Vol.2022-ARC-248 No.12
Vol.2022-SLDM-198 No.12
Vol.2022-EMB-59 No.12
2022/3/10

深層の学習済み重みを利用した CNNの計算量削減に関する初期検討

黒田 大貴[1]　津邑 公暁[1]

概要：画像認識等において，CNN（Convolutional Neural Network）と呼ばれるニューラルネットワークが高い認識精度を示し，広く利用されているが，計算量が大きく，この計算量を削減することが，CNN の大きな課題となっている．計算量削減手法として，ランダムな値によるパラメータ固定化を行った先行研究として LBCNN がある．また，これに対し我々は，CNN が 1 層目の畳み込み層で単純な特徴パターンの抽出を行うという特性に基づきパラメータ固定化を行う，Functionally-Predefined Kernel を提案している．しかし，複雑な特徴パターンの抽出を行う 2 層目以降の畳み込み層に含まれるカーネルは固定できておらず，計算量の削減が十分ではない．そこで本稿では，学習が十分に行われた CNN の，2 層目以降の畳み込み層に含まれるカーネルに，共通して存在する特徴を機械的に分析し，抽出する方法を検討する．CNN の 2 層目の畳み込み層に含まれるカーネルを事前定義することで，先行研究で達成される認識精度を維持したまま，更に計算量を削減する方法について考察・検討する．

1. はじめに

画像認識等において，CNN（Convolutional Neural Network）と呼ばれるニューラルネットワークが高い認識精度を示し，文字認識や顔認証，自動運転における歩行者認識などのアプリケーションに広く利用されている．一方で，この CNN では，学習・推論において膨大な回数の積和演算が必要であり，この計算量の大きさが問題となっている．

この問題に対し，CNN に含まれるパラメータ値や計算を近似することで計算量を削減する研究が盛んに行われている．しかし，近似方法や近似度によっては，元のパラメータが持つ情報が大きく損なわれ，CNN の認識精度が有意に低下してしまう．このような認識精度の低下は，アプリケーションによっては無視できない場合も多いため，認識精度を低下させることなく計算量を削減することが，CNN の大きな課題となっている．

この問題に対する別の取り組みとして，パラメータ固定化による計算量削減手法 [1] がある．この手法は，CNN に含まれるパラメータの一部を学習中固定し，これらパラメータに対する更新処理を不要とすることで，認識精度を維持したまま計算量の削減を目指すものである．この手法については，先に述べた課題に対する有用性が示されているが，どのようなパラメータ配置にするか，すなわち，どのようなカーネルを用いるかについては議論されていない、

これに対し我々は，カーネルテンプレート化 [2] という考え方を過去に提案している．カーネルテンプレート化では，パラメータ固定化の際，ランダム生成されたカーネルではなく，単純な特徴パターンを抽出することに特化したカーネルを用いることで認識精度を維持したまま計算量を削減する．しかしカーネルテンプレート化は，1 層目の畳み込み層で単純な特徴パターンの抽出を行うという，CNN の特性に基づいたものであるため，2 層目以降のカーネルは固定できておらず，計算量の削減が十分ではない．

学習が十分に行われた CNN の，1 層目の畳み込み層に含まれるカーネルと，2 層目以降の畳み込み層に含まれるカーネルには，特徴抽出における役割の違いが存在することを示した先行研究 [3] がある．1 層目の畳み込み層では，ブロブ検出や，縦・横・斜め方向のエッジ検出など，基本的な特徴を抽出していると考えられるカーネルが多く見られることが，この研究では示されている．したがって，基本的な特徴を抽出することに特化したカーネルを事前に作成する，という方針をとることができた．一方，2 層目以降の畳み込み層では，複雑な特徴を抽出していると考えられるカーネルが多く見られることも，この研究では示されており，カーネルの事前定義は困難であると考えられる．

そこで本研究では，学習が十分に行われた CNN の，2 層目以降の畳み込み層に含まれるカーネルに共通して存在する特徴を機械的に分析し，抽出する方法を検討する．その

[1] 名古屋工業大学
Nagoya Institute of Technology

Vol.2022-ARC-248 No.12
Vol.2022-SLDM-198 No.12
Vol.2022-EMB-59 No.12
2022/3/10

初期検討として，本研究では 2 層目の畳み込み層を対象とし，クラスタリングによりそれを分類することで，カーネルの事前定義を試みる．

2. 関連研究

本章では，CNN の計算量を削減することを目的とした様々な研究について述べる．

2.1 パラメータ量子化によるアプローチ

CNN の計算量削減に向けた一つのアプローチとして，パラメータを量子化する方法が挙げられる．パラメータ量子化について様々な研究が提案されているが，代表的な論文として，XNOR-NET[4] がある．この論文が提案している手法の一つである Binary-Weight-Networks については，パラメータを +1 または −1 の二値に量子化する事で，メモリ使用量を 32 分の 1 にまで抑えられることが確認されている．これにより，携帯機器のようなメモリ容量が限られた環境においても，大規模な CNN の処理が可能になると著者らは主張している．また，この論文が提案しているもう一つの手法である XNOR-Networks については，パラメータだけでなく入力特徴マップも二値に量子化する．これにより，CNN 内の乗算を単純な XNOR 演算で置き換えることが可能となり，メモリ使用量を削減できるだけでなく，畳み込み処理を 58 倍高速に実行できることが確認されている．そして著者らは，最先端の大規模な CNN の推論を，GPU 上ではなく CPU 上でもリアルタイムで実行できるようになると主張している．一方で，認識結果の第一候補が正解と一致する割合については，例えば AlexNet が ImageNet データセットに対して推論した際に 56.7%であるのに対し，AlexNet に対して XNOR-Networks の手法を適用した際には 44.2%となり，10%以上の大幅な低下を引き起こしてしまうことが確認されている．

同様にパラメータを量子化する研究としては，パラメータを対数表現するものも提案されている．これらの研究では，対数表現を用いることで乗算をシフト演算に置き換え可能とし，これにより処理の高速化を目指している．また，より少ないビット数でパラメータを表現したとしても，認識精度の低下を抑えることができると主張している．実際に，これらの手法の評価結果からは，認識精度に大幅と言えるほどの低下は見られないものの，やはりある程度の認識精度の低下が確認できる．以上に示すように，パラメータ量子化について様々な手法が提案されているが，いずれの手法においても，量子化により有意に認識精度が低下してしまう部分を考慮せずに量子化を行うため，CNN の認識精度が有意に低下してしまう場合が多い．そして，アプリケーションによってはこの認識精度の低下が無視できなくなる可能性がある．

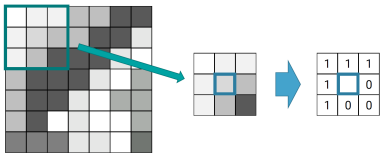

図 1 LBP 特徴量の算出 (1)

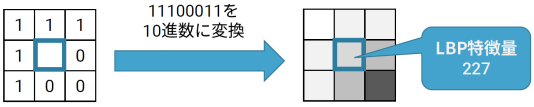

図 2 LBP 特徴量の算出 (2)

2.2 パラメータ固定化によるアプローチ

この問題を解決するために， パラメータ固定化を用いた CNN 計算量削減手法が提案されている．この手法では，CNN に含まれるパラメータの一部を学習中固定し，これらパラメータに対する更新処理を不要とすることで，CNN の学習に要する計算量を削減する．

2.2.1 LBCNN

パラメータ固定化に関する研究として，Juefei-Xu らが提案する Local Binary Convolution（LBC）[1] がある．この研究では，標準的な CNN における畳み込み層を，LBC 層と呼ばれる層で代替することで，計算量削減を実現している．LBC 層は，学習中に更新されない三値のパラメータで構成されるスパースなカーネルのセット，非線形活性化関数，および学習により重みが更新される線形パラメータのセットとで構成される．なお，標準的な CNN における畳み込み層を LBC 層で代替した CNN は，LBCNN（Local Binary Convolutional Neural Network）と呼ばれる．LBCNN の学習においては，カーネルのパラメータを最適化するのではなく，線形パラメータのみを最適化することとなる．そのため，LBC 層は標準的な CNN 層と比較して，学習させる必要のあるパラメータ数を大幅に削減可能である．

LBC 層におけるパラメータは，Local Binary Patterns（LBP）の考えに基づいて設定される．LBP とは，顔認識分野から生まれたシンプルかつ強力な局所特徴量であり，パターン認識や画像処理アプリケーションなどで広く用いられている．ここで，LBP 特徴量の算出方法を図 1 に示す．LBP 特徴量を算出する際，まず画像の一部をパッチとして切り出し (図 1 左，図 1 真中)，そのパッチにおける中心画素の輝度と，その周辺画素の輝度とを比較する．このとき，周辺画素の輝度が中心画素の輝度よりも高ければ 1，低ければ 0 とする (図 1 右)．そして、この比較した結果を図 2 に示すようにビット列に変換し，さらにそれを十進数に変換した値が，パッチにおける中心画素の LBP 特徴量となる．この研究ではこの考え方を応用し，指定されたスパーシティに応じてパラメータ値に 0，あるいは 1 か −1 の三値をランダムに割り当ててカーネルを生成する．そし

Vol.2022-ARC-248 No.12
Vol.2022-SLDM-198 No.12
Vol.2022-EMB-59 No.12
2022/3/10

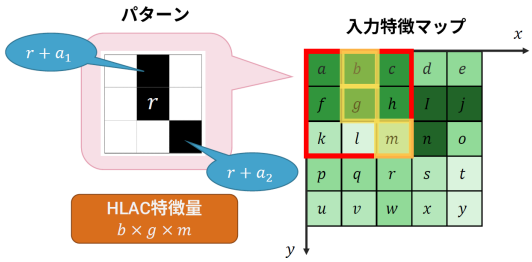

図 3　LBP 特徴量の算出 (2)

て，このランダムに生成されたカーネルを，LBC 層中の学習されないカーネルとして用いる．

LBCNN は非常にシンプルなモデルであるにもかかわらず，過学習が起こりにくく，また，リソースが制約された環境における学習および推論に適していることが示されている．そして，LBC 層が CNN 層を適切に近似できていることが，理論的に，かつ経験的に証明されている．LBCNN は，標準的な CNN と比較して計算量を大幅に削減しながら，MNIST データセット [5]，SVHN データセット [6]，CIFAR-10 データセット [7]，および ImageNet データセット [8] を用いた評価では，通常の CNN と同等の認識精度を達成している．

2.2.2　カーネルの共通性とその活用

2.2.1 節で示した LBCNN に対し，我々は Functionally-Predefined Kernel[2] という考え方を過去に提案している．Functionally-Predefined Kernel では，パラメータ固定化の際，LBCNN のようにランダム生成されたカーネルを用いるのではなく，ブロブ検出や，縦・横・斜め方向のエッジ検出などの，単純な特徴パターンを抽出することに特化したカーネルを用いることで，LBCNN で達成される認識精度を維持したまま計算量を削減した．

この手法では，Functionally-Predefined Kernel の設計において，高次局所自己相関特徴（HLAC: Higher-order Local AutoCorrelation）[9] を利用している．HLAC とは，任意の次数・領域に対して定義可能な画像特徴量であり，画像認識に対する基本的な要望としての位置不変性および加法性を満たすという特徴を持つ．HLAC は，以下に示す式で定義される．

$$x(a_1,\cdots,a_N) = \int f(r)f(r+a_1)\cdots f(r+a_n)dr$$

なお，式中の r は対象となる画像領域内の位置 (x,y) を，a_n は位置 r からの変位を，N は相関の次数を，$f(r)$ は位置 r における画素値を，それぞれ表している．例えば，図 3（右）に示す入力特徴マップ上に位置する赤色の枠で囲まれた領域に対し，図 3（左）に示すような，位置 r=(2,2) の画素と位置 r から変位 a_1= (0,−1) および a_2=(1,1) の位置にある画素との相関について考える．この時，位置 r の画素の画素値である $f(r)$=g，および位置 r から変位 a_1 および a_2 の位置にある画素の画素値，すなわち $f(a_1)$=b

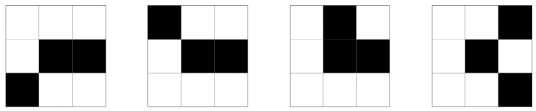

図 4　3 × 3 の領域に対して二次相関で定義できる HLAC パターンの一部

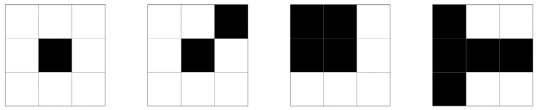

図 5　その他の次数について 3 × 3 の領域に対して定義できる HLAC パターンの一部

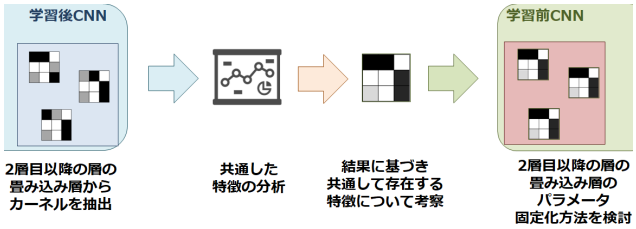

図 6　手法の全体像

および $f(a_2)$=m を全てかけ合わせた bgm が位置 r における HLAC 特徴量となる．この例では，3 × 3 の領域に対して位置 r の画素とその他二つの画素との相関，すなわち二次相関について示したが，二次相関では他にも図に示すようなパターンを定義できる．同様に，HLAC はその他の次数についても，3 × 3 の領域に対し図に示すようなパターンを定義できる．このように，HLAC は局所的な特徴量を抽出可能であり，かつ様々なパターンを定義可能であるため，HLAC に基づいた Functionally-Predefined Kernel を作成した．

CNN は 1 層目の畳み込み層で，単純な特徴パターンの抽出を行うという特性を持つことが知られており [3]，Functionally-Predefined Kernel は，これを活用している．そのため複雑な特徴パターンの抽出の役割を果たす，2 層目以降の畳み込み層に含まれるカーネルは固定できていない．なお，2 層目以降の畳み込み層が持つパラメータは，全ての層の畳み込み層が持つパラメータの内約 85.7%を占めており，2 層目以降の畳み込み層で必要となる計算量は，全ての層の畳み込み層で必要となる計算量の中で大きな割合を占めている．そのため，計算量の削減がまだ十分ではない．

3.　提案手法

3.1　2 層目以降の Predefined Kernel の設計方針

本稿では，2 層目以降の畳み込み層に対して，適切な Predefined Kernel を作成する方法について検討する．この初期検討として，まずは 2 層目の畳み込み層を対象とする．

手法の全体像を図 6 に示す．まず初めに，学習処理が十

Vol.2022-ARC-248 No.12
Vol.2022-SLDM-198 No.12
Vol.2022-EMB-59 No.12
2022/3/10

分に行われた CNN の 2 層目以降の畳み込み層から，カーネルを抽出する．そして，抽出した層毎に，カーネルのクラスタリングを行い，類似した特徴を持つカーネルごとに分類する．そして，各クラスタを代表するカーネルを作成し，それを学習前の CNN に Predefined Kernel として適用することで，計算量を削減する方法について検討する．

3.2 クラスタリング手法の選択

まずはカーネルの分類に適切なクラスタリング手法を採用する必要がある．なお，ここで言う「適切」とは，CNN の学習及び推論時の畳み込みにおける特徴抽出において，類似した役割を果たしているカーネルを同一のクラスタに分類できることを指す．

一般的なクラスタリング手法は，非階層型クラスタリングと呼ばれる，予め設定されたクラスタ数になるようにデータを分類する手法であるため，適切なクラスタ数を事前に適切に設定することが必要となる．しかし，設定したクラスタ数が十分に大きな値でない場合，特徴抽出の上で異なる役割を果たしていると考えられるカーネルが，同一クラスタに分類されてしまう．逆に，過剰に大きな値であった場合には，特徴抽出の上で類似した役割を果たしていると考えられるカーネルであっても，別のクラスタに分類されてしまう．この適切なクラスタ数を事前に特定することは困難である．

これに対し，予めクラスタ数を設定する必要のない，階層型クラスタリングがある．階層型クラスタリングの初期状態では，データ 1 つ 1 つをクラスタとみなす．そして，全クラスタ間の類似度を計算し，類似度の高い 2 つを併合する．このような類似度の計算およびクラスタの併合を繰り返すことで，次第に大きなクラスタが形成されていき，最終的に全てのデータを含んだ 1 つのクラスタができ，この併合の過程はひとつの樹形図で表現されることとなる．本研究ではこの階層型クラスタリングを用いて，学習後カーネルを分類することとする．

3.3 クラスタリングの手順

カーネルのクラスタリングを行うにあたり，類似度の計算には Ward 法を採用した．概要は以下のとおりである．いま，クラスタ A，B があるとする．また，この A および B を併合した場合に生成されるクラスタを C とする（**図 7**）．この A，B，C 各クラスタにおいて，そのクラスタの重心と，そのクラスタに含まれる全データとの差の 2 乗誤差和 S_A，S_B，S_C を計算する．ここで，$\Delta = S_C - S_A - S_B$ として計算された Δ が小さいほど，A と B の類似度が高いとみなし，そのようなクラスタから順に併合する．

併合の過程は，**図 8** に示すような樹形図で表現できる．この樹形図において，葉に近い位置で結合されているものほど，早期に併合された，つまり類似度が高いことを意味

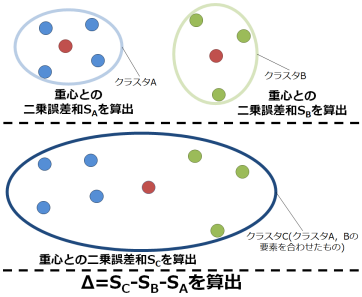

図 7 Ward 法における 2 つのクラスタ間の類似度算出の流れ

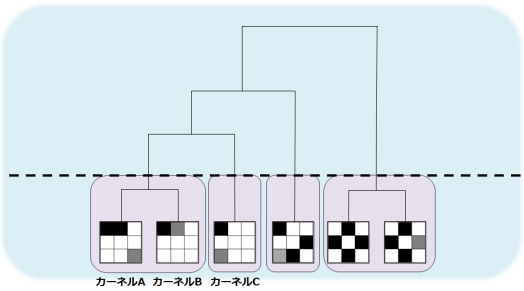

図 8 樹形図とその切断により生成されるクラスタ

する．この樹形図を，ある高さで切断することで，それぞれのクラスタに含まれるカーネルの類似度が高い状態でのクラスタリング結果が得られる．図 8 の例では，破線で示した高さで切断することで，カーネルが 4 つのクラスタに分類されている．階層型クラスタリングでは，満たしたい類似度（樹形図上における高さ）を決めることでクラスタ数も決定されることとなる．よってこの類似度（高さ）を，何らかの基準で適切に設定することが必要であるが，本稿では，切断する高さを変化させた場合の評価を行う．

3.4 Representative Kernel の作成方法

次に，クラスタリング結果に基づいて Representative Kernel を作成する方法について説明する．本研究では LBCNN の学習後カーネルを用いた上で，LBCNN のカーネルを事前固定することを前提とするため，Representative Kernel の要素も -1，0，1 の 3 値とする必要がある．そのため，以下のような手順で Representative Kernel を決定する．まず，クラスタに含まれるカーネルのうち，樹形図上の最も下位でつながっているカーネル，すなわち最も類似度が高いと判断されているカーネルを 3 つ選び，それらのカーネルのパラメータの最頻値からカーネルを 1 つ作成する．次に，その作成したカーネルと，それを作る際に用いた 3 つのカーネルに最も類似度が高いと判断されている 2 つのカーネルとを用いて，先ほどと同じように最頻値からカーネルを 1 つ作る．もし，まだクラスタ内にカーネルが残っていればこの処理を繰り返す．最終的に、クラスタ内のカーネルの数が 3 つ未満になったら，残ったカーネル全てを Representative Kernel にする．このような手順によ

Vol.2022-ARC-248 No.12
Vol.2022-SLDM-198 No.12
Vol.2022-EMB-59 No.12
2022/3/10

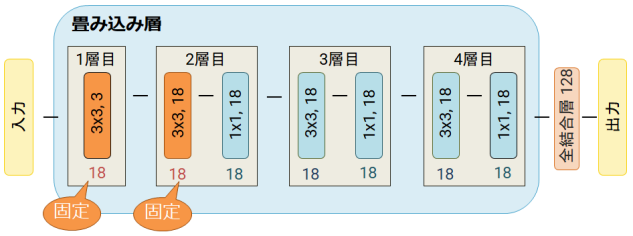

図 9 評価に用いた CNN アーキテクチャ

り，各クラスタから 1 ないし 2 個の Representative Kernel を作成する．

4. 評価

4.1 評価方法

今回は LBCNN の 2 層目を構成するカーネルを対象とし，クラスタリング結果から生成した Representative Kernel を用いてカーネルを事前定義し固定した場合の認識精度を調査する．クラスタリングの基準とするカーネル間類似度を変化させならが認識精度との関係を調査した．具体的には，2 つのカーネル間において，要素の二乗誤差和をカーネルの類似度と定義し，同一クラスタに分類されてよいカーネル同士の類似度に閾値を設ける．そしてこの閾値を変化させることで，クラスタの粒度，ひいては生成される Representative Kernel の個数を変化させ，認識精度との関係を確認する．

なお LBCNN では，二乗誤差和の値が取りうる値の範囲は，比較する 2 つのカーネルの，同座標に位置するパラメータ値が全て同じ場合である 0 から，同座標に位置するパラメータ値の差が全て 2 である場合の $2^2 \times 9 = 36$ までである．そのため閾値は 0 ～ 36 の範囲で変動させた．なお，今回は LBCNN の 2 層目には 18 個のカーネルが存在する構成で評価したため，Representative Kernel の個数が 18 より少ない場合の，事前定義されないカーネルについては，従来どおり学習によりパラメータが更新されるようにした．

また，Representative Kernel の汎用性を確認するにあたり，MNIST[5] データセットで学習済のカーネルから生成した Representative Kernel を用いて事前定義したネットワークを用いて，CIFAR-10[7] および SVHN[6] を学習させた場合についても評価した．

4.2 評価環境

評価には，ディープラーニングフレームワークとして Torch を用いた．評価に用いた CNN のアーキテクチャは，**図 9** に示すような，4 層の畳み込み層を持つ LBCNN である．図 9 のオレンジ色で示す，1 層目の畳み込み層と，2 層目の畳み込み層がパラメータ固定化の対象である．1 層目を先行研究である Functionally-Predefined Kernel で，2 層目を今回クラスタリング結果から作成した Representative Kernel で，それぞれ固定する．

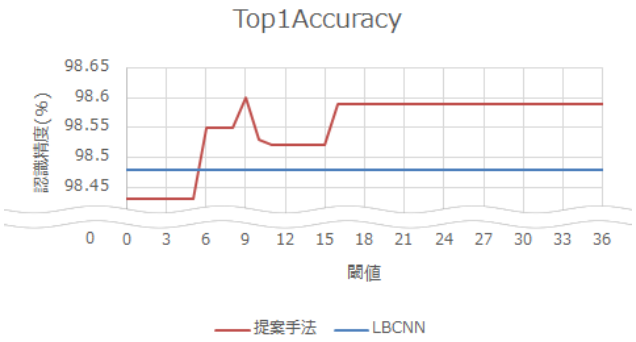

図 10 閾値と認識精度の関係 (MNIST・Top1Accuracy)

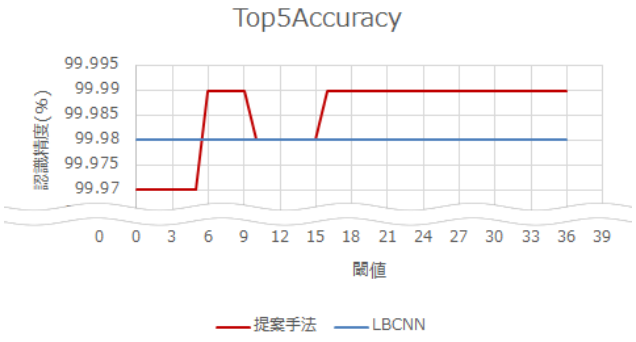

図 11 閾値と認識精度の関係 (MNIST・Top5Accuracy)

Representative Kernel を作成するために使用する学習後カーネルには，学習に必要となる計算量が小さい MNIST データセットを用いた．MNIST を用いて学習を行った CNN の，2 層目の畳み込み層から取り出した学習後カーネルに対して階層型クラスタリングを行い，カーネル類似度の閾値を 0 ～ 36 の範囲で変化させて Representative Kernel を作成し，それを用いて 2 層目のカーネルを事前定義により固定して，30 エポック学習させた結果の Top1Accuracy および Top5Accuracy を調査した．

なお，同一クラスタに含むことができるカーネル類似度の閾値を変化させても，結果としてクラスタ数ひいては Representative Kernel の数が同一となる場合があり，Representative Kernel の個数は，閾値が 0 ～ 5 の場合 18 個（2 層目のカーネル全て固定），6 ～ 8 の場合 14 個（2 層目のうち 4 個のカーネルが学習対象），9 の場合 8 個（8 個のカーネルが学習対象），10 の場合 6 個（12 個のカーネルが学習対象），11 ～ 15 の場合 4 個（14 個のカーネルが学習対象），16 ～ 36 の場合 2 個（16 個のカーネルが学習対象）となる．

4.3 評価結果

まず，MNIST の学習済カーネルから生成した Representative Kernel を用いてカーネルを固定したネットワーク

Vol.2022-ARC-248 No.12
Vol.2022-SLDM-198 No.12
Vol.2022-EMB-59 No.12
2022/3/10

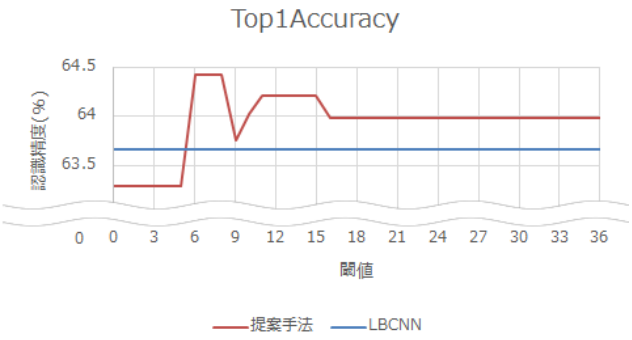

図 12 閾値と認識精度の関係 (CIFAR-10・Top1Accuracy)

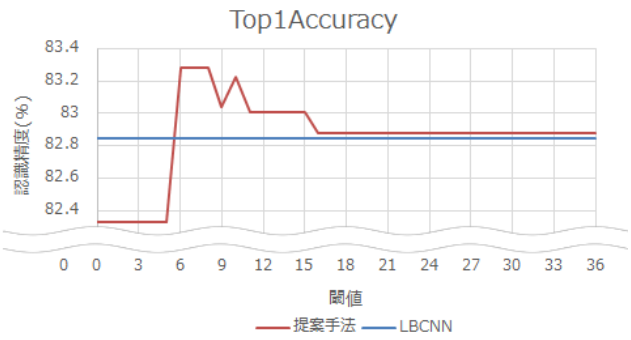

図 14 閾値と認識精度の関係 (SVHN・Top1Accuracy)

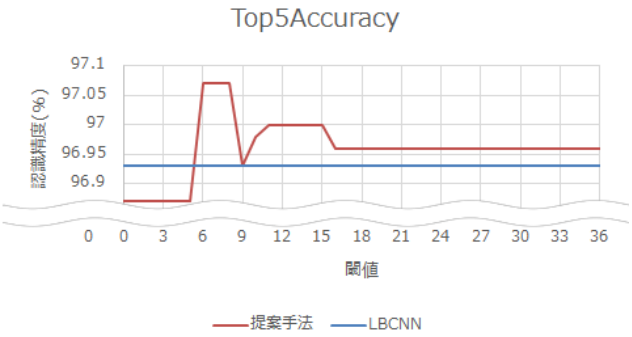

図 13 閾値と認識精度の関係 (CIFAR-10・Top5Accuracy)

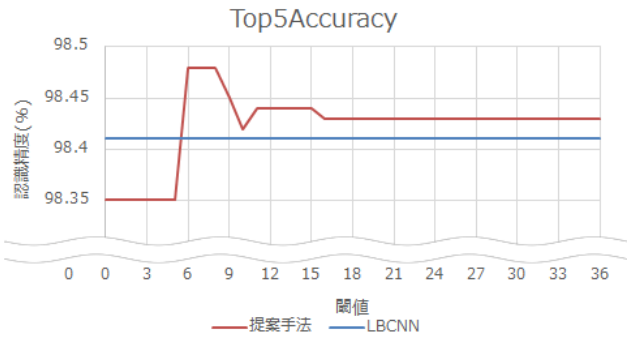

図 15 閾値と認識精度の関係 (SVHN・Top5Accuracy)

で，MNIST を学習させた結果を，**図 10** および**図 11** に示す．閾値が 5 以下，すなわち全カーネルを固定した場合を除き，LBCNN よりも高い認識精度を示していることが確認できる．

次に，MNIST の学習済みカーネルから生成した Representative Kernel を用いてカーネルを固定したネットワークで，CIFAR-10 および SVHN を学習させた結果を，**図 12**，**図 13**，**図 14**，**図 15** に示す．MNIST の結果と同様，全カーネルを固定した場合を除いて LBCNN よりも高い認識精度を示しており，MNIST の学習結果から生成した Representative Kernel が，他のデータセットに対するある程度の汎用性を持っていることが確認できる．ただし，閾値の値が大きくなるほど，認識精度の低下が見られる．この原因としては，閾値を大きくした場合，Representative Kernel の数が十分でなくカバーできるパターンが減少すること，また，クラスタの併合が進みすぎたことで Representative Kernel が学習済カーネルから乖離してしまったことが考えられる．

以上の結果から，本稿で示した方法により生成した Representative Kernel が異なるデータセットに対する汎用性をある程度保持していることが確認できた．また，今回評価した環境においては，Representative Kernel の数は多いほど認識精度が高くなること，ただし，全てのカーネルを固定すると却って精度は悪化することを確認した．Representative Kernel の数が多いということは，多くのカーネルを固定できるということであり，計算量削減の面でも都合がよい．たとえば図 12 ～ 図 15 で最も高い認識精度を示している閾値 6 ～ 8 の場合，2 層目のカーネル 18 個のうち 14 個にあたる約 78%のカーネルを学習対象から除外できる．

5. おわりに

本研究では CNN の計算量削減を目的とし，CNN の 1 層目の畳み込み層に含まれるカーネルをパラメータ固定化した先行研究に加え，2 層目以降の畳み込み層に含まれるカーネルをパラメータ固定化する方法を検討した．学習コストの小さいデータセットである MNIST を用いて学習を行った CNN の，2 層目の畳み込み層に含まれるカーネルに対して階層型クラスタリングを行い，その結果から生成した Representative Kernel を用いてカーネルを固定した場合の認識精度の変化について調査した．

調査の結果，Representative Kernel は，データセット不問で計算量を削減できるだけでなく認識精度を向上すらさせられること，また，Representative Kernel が多いほど認識精度が高くなるが，固定せず自由に学習できるカーネルも一定数必要であることを確認した．

今後は，Representative Kernel の汎用性のさらなる調査と意味的解析，また，最適な Representative Kernel 数の

決定方法などについて検討していきたい.

謝辞 本研究の一部は，JSPS 科研費 21H03408 の助成を受けたものである.

参考文献

[1] Juefei-Xu, F., Boddeti, V. N. and Savvides, M.: Local Binary Convolutional Neural Networks, arXiv:1608.06049 [cs.LG] (2016).
[2] Inouchi, Y., Yamaki, H., Miwa, S. and Tsumura, T.: Functionally-Predefined Kernel: a Way to Reduce CNN Computation, *Proc. IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing (PacRim 2019)*, pp. 1–6 (2019).
[3] Mahendran, A. and Vedaldi, A.: Understanding Deep Image Representations by Inverting Them, arXiv:1412.0035 (2014).
[4] Rastegari, M., Ordonez, V., Redmon, J. and Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks, *European Conference on Computer Vision*, Springer, pp. 525–542 (2016).
[5] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-based Learning Applied to Document Recognition, *Proc. of the IEEE*, Vol. 86, pp. 2278–2324 (1998).
[6] Netzer, Y. et al.: Reading Digits in Natural Images with Unsupervised Feature Learning, *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011* (2011).
[7] Krizhevsky, A.: *Learning Multiple Layers of Features from Tiny Images* (2009).
[8] Krizhevsky, A., Sutskever, I. and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pp. 1097–1105 (2012).
[9] Otsu, N. and Kurita, T.: A New Scheme for Practical Flexible and Intelligent Vision Systems, *Proc. IAPR Workshop on Computer Vision*, pp. 431–435 (1988).