end{cases} \quad (1)$$

$$\text{obj-CNN}^b(\text{img}) = \begin{cases} 1 & (\text{obj-CNN}(\text{img}) \geq 0.5) \\ 0 & (\text{obj-CNN}(\text{img}) < 0.5) \end{cases} \quad (2)$$

$$\begin{aligned} \text{eval}(\text{img}) &= \text{lap}^b(\text{img}) \cdot \text{obj-CNN}^b(\text{img}) \\ &\cdot \{(1 - \alpha) \text{req-CNN}(\text{img}) + \alpha \text{web-CNN}(\text{img})\} \end{aligned} \quad (3)$$

$$(0.0 \leq \alpha \leq 1.0)$$

3. 評価実験: 線形結合パラメータの依存性

本章では、実際に実験を行う際に用いたデータセットや設定について述べた上で、線形結合のパラメータ α の依存性の実験を行い、総合評価値を用いてランキングし、上位10件を取り出した場合の適合率と上位 k 件の再現率の結果について述べる。

まず、GANは『CUB-200-2011』[4]と呼ばれるカリフォルニア大学提供の、鳥の種類が200種登録されている11788枚の画像データセットと、このデータセットの画像それぞれに説明が書かれているデータ[5]を学習させたものとし、このようなGANから出力される画像をフィルタリングの対象として想定する。次に、処理2のobj-CNNは『鳥らしさ』を分析するために、鳥のカテゴリが含まれているデータセットである『cifar10』[6]というデータセットを用いて、鳥と鳥以外とに分けて学習させた。処理3のreq-CNNは、GANの学習の際に用いる『CUB-200-2011』とその説明データを使って、ユーザリクエストとして入力されたキーワードで説明データを基に検索された画像と残りの『CUB-200-2011』からランダムに同数分取り出した画像とを学習させ、『キーワードの反映度』を分析できるようにする。さらに、キーワードに該当するものをウェブから検索して、手動で集めた画像でweb-CNNを構築した。

3.1 実験用サンプルの概要

実際のGANから得られる生成画像をリアルタイムに実験するとランダム性に影響されるため、本稿ではGANから得られるような画像を集めた実験用サンプルを作成して、これを用いることとする。画像毎の特徴は以下のように決める。

- (1) 精度の低いGANから得られるような画像
- (2) 精度が中程度のGANから得られるような画像
- (3) 精度が高いGANから得られるような画像



図5 精度の低い GAN から得られるような画像



図6 精度が中程度の GAN から得られるような画像



図7 精度が高い GAN から得られるような画像



図8 ユーザが求めるような実際に撮影された画像

表1 キーワード5種及び全体の適合率と全再現する Top k

	適合率			全再現する Top k		
	平均値	最大値	最小値	平均値	最大値	最小値
全体	0.50	0.70	0.30	61	77	21
black body	0.70	0.70	0.70	56	56	56
blue wings	0.60	0.60	0.60	23	25	21
curved beak	0.31	0.40	0.30	73	73	73
red wings	0.49	0.50	0.40	76	76	76
white belly	0.42	0.50	0.40	77	77	77

(4) ユーザが求めるような実際に撮影された画像

図5のような画像を16枚、図6のような画像を10枚、図7のような画像を10枚、図8のような画像を41枚用意し、合計77枚の実験用サンプルを作成する。

また、ユーザーリクエストのキーワードに依って結果が変化するため様々な状況での評価を行うために、入力されるキーワードを複数想定して実験を行う。キーワードは「black body」「blue wings」「curved beak」「red wings」「white belly」の5種類と仮定し、実験用サンプルの中にキーワード毎の正解として筆者らが合っていると判断した画像を各10枚ずつ用意した。これらの画像は合計77枚の中に含まれている。

3.2 実験結果

各々のキーワードにおける10番目までの「適合率」と、何番目の出力ですべての正解が出揃うかの「再現率」を結果として記述する。初めに、 $0.0 \leq \alpha \leq 1.0$ における10番目までの適合率の平均と最大値、最小値、さらに再現率が1.0に達した際の Top k の値の平均と最大値、最小値についてキーワード毎に記述し、加えてキーワード5種の全体についても同様に表1にまとめる。また、全体の上位 k 件の再現率の変化を図9に示す。

「black body」を入力した際の10番目までの適合率は、線形結合パラメータ α を変化させても0.7で一定であり、適合率に変化はなかった。図10は「black body」を入力したときの再現率を示している。図10から上位22番付近までは、再現率が増加傾向にあるものの全体で見ると線形結合パラメータ α の値に関わらず、すべての正解の画像を拾い上げた Top k の値は表

1を見てもわかる通り56番目という結果になった。正解の画像のうち2枚が処理1もしくは処理2の閾値によって、総合評価が0.0になっていることがわかり、優先表示から大きく外されてしまったことから、すべての正解を出すまでに後半まで出力する必要があったため、このような結果になったと考えられる。

「blue wings」についても「black body」と同様に、線形結合パラメータ α を変化させても10番目の適合率は0.6で一定という結果になった。次に図11の「blue wings」の再現率を示す。図11から、「black body」の再現率とは違い正解データの総合評価が0.0となる評価は付けられておらず、早い段階ですべての正解を拾い上げていることがわかる。しかし、上位25番前後までは出力しなければならず、処理3の「req-CNN」と「web-CNN」の評価付けにおいて、正解の画像よりも高い評価を付けている画像が存在したことが原因でこのような結果になったと考えられる。

「curved beak」の適合率については、線形結合パラメータ $\alpha = 0.0$ の場合のみ0.4という最大値になり、その他の α の値ではすべて0.3という結果になった。次に「curved beak」についての再現率を図12に示す。正解の画像中の4枚が処理1と処理2の閾値により総合評価が0.0になり、優先表示から大きく外されていることや、図12の結果から、前半の推移を見ても残りの正解の画像6枚が出力されるまで34番目まで出力を要していることと、表1の平均適合率が低い値であるという点から処理3の「req-CNN」と「web-CNN」の分類能力が低いことがわかる。これは、「curved beak」という鳥のパーツとしても小さな特徴についてCNNが規則性を見つけられず、分類が難しくなったためであると考えられる。

「red wings」の適合率は、線形結合パラメータ $\alpha = 0.0$ の場合のみ0.4という最小値になり、その他の α の値では0.5という結果になった。次に「red wings」についての再現率を図13に示す。「red wings」については、正解の画像中から4枚の画像が総合評価値に0.0を付けられており、優先表示から外れている。図13の結果から線形結合パラメータ $\alpha = 1.0$ の場合で、他の値の時よりも正解を優先していることが確認できる。これは、処理3の「req-CNN」で学習用に「red wings」のキーワード検索をして抽出された画像に「red wings」の要素が含まれていない（または、少ない）画像が多かったため精度が出ず、「web-CNN」に重みを置いて用いた結果の方が正解を優先的に表示できたと考えられる。

「white belly」の適合率は、線形結合パラメータ $\alpha = 0.0$ と $\alpha = 0.1$ の時に0.5の最大値になり、その他の値では0.4という結果になった。次に「white belly」についての再現率を図14に示す。「white belly」については、正解の画像中から4枚の画像が総合評価値に0.0を付けられており、優先表示から外れている。表1の結果からは平均適合率が低いことがわかるが、図14の再現率推移から、28番目付近まで出力することで線形結合パラメータ α の値に関わらず、比較的早い段階で残りの6枚は優先的に表示されていることが確認できる。つまり、処理1及び処理2の閾値によって評価が下げられていることが原因で適合率が下がっていると考えられる。

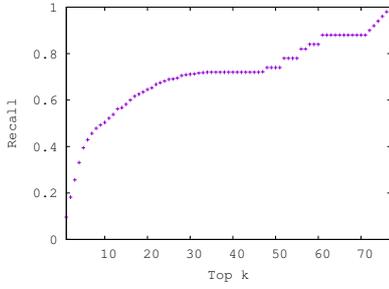


図 9 キーワード 5 種全体の再現率

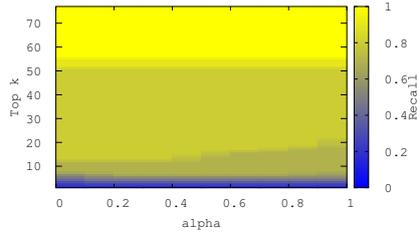


図 10 キーワード「black body」の再現率

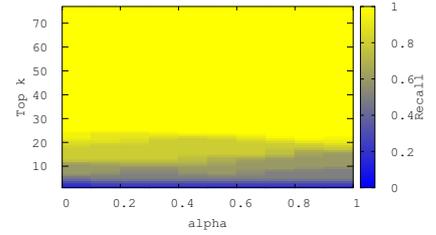


図 11 キーワード「blue wings」の再現率

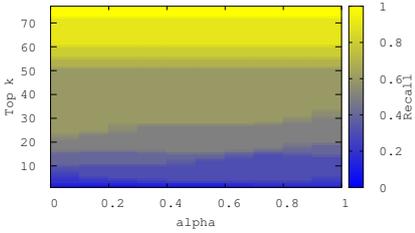


図 12 キーワード「curved beak」の再現率

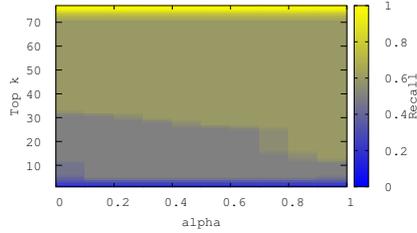


図 13 キーワード「red wings」の再現率

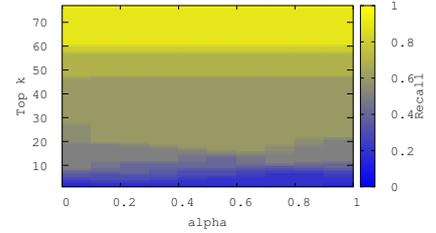


図 14 キーワード「white belly」の再現率

3.3 考察

どのキーワードの適合率の結果を見ても、線形結合パラメータ α を変化させることによる 10 番目までの適合率の大きな変化は見込めない結果となった。これは、処理 3 の「req-CNN」及び、「web-CNN」によるリクエスト反映度の評価によって高い評価値を得られているものの、lap、obj-CNN による閾値分類の評価によって、正解の画像が各再現率推移の結果から間引かれており、10 番目までの適合率を表示すると、lap、obj-CNN で除外された正解の画像が優先から外れ、10 番以内で連続で正解を当てていたとしても、10 番目まで表示すると総合的に適合率が同じになってしまったためであると考えられる。また、各キーワードの再現率推移が中間の件数の際に大きく間隔を空けてから再現率が 1.0 に収束している点からも見受けられる。さらに全体の再現率の平均である図 9 からも、ほとんどのキーワードが正解の画像を除外されていることで後半まで出力する必要がある状態であることが見て取れる。

4. 評価実験: 2つの閾値への依存性

3 章の「線形結合パラメータ α への依存性」に関する評価実験の結果は、処理 1 の lap、処理 2 の obj-CNN により正解の画像の一部が除かれていたことが原因で、処理 3 の req-CNN と web-CNN とを線形結合するパラメータ α を変化させても適合率に差が生じず、複数の CNN を組み合わせることによるロバスト性の向上にはつながらなかった。そこで本章では、各キーワードについて、2.4 節の総合評価の数式 (1) 及び数式 (2) の閾値を変化させて最も高い適合率となる組を求める。

4.1 実験結果

キーワード毎、平均的に適合率の高い値を出している時の線形結合パラメータ α の値に固定して、数式 (1) の閾値と数式 (2) の閾値を変化させた場合の上位 10 件までの適合率の変化

を 2 次元カラーマップとして表示する。各図のラベル「lap th」「obj-CNN th」「Precision」はそれぞれ、「数式 (1) の閾値」「数式 (2) の閾値」「上位 10 件までの適合率」を示している。

4.2 考察

いずれのキーワードの適合率の結果を見ても「lap th」の値が低い場合に適合率が高くなる傾向が見受けられる。また、「obj-CNN th」についても、obj-CNN th = 0.6 付近よりも小さい場合に適合率が高くなる傾向が見受けられ、それ以上になると低く推移していることがわかる。

さらに、キーワード「blue wings」の図 17 と「white belly」の図 20 から、適合率が比較的高い分布を示しているのは「lap th」「obj-CNN th」の値がお互いに小さい部分に集中しており、処理 1 と処理 2 によって除外する画像の制限を緩める方が優先表示できていることがわかる。これは、3 章の線形結合パラメータ α の依存性についての実験結果で、正解の画像の総合評価が処理 1 と処理 2 によって 0.0 と付けられ、優先表示できなかった状況に陥っていた結果になったことや図 15 のキーワード 5 種の平均適合率からも裏付けられる。つまり、処理 3 の「req-CNN」及び「web-CNN」に画像を与えることで図 5 や図 6 のような画像の評価を低く付ける、もしくは正解の画像に対して比較的高い評価を付けることができていることがわかり、処理 1 や処理 2 で除外を行うよりも、直接処理 3 を行う方が優先表示ができていることがわかった。

5. まとめと今後の研究課題

本稿では、GAN から想定される生成画像を処理 1 のぼやけ具合の評価、処理 2 の生成対象物の評価、処理 3 のユーザーリクエストの反映度の評価 (2 種類) によって 4 種類の評価値を算出し、処理 4 で 4 種類の評価値を用いた総合評価を行うことで、ユーザのリクエストに合った画像を優先的に表示できるような

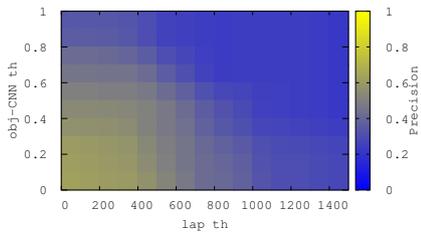


図 15 キーワード 5 種の平均適合率
(α はキーワード毎の最良の値)

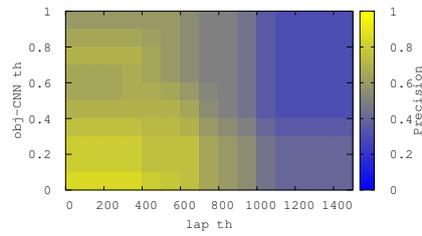


図 16 キーワード「black body」の
適合率 ($\alpha = 0.3$)

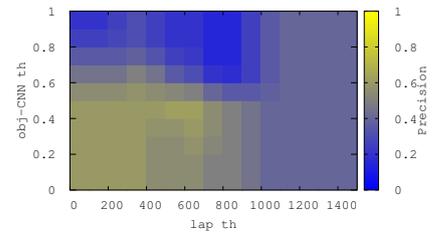


図 17 キーワード「blue wings」の
適合率 ($\alpha = 0.0$)

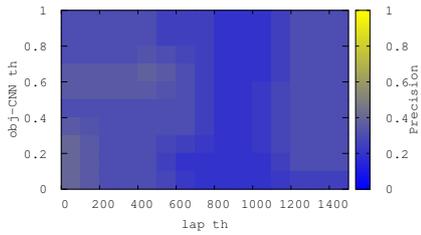


図 18 キーワード「curved beak」の
適合率 ($\alpha = 0.4$)

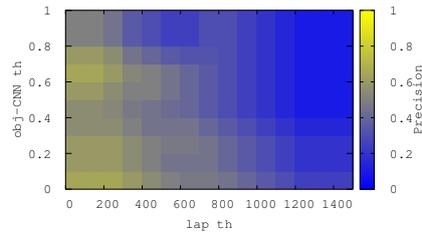


図 19 キーワード「red wings」の
適合率 ($\alpha = 1.0$)

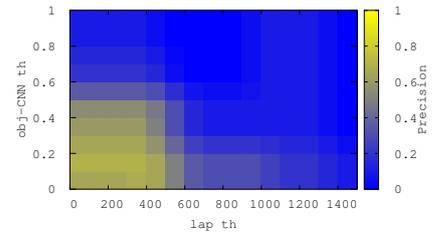


図 20 キーワード「white belly」の
適合率 ($\alpha = 0.0$)

フィルタリング手法を提案した。次に、処理 4 で「req-CNN」「web-CNN」の評価値を線形結合するパラメータ α の依存性についての実験を行い、10 番目までの適合率を出力させたところ、 α の大きな依存性は認められず、処理 1 と処理 2 の閾値設定によって適合率が下がっている可能性があるという結果が得られた。そこで、処理 1 と処理 2 の最適な閾値を求めるために、閾値への依存性を実験した結果、図 16 と図 19 のみ「obj-CNN th」の値が 0.6~0.8 付近で適合率が上昇していることを確認できるが、その他は処理 1 及び処理 2 の閾値に依って正解の画像を除外してしまう場合が多く、数式 (3) の評価方法によって、10 番目までの適合率が α の値を変えても変化が微小になってしまうといった結果になり、優先順位を付けても正解を拾い上げることができなくなってしまうといった課題が判明した。

改善案としては、根本的に処理 1 を取り除くか、総合評価方法に工夫を加えることで改善する必要があると考えられる。例えば、数式 (4) のように閾値を用いず、各評価値の重みを調節することや、直接のスコア同士の掛け算によって処理 1 や処理 2 での除外を行わなければ、正解の画像の脱落を防ぎつつ、ぼやけた画像の評価を下げながらユーザーの要求順に並べることが可能になると考える。

$$\begin{aligned} \text{eval}(img) &= \text{lap}(img)^\beta \cdot \text{obj-CNN}(img)^\gamma \\ &\cdot \{(1 - \alpha) \text{req-CNN}(img) + \alpha \text{web-CNN}(img)\} \end{aligned} \quad (4)$$

$$(0 \leq \alpha, \beta, \gamma \leq 1)$$

また、処理 2 については、本稿では鳥のカテゴリを含むデータセットを用いたが、「cifar10」の画像サイズが 32×32 ピクセルと小さいことで、誤認評価を発生させる原因になっていることも否定できない。さらに、鳥であったにもかかわらず、他

のオブジェクト（人の手など）が写っている画像や、鳥の一部（頭や脚のみ）のみが写っている画像などを鳥ではないと誤認し、評価値が下げられている可能性も考えられる。これについては、画像サイズがより大きいデータを利用して学習させることや、様々な写り方をした鳥の画像を用意する必要があると考えられる。

さらに、本稿では、キーワード毎の正解の画像を 10 枚に揃え、総合評価の値でランキングして 10 番目までの適合率を求めてきたが、実際に GAN の生成画像を直接フィルタリングする際は、フィルタリング処理にかかる時間の短縮や、何枚まで画像を表示すればユーザーが一度に得る情報量が多過ぎず、ユーザーの要求に合った画像を妥当な数表示できるかといったインターフェース的な観点の課題も残されている。

文 献

- [1] Jorge Agnese, Jonathan Herrera, Haicheng Tao, Xingquan Zhu, “A Survey and Taxonomy of Adversarial Neural Networks for Text-to-Image Synthesis” Computer Vision and Pattern Recognition (CVPR), arXiv.org, pp.1–26 (2019).
- [2] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, Jieping Ye, “A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications,” Journal of L^AT_EX Class Files, vol.14, no.8, pp.1–28 (2015). <https://arxiv.org/pdf/2001.06937.pdf>
- [3] OpenCV ライブラリ, <https://opencv.org/> (2020).
- [4] Caltech-UCSD Birds-200-2011, <http://www.vision.caltech.edu/visipedia/CUB-200-2011.html> (2020).
- [5] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, Xiaodong He, “AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks,” Computer Vision and Pattern Recognition (CVPR), arXiv.org, pp.1–9 (2018).
- [6] Alex Krizhevsky, “The CIFAR-10 dataset,” <https://www.cs.toronto.edu/~kriz/cifar.html> (2020).