情報処理学会論文誌　数理モデル化と応用　Vol.10　No.2　1–13 (July 2017)

畳み込みニューラルネットワークを用いた画像分類タスクの直感的可視化方法

荒井 敏[1,a)]　長尾 智晴[1]

受付日 2016年11月13日，再受付日 2017年1月2日，
採録日 2017年1月18日

概要：深層ニューラルネットワークは画像認識の様々な分野で目覚ましい成果をあげているが，今後の産業応用を考えるうえでは解決すべき課題も残されている．一例として，画像分類のタスクでは分類結果をラベルとして出力するだけではなく，画像中のどの部位に着目して分類がなされたか，分類の根拠を示すよう求められる場合がある．筆者らはこの問題を解決するシンプルな構成のネットワークを提案する．提案手法では分類スコアと直接対応する可視化用のマップが分類タスクの過程で生成され，視覚的に確認可能なマップが分類結果に自然な形で反映される．ベンチマーク用画像を用いて実験を行い，本手法が可視化手法として有効であることを示す．

キーワード：深層学習，畳み込みニューラルネットワーク，画像分類，可視化

Intuitive Visualization Method for Image Classification Using Convolutional Neural Networks

Satoshi Arai[1,a)]　Tomoharu Nagao[1]

Received: November 13, 2016, Revised: January 2, 2017,
Accepted: January 18, 2017

Abstract: Deep neural networks show excellent performance in various image recognition field. However, some issues remain for future industrial applications. For example, in image classification tasks, users might request not only to estimate class label but also to answer where the system give attention to classify. We propose novel network architecture to solve this issue. Our method generates 2D maps directly related to classification scores during classification, and generated maps are visually recognizable and reflected to classification result naturally. We empirically indicate effect of our method for existing datasets.

Keywords: deep learning, convolutional neural networks, image classification, visualization

1. はじめに

近年，深層学習 [1], [2], [3] の発展にともなって認識処理の性能が急速に向上し，コンシューマ分野から産業分野に至るまで様々な活用の機運が高まっている．

画像認識の分野では，長らくヒト視覚系が究極の目標であったが，畳み込みニューラルネットワーク（CNN）と大規模データによる学習を組み合わせることで認識精度が大きく向上し [4]，一般画像の分類においてヒト視覚系の認識精度 [5] を超えるような結果も得られるようになった [6], [7]．

CNN は畳み込み層を構成要素の１つとして用いるニューラルネットワークであり，特に画像認識や画像生成の分野で広く用いられている．実際には畳み込み層に加え，正規化層，プーリング層，全結合層といった構成要素を複雑に積み上げた構成であり，ネットワーク規模に応じて表現力が向上するため，高度なタスクに応用する際はより大規模でより層数の多い（深い）ネットワークが求められる傾向

[1] 横浜国立大学大学院環境情報学府
Graduate School of Environment and Information Sciences, Yokohama National University, Yokohama, Kanagawa 240–8501, Japan
[a)] arai-satoshi-rw@ynu.jp

にある．このような大規模なネットワークを学習によって全体最適化できることが深層学習の強みである反面，全体的な動作の把握を難しくしている．

現在，深層学習はその高い性能に牽引される形で普及が進んでいるが，その動作に関しては十分理解が進んでいるとはいえず，いまだにブラックボックスであるといってよい．これはコンシューマ分野ではあまり重視されないかもしれないが，産業分野，特に品質検査や医療などの安全性に関わる分野に応用する際は無視できない課題となりうる．

産業分野への応用においては単に正しい認識結果を返すだけでは不十分で，どのような観点で認識処理を行ったのか，何らかの根拠を示すように求められる場合がある．特に画像入力に対してラベルのみを出力するような画像分類のタスクでは，認識処理が想定した対象に対して正しく行われているかという懸念がつねにあるため，これを確認する意味でも実際に画像中のどの部位に着目して分類が行われたかという情報は重要である．

このような背景をふまえ，本稿では CNN を用いて単に画像を分類するだけではなく，認識結果に関する適当な根拠を提示する手法に焦点を当てる．

2. 関連研究

2.1 畳み込みニューラルネットワーク（CNN）

CNN はフィルタの畳み込み演算（convolution）を用いた多層のフィードフォワード型ニューラルネットワークである．畳み込み演算を用いた画像認識系の着想は Fukushima [8] の Neocognitron に端を発する．LeCun ら [9] は手書き数字画像分類用の処理系として，逆伝播法を用いた end-to-end 学習が可能であり現在の CNN の原型となる LeNet を提案した．

LeNet 以降様々なバリエーションが提案されており現在も活発な研究が続いているが，画像認識用のネットワークに関しては基本的な骨格はおおむね共通している（**図 1**）．まずネットワークの前半は主に特徴量を算出する役割を担う．ここでは畳み込み演算と活性化関数を用いた非線形変換が多段に適用されるが，その途中，プーリング層によって空間サイズが縮小されるほか，正規化処理などが行われる．特徴量は入力画像に比べて空間サイズが小さく，チャネル数が大幅に増加するのが普通であり，活性化関数として ReLU [10] を用いたネットワークでは値が 0 となる要素を多く含む．すなわち，疎表現の符号化（encode）が行われている．そしてネットワークの後半は特徴量を集約しクラスの判別を行う．前半部分に比べると後半のバリエーションは少なく，GAP 層（Global Average Pooling）や全結合層（Full Connect）などを組み合わせ，最後に Softmax 回帰 [11] を行うのが普通である．

2.2 Encoder-Decoder モデル

Encoder-Decoder モデル（**図 2**）は画像から特徴量を算出（encode）する処理と特徴量から画像を再構成（decode）する処理を組み合わせたネットワークで，画像生成系（generative model）の処理などで広く用いられている [12], [13], [14]．一般に CNN は特徴量を算出する過程で空間サイズを縮小するので，再構成処理においては空間サイズの拡大が必要になる．拡大処理には単純なアップサンプリング（upsampling）のほか，逆畳み込み演算（deconvolution）[15] がよく用いられる．

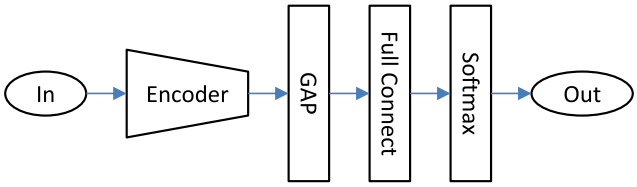

図 1　畳み込みニューラルネットワーク（CNN）の基本的な構成
Fig. 1　Basic structure of convolutional neural network (CNN).

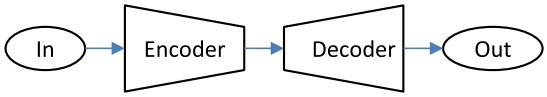

図 2　Encoder-Decoder モデル
Fig. 2　Encoder-Decoder model.

2.3 Attention ベースモデル

Attention ベースモデルは入力データに含まれるすべての情報を 1 度に扱うのではなく，その一部の情報に注目（attend）して処理を行う手法である．入出力の規模が大きい，あるいは不定サイズのような場合でも，計算リソースを大きく増加させることなく精度の良い処理が可能となる．ただし，1 度に得られる情報が部分的になるため，入力データに含まれるすべての情報を得るために注目範囲を変えながら複数回，系列的に情報を取得する必要がある．また，注目範囲を系列的に制御する仕組みもあわせて必要である．

Bahdanau ら [16] は機械翻訳に attention ベースモデルを適用することで，不定長，特に学習データより長い入力テキストを扱う場合に固定長の特徴ベクトルでは表現力が不足する問題を解決する手法を提案している．

Xu ら [17] は静止画像を入力として説明文（caption）を生成するタスクに attention ベースモデルを適用している．この手法では Encoder（CNN）が算出した特徴量マップを重み付き加算してコンテキストベクトルを生成する．コンテキストベクトルは空間的な注目領域を表しており，情報を空間的に絞り込んでいる．重みを系列的に制御することで注目領域が画像中を巡回し，あわせて説明文が生成される．また，コンテキストベクトルを拡大し入力画像にオーバレイ表示することで，生成された説明文の各単語に関連する画像領域を可視化している．

2.4　可視化手法

画像分類ネットワークの内部状態あるいは分類結果を描写することを目的とした，複数の可視化手法がこれまでに提案されている．以下に代表的なものをあげる．

2.4.1　ユニットの反応マップを生成する手法

Zeiler ら [18] は CNN に画像を入力した際に max pooling や ReLU がスイッチのように振る舞うことに着目し，逆畳み込み演算（deconvolution）を用いて中間層のユニットの反応を可視化する手法を提案している．この手法はある入力を与えた場合の注目ユニットの反応を逆畳み込み演算の反復によって入力方向に逆伝播させ，最終的に入力画像と同じサイズ（空間解像度）の反応マップを生成し，これを可視化用のマップ（可視化マップ）とする．注目ユニットを変えながら繰り返し可視化マップを生成することで，ネットワークを構成する全ユニットの反応特性を可視化することが理論的には可能である．

Springenberg ら [19] は勾配逆伝播の仕組みを利用することで反応マップをより簡便に生成する手法（Guided Back-propagation）を提案している．この手法は学習の完了した認識ネットワークに含まれるすべての ReLU に対して，逆伝播時に通過する勾配を非負値にクリップする制約を追加する．そのうえで，入力に関する注目ユニットの微分を計算することで入力画像と同じサイズの反応マップを求め，これを可視化マップとしている．

これらの手法は，可視化の対象が単一のユニットであるためネットワーク全体の挙動を総合的に把握することが難しく，最終的な分類結果との関連性を理解するのが困難であるという問題がある．また，生成される可視化マップが微分画像の様相を示すため視認しにくく，入力画像との対応を把握しにくいのも難点である．

2.4.2　中間層の出力をそのまま可視化する手法

Lin ら [20] は画像分類ネットワークの過学習を避ける観点から，全結合層を用いない構成を提案している．これは畳み込み層の最終チャネル数をクラス数と同一に揃え，その出力を global average pooling を用いて集約してクラスごとのスコア（クラススコア）を得るものである．また Lin らは，畳み込み層から出力される特徴量マップ（feature maps）が各クラスの信頼度マップ（categorical confidence maps）として解釈可能であることを示している．

この手法は CNN が生成する特徴量マップをそのままシンプルに可視化マップとして利用するものである．しかし，一般的な CNN では位置不変性を高めるために使用されるプーリング層やストライドの影響で特徴量マップの空間サイズが縮小されるため，入力画像と比較して空間解像度の低い情報しか得られないという問題がある．

2.4.3　物体の概略位置を示すマップを生成する手法

Zhou ら [21] は，ある画像を学習済みの画像分類ネットワークに入力した場合の畳み込み層の出力（特徴量マップ）を注目クラスに対応する全結合層の重みを用いて重み付き加算することで，注目クラスに関する物体の概略位置を示す重みマップを生成し可視化する手法（Class Activation Mapping; CAM）を提案している．

Selvaraju ら [22] は Zhou らの考え方を発展させた手法（Grad-CAM）を提案している．これは全結合層における注目クラスの出力を特徴量マップで偏微分することで，注目クラスに対する特徴量マップ各チャネルの重要度を求め，この重要度を重みとして重み付き加算することで，CAM と同様に注目クラスに関する物体の概略位置を示す重みマップを生成し可視化するものである．

これらの手法はいずれも物体の概略位置を示す重みマップを生成できるという点で有効であるが，2 つの問題をかかえている．1 つは空間解像度の低下である．特徴量マップは入力画像と比較して空間解像度が低く，これを組み合わせて生成した重みマップも同様に解像度が低いものとなる．もう 1 つは可視化マップの生成方法である．どちらの手法も，生成された重みマップを拡大したうえで疑似カラー化し，さらに入力画像にオーバレイ表示することで可視化を行っているが，この生成手順には，

(1) 重みマップの拡大処理
(2) 重みマップの疑似カラー化
(3) 入力画像との加重混合によるオーバレイ表示

という 3 つのヒューリスティックな処理が含まれる．(1)〜(3) の処理はいずれも可視化結果に影響を与える一方，基となった画像分類ネットワークの分類結果とは無関係に調整が可能である．すなわち，同一の分類結果であっても作為によって可視化結果は変化する．したがって，このようなヒューリスティックな可視化方法は分類結果を忠実に反映しているとはいいがたい．可視化マップから分類結果を直感的に理解することも同様に難しいといえる．

以上の説明から分かるとおり，これまでに提案された可視化手法には以下の 2 つの問題があり，筆者らの知る限りこれらを同時には解決した手法は存在しない．

- 可視化マップの空間解像度が低下する．
- 可視化結果が分類結果と直接対応していない．

画像分類のタスクにおいて分類結果とともに根拠を示すためには，これらの課題を解決した可視化手法が必要である．

このような問題点をふまえ，本稿では画像分類の根拠を直感的に可視化できる新たな画像分類手法 Generative Contribution Mappings（GCM）を提案する．

3.　Generative Contribution Mappings

本章では提案手法について詳説する．まず基本的な考え方を説明し，それから実際のネットワーク構成とバリエー

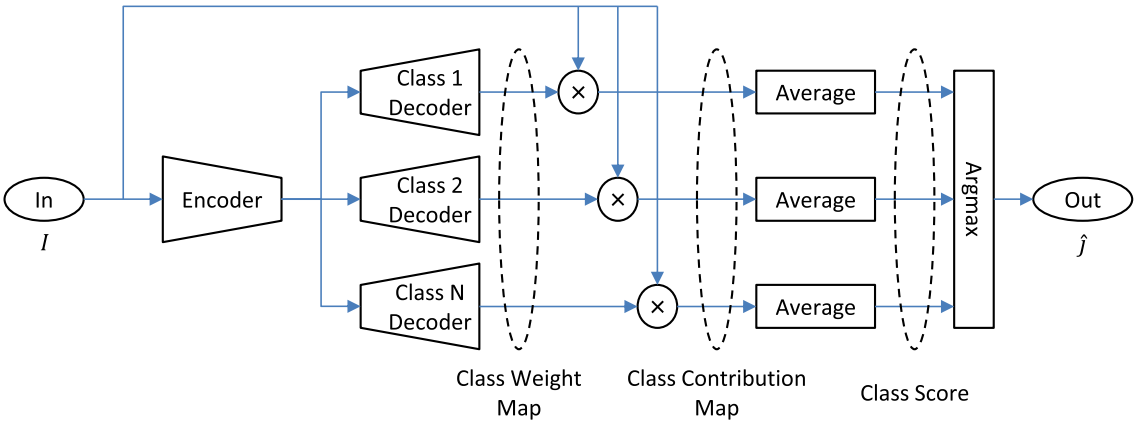

図 3　Generative Contribution Mappings のネットワーク基本構成
Fig. 3　Basic network structure of Generative Contribution Mappings.

ションについて記述する．さらに理論的な解釈について解説する．

3.1　基本的な考え方

提案手法の目的は，入力画像をクラス分類すると同時にその分類が画像中のどの部位に着目してなされたかという分類の根拠をユーザに提示することである．これを実現するために以下の方針を採用する．

1. 画像中のどの部位に注目して分類が行われたか，根拠となる情報を二次元のマップとして提示する．マップは入力画像と同じ解像度で生成し，比較を容易にする．
2. 提示する情報は分類結果と直接関連したものとする．情報を見たユーザがそこから分類結果を直感的に推定できるようなものが好ましい．
3. 方針 1，2 を実現するための構成を初めから画像分類ネットワークに組み込んでおく．これにともなうネットワーク規模の増大は許容する．

3.2　ネットワーク構成

3.1 節の方針を実現するために提案手法で用いる画像分類ネットワークの基本的な構成を**図 3** に示す．このネットワークは，一般的な画像分類ネットワークと比較して大きく以下の 3 点が異なっている．

1. Encoder の後続処理として decoder を有する．
2. Decoder の出力と入力画像との直接的な乗算経路を有する．
3. 乗算後，単純な平均処理を用いてクラススコアを算出し，全結合層は用いない．

これらの構成要素が持つ意味は 3.5 節で改めて考察する．

以下，ネットワークの動作について詳細に説明する．

入力画像 $I \in \mathbb{R}^{R\times C\times D}$ は $R\times C\times D$ の次元数，すなわち垂直画素数 R，水平画素数 C，チャネル数 D を持つとする．特に RGB 画像の場合は $D=3$ である．ネットワークはクラス数 N の画像分類を行い，いずれかのクラスラベル j（$j=1,2,\ldots,N$）を出力するものとする．

I はエンコーダ（Encoder）で次元数任意の特徴量に変換された後，デコーダ（Decoder）によって $R\times C$ の次元数を持つマップに再構成される（式 (1)）．このマップは入力画像の各位置が注目クラスに関してどの程度そのクラスらしいかを表す空間的な重みマップであり，Class Weight Map（CWM）と呼ぶ．CWM は正負の値をとり，値が負になる場合はそのクラスらしくない程度を表している．デコーダは分類する各クラスに対して 1 つ，合計 N 個が用意され，したがって CWM もクラス数 N と同数が生成される．

$$M^{(j)}_{CWM} = F^{(j)}_{Decoder}(F_{Encoder}(I)) \tag{1}$$

ただし，$M^{(j)}_{CWM} \in \mathbb{R}^{R\times C}$ と $F^{(j)}_{Decoder}$ はクラス j（$j=1,2,\ldots,N$）の CWM とデコーダを，$F_{Encoder}$ はエンコーダそれぞれ表す．

次に $R\times C$ の次元数を持つ各クラスの CWM を D 回コピーしてチャネル方向に連結し，入力画像 I と同じ次元数 $R\times C\times D$ に拡張する．この演算を Tile と表記し，Tile 演算の結果を $W^{(j)} \in \mathbb{R}^{R\times C\times D}$ とする（式 (2)）．

$$W^{(j)} = \mathrm{Tile}\big(M^{(j)}_{CWM}\big) \tag{2}$$

さらに $W^{(j)}$ と入力画像 I を要素ごとに乗算することで新たなマップを得る（式 (3)）．このマップは入力画像 I からの情報と CWM からのクラスらしさの情報の両方をあわせ持ち，入力画像のどの部位が注目クラスらしいかという情報を一目で把握可能になっている．これを Class Contribution Map（CCM）と呼び，ユーザに提示するための可視化情報（可視化マップ）とする．

$$M^{(j)}_{CCM} = W^{(j)} \otimes I \tag{3}$$

ただし，$M^{(j)}_{CCM} \in \mathbb{R}^{R\times C\times D}$ はクラス j の CCM を表し，入力画像 I と同じ $R\times C\times D$ の次元数を持つ．演算子 $\otimes$

は要素ごとの積を表す．

さらに CCM を空間およびチャネルのすべての軸に関して平均（global average）することでクラス j に関するスカラ値のスコア（Class Score）$V_{SC}^{(j)} \in \mathbb{R}$ を得る（式 (4)）．

$$V_{SC}^{(j)} = \text{global\_average}\left(M_{CCM}^{(j)}\right) = \frac{1}{RCD}\sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{d=1}^{D} M_{CCM}^{(j)}(r,c,d) \tag{4}$$

ただし，$M_{CCM}^{(j)}(r,c,d) \in \mathbb{R}$ は $M_{CCM}^{(j)}$ の位置 (r,c)，チャネル d における要素を表す．

最終的にクラススコア $V_{SC}^{(j)}$ の最も高いクラス $\hat{j}$ を分類結果として出力する（式 (5)）．

$$\hat{j} = \underset{j}{\text{argmax}}\, V_{SC}^{(j)} \tag{5}$$

以上が提案手法の基本的な処理の流れであり，Class Contribution Map を生成的（generative）に求めることから Generative Contribution Mappings（GCM）と呼ぶ．

3.3 Shared Decoder

GCM は分類する各クラスに対して 1 つのデコーダを割り当てるため，通常の CNN と比較してネットワーク規模が増大し，これは特にクラス数が多い場合に問題となる．この問題を緩和するため，デコーダの一部をクラス間で共有する構成（Shared Decoder）を用いる．すなわち，デコーダを大きく前半と後半に分割し，前半は単一のデコーダを全クラスで共通して使用し，後半はこれまでどおり各クラスに 1 つのデコーダを割り当てる構成とする．

3.4 動的な重み生成ネットワークとしての解釈

GCM は画像分類ネットワークとしては複雑な構成に見えるかもしれないが，以下のように変形すると単純な形に整理できる．

まず式 (3) を式 (4) に代入し，要素ごとの表現に改めると式 (6) を得る．

$$V_{SC}^{(j)} = \text{global\_average}(W^{(j)} \otimes I) = \frac{1}{RCD}\sum_{r=1}^{R}\sum_{c=1}^{C}\sum_{d=1}^{D} W^{(j)}(r,c,d) I(r,c,d) \tag{6}$$

ただし，$W^{(j)}(r,c,d) \in \mathbb{R}$ と $I(r,c,d) \in \mathbb{R}$ はそれぞれ $W^{(j)}$ と I の位置 (r,c)，チャネル d における要素を表す．

さらに $W^{(j)}$ と I の要素ごとの積と加算を内積演算としてまとめ，定数除算（$1/RCD$）はクラススコアの大小関係に影響しないことからこれを無視すると，単純な形の式 (7) を得る．

$$V_{SC}^{(j)} = (W^{(j)}, I) \tag{7}$$

ただし，$(\boldsymbol{a}, \boldsymbol{b})$ は $\boldsymbol{a}$ と $\boldsymbol{b}$ の内積を表す．

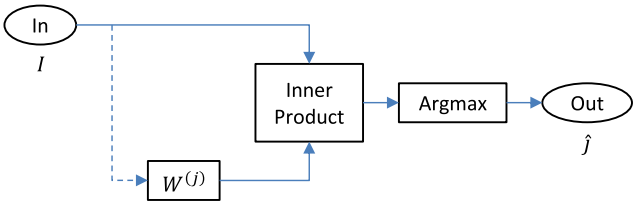

図 4　動的重み生成ネットワーク
Fig. 4　Dynamic weight generation network.

$W^{(j)}$ は入力画像に応じて式 (1) および式 (2) から動的に生成され，I と同じ次元数 $R \times C \times D$ を持つ．式 (7) とあわせて考えると，GCM とは動的に生成された重み $W^{(j)}$ と入力画像 I の内積によってクラススコアを求める動的重み生成型の画像分類ネットワーク（**図 4**）であると解釈できる．

3.5 ネットワーク構成に関する考察

3.2 節の冒頭でも述べたとおり，GCM のネットワーク構成は一般的な画像分類ネットワークの構成と大きく 3 点が異なっている．以下，それぞれの意味について考察する．

3.5.1 Decoder の存在

入力画像に encoder を適用して得られる中間出力は，encoder に含まれるプーリング層やストライドの影響で空間解像度が低下しているため，入力画像との比較や可視化を行う際は何らかの拡大処理が必要となる．GCM では encoder に引き続いて decoder を適用することで中間出力を拡大し，入力画像と同じサイズ（空間解像度）の重みマップである CWM を生成している．中間出力を他のヒューリスティックな処理で拡大する方式も考えられるが，その場合，拡大手法の選択が問題になる．GCM では decoder を学習によって最適化することで，この問題をシンプルに解決している．

3.5.2 入力画像との直接的な乗算経路の存在

GCM では，decoder によって生成された重みマップである CWM を入力画像と直接的かつ要素ごとに乗算し，可視化マップである CCM を生成している．すなわち，重みマップと入力画像の乗算混合によって可視化マップを生成している．

乗算混合によって生成された可視化マップはスポットライトのような視覚効果を持ち，重みの小さい領域が暗色で提示されるため，重みの大きい画像領域を視覚的に確認しやすいという特徴を持つ．

従来の可視化手法 [21], [22] では重みマップを可視化する際に，重みマップと入力画像との重み付き加算，すなわち，加重混合を用いている．加重混合では重みマップとは別に混合係数を用意し，ヒューリスティックな手法などで調整する必要がある．これに対して乗算混合の場合は重みマップを単に入力画像に乗じればよく，追加の混合係数を

必要としない．これはパラメータ数を増加させず処理系をシンプルに保つことに貢献している．

3.5.3　平均処理によるクラススコアの算出

2.1 節で述べたとおり，一般的な画像分類ネットワークは GAP 層を用いて特徴量マップを空間集約したのち，全結合層を適用してクラススコアを算出することが多い．これに対して GCM では可視化マップである CCM を空間およびチャネルのすべての軸に関して平均することでクラススコアを算出しており，全結合層のようにパラメータを有する処理を使用しない（パラメータレス）．これは可視化マップとクラススコアの対応関係を一定かつ直感的に保つために重要である．すなわち，可視化マップから「視覚的な暗算」によってクラススコアを把握することを可能としている．仮に，可視化マップに全結合層を適用してクラススコアを算出する構成の場合，全結合層のパラメータは学習によって変化するため，可視化マップとクラススコアの関係は一定ではなくなる．その結果，ユーザは可視化マップからクラススコアを直感的に把握することが難しくなる．

全結合層を用いない画像分類ネットワークは Lin ら [20] によって提案されたが，Lin らの目的はパラメータ数の削減とそれにともなう過学習の抑制であり，GCM における目的，すなわち，可視化マップとクラススコアの対応関係を一定に保つ，とは異なる．また，Lin らは空間的な平均のみを算出しており，空間およびチャネルのすべての軸に関して平均を算出する GCM とは構成上も異なる．

3.6　従来の可視化手法との相違

2.4 節で述べたとおり，従来の可視化手法は以下の 2 つの問題をかかえており，これらを同時に解決する手法はこれまでに提案されていない．

- 可視化マップの空間解像度が低下する．
- 可視化結果が分類結果と直接対応していない．

しかし，GCM ではこれらの問題を同時に解決している．まず，decoder を用いて重みマップ（CWM）を生成し，入力画像との乗算混合によって可視化マップ（CCM）を生成することで，第 1 の問題である可視化マップの空間解像度の低下を防ぎ，入力画像と同じサイズの可視化マップを生成している．さらに，可視化マップの単純な平均によってクラススコアを算出し，このクラススコアが最大となるクラスに分類することで，可視化結果と分類結果を直接的に対応させ，第 2 の問題を解決している．

GCM はこれら 2 つの問題を同時に解決することで「画像分類のタスクにおいて分類結果とともに根拠を示す」ことをより高いレベルで実現するものであり，従来の可視化手法にはない新しい価値を提供している．

3.7　Attention ベースモデルとの相違

新たな画像分類ネットワークの提案という本稿の目的とはやや視点が異なるが，ここで attention ベースモデルとの相違について論じたい．その理由は，attention ベースモデルの定式化において GCM の定式化と一部類似する構成が現れるため，両者の違いを論じることは GCM の独自性を確かめるうえで有益であると思われるからである．

Attention ベースモデルは 2.3 節で述べたように，入力データに含まれる一部の情報に注目（attend）して処理を行い，計算リソースを大きく増加させることなく精度の良い処理を可能とする手法である．ただし，注目内容を変えながら複数回，系列的に情報を取得する必要があり，そのための制御が必要であることも述べた．以下，具体的な定式化について比較に必要な範囲で簡単に説明する．

3.7.1　Attention ベースモデルの定式化

Attention ベースモデルでは，まず入力データに encoder を適用し，複数の特徴量ベクトル $\mathbf{a}_\mathrm{i}$ を生成する．特徴量ベクトルの数を L，次元数を D でそれぞれ表す（式 (8)）．

$$\{\mathbf{a}_1, \ldots, \mathbf{a}_L\}, \quad \mathbf{a}_i \in \mathbb{R}^D \tag{8}$$

次にすべての特徴量ベクトルを用いて時刻 t におけるコンテキストベクトル $\boldsymbol{c}_t \in \mathbb{R}^D$ を式 (9) のように算出する．

$$\boldsymbol{c}_t = \sum_{i=1}^{L} w_{ti}\mathbf{a}_i \tag{9}$$

ここで $w_{ti} \in \mathbb{R}$ は時刻 t において特徴量ベクトルを重み付けするための正の重みであり，特徴量ベクトル $\mathbf{a}_i$ および前時刻の処理系の内部状態 $\boldsymbol{h}_{t-1}$ を用いて式 (10) のように算出される．ここで ϕ は非線形関数を表す．

$$w_{ti} = \phi(\mathbf{a}_i, \boldsymbol{h}_{t-1}) \tag{10}$$

時刻 t における処理系の出力 $\boldsymbol{y}_t$ は前時刻の出力 $\boldsymbol{y}_{t-1}$，現時刻の内部状態 $\boldsymbol{h}_t$ およびコンテキストベクトル $\boldsymbol{c}_t$ を用いて式 (11) のように算出される．ここで f は非線形関数を表す．

$$\boldsymbol{y}_t = f(\boldsymbol{y}_{t-1}, \boldsymbol{h}_t, \boldsymbol{c}_t) \tag{11}$$

また，処理系の内部状態は式 (12) のように更新される．ここで g は非線形関数を表す．

$$\boldsymbol{h}_t = g(\boldsymbol{y}_{t-1}, \boldsymbol{h}_{t-1}, \boldsymbol{c}_t) \tag{12}$$

式 (10)〜式 (12) における非線形関数 ϕ，f，g は，具体的には LSTM [23] などの RNN を用いて実装される．

Attention ベースモデルでは，中核となる式 (9) によって L 個の特徴量ベクトル $\mathbf{a}_i$ が重み付き加算され，時刻 t における単一のコンテキストベクトル $\boldsymbol{c}_t$ に集約される．すなわち特徴量を表すパラメータ数は $1/L$ に削減され，情報が絞り込まれる．その代わり $\boldsymbol{c}_t$ は系列的に複数生成され，特徴量ベクトルの持つ情報を順次取得する構成となっている．また，$\boldsymbol{c}_t$ を系列的に生成するため，式 (10) に従って各時刻 t における重み w_{ti} を制御している．

表 1　両手法の定式化における対応関係
Table 1　Correspondence in formulation of Attention-based Model and GCM.

Attention ベースモデル	GCM
$\mathbf{a}_i \in \mathbb{R}^D$	$I \in \mathbb{R}^{R\times C\times D}$
$w_{ti} \in \mathbb{R}$	$W^{(j)} \in \mathbb{R}^{R\times C\times D}$
$\boldsymbol{c}_t \in \mathbb{R}^D$	$V_{SC}^{(j)} \in \mathbb{R}$
$\phi(\cdot, \boldsymbol{h}_{t-1})$	$\mathrm{Tile}\left(F_{Decoder}^{(j)}(F_{Encoder}(\cdot))\right)$
$\boldsymbol{h}_t$	該当なし

3.7.2　提案手法（GCM）との比較

Attention ベースモデルの定式化における式 (9) と GCM の定式化における式 (6) を比較すると，どちらも入力テンソルの重み付き加算を算出する形式となっている．さらに式 (10) と式 (1) および式 (2) を比較すると，重み算出の際に入力テンソルを用いているという共通点が見られる．これらをまとめると**表 1** のようになる．

したがって，Attention ベースモデルと GCM は定式化において部分的な類似性が認められる．しかしながら，GCM は Attention ベースモデルとは以下に示す点で異なっており，両者は明確に区別されるべきものである．

まず Attention ベースモデルにおける重み w_{ti} は特徴ベクトルを重み付き加算してコンテキストベクトル $\boldsymbol{c}_t$ を算出するために用いられる．コンテキストベクトルは特徴ベクトルから情報を部分的に抽出したものであるため，特徴ベクトルの持つ情報全体を取り出すためには系列的（sequential）に複数回生成される必要がある．そのため，これに用いられる重み w_{ti} も系列的に生成する必要があり，さらにこれを制御する仕組みとして内部状態 $\boldsymbol{h}_t$ を有するネットワークが必須である．また，各時刻において生成される重みの数は入力となる特徴ベクトルの数 L と等しく，出力サイズとは無関係である．

一方 GCM では，重み $W^{(j)}$ は入力画像 I との内積を求めることでクラススコア $V_{SC}^{(j)}$ を算出するために用いられる．重み $W^{(j)}$ は各クラス j に対して独立かつ並列的（parallel）に生成され，1 度の適用でクラス j の分類に必要な情報をすべて抽出するように最適化される．すなわち，Attention ベースモデルのように情報を部分的かつ系列的に抽出する意図はなく，内部状態を用いて重みの生成を制御する仕組みを必要としない．また，各入力画像に対して生成される重み $W^{(j)}$ の数は分類結果として出力されるクラス数 N と等しく，画像や特徴ベクトルなどの入力サイズとは無関係である．

以上の相違点を**表 2** にまとめた．Attention ベースモデルと GCM が明らかに異なる手法であることが分かる．

また繰返しになるが，GCM は，第 1 に decoder による入力画像と同解像度の重みマップ（CWM）の生成，第 2 に CWM と入力画像との直接的な乗算による可視化マップ（CCM）の生成，第 3 にパラメータレスの集約演算（空間・チャネル平均）を用いたクラススコアの生成，という 3 つの構成が揃って効果を発揮する手法であり，単に特徴量マップの重み付き加算を用いただけでは同様の効果は得られない点を強調しておきたい．

表 2　両手法の相違点
Table 2　Differences between Attention-based Model and GCM.

比較項目	Attention ベースモデル	GCM
重みの生成方法	系列的	並列的
重みの生成回数	複数回	1 回
重みの数	入力ベクトル数と等しい	出力クラス数と等しい
内部状態を用いた重みの制御	必要	不要

4.　実験

提案手法の有効性を検証するため，ベンチマークテスト用の画像データを用いて実験を行った．今回使用した画像データは CIFAR-10 [24] および SVHN Format 2 [25] である．

4.1　実験設定

本稿の実験に共通して用いられる設定についてまず説明する．

(1)　**ネットワーク構成**

基本的な構成は図 3 に示すとおりであるが，ネットワーク規模抑制のためデコーダを Shared Decoder（3.3 節）に置き換える．また出力の前に Softmax 回帰を加え，損失関数として Categorical Cross-Entropy（CCE）を用いる（**図 5**）．エンコーダとデコーダの詳細な構成を**表 3** に示す．パラメータ欄の 3x3，c16，s1 はフィルタサイズ 3×3，出力チャネル数 16，ストライド 1 を表す（他の場合も同様）．BN は Batch Normalization [26]，ReLU は Rectified Linear Unit の適用をそれぞれ表す．デコーダは 5 層からなるが，層 11 から 14 は Shared Decoder であり全クラス共通で使用される．層 15 はクラス別に用意される．

(2)　**学習条件**

ネットワークの学習条件を**表 4** に示す．これらの学習条件は CIFAR-10 に含まれる 50,000 枚の学習データを 45,000 対 5,000 に分割し，前者を学習データ，後者を検証データとする予備実験を行って事前に決定した．

(3)　**実験環境**

実験に用いたソフトウェアおよびハードウェアを**表 5** に示す．実験プログラムはすべてスクリプト言語 Python で

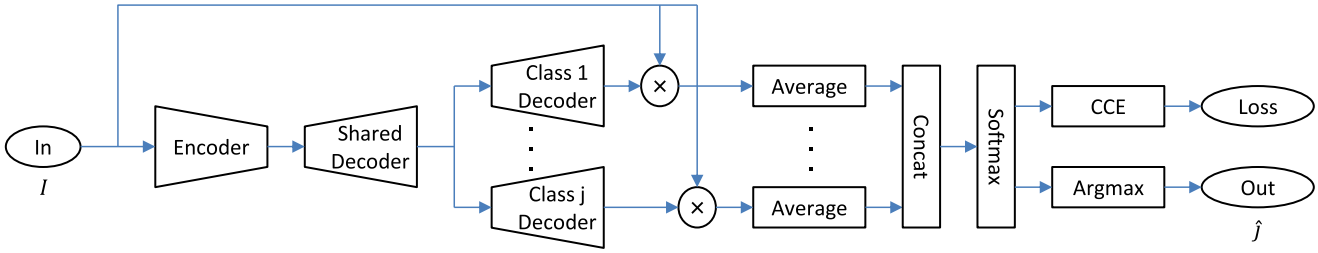

図 5　実験に用いたネットワーク構成
Fig. 5　Network structure used in experiments.

表 3　実験に用いたエンコーダ/デコーダの構成
Table 3　Structure of Encoder-Decoder used in experiments.

	層番号	種別	パラメータ
Encoder	1	Conv	3x3, c16, s1, BN, ReLU
	2〜4	Conv	3x3, c16, s1, BN, ReLU
	5	Conv	3x3, c32, s2, BN, ReLU
	6, 7	Conv	3x3, c32, s1, BN, ReLU
	8	Conv	3x3, c64, s2, BN, ReLU
	9, 10	Conv	3x3, c64, s1, BN, ReLU
Decoder	11	Deconv	3x3, c32, s2, BN, ReLU
	12	Conv	3x3, c32, s1, BN, ReLU
	13	Deconv	3x3, c16, s2, BN, ReLU
	14	Conv	3x3, c16, s1, BN, ReLU
	15	Conv	5x5, c1, s1

表 4　学習条件
Table 4　Configurations for training.

エポック数	180
バッチサイズ	128
最適化手法	SGD
学習率	0.1（エポック 100 と 150 でそれぞれ 1/10 に切り下げ）
モーメント	0.9
前処理	なし
データ拡張	画像の上下左右に 4 画素 0 padding. ±4 画素のランダムシフト切り出し＋水平方向ランダムフリップ.

実装している [27], [28], [29].

4.2　CIFAR-10

CIFAR-10 は 50,000 枚の学習データと 10,000 枚のテストデータから構成される画像分類タスクのデータベースであり，各画像に 10 クラスのいずれかのラベルが付与されている．画像はすべて 32 × 32 画素の RGB 3ch データである．

4.1 節の実験設定を用いて学習とテストを行い，テスト画像に対して 90.43%の分類精度を得た．学習パラメータ数は約 162,000 個であった．

表 5　実験に使用した環境
Table 5　Environments for experiments.

OS	Windows 10 Pro
CPU	Intel Corei7-6700K 4GHz
GPU	NVIDIA TITAN X
開発環境	Visual Studio 2013, CUDA 8.0, cuDNN 5.1 Python(Miniconda2) / Theano / Lasagne

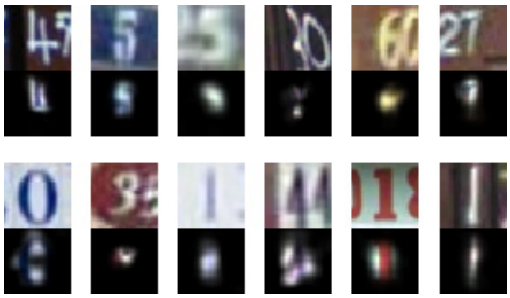

図 6　提案手法を SVHN 画像に適用した結果
Fig. 6　Results of proposed method applied to SVHN images.

図 7 に正しく分類されたテスト画像の例とその CCM を示す．CCM は正負の値をとるため，正の成分と負の成分に分けて表示している．奇数段の一番左が原画像，二番目以降が各クラスに対応した CCM の正の成分である．偶数段は同じく CCM の負の成分を示している．下側の数値は CCM から算出したクラススコアである．また，正解クラスの CCM には赤枠を付けた．GCM では CCM の空間平均がそのままクラススコアになるので，どのクラスのスコアが高いかを目視でおおよそ読み取ることができる．

図 9 にさらにいくつかの例について原画像と正解クラスの CCM（ただし正の成分のみ）を示す．

4.3　CCM の負の成分をクリップした場合

3.2 節で述べたように CCM は正負の値をとりうるが，あえて負の成分をクリップし，0 または正の成分に限定した場合の挙動を観察した．そのため表 3 に示すネットワーク構成の層 15 に ReLU を追加し，学習をやり直した．このときテスト画像に対する分類精度は 89.91%であった．**図 8** に原画像および各クラスの CCM を示す．

図 7　提案手法を CIFAR-10 画像に適用した場合の Class Contribution Map（CCM）それぞれ上段が正の成分，下段が負の成分，数値がクラススコアを表す．正解クラスを赤枠で示す．

Fig. 7　CCMs generated by proposed method applied to CIFAR-10 images.

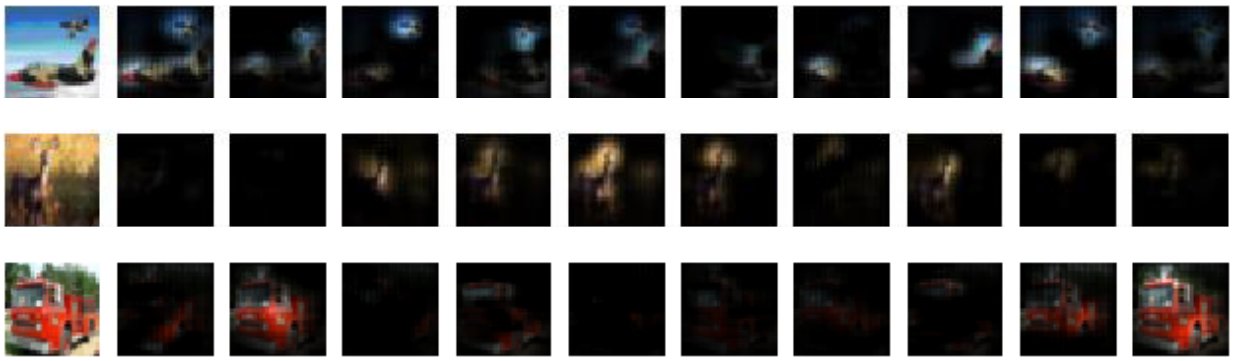

図 8　負の成分をクリップして学習した場合の CCM

Fig. 8　CCMs when training by clipping negative components.

4.4　CNN との分類精度の比較

GCM は通常の CNN とは異なりデコーダにもパラメータを割り当てなければならないため，同じパラメータ数の CNN と比較して分類精度が低下する可能性が懸念される．その程度を確認するため，CNN との比較実験を行った．

比較に使用した CNN の構成を**表 6**，パラメータ数と分類精度の比較結果を**表 7** に示す．学習条件は表 4 のとおりだが，CNN 側のみ前処理として中心化を行っている．

4.5　SVHN Format 2

SVHN は Google Street View から抽出された家屋番号の画像データベースで，73,257 枚の学習データと 26,032 枚のテストデータを含む．画像は 32×32 画素の RGB 3ch で，中央の数字を推定する 10 クラス分類のタスクである．

4.1 節の実験設定を用いて学習とテストを行い，テスト画像に対して 96.19%の分類精度を得た．

図 6 に正しく分類されたテスト画像の例とその CCM（ただし正の成分のみ）を示す．

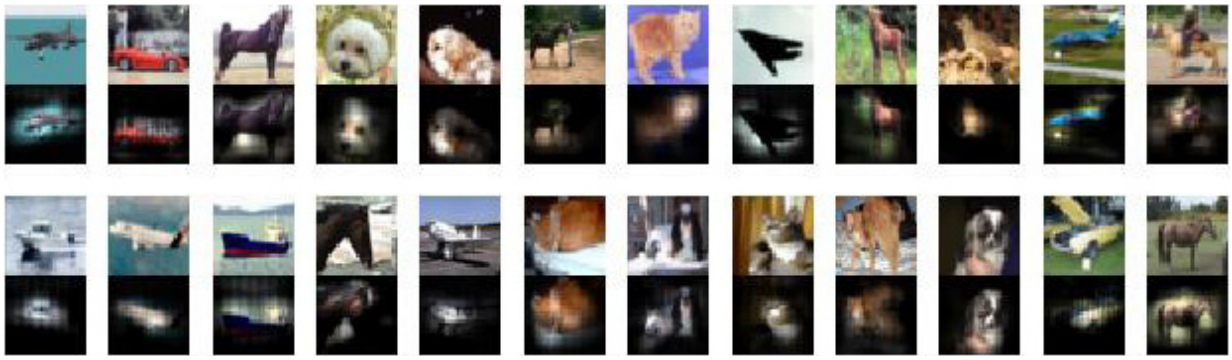

図 9　原画像および正解クラスの CCM
Fig. 9　Original images and CCMs of correct classes.

表 6　比較実験に使用した CNN の構成
Table 6　Structure of CNN used for comparison.

層番号	種別	パラメータ
1	Conv	3x3, c16, s1, BN, ReLU
2〜4	Conv	3x3, c18, s1, BN, ReLU
5	Conv	3x3, c37, s2, BN, ReLU
6, 7	Conv	3x3, c37, s1, BN, ReLU
8	Conv	3x3, c74, s2, BN, ReLU
9, 10	Conv	3x3, c74, s1, BN, ReLU
11	GAP	
12	FC+Softmax	

表 7　CNN との比較結果
Table 7　Comparison result between CNN and GCM.

手法	パラメータ数	処理時間比	分類精度
CNN	164K	1	90.39%
GCM	162K	1.15	90.43%

4.6　従来の可視化手法との比較

CIFAR-10 の代表的な画像に関して，GCM および従来の可視化手法を用いて可視化マップを生成し，その内容を比較した．比較した可視化手法は，Guided Backpropagation（GB）[19]，Guided Grad-CAM（GGCAM）[22]，CAM [21]，Grad-CAM [22]，および GCM の 5 つである．GCM は 4.2 節で学習したネットワークをそのまま使用し，それ以外の手法は 4.4 節で学習した CNN を基にして可視化マップを生成した．

GB および GGCAM では正解クラスに対応する全結合層のユニットに対する反応マップを生成し，これをそのまま可視化マップとして使用した．

CAM および Grad-CAM では正解クラスに関して生成した重みマップを bicubic 補間によって入力画像と同じサイズに拡大（縦横各 4 倍）し，さらに疑似カラー化（jet）したうえで入力画像と加重混合して可視化マップを生成した．加重混合の割合は重みマップ : 入力画像が 0.7 : 0.2 である．この一連の可視化手順は Zhou ら [21] が公開している CAM の実証コード [30] に含まれるものであり，CAM および Grad-CAM における標準的な可視化マップ生成方法といえる．**図 10** では拡大処理を行っていない原状態の重みマップと上記手順で生成した可視化マップを提示した．

GCM では正解クラスに関して生成された CWM および CCM をそのまま提示した．

生成された可視化マップを図 10 に示す．なお，本実験に使用したコードはすべて筆者らが表 5 の環境を用いて実装した．

4.7　乖離率を用いた定量評価

2.4 節でも述べたとおり，画像分類結果とともにその根拠を示すためには，可視化結果と分類結果が直接対応していることが求められる．

したがって，可視化結果と分類結果の乖離度合いを測ることで，分類の根拠を示すという観点での可視化手法の優劣を論じることが可能と考えられる．そこで以下のような評価実験を行った．

1. 画像分類ネットワークを用いてクラス分類を行った場合の分類精度を算出する．
2. 1. で用いた画像分類ネットワークを基に，可視化手法で可視化マップを生成する．
3. 2. で生成した可視化マップを集約し，これをクラススコアとしてクラス分類を行って，分類精度を算出する．集約手法としては最も単純な可視化マップの全要素の加算を用いる．
4. 1. および 3. で算出された分類精度を用いて乖離率（後述）を算出する．
5. 算出された乖離率を用いて複数の可視化手法を比較評価する．

本実験では乖離率を式 (13) のように定義する．

$$\text{乖離率} = \frac{(A) - (B)}{(A)} \tag{13}$$

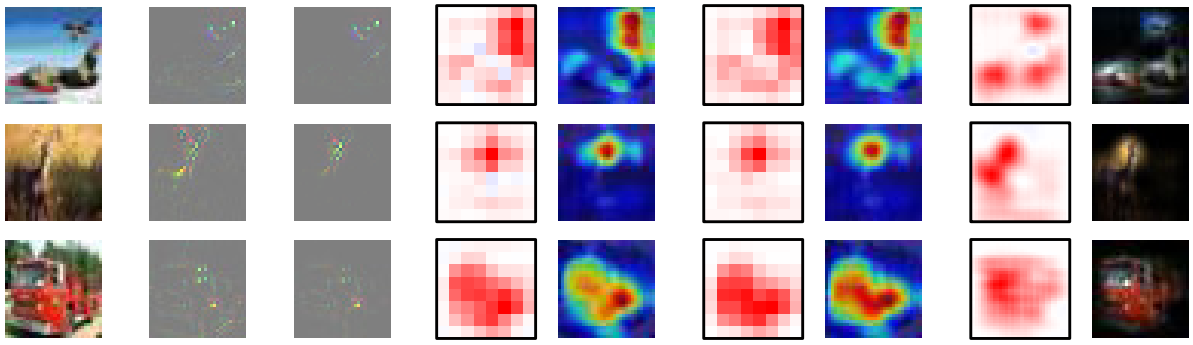

図 10　従来の可視化手法および提案手法における可視化マップの比較 (a) 原画像　(b) Guided Backpropagation　(c) Guided Grad-CAM　(d) CAM の重みマップ　(e) CAM の可視化マップ　(f) Grad-CAM の重みマップ　(g) Grad-CAM の可視化マップ　(h) 提案手法の重みマップ（CWM）　(i) 提案手法の可視化マップ（CCM）

Fig. 10　Visual maps generated by prior art and proposed method.

表 8　可視化マップから算出したスコアを用いてクラス分類を行った場合の分類精度と乖離率

Table 8　Accuracy and decline ratio in classification using class score calculated from visual maps.

可視化手法	基準精度	(イ) 重みマップ		(ロ) 重みマップの絶対値		(ハ) 入力画像との加重混合		(ニ) 入力画像との乗算混合	
		分類精度	乖離率	分類精度	乖離率	分類精度	乖離率	分類精度	乖離率
Guided Backprop	90.39%	84.76%	6.23%	84.85%	6.13%	84.76%	6.23%	23.98%	73.47%
Guided Grad-CAM	90.39%	88.33%	2.28%	88.18%	2.44%	**88.33%**	**2.28%**	87.91%	2.74%
CAM	90.39%	89.80%	0.65%	87.11%	3.63%	87.39%	3.32%	30.64%	66.10%
Grad-CAM	90.39%	88.21%	2.41%	86.74%	4.04%	86.37%	4.45%	33.00%	63.49%
提案手法 (GCM)	90.43%	**90.13%**	**0.33%**	**88.52%**	**2.11%**	88.14%	2.53%	**90.43%**	**0.00%**

- 分類精度は数値が大きいほど良く，乖離率は数値が小さいほど良い．
- 太字は各列で最も良い分類精度あるいは乖離率を表す．
- 下線は各手法の標準的な可視化マップを用いた場合の分類精度を表す．

ここで (A) および (B) は，

(A) 画像分類ネットワークによる分類精度（基準精度）

(B) 可視化マップを集約してクラススコアとした場合の分類精度

をそれぞれ意味する．

評価データとして CIFAR-10 を使用し，4.6 節と同様に，GB，GGCAM，CAM，Grad-CAM，GCM の 5 つの可視化手法に関する評価を行った．ただし本実験では，各可視化手法の標準的な可視化マップ生成手法に加え，他の可視化手法で用いられる可視化マップ生成手法もそれぞれ実施し，直交性を持たせた．具体的には，各可視化手法で生成された重みマップを用いて以下の 4 つの方法でそれぞれ可視化マップを生成した．

(イ) 重みマップをそのまま可視化マップとして用いる．

(ロ) 重みマップの絶対値を可視化マップとして用いる．

(ハ) 重みマップと入力画像の加重混合によって可視化マップを生成する．4.6 節で述べた CAM の標準的な可視化マップ生成手順を用いる．

(ニ) 重みマップと入力画像の乗算混合によって可視化マップを生成する．

ただし GB および GGCAM では，反応マップをそのまま重みマップとみなして用いる．

評価の結果得られた分類精度と乖離率を**表 8** に示す．乖離率が小さいほど可視化結果は分類結果をより忠実に反映しているといえる．乖離率が大きい場合，可視化結果には分類結果とのギャップがあることを意味する．

5. 考察

(1) GCM を用いた可視化の効果

図 7 (A) の画像は複数の物体を含むため，通常の画像分類の場合，どちらの物体を分類対象としたかという疑問が残る．しかし CCM を見れば答えは一目瞭然で，両方の物体を対象としたスコアの合算によって分類していることが容易に読み取れる．

図 7 (C) では正解である truck のほか，automobile にも比較的強い反応が見られる．車両前面は両者に共通して反応している一方，荷台部分は truck のみ強く反応し，スコアを増加させている．すなわち，処理系が truck へと分類した決め手は荷台部分の有無であったことが分かる．CCM を観察することでこのような分析も可能となる．

また，図 6 において画像に含まれる複数の数字の中から画像中央付近の数字が正しく注目され，分類に反映されていることが確認できる．

(2)　**CCM における負の成分の効果**

図 7 および図 8 を比較すると，正解クラスの CCM にはそれほど大きな差は見られないが，正解以外のクラスでは後者がより強い反応を示し，クラス分類の観点でノイズが多い状態といえる．CCM の負の成分は可視化情報の質を向上させる効果があるといえる．

(3)　**分類精度の低下について**

表 7 によれば，少なくとも今回の実験設定では GCM には懸念されたような分類精度の低下は見られず，同程度のパラメータ数の CNN と同等の分類精度が得られている．ただし，デコーダを持つことで層数が増加するため，処理時間は若干増大している．

(4)　**従来の可視化手法との定性的評価**

図 10 を用いて各手法の可視化マップを比較すると，(b) 列の GB および (c) 列の GGCAM では (a) 列の原画像と同じサイズのマップが得られているものの，その内容は視認しやすいものとはいえない．(d) 列の CAM および (f) 列の Grad-CAM では解像度の低いマップしか生成されておらず，これらから生成された (e) 列および (g) 列の可視化マップとは見た目が大きく異なることが分かる．すなわち，その外見は可視化手順に負うところが大きい．GCM では (h) 列のように原画像と同じサイズのマップが生成され，さらに (i) 列のように原画像との関連を視認しやすい可視化マップが生成されている．比較した 5 つの手法中，GCM で生成した可視化マップ（CCM）が原画像の内容を最も明確に視認可能である．その結果，可視化マップと原画像との対応を把握しやすくなるため，可視化手法としては好ましい性質といえる．

(5)　**乖離率を用いた定量的評価**

表 8 によれば，今回検討した 4 つの可視化マップ生成方法のうち，3 つの方法において GCM の乖離率が最も良い（値が小さい）結果となった．GCM は（ニ）乗算混合によって可視化マップを生成する前提で最適化を行うため，乗算混合において乖離率が低いのは当然であるが，実際はそれ以外にも（イ）重みマップをそのまま用いた場合や（ロ）重みマップの絶対値を用いた場合にも他の手法より良い結果（乖離率が小さい）を示した．これは CCM の最適化を通じて得られる CWM もまた CCM と同様にクラス分類に適した性質を獲得するためと考えられる．一方，（ハ）加重混合では GGCAM が最も良い結果となった．今回用いた加重混合は 4.6 節で示したようにヒューリスティックな要素を含むため，最適化によって獲得した性質とは適合しなかったと考えられる．

GCM を用いて生成した可視化マップ（CCM）は他の手法と比較して可視化結果と分類結果の乖離が小さく，分類の根拠を示すための可視化手法として GCM が有効であることが分かる．

(6)　**結論**

以上の考察をふまえると，提案手法（GCM）は画像分類の根拠を提示するための可視化手法として有効であり，分類精度の観点でも従来の CNN に劣らず同等であると結論づけられる．

6. まとめ

本稿では画像分類の根拠を提示するための新たな可視化手法を提案した．また，ベンチマークテスト用の画像データを用いてその有効性を確認した．

今後は ImageNet などの大規模なデータベースを用いて検証を進めるほか，実際の応用問題に適用しながら有効性を確認していくことが必要である．

参考文献

[1] 中山英樹：ディープラーニングの発展と最新動向，画像電子学会 (2016/06/01) (2016).

[2] 山下隆義：ディープラーニングによる画像認識と応用事例，DeepLearningDay2016 (2016).

[3] 岡谷貴之：ディープラーニング，映像情報メディア学会誌，Vol.68, No.6, pp.466–471 (2014).

[4] Krizhevsky, A. et al.: ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems 25* (*NIPS 2012*) (2012).

[5] Russakovsky, O. et al.: ImageNet Large Scale Visual Recognition Challenge, *International Journal of Computer Vision*, December 2015, Vol.115, Issue 3, pp.211–252 (2015).

[6] He, K. et al.: Deep Residual Learning for Image Recognition, arXiv:1512.03385 (2015).

[7] Szegedy, C. et al.: Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, arXiv:1602.07261 (2016).

[8] Fukushima, K.: Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position, *Biological Cybernetics*, Vol.36, Issue 4, pp.193–202 (1980).

[9] LeCun, Y. et al.: Backpropagation Applied to Handwritten Zip Code Recognition, *Neural Computation*, Vol.1, No.4, pp.541–551 (1989).

[10] Nair, V. and Hinton, G.E.: Rectified Linear Units Improve Restricted Boltzmann Machines, *Proc. 27th International Conference on Machine Learning* (2010).

[11] Heckerman, D. and Meek, C.: Models and Selection Criteria for Regression and Classification, *Proc. 13th Conference on Uncertainty in Artificial Intelligence* (1997).

[12] Vincent, P. et al.: Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with

a Local Denoising Criterion, *Journal of Machine Learning Research 11*, pp.3371–3408 (2010).
[13] Dong, C. et al.: Learning a Deep Convolutional Network for Image Super-Resolution, *13th European Conference on Computer Vision - ECCV 2014* (2014).
[14] Kulkarni, T.D. et al.: Deep Convolutional Inverse Graphics Network, arXiv:1503.03167 (2015).
[15] Noh, H. et al.: Learning Deconvolution Network for Semantic Segmentation, arXiv:1505.04366 (2015).
[16] Bahdanau, D. et al.: Neural Machine Translation by Jointly Learning to Align and Translate, arXiv:1409.0473 (2014).
[17] Xu, K. et al.: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, arXiv:1502.03044 (2015).
[18] Zeiler, M.D. and Fergus, R.: Visualizing and Understanding Convolutional Networks, arXiv:1311.2901 (2013).
[19] Springenberg, J.T. et al.: Striving for Simplicity: The All Convolutional Net, arXiv:1412.6806 (2014).
[20] Lin, M. et al.: Network In Network, arXiv:1312.4400 (2014).
[21] Zhou, B. et al.: Learning Deep Features for Discriminative Localization, arXiv:1512.04150 (2015).
[22] Selvaraju, R.R. et al.: Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization, arXiv:1610.02391 (2016).
[23] Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, *Neural Computation*, Vol.9, Issue 8, pp.1735–1780 (1997).
[24] Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images (2009). available from ⟨http://www.cs.toronto.edu/~kriz/cifar.html⟩.
[25] Netzer, Y. et al.: Reading Digits in Natural Images with Unsupervised Feature Learning, *NIPS Workshop on Deep Learning and Unsupervised Feature Learning* (2011). available from ⟨http://ufldl.stanford.edu/housenumbers/⟩.
[26] Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, arXiv:1502.03167 (2015).
[27] van Rossum, G. et al.: Python Reference Manual (2001), available from ⟨https://www.python.org/⟩.
[28] Theano Development Team: Theano: A Python framework for fast computation of mathematical expressions, arXiv:1605.02688 (2016).
[29] Dieleman, S. et al.: Lasagne: First release, doi:10.5281/zenodo.27878 (2015).
[30] Zhou, B.: Class Activation Mapping, 2016, available from ⟨https://github.com/metalbubble/CAM⟩.

荒井 敏　（正会員）

1995 年東京工業大学大学院総合理工学研究科博士課程前期修了．精密機器メーカー勤務．横浜国立大学大学院環境情報学府に在学．

長尾 智晴　（正会員）

1985 年東京工業大学大学院総合理工学研究科博士課程後期中退．同年同大学助手．同大学助教授を経て，2001 年横浜国立大学大学院環境情報研究院教授．工学博士．画像処理，進化計算法等の知能情報学の研究に従事．電子情報通信学会，人工知能学会，進化計算学会，IEEE 等各会員．