CH-015

GridNet: アンサンブル学習に着目した画像認識のための畳み込みニューラルネットワーク

GridNet: Deep Convolution Neural Network with Ensemble Learning for Image Recognition

武田 敦志 †
Atsushi Takeda

1. まえがき

2012 年に開催された画像認識精度を競う大会 ILSVRC 2012[1] において畳込みニューラルネットワーク (CNN) である AlexNet[2] を用いたチームが優勝し，CNN が画像認識技術として有効であることが明らかとなった．この大会以降，画像認識を目的とした CNN の研究が以前よりも盛んに行われるようになり，現在までに多くの CNN が提案されている．一般的に，CNN の畳み込み層を増やせば CNN の表現力が向上するため，画像認識の精度も向上すると考えられる．そのため，多層 CNN の構成方法や学習方法の研究開発が行われ，100 層以上の CNN を用いて画像認識を行う方法が多数提案されている [3, 4, 5]．

一方，多層 CNN には過学習に陥りやすいという問題があり，画像認識精度を向上させるためには CNN の汎化能力を向上させることが重要となる [6, 7]．機械学習の過学習を防ぎ，汎化能力を向上させるための手法としてアンサンブル学習がある．アンサンブル学習は CNN の汎化能力の向上に対しても有効であり，異なる初期値から学習を行った複数の CNN の計算結果を統合することで，画像認識の精度を向上させる方法が提案されている [3, 8, 9]．また，CNN の一部のニューロンを切り離した状態で学習を行うことにより，CNN の過学習を抑制し，CNN の画像認識精度が向上することが分かっている [10, 11, 12]．これは，一部のニューロンを切り離すことが異なる条件下での学習に相当するため，アンサンブル学習と同等の効果により CNN の汎化性能が向上していると考えられる [13]．さらに，画像認識を目的とした最新の CNN では，入力と出力の間に複数の演算経路を設けることにより画像認識の精度を向上させている [14, 15, 16, 17, 18]．これらの CNN は，異なるパラメータを持つ複数の演算経路によるアンサンブル学習を行っており，その結果として高い汎化性能を有している．

そこで，本稿では，高い精度の画像認識技術を実現するため，アンサンブル学習に着目した CNN である GridNet を提案する．GridNet では，畳込み演算を行う計算ユニットをグリッド状に配置した CNN であり，入力と出力の間に複数の演算経路が存在するように設計されている．GridNet 内の個々の演算経路は異なるパラメータを用いて計算を行い，これらの計算結果を統合することで GridNet の出力を計算する．つまり，GridNet は複数の演算経路によるアンサンブル学習を行う CNN であり，過学習を抑制して高い汎化性能を持つことができる．本稿では，GridNet の詳細設計について述べ，GridNet に含まれる演算経路について考察する．また，画像認識のデータセットである CIFAR-10[19] の画像分類を目的とした GridNet の実装について説明する．さらに，この実装を用いた実験結果より，GridNet が最新技術 (state-of-the-art) の CNN と同等の画像認識精度があることを示す．

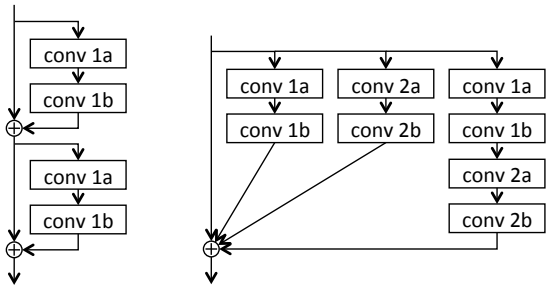
以下，2 章では画像認識を目的とした最新の CNN を紹介し，それらの CNN がアンサンブル学習の仕組みを有することを説明する．また，3 章では GridNet の詳細設計について述べ，GridNet が複数の演算経路によるアンサンブル学習を行っていることを説明する．さらに，4 章では CIFAR-10 の画像分類のための GridNet の実装を説明し，この実装を用いた実験結果から GridNet が最新の CNN と同等の画像認識精度があることを示す．最後に，5 章ではこの研究成果を考察し，今後の課題について述べる．

2. 関連研究

画像認識を目的とした CNN の研究では，CNN の畳み込み層の層数を増やすことにより，CNN の表現力を向上させることが重要であると考えられてきた．そのため，多層 CNN の構成方法に関する多くの研究が行われ，現在までに 100 層以上の CNN を用いて画像認識を行うことが可能となった [3, 4, 5]．しかし，CNN の層数を増やすだけでは画像認識精度を大きく向上させることは難しく，極端に層数を増やすよりもチャンネル数を増やす方が効果的であると報告されている [20]．これは多層 CNN が過学習に陥りやすいことが原因であると考えられるため，CNN を用いた画像認識の精度を向上させるためには高い汎化能力を有する CNN を構成する必要がある [6, 7]．

アンサンブル学習とは，複数の異なる条件下での計算結果を統合することで汎用的な計算結果を得る方法であり，機械学習分野で識別器の汎化能力を向上させる手法として用いられてきた．CNN を用いた画像認識の分野においてもアンサンブル学習は有効であり，異なる初期値からの学習を行った複数の CNN を用いることにより画像認識の精度を向上できることが分かっている [3, 8]．また，CNN の学習途中のパラメータを複数回記録し，これらのパラメータを使って計算した

†東北学院大学教養学部情報科学科
Department of Information Science, Tohoku Gakuin Univ.



(a) 2 Residual Blocks (b) Unraveled Network of 2 Residual Blocks

図 1: Residual Network の構成要素 (Residual Block) とその構成要素を展開したネットワーク

複数の結果を統合することで，単一の CNN よりも高い精度で画像を分類できることが報告されている [9]．

画像認識を目的とした最新の CNN では，1 個の CNN の中に複数の演算経路を設定しており，これら複数の演算経路がアンサンブル学習を行うことにより高い汎化性能を実現していると考えられる．Fractalnet[17] は，フラクタル構造をもつ多層 CNN であり，入力から出力までの間に独立した複数の演算経路が設定されている．また，Xception[15] や ResNeXt[16] では，入出力間の演算経路を途中で分岐させ，それぞれの分岐先に異なる畳み込み層を設定することで複数の演算経路を実現している．近年，Residual 構造 [3] を有する多くの CNN が提案されているが，この Residual 構造は複数の異なる演算経路に展開できることが分かっている [21]．例えば，図 1(a) に示す Residual 構造のネットワークは図 1(b) に示す複数の演算経路の計算結果を統合するネットワークと同等の計算を行う．このように Residual 構造が複数の演算経路を含んでいるため，Residual 構造を持つ多くの CNN[3, 5, 11, 16, 20, 22] は複数の演算経路によるアンサンブル学習の効果により高い汎化性能を実現しているものと考えられる．

3.GridNet: アンサンブル学習に着目した CNN

1 個の CNN の中に複数の演算経路を設定できれば，これら複数の演算経路がアンサンブル学習を行うことで高い汎化性能を持つ CNN を実現できる．そこで，本稿では，複数の演算経路を有する多層 CNN である「GridNet」を提案する．GridNet は，畳込み計算ユニットをグリッド状に配置した CNN であり，入力から出力までの間に複数の演算経路を有する．それぞれの演算経路では個別のパラメータに基づいて計算が行われ，その結果を統合することで最終的な計算結果を得る．このように，GridNet は複数の演算経路によるアンサンブル学習を行うため，高い汎化能力を有する CNN となっている．

図 2 に，CIFAR[19] データセットの画像分類を目的とした GridNet の構成を示す．ここで，図中の *BN* は Batch Normalization[23] を示し，*ReLU* は Rectified Linear Unit[24] を示し，*ave-pool* は Average Pooling を示す．また，図中の $\oplus$ は入力データの要素ごとの加

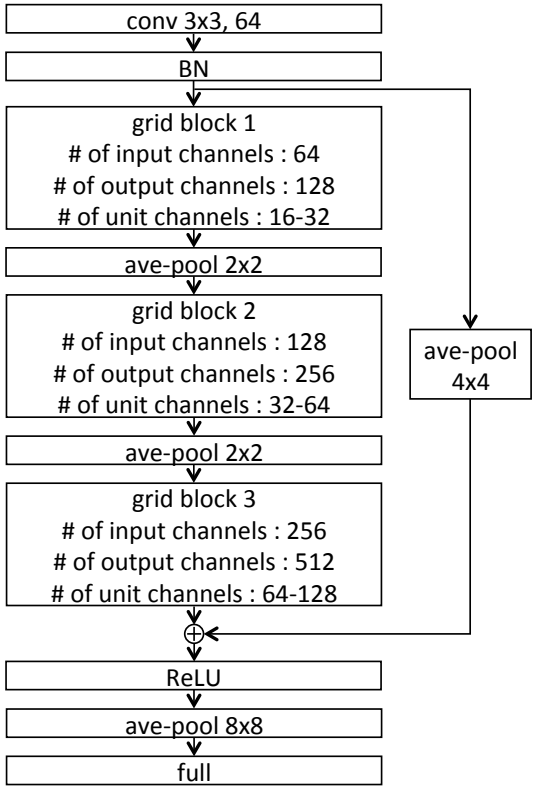


図 2: GridNet の構成

算を示す．GridNet は，Residual Network[3] と同様に，一定の計算処理を行うごとに画像サイズを縮小して畳み込み層のチャンネル数を増加させるネットワーク構成となっている．ここで，Grid Block は畳み込み計算ユニットをグリッド状に配置したネットワークであり，Grid Block の中に複数の演算経路が存在することで GridNet は高い汎化性能を実現している．

図 3 に，Grid Block の構成を示す．Grid Block は入力データを分割する Split Layer，畳込み計算を行う Grid Unit，及び，計算結果を統合する Join Layer で構成されている．Grid Block に入力されたデータは Split Layer によって分割され，分割されたデータは各 Grid Unit の入力データとなる．それぞれの Grid Unit では入力データに対する畳み込み計算が行われ，計算結果は隣接する Grid Unit と Join Layer の入力データとなる．最後に，Join Layer が Grid Unit の計算結果を統合することで Grid Block の出力データを作成する．

Grid Unit は N 次元のグリッド状に配置されており，グリッドの辺の長さを L とすると，1 個の Grid Block の中に L^N 個の Grid Unit が配置されている．それぞれの GridUnit は個別のパラメータに従って計算を行い，その Grid Unit の出力データは別の Grid Unit と Join Layer の入力データとなる．ここで，座標 $(k_0, k_1, \cdots, k_{N-1})$ に配置されている Grid Unit への入力データを $\boldsymbol{I}_{k_0,k_1,\cdots,k_{N-1}}$ とし，この Grid Unit からの出力データを $\boldsymbol{O}_{k_0,k_1,\cdots,k_{N-1}}$ とすると，その Grid Unit の順伝播の計算 f は

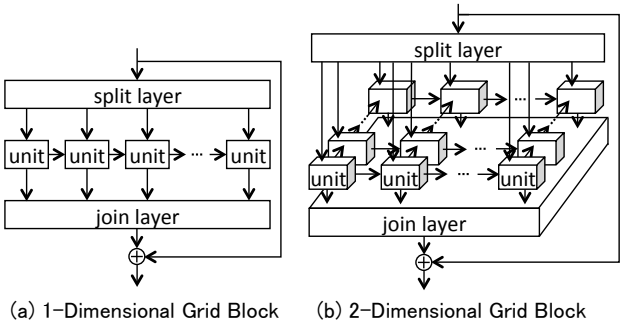


図 3: Grid Block の構成

$$
\begin{aligned}
&\boldsymbol{O}_{k_0,k_1,\cdots,k_{N-1}} \\
&\quad = f(\boldsymbol{I}_{k_0,k_1,\cdots,k_{N-1}}, \boldsymbol{\theta}_{k_0,k_1,\cdots,k_{N-1}}) \\
&\boldsymbol{I}_{k_0,k_1,\cdots,k_{N-1}} \\
&\quad = (\boldsymbol{S}_{k_0,k_1,\cdots,k_{N-1}}, \boldsymbol{M}_{k_0,k_1,\cdots,k_{N-1}}) \\
&\boldsymbol{M}_{k_0,k_1,\cdots,k_{N-1}} \\
&\quad = \{\, \boldsymbol{O}_{k_0,\cdots,k_m-1,\cdots,k_{N-1}} \mid 0 \le m < N \wedge k_m > 0 \,\}
\end{aligned}
$$

となる．ここで，$\boldsymbol{\theta}_{k_0,k_1,\cdots,k_{N-1}}$ は各 Grid Unit のパラメータであり，$\boldsymbol{S}_{k_0,k_1,\cdots,k_{N-1}}$ は Split Layer からの入力データであり，$\boldsymbol{M}_{k_0,k_1,\cdots,k_{N-1}}$ は他の Grid Unit からの入力データである．順伝播のときは，座標 $(0,0,\cdots,0)$ に配置されている Grid Unit から順伝播の処理を実行することで，すべて順伝播の計算のを矛盾なく実行できる．また，逆伝播のときは，座標 $(L-1,L-1,\cdots,L-1)$ の Grid Unit から逆伝播の処理を実行することで，すべての Grid Unit に誤差情報を伝播させることが可能である．

図 4 に Grid Block の各要素の詳細を示す．ここで，図中の *conv* は畳み込み層を示し，*mean* は要素ごとの平均計算を示す．Grid Block は Grid Unit の畳み込み層 (conv 3x3) で画像認識のための畳込み計算を行う仕組みとなっている．この畳み込み層は Grid Unit の数だけ存在するが，効果的にアンサンブル学習を行うためには，各 Grid Unit の畳込み層のチャンネル数が異なり，それぞれが異なった特性について計算することが望ましい．そこで，それぞれの畳み込み層のチャンネル数が異なるように設定するため，N 次元の Grid Block の座標 $(k_0,k_1,\cdots,k_{N-1})$ に配置された Grid Unit の畳み込み層の入力チャンネル数 $s_{k_0,k_1,\cdots,k_{N-1}}$ と出力チャンネル数 $d_{k_0,k_1,\cdots,k_{N-1}}$ を

$$
s_{k_0,k_1,\cdots,k_{N-1}} = c_{min} + (c_{max} - c_{min}) \frac{\sum k_i}{1 + N(L-1)}
$$

$$
d_{k_0,k_1,\cdots,k_{N-1}} = c_{min} + (c_{max} - c_{min}) \frac{1 + \sum k_i}{1 + N(L-1)}
$$

とする．ここで，c_{min} は 1 個の Grid Block に含まれる Grid Unit の最低チャンネル数であり，c_{max} は最大チャンネル数である．1 個の Grid Block の中に様々なチャンネル数の Grid Unit を混在させることにより，GridNet の汎化性能を向上させることができる．Split Layer の出力データはすべての Grid Unit の入力データとなるため，Split Layer の出力チャンネル数は $\sum s$ となる．また，すべての Grid Unit の出力データは Join Layer の入力データとなるため，Join Layer の入力チャンネル数は $\sum d$ となる．

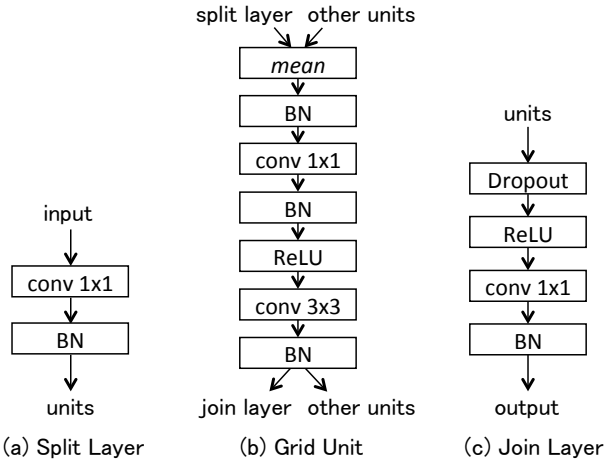


図 4: Grid Block の各要素の詳細

Grid Block には入力と出力の間に複数の演算経路が存在する．図 5 に $N=1, L=4$ の Grid Block に含まれる演算経路を示す．例えば，$N=1, L=4$ の Grid Block の場合，1 個の Unit を通過する演算経路が 4 通り，2 個の Unit を通過する演算経路が 3 通り，3 個の Unit を通過する演算経路が 2 通り，4 個の Unit を通過する演算経路が 1 通り存在する．これらの演算経路の計算結果を Join Layer で統合したものが Grid Block の出力となる．そのため，これらの演算経路のアンサンブル学習を効果的に行い，GridNet の汎化性能を向上させるためには，画像認識の精度を向上に寄与する演算経路が多く存在するように Grid Block の次元数 N とグリッドの辺の長さ L を設定しなくてはならない．

表 1 に，Grid Block の次元数 N と Grid Block に含まれる演算経路の数の関係を示す．ここでは，Grid Unit の数が常に 16 個となるよう L の値を設定している．Grid Block の次元数が増えると，少数の Grid Unit のみを通過する浅い演算経路の数が増加する．一方，Grid Block の次元数の増加にともない，多数の Grid Unit を通過する深い演算経路の数が減少する．例えば，$N=1, L=16$ の Grid Block には 16 個の Grid Unit を通過する演算経路が存在するが，$N=4, L=2$ の Grid Block には 6 個以上の Grid Unit を通過する演算経路は存在しない．現在までの研究より，CNN による画像認識の精度を向上させるためには，浅い演算経路によるアンサンブル学習が重要ではあるが，深い演算経路による計算も必要だと考えられる [21]．つまり，アンサンブル学習により GridNet の汎化性能を向上させるためには，浅い演算経路と深い演算経路の両方がバランス良く存在する N と L の値を設定する必要がある．

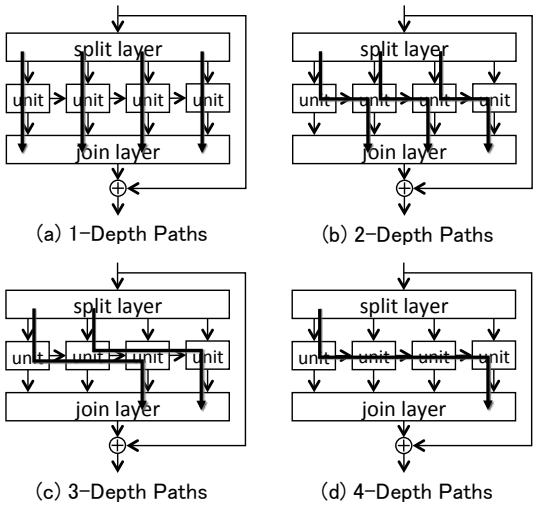


図 5: Grid Block($N = 1, L = 4$) に含まれる演算経路

表 1: Grid Block に含まれる演算経路の個数

	# of paths				
	depth=1	2	3	4	5
N=1, L=16	16	15	14	13	12
N=2, L=4	16	24	34	44	48
N=4, L=2	16	32	48	48	24

4. 実験評価

ニューラルネットワークの実装フレームワークである Chainer 2.0* を用いて図 2 で示した GridNet を実装し，この実装を用いて GridNet の画像認識精度の評価実験を行った．この実験では，GridNet の画像分類精度を計測するため，画像のデータセットである CIFAR-10[19] を使用した．CIFAR-10 は大きさが 32x32 のカラー画像のデータセットであり，それぞれの画像が 10 種類のカテゴリに分類されている．CIFAR-10 には 60,000 個の画像データが存在するため，50,000 個の画像データを学習用データとして，残りの 10,000 個の画像データをテスト用データとして使用した．GridNet の学習では，MSRA[25] でパラメータを初期化し，Momentum SGD(momentum=0.9, weight decay=1e-4) でパラメータの更新を行った．また，学習時のミニバッチサイズは 128 であり，300 epoch の学習を実施した．学習開始時の学習係数は 0.1 であり，150 epoch と 225 epoch に学習係数をそれぞれ 0.01 と 0.001 に変更した．さらに，学習効率を向上させるため，学習用の画像データに対していはランダムサンプリングと反転を行い，学習用の画像データを増加させている [3, 5, 16]．以上の GridNet の実装と GPU(NVIDIA Geforce 1080Ti) を用いて CIRAR-10 の画像分類タスクの実験を行い，GridNet の画像認識精度を検証した．

まず，Grid Block の次元数と画像認識精度の関係を評価するため，Grid Block の次元数 N を $1, 2, 4$ に変更

*https://chainer.org

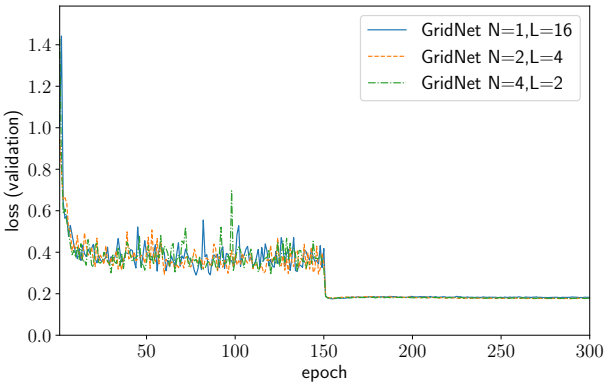


図 6: CIFAR-10 の画像識別タスクの識別誤差

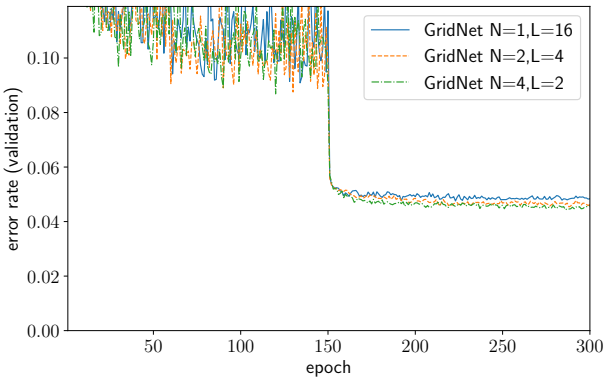


図 7: CIFAR-10 の画像識別タスクの識別エラー率

したときの画像認識精度を測定した．ただし，次元数によって Grid Unit の数が変化しないようにするため，$L^N = 16$ となるように L の値を設定した．このとき，それぞれの GridNet に含まれるパラメータ数は 3.7M となった．図 6 にテスト画像の識別誤差 (正解信号との softmax-cross-entropy) を示し，図 7 にテスト画像の識別エラー率を示す．また，Grid Block の次元数を変化させたときのテスト画像の分類精度を表 2 にまとめる．同数の Grid Unit を持つ GridNet の場合，Grid Block の次元数 N を増やすことで画像識別精度を向上する結果となった．これは，3 章で述べたように，Grid Block の次元数を増やすことで入力から出力までの演算経路の数が増加し，これらの演算経路によるアンサンブル学習が有効に機能したため，画像識別精度が向上したと考えられる．一方，Grid Block の次元数を増やすと，多くの畳み込み演算を行う深い演算経路の数が減少するという問題がある．しかし，いくつかの CNN を用いた画像識別の研究 [20, 21] より，CIFAR-10 の画像分類精度を向上するためには深い演算経路よりも浅い演算経路の方が重要であることが分かっている．そのため，この実験結果でも，深い演算経路を多く持つ $N = 1$ の GridNet よりも，浅い演算経路を多く持つ $N = 4$ の GridNet の方が正確に画像を分類できたと考えられる．

表 2: 次元数と画像分類精度 (CIFAR-10) の関係

model	error (%)
GridNet (N=1,L=16)	4.76
GridNet (N=2,L=4)	4.54
GridNet (N=4,L=2)	4.43

表 3: 画像分類精度 (CIFAR-10) の比較

model	# of params	error (%)
PyramidNet-110 [5]		
$\alpha = 84$	3.8M	4.26
$\alpha = 270$	28.3M	3.73
PyramidNet-164 [5]		
$\alpha = 270$ (bottleneck)	27.0M	3.48
ResNeXt-29 [16]		
8x64d	34.4M	3.65
16x64d	68.1M	3.58
Shake-Shake-26 [12]		
2x32	2.9M	3.55
2x96	26.2M	2.86
GridNet (proposal)		
N=2,L=4,dropout=0.0	3.7M	4.54
N=3,L=4,dropout=0.0	14.6M	3.92
N=3,L=4,dropout=0.2	14.6M	3.57

次に，GridNet と最先端 (state-of-the-art) の CNN との画像認識精度の比較を行った．パラメータ数が 14.6M の GridNet(N=3,L=4,Dropout=0.2) のエラー率は 3.57%となった．これは 2016 年の最先端の CNN(PyramidNet,ResNeXt) と同等の低いエラー率であり，GridNet が他の最先端 CNN に匹敵する画像認識精度を有していることを示している．一方，2017 年に提案された CNN である Shake-Shake[12] は 3%以下のエラー率を達成している．これは，Shake-Shake が従来とは根本的に異なる学習アイデアを導入しているためであり，この学習アイデアは GridNet にも応用可能である．2017 年以降に報告されている新しい学習方法を GridNet に応用することにより，GridNet のエラー率をさらに改善できると考えられる．

5. むすび

近年の研究報告により，CNN を用いた画像認識の精度を向上させるためには CNN の汎化能力を向上させることが重要であることが判明した．そこで，本稿では，アンサンブル学習を行うことにより高い汎化能力を持つ GridNet を提案した．GridNet は，畳込み演算を行う計算ユニット (Grid Unit) をグリッド状に配置した CNN であり，入力と出力の間に複数の演算経路が存在するように設計されている．GridNet ではこれらの複数の演算経路によるアンサンブル学習が行われるため，その結果として GridNet は高い汎化能力を有する CNN となっている．また，本稿では，CIFAR-10 のデータセットを用いた実験を通じて，GridNet の画像認識精度が最先端の CNN と同等であることを確認した．

今後の課題としては，GridNet の構成要素の改良が挙げられる．GridNet を構成する要素として Split Layer，Grid Unit，及び，Join Layer があるが，これらの構成要素にとって最も良いと思われる計算手順は分かっていない．Residual Network では，Residual Block の計算手順を変更することで画像認識精度を大幅に向上できることが報告されている [3, 4]．GridNet においても，それぞれの構成要素の計算手順を改良することで，現在よりも画像認識精度を向上できると考えられる．また，近年，Stochastic Depth[11] や Shake-Shake[12] などの従来とは異なる CNN の学習手法が提案されている．GridNet にこれらの学習手法を適用することによって，GridNet の画像認識精度をさらに向上できると考えられる．

本稿では，CIFAR-10 のデータセットを用いて GridNet の画像認識精度の評価を行った．ただし，本稿で示した実験結果は 1 回の試行のみの結果であり，より正確に画像認識精度の評価を行うためには複数回の試行の平均値を用いる必要がある．GridNet の実験回数を増やし，より正確な評価を行うことも今後の課題である．また，CIFAR-10 以外の画像認識評価のためのデータセットとして，CIFAR-100 や ImageNet などの画像データセットが公開されている．これらのデータセットを用いることで，より多くの側面から GridNet の画像認識精度を評価することも今後の課題として挙げられる．

参考文献

[1] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[5] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. *arXiv preprint arXiv:1610.02915*, 2016.

[6] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[7] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.
[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.
[9] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.
[10] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
[11] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, pages 646–661. Springer, 2016.
[12] Xavier Gastaldi. Shake-shake regularization of 3-branch residual networks. In *In the 5th International Conference on Learning Representations Workshop*, 2017.
[13] Pierre Baldi and Peter Sadowski. The dropout learning algorithm. *Artificial intelligence*, 210:78–122, 2014.
[14] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.
[15] François Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint arXiv:1610.02357*, 2016.
[16] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016.
[17] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.
[18] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.
[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Tech Report*, 2009.
[20] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.
[21] Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Advances in Neural Information Processing Systems*, pages 550–558, 2016.
[22] Ke Zhang, Miao Sun, Xu Han, Xingfang Yuan, Liru Guo, and Tao Liu. Residual networks of residual networks: Multilevel residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.
[23] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.
[24] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.
[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.