特　集「ニューラルネットワーク研究のフロンティア」

画像認識のための深層学習の研究動向
―畳込みニューラルネットワークとその利用法の発展―

Research Trend in Deep Learning for Visual Recognition
― Advances of Convolutional Neural Networks and Their Use ―

岡谷　貴之　東北大学大学院情報科学研究科
Takayuki Okatani　Graduate School of Information Sciences, Tohoku University.
okatani@vision.is.tohoku.ac.jp, http://www.vision.is.tohoku.ac.jp/

Keywords: deep learning, convolutional neural network, image recognition.

1. はじめに

深層学習の登場はコンピュータビジョンに革命をもたらした．2012年の画像認識のコンテストILSVRCの結果は多くの研究者に衝撃を与え[Krizhevsky 12]，それ以来，畳込みニューラルネットワーク（Convolutional Neural Network，以下CNN）とその応用は，分野の中心的な研究テーマになった．

衝撃が大きかった理由はいくつかある．第一にその高い性能である．物体カテゴリー認識をはじめ，いくつかの画像認識のタスクで人と同じか人を上回る認識精度を達成している[He 15b, Russakovsky 14, Taigman 14]．さらに驚くべきは，それまでの常識を超える大規模な学習―100万を超える学習サンプルを用いて，1億以上の数のパラメータをもつCNNを，最新のGPUを使って数日以上かけて学習する―によりその性能が実現されていることである．

CNNの有効性は幅広い問題に及ぶこともわかり，日々応用範囲が拡大している．その一方で多くの疑問も残されている．例えば高い性能をあげられる理由，性能と学習方法やネットワークのデザインとの関係，数多ある画像認識や画像処理のどれに有効で有効でないか，などである．転換点となった2012年からわずか3年あまりの間に，数えきれない多くの研究が行われ，論文が出版された．本稿ではそのうち重要なものを紹介し，今後の発展の方向を議論する．

2. 畳込みニューラルネットワーク

2·1 基本構造

CNNは多層のフィードフォワードネットで，画像を受け取る入力層から，畳込み層とプーリング層を何度か繰り返した後，何層かの全結合層を経て出力に至る構造を基本とする．

畳込み層では，入力される画像にフィルタを畳み込む演算を行う．畳込み層の入力および出力はともに画像のフォーマットをもち，三次元配列で表される．そのサイズを$W\times H\times K$と表記すると，WおよびHは画像の縦横サイズ（画素数），Kはチャネル数である．例えば640×480のカラー画像を入力に受け取る最初の畳込み層の入力は$W\times H\times K=640\times480\times3$となる．中間にある畳込み層の入力は後述のプーリングにより，層を経るにつれWおよびHは小さく，ただしKは数十から数百程度となる．畳み込むフィルタは入出力と同様の三次元配列で，$P\times P\times K$の形をもつ．縦横サイズPは入力のサイズ（W, H）より普通ずっと小さく，例えば$P=3$から11程度であり，チャネル数（K）は入力と一致する．このようなフィルタを一つの畳込み層で複数個利用する．

第l層にある畳込み層がM個のフィルタをもち，$W\times H\times K$の配列z^{l-1}_{ijk}を入力に受け取るとする．全フィルタの要素をまとめて$h_{pqkm}(p, q=0\cdots P-1, k=0\cdots K-1, m=0\cdots M-1)$と書くと，この畳込み層の1ユニットの出力$z^{l}_{ijm}$は，次のようにまずユニットへの総入力$u^{l}_{ijm}$を

$$u^{l}_{ijm}=\sum_{k=0}^{K-1}\sum_{p=0}^{P-1}\sum_{q=0}^{P-1}z^{l-1}_{i+p,j+q,k}h_{pqkm}+b_m \qquad (1)$$

と計算した後，

$$z^{l}_{ijm}=f(u^{l}_{ijm}) \qquad (2)$$

のように活性化関数を適用して得られる．活性化関数にはrectified linear関数すなわち$f(u)=\max(u, 0)$が普遍的に使われている．こうして得た出力z^{l}_{ijm}のサイズは$W\times H\times M$になる．また，すべての位置(i, j)で積和を計算する代わりに一定間隔（ストライドと呼ぶ）s，つまり$i, j=0, s, 2s, \cdots$で行う場合もある．この場合，出力配

列の縦横サイズは(約)$1/s$倍に縮小され，$\lfloor W/s \rfloor \times \lfloor W/s \rfloor \times M$となる．

プーリング層は畳込み層の直後に配置される．畳込み層と同様のフォーマットの入出力をとり，入力を縦横方向に間引く働きをもつ．例えば2×2の最大プーリングでは，$W\times H\times M$の入力を縦横方向に2×2の小領域に分割し，各領域内の出力の最大値を一つ選んで出力する．小領域は互いに重複するようにとることもでき，隣接小領域のストライドsが間引き率($1/s^2$)を決める．なおプーリングは通常，チャネルごとに独立に行う．したがってプーリング層の入出力間でチャネル数は不変である．入力サイズを$W\times H\times M$，ストライドをsとすると出力のサイズは$\lfloor W/s \rfloor \times \lfloor H/s \rfloor \times M$となる．各小領域の代表値を選ぶ選び方にはバリエーションがあり，最大値を選ぶ最大プーリング（max pooling）が最も一般的である．平均プーリングが一部（後述の出力段の全体プーリングなど）で使われる程度である．

畳込み層とプーリング層の繰返しの後には全結合層―2層の全ユニット間に結合をもつ層―を何層か配置し，出力層に接続するのが一般的である．最後の出力層は，通常のフィードフォワードネット同様，目的に応じて設計する．すなわち多クラス分類ではクラス数と同数のユニットを配置し，活性化関数にソフトマックス関数を用い，各ユニットの出力を対応クラスの尤度とみなす．回帰の場合，目的変数と同じ数のユニットを配置し，適切な値域をもつ活性化関数を選ぶ．

畳込み層での重み（パラメータ）の数と計算量は次のとおりである．サイズ$L\times L\times K$のフィルタをM個もち，入力サイズ$W\times H\times K$，出力サイズ$W\times H\times M$の畳込み層を考える．この層の重み（パラメータ）の数はフィルタの要素数に一致し，つまりKL^2Mである．式(1)に示した演算を全i,j,mについて行うとき，これは$M\times(KL^2)$の行列と$(KL^2)\times WH$の行列の行列積として表せる．したがって，畳込みに要する積の演算回数はKL^2MWHになる．つまり畳込み層では通常，重みの数より演算回数のほうがずっと大きい．これは全結合層と対照をなし，入力層Kユニットと出力層Mユニットの全結合層では，重みの数と演算数はともにKMとなる．

2·2 学　　習

CNNの学習は通常のフィードフォワードネットの場合とほとんど変わらない．クラス分類では上述のように出力層を設計しておいて，分類誤差に交差エントロピーを選び，これをパラメータの目的関数として最小化を行う．最小化には確率的勾配降下法（SGD：Stochastic Gradient Descent）を使うのが一般的である．上述のようにCNNで可塑性をもつ（学習で修正される重みをもつ）層は畳込み層と全結合層の2種類あるが，いずれの層の重みもその勾配の計算には，誤差逆伝播法（back-propagation）を用いる．SGDによる重みの更新の際には，モメンタムや重み減衰（重み＝フィルタの2乗和）などを一般に用いる．

以上の学習を，よりうまく行うためのトリックが多数提案されている．それらはCNNのためだけに考案されたわけではないが，CNNでも使えるものがほとんどである．なお，深層学習の端緒をつくったともいえる事前学習は，音声認識で一般的な全結合層のみからなる多層ネットワークでは高い有効性をもつが，CNNではもともと不要であり，ほとんど使われない．

一つ目は重みの初期化の方法である．重みの初期値は普通，ガウス分布に基づいてランダムに生成した値をセットする．ガウス分布の平均は0でよいが，分散の選択は極めて重要で，不適切な決定は学習が進まない要因になる．最もオーソドックスな決め方は，対象とする入力画像サンプルの集合に対し，各中間層の出力の値の分布の分散が一定になるように決める方法である[Glorot 10]．最近，rectified linear関数をターゲットにした方法が提案され[He 15b]，それ以前は難しかった30層ほどの多層CNNの学習が可能になった．関連して，入力サンプルの集合に対する中間層の出力の共変量シフト（covariate shift）[Shimodaira 00]を小さくする，バッチ正規化（batch normalization）と呼ばれるトリックが提案され[Ioffe 15]，その高い有効性から一般化しつつある．

さらに最近，残差学習（residual learning）[He 15a]という方法が提案され，学習可能なCNNの層数を飛躍的に増やせるようになった．この方法は，ネットワーク内の数層を一括りにした部分ネットを考え，その入出力間にショートカットを挿入するというものである(図1)．部分ネットの出力側では，本来の部分ネットの出力にショートカットした部分ネットの入力を加算し，これが次の層の入力になる．ネットワークの中間層をこのような部分ネットに切れ目なく分割し，各部分ネットの入出力間に同様のショートカットを与える．基本的には以上の拡張に，上述の重みの初期化およびバッチ正規化を同時利用するだけで，実に100～1 000層にも及ぶ多層ネットの学習がきちんと行えるようになることが示された．

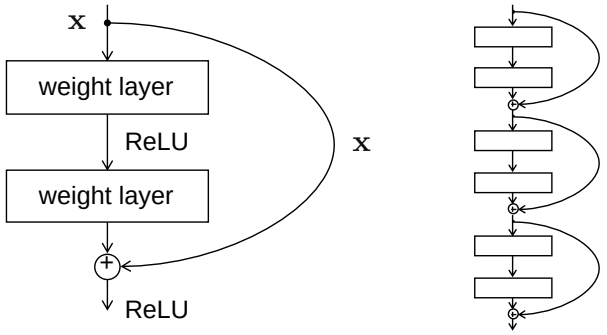

図1　残差学習 [He 15a] の概要．
左：数層をまたいだショートカットの構造．右：全体の構造．ResNet はこの構造をもつ極めて多層（100～1 000）のネットワークである

この方法を使って最適化された152層のCNN（ResNet）は，ILSVRC 2015の物体カテゴリー分類など各部門の勝者となった．

ドロップアウト―学習時，各層のユニットをミニバッチ単位で一定の確率でランダムに選んで無効化する―はCNNでも有効であることが知られている [Srivastava 14]．畳込み層やプーリング層にも原理的には適用可能だが，全結合層と入力層のみに用いるのが良いようである．後述のように，CNNの後段には大きな全結合層を置かず，大域平均プーリング層を使って直接出力層につなげるのが最近のCNNのトレンドであり，CNNでのドロップアウトの使用頻度は今後減っていくと思われる．

SGDによる学習時，重みの更新はミニバッチ（＝数百程度の学習サンプル）単位で行う．そのミニバッチのサイズは，利用する計算機システムのメモリ容量や並列演算装置の効率を考慮しつつ決定するが，結果的にモメンタムや学習率（learning rate），重み減衰のハイパーパラメータの選択にも影響する．これらを（バッチサイズに応じて）どう選ぶべきかにはいくつかの哲学があり得て，それらに基づいて理論的に決めることもあるが，経験に頼っている状況である．[Krizhevsky 14] には，バッチサイズを k 倍するとき，学習率は k 倍，モメンタムおよび重み減衰は変更しないという経験則が示されている．

以上とは別に，最適化方法そのものの改良も各種提案されており，AdaGrad [Duchi 11]，RMSProp，ADAM [Kingma 14]，自然勾配 [Desjardins 15] などがあるが，詳細は省略する．

2·3 構造の設計方法

CNNの構造をどのように決めたら良いか，つまり層数や各層のフィルタ数をどう決めると良いかについては，定見はまだないと言ってよい．ただし，少なくとも物体カテゴリー認識を例にとれば，年々層数が増える傾向にあり，性能向上はそこに負うところが大きいように思われる．では，ネットワーク全体の自由度を一定にしたとき，層数は多ければ多いほど性能は向上するのだろうか？ 層数を増やそうとすると通常，学習はより難しくなるため，純粋に層数と認識性能の効果を実験的に確かめることは難しい．

上述した残差学習は，極めて多層のCNNの学習を可能にし [He 15a]，この難しさを打ち破った可能性がある．上述のようにILSVRC 2015では，152層のCNNが最も性能が良かったという．また同じ論文で，より小さな画像サイズの物体認識のベンチマークテストであるCIFAR-10/100を対象に，1 000層を超えるCNNの学習が試され，層数と認識性能の関係が実験的に評価されている．結果は110層のCNNが最も高精度で，1 000層のものよりも良く，必ずしも層数が多ければ多いほど性能が高くなるわけではないということであった．この研究以前に，同様の問に実験的に答えを見いだそうとした研究に [Eigen 14, He 15b] がある．

なお，ニューラルネットワークの層数とその性能の関係については，以前から多くの議論がある．中間層を1層しかもたない2層のネットワークであっても，中間層のユニット数を自由に増やせるならば，任意の関数を表現できること [Cybenko 89, Hornik 89] が，以前からよく知られている．近年，Montufarらは，rectified linear関数を活性化関数にもつ多層ネットワークの表現力に関する研究を行っている [Montufar 14]．rectified linear関数の性質から，このネットの入出力関係は入力の空間の局所領域で線形となる．この研究は，そのような局所領域の数が層数に対し指数的に増加することを導いた．つまり同じユニット数の浅いネットと深いネットでは，後者が圧倒的に効率良く，目標とする関数を表現し得ることを示唆する．つまり潜在的には，層数が多いほどニューラルネットは高い表現力をもち得るということである．

一方で，通常の方法で学習した多層のネットワークは，高い冗長性をもつことも知られるようになっている．すなわち，学習済みの重みは冗長であり，学習後に圧縮可能である [Jaderberg 14, Neyshabur 15]．また層数についても同様であり，知識抽出（knowledge distillation）という考え方により，学習済みの大規模モデルを，より小規模なモデルで置換することが可能である [Hinton 15]．これを上述の多層性の利点と考え合わせると，現在の多層ネットワークの学習方法に，改善の余地があることを示唆するといえる．

2·4 代表的なモデル

近年のCNNの発展は，ImageNetの画像サンプルを利用した物体カテゴリー認識のコンテストILSVRCによってもたらされた．後述する転移学習の考え方により，このILSVRC用にデザインされデータを学習したモデルが，いろいろな画像認識のタスクに流用されるようになり，CNNの応用範囲は広まった．Caffe [Jia 14] に代表される深層学習の計算ライブラリが一般化し，ネットワークの構造そのものを記述したファイルが，ILSVRCの物体認識タスクを学習した重みと一緒に頒布されるなど，誰でも簡単に使えるようになった．こうした利用がなされている代表的なCNNに，AlexNet [Krizhevsky 12]，VGGNet [Simonyan 14b]，GoogLeNet [Szegedy 15] がある．

AlexNetはCNNが物体カテゴリー認識に極めて有効であることを最初に示したモデルである [Krizhevsky 12]．その後，AlexNetに代わる新しいモデルの探求が行われ，大幅に認識性能を向上したモデルがいくつか提案された．その代表格がSimonyanらが発表したVGGNetと，SzegedyらのGoogLeNetである．両者に共通するAlexNetとの差は，層数を増やした（8層→

表1 代表的なモデルのパラメータ数および演算回数．畳込み層および全結合層での合計と総計

モデル		AlexNet	VGGNet	GoogLeNet	ResNet
畳込み	層	5	13	21	151
	重み	380万	0.15億	580万	−
	演算	10.8億	153億	15億	113億
全結合	層	3	3	1	1
	重み	0.59億	1.24億	100万	200万
	演算	0.59億	1.24億	100万	200万
合 計	重み	0.62億	1.38億	680万	−
	演算	11.4億	155億	15億	113億

20層前後）こと（表1），および局所正規化層（Local Response Normalization：LRN）を廃したことである．LRN（あるいはLocal Contrast Normalization [Jarret 09]）は従前は重要な役割を果たすと考えられていたが，今では（認識性能のみを考える限り）必要ないという共通認識である．

VGGNetはAlexNetの構造をベースに規模（層数）を拡張したものと見ることができ，16層と19層のモデルが良く用いられている（表は16層のモデル）．畳込み層のフィルタサイズを全層通して3×3に統一し，これを2〜3層積み重ねる構造をとった点に新しさがある．AlexNetでは例えば第1層で11×11という大きなフィルタが使われているが，この1層を3×3畳込み層の2〜3層の積み重ねで置換する．この二つを比べたとき，出力ユニットの受容野のサイズは同じままで，多層化による表現能力向上と，重み数の削減が利点であるとされる．また，目的の多層CNNを学習するために，層数の少ないモデルをまず学習し，その後新たに層を追加し再学習する方法をとる．表1に示すとおり，VGGNetは非常に規模が大きいが，以上の構造の均質性と高い性能から，現在最も広く利用されている．

なお，AlexNetやVGGNetでは，重みの数では全結合層が全体の9割を占める一方，演算回数では畳込み層が全体の9割以上を占めており，いびつな構造をもつといえる．GoogLeNetはこのいびつさを解決しており，その結果サイズは極めて小さく，AlexNet比でも重みの数は12分の1しかない．しかし層数は22とVGGNetよりも多く，また性能で上回る．

GoogLeNetの最大の特徴は，Inceptionモジュールと呼ぶ1×1，3×3，5×5畳込み，およびプーリングを同じ層で並行して行う構造を単位に，これを積み重ねた構造をもつことである．Inceptionモジュールの3×3と5×5畳込みの直前（およびプーリングの直後）には，1×1畳込み層が挿入してあり，その後の畳込み層の入力チャネル数を削減し，重みの数と演算回数を削減している．このInceptionモジュールとプーリング層を積み重ねて全体が構成される．また上述のようにGoogLeNetは，AlexNetやVGGNetのような大きな全結合層をもたない．最後の畳込み層の出力は，全体平均プーリングでまとめられ，カテゴリー尤度を出力する出力層に結合される．畳込み層部分の構造を凝ったものとし，それによって表現力を高めることで，後段の全結合層が不要になっていると見ることができる．なお，論文に記載のモデルではこの間に全結合層が挿入されているが，主に転移学習などでの利用を意図したもので，認識精度向上への寄与は小さいとされている．これはILSVRC 2015の勝者となったResNetでも同じである．最終畳込み層の出力は全体プーリングでまとめられ，一層の全結合層を経て出力層につながる．またResNetでは最初の畳込み層の後，3×3でストライド2の最大プーリングを一度行う以外，プーリング層が入っていない．後述のAll-CNNのようなストライド2の畳込み層でプーリング層を代替している．

3. 転移学習・ファインチューニング

転移学習（transfer learning）とは，ある特定の問題を解くために獲得した知識を，異なるが関連はある問題に適用する機械学習の一般的方法である．ILSVRCのカテゴリー認識（classification）のタスクを学習したCNNは，この転移学習に向くことがよく知られている [Donahue 14, Oquab 14, Razavian 14]．すなわち，ILSVRC向けに学習済み（pretrained）のAlexNetやVGGNetを，例えばシーン認識 [Xiao 14]，ポーズ推定 [Toshev 14]，テクスチャマテリアル認識 [Cimpoi 15] などのタスクに流用できる．

流用とは，そのネットのアーキテクチャのみならず，学習で得た重み（の一部）を，手元のタスクで使用することである．この方法が強力なのは，目的タスクの学習サンプルがあまりたくさん用意できない場合でも，ImageNetの100万以上のサンプルから学習した知識（＝学習した重み）の助けを借りて，高い予測（分類）性能を実現できてしまうことである．

転移学習の最も簡単な方法は，学習済みのCNNをそのまま使い，目的の画像をそのCNNに入力して得られる中間層の出力を，その画像から抽出した「特徴」とみなす方法である．こうして取り出した特徴は，例えばサポートベクタマシンなどの適当な分類器に入力し，目的とする分類を行う．通常，AlexNetやVGGNetの全結合層を一つ選び，その層の出力を全部合わせたものを特徴ベクトルとみなす．後段の全結合層の出力を特徴ベクトルとすることが多いが，その成分数は数千（通常4 096）個程度しかなく，深層学習以前の古典的な画像特徴と比べてもコンパクトである．この場合，SVMを学習できる程度のサンプルがあればよく，学習データはかなり少なくてもうまくいく．

より進んだ方法は，目的タスクの学習サンプルを用いて学習済みのCNNを再学習する，いわゆるファインチューニングである（図2）．学習済みCNNの上位層

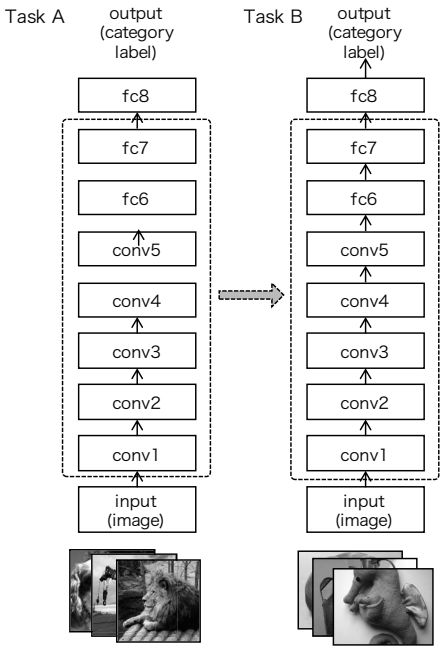

図 2　転移学習・ファインチューニングの概要．タスク A（典型的には ILSVRC の分類タスク）を学習した CNN を重みとともにコピーした CNN を使って，タスク B の学習を行う

だけを目的タスク用に設計し直し，下位層は重みを含めてそのままとしておき，これを新たなタスクについて通常の方法（交差エントロピー誤差の SGD による最小化）で再学習する．タスクが違うので，少なくとも出力層は新設し，その重みはランダムに初期化することになる．例えば目的とするタスクが 10 クラス分類なら，AlexNet の出力層のユニット数を 1 000 から 10 にする．

再学習のとき，すべての層の重みを均等に学習によって変化させる必要はない．上位層から何層かを選び，それらのみを更新し，下位層はそのままにするか，あるいは層の深さに応じて学習率に差をつける．上から何層まで再学習するかは，利用できる手元のサンプル数に応じて決める．更新する層数が多くなれば，その分モデルのパラメータが増えるわけで，見合った（過適合を生じさせない）量の学習サンプルを要する．

このように上位層から順に選ぶのは，CNN（およびすべての多層ニューラルネット）に対する仮説（あるいは経験知）に基づく．すなわち，下位層では単純な特徴が，上位層ではより複雑な特徴が抽出され，そこには階層構造があり，上位層の高度な特徴抽出は下位層のより単純な特徴の「再利用」により実現されている，という見立てである．そうであれば，学習済みのタスク内では，基礎的で何度も繰り返し使われる特徴ほど，下位層に存在することになる．そんな下位層の基礎的特徴は，目的タスクでもやはり（上位層から）必要とされる可能性が高く，再学習時に更新の必要は少ない．逆に上位層では，タスクごとに専門性の高い特徴抽出がなされると考えられ，したがって更新の必要性が高いということになる．

4. CNN の基本構造の改良

4·1　畳込み層の改良

Lin らは，畳込み層の表現能力を高める目的で，多層パーセプトロン（Multi-Layer Perceptron）を畳込み層の出力部分に組み込んだ MLPconv 層を提案し，これを Network-in-Network（NIN）と呼んだ [Lin 14]．MLPconv 層は，二つの畳込み層を重ねたものとも捉えることができる．1 番目は $H \times H$（$\times K$）のフィルタをもつ通常の畳込み層で，2 番目は 1×1（$\times K'$）のフィルタをもつ畳込み層である．Lin らの論文には，後述する出力層前段での全体平均プーリング（global average pooling）の導入という，もう一つの貢献もある．これら 1×1 畳込みと全体平均プーリングの考え方は，GoogLeNet を始めとする後の研究に影響を与えた．

CNN のある層の一つのユニットを考えるとき，このユニットの出力に影響を及ぼし得る入力画像平面の領域（画素の集合）のことを，受容野（receptive field）と呼ぶ*1．通常，CNN の畳込み層でのフィルタの形状は正方形であり，プーリングも正方領域である．その場合，どの層のユニットであれ，入力画像上にもつ受容野は正方形となる．任意の位置にものが写る画像からものを認識する場合，認識結果に影響を及ぼす画像領域の選択は本質的な問題といえる．そういった文脈での受容野の柔軟な選択に関する研究は，深層学習以前のほうがより盛ん（例えば [Coates 11, Jia 12, Kong 14]）であり，CNN についてはそういった検討はほとんどなされていない．そもそも VGGNet のように，すべての畳込み層のフィルタサイズを 3×3 としても良い性能が出ることから，フィルタの形状を正方形以外にとる余地があるようには見えなかった．

ところが最近の我々の研究で，この形状をうまくデザインすると性能を向上させられることを確かめている

表 2　物体カテゴリー認識のベンチマークデータ CIFAR-10/100 による各手法の性能比較．
データ拡張なし，単一モデルによる結果

モデル	CIFAR-10	CIFAR-100	重み数
NIN [Lin 14]	10.41	35.68	$\approx 1\,M$
DSN [Lee 15]	9.69	34.57	$\approx 1\,M$
ALL-CNN [Springenberg 14]	9.08	33.71	$\approx 1.4\,M$
RCNN [Liang 15]	8.69	31.75	$\approx 1.9\,M$
Spectral pooling [Rippel 15]	8.6	31.6	–
FMP [Graham 14]	–	31.2	$\approx 12\,M$
Hex kernel [Sun 15]	8.54	30.54	$\approx 1.4\,M$
Hex kernel	8.42	29.77	$\approx 2.4\,M$

*1 受容野とは神経科学の用語で，感覚系のある神経細胞の応答に変化を与える刺激の空間領域をいう．

[Sun 15]．各層のフィルタを3 × 3から二つの画素を除去した“凸”形状とし，この“凸”の向きを各層でバラバラになるようにセットしたCNNは，CIFAR-10/100において高い性能を示す（表2の“Hex kernel”）．性能向上の理論的な理由はまだ定かでないが，このCNNでは，上位層のユニットが入力画像上にもつ受容野の形状が円形に近くなることで，何らかの効率の向上が果たされていると考えられる．

4·2 プーリング層の改良

プーリング層の改良の試みは畳込み層よりも多数ある．プーリング層では入力の縦横サイズが縮小されるが，プーリングによるサイズの縮小率は最も緩やかな場合でも縦横1/2であり（2×2→1），実に75%もの情報を捨てていることになる．またプーリングを何度も行うと急速にサイズが小さくなるため，入力画像サイズを大きくしないかぎり，ネット全体でのプーリング層の数は増やせない．

このような問題を解決するため，Grahamはフラクショナル最大プーリング（Fractional Max Pooling，以下FMP）[Graham 14]を提案している．FMPは，プーリング領域をレギュラーな正方格子状にとるのではなく，格子の間隔を縦横それぞれ1あるいは2を一定の確率でランダムにとるように決め，その格子が切り分ける各領域をプーリング領域とする方法で，1/2ではなく例えば1/1.5のような，1と2の間の分数（fraction）を比としてサイズが縮小されるプーリングを実効的に実現する．CIFAR-10/100で高い性能を示す（表2）．

CNNは通常プーリング層を不可欠の構成要素として含む．Springenbergらはこのプーリング層（および全結合層）を廃したAll CNNを提案した[Springenberg 14]．All CNNでは，ストライドを1より大きい値にセットした畳込み層でプーリング層を代替する．さらに，通常のプーリング層ではチャネルを横断したプーリングは行わないが，この方法はチャネルを横断した計算も行うことになり自由度が増す（ただし，それがそのまま利点になるわけではない）．またAll CNNでは，NIN同様に出力層前段での全体平均プーリングを採用することで，全結合層を完全に廃している．この構成で，CIFAR-10/100やImageNetなどのベンチマークで，同じかより少ないパラメータ数で従来法と同等以上の性能を示す（表2）．

画像をフーリエ変換した周波数領域では，画像へのフィルタの畳込みは，画像とフィルタのフーリエ変換の要素ごとの積として実現される．離散フーリエ変換（DFT）を活用し，畳込み層の計算をスピードアップする試みがなされており，一定の成功を収めている[Mathieu 13, Vasilache 14]．Rippelらは，計算速度向上ではなく，性能向上のためにプーリングを周波数領域で行う方法を提案している[Rippel 15]．方法は単純で，プーリング層への入力のDFT変換を，低周波成分に対応する矩形領域を切り出し，これを逆DFT変換するというものである．彼らはこの方法をspectral poolingと呼んでいる．最大プーリングの圧縮率は最も緩やかな場合（2×2→1）でも4：1と情報の損失が大きいが，spectral poolingはこれをよりマイルドにできる．この方法もCIFAR-10/100でかなり高い性能を示す（表2）．

4·3 全結合層の縮小

AlexNetや過去の典型的なCNNでは，後段に数層の全結合層をもち，これが最後の畳込み層（プーリング層）と出力層の間に入る．先述のとおり，結果的に全体の重みの9割以上がこれら全結合層に集中し，その規模がモデルのパラメータ（重み）数を左右するといういびつさがあった．最近はこれを改め，NIN，GoogLeNet，ResNetのように，畳込み層部分を強化して表現力・識別力を高めることで，後段の巨大な全結合層を廃するのがトレンドとなっている．

このような考え方に基づくモデルでは，最後の畳込み層部分の出力に全体平均プーリングが適用される．NINでは，最後の畳込み層の出力（通常6×6×チャネル数程度）のチャネル数を，目的クラス数と同数にしておき，チャネルそれぞれの全体を平均プーリングした結果を（ソフトマックス関数で正規化し）当該クラスの出力とする．

5. 画像の入力方法の多様化

5·1 畳込みとプーリングの位置不変性

一般的なCNNは固定サイズの画像（例えば224×224）を入力に受け取り，出力は固定長（分類タスクならカテゴリー数と同数のユニット）である．しかし入力画像はさまざまな解像度，アスペクト比をもち得る．この違いを吸収するにはいくつかの方法があり得る．

畳込みやプーリング自体は，シフト不変な演算，つまり画像の位置によらず計算そのものは同一である．したがって，CNNの入力から全結合の前までの部分ネットワークに関する限り，入力サイズの変化は出力サイズを変えることで吸収できる（なお，CNNのこの使い方は古くから知られたものである[Wolf 94]）．画像をリサイズすることなくそのままCNNに入力し，上述のように畳込み層（あるいはプーリング層）部分の出力（特徴マップなどと呼ぶ）をサイズ可変の形で得，その後の計算で用いることができる．

例えばVGGNetは，224×224の画像を入力に受け取り，全結合層に至るまでに2×2の最大プーリングを5回行う．5番目のプーリング層の出力サイズは1辺が$224/(2^5)=7$，つまり7×7である（図3）．この49個の要素は，それぞれ同一の部分ネットワーク（構造および重みが同じであるという意味）を，画像の異なる部分に適用して計算されたものと解釈できる．つまりこのネッ

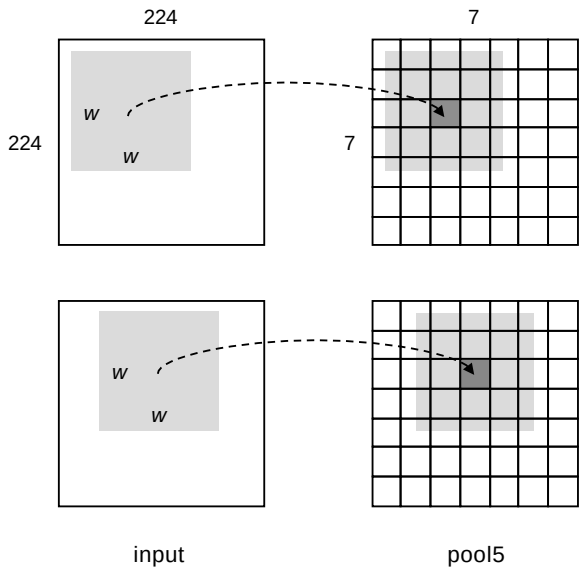

図 3　VGGNet の入力画像（224×224）と 5 番目のプーリング層（7×7）の関係.
プーリング層の各ユニットの出力は，同一のネットワークを入力画像上の正方形（$w \times w$）に適用し計算されている．適用位置のストライドは $2^5=32$ となる

トワークを，入力画像上 $f=2^5=32$ のストライドで $7\times7=49$ か所に適用したときの出力が，5 番目のプーリング層の出力に一致する．入力画像のサイズがどうであれ，同じ部分ネットワークをストライド f でこれに適用すると，その入力サイズに応じたサイズのプーリング層の出力を得られる．

5·2　単一出力の場合

物体カテゴリー分類などで一般的な，入力画像に対し単一の出力を得たい場合を考える．最も簡単なのは入力画像を CNN の入力サイズにリサイズする方法である [Krizhevsky 12]．ただし，入力画像が長方形の場合，リサイズによって縦横アスペクト比が変化し，その影響が懸念される．

上述の方法によれば，入力のサイズに応じて畳込み層部分の出力のサイズを変化させられるが，そのままでは全結合層には入力できない．対処法の一つは，畳込み層部分の出力の全体を対象に全体プーリングを行う方法である．Spatial Pyramid Pooling（SPP）は，ただ全体をプーリングするだけでなく全体を 2×2，4×4 と繰り返し分割し，その各領域の畳込み層の出力をプーリングする [He 14]．これは CNN 登場以前，物体認識のための特徴抽出の標準的方法だった BoF（Bag-of-feature）で一般的であった方法でもある．このピラミッド式のプーリングにより，最終畳込み層のチャネル数を K とすると，その縦横サイズにかかわらず，$K+4K+16K$ のような固定長のベクトルを得る．これを従来同様の全結合層に入力し，後は同じである．

5·3　可変サイズの出力（特徴マップ）

さまざまなサイズの画像 I に対し，全体プーリングを行って固定長の出力 $I\rightarrow y$ を得るのではなく，与えられた画像から部分領域 $R_{ij}\subset I$ を切り取り，これを CNN に（場合によってはリサイズした後）入力し，出力 $R_{ij}\subset y_{ij}$ を得る方法もある．さまざまな利用方法があり，切り出す R_{ij} が疎（まば）らか，密かで分類できる．

疎らに R_{ij} を切り出す一例は物体検出のための R-CNN（後述）である．なおカテゴリー分類でも，いわゆる multi-view test ―精度向上のために画像の部分領域（例えば画像中央と 四隅の 5 か所）を切り出し，それらのカテゴリー予測の平均を求める―で同様の処理が行われる．

一方，R_{ij} を（i, j）を変えつつ切出し位置を密にとり，それぞれについて出力を計算すると，画像のフォーマットの出力マップ y_{ij} を得る．この方法は，入力画像の画素（i, j）ごとに物体カテゴリーのクラス y_{ij} を推定するセマンティックセグメンテーションで用いられる [Long 15]．なお後段に全結合層をもつ CNN では，全結合層を畳込み層と読み替えると見通しが良くなる．つまり，最後の畳込み層（あるいはプーリング層）と接続する全結合層を，前者の出力と同じサイズのフィルタを畳み込む畳込み層とみなすのである．この読み替えた畳込み層は通常とは違い，入力の 1 か所のみにフィルタを適用する．さらに元の全結合層がもつユニット数が，読み替えた畳込み層のフィルタ数となり，これを M とすると，読み替えた畳込み層の出力サイズは $1\times1\times M$ になる．さらに全結合層が続く場合，この 1×1 サイズの画像に，さらに $1\times1\times M$ のサイズのフィルタを M' 個適用する畳込み層と読み替える．

入力画像の R_{ij} の位置（i, j）を縦横 1 画素ずつ密に動かすと，出力マップ y_{ij} は入力画像と同じサイズ（解像度）になる．この計算を効率良く行う方法が，さまざまな位置・スケールの R_{ij} についての出力マップ（特徴マップ）を得る方法（Overfeat）とともに提案された [Sermanet 13]．この方法はシフト & スティッチとも呼ばれ，入力画像を縦横に $d\in\{0\cdot f-1\}$ 画素ずらして（shift）CNN に入力し，得られる f^2 通りの出力マップ（図 3 の 7×7 出力）を，使用した d に応じて互い違いに並べ直す（stitch）．ただし f は，プーリング層の隣接ユニットの受容野間の距離である．一方，fully-CNN（FCNN）では，CNN の畳込み層部分を入力画像に適用して得られる可変サイズの出力マップ―プーリングの回数だけ解像度が低下する―をそのまま利用し，セマンティックセグメンテーションに応用している [Long 15]．そこでは密な出力を得るために，シフト & スティッチではなく出力マップのアップサンプリングが行われている．

5·4　物体検出への応用

物体検出（object detection）とは，画像が 1 枚与え

られ，その中にどんな物体がどこに写っているか，位置とカテゴリーを同時に認識する問題をいう（類似の，より簡単な問題に，カテゴリーが未知あるいは既知の物体一つの画像内の位置を推定するlocalizationがある）．物体検出は，高精度な画像検索 [Johnson 15] や画像記述 [Karpathy 15] などのより高度なタスクの基盤技術となる．この点で以前から最も必要とされてきたが，しかし解決が難しかった問題の一つであった．CNN登場以後，難しい問題であることに変わりはないが，CNNの使い方が洗練されてくるにつれ急速に性能を向上させつつある．

Overfeatはその初期の研究である [Sermanet 13]．入力画像から位置・サイズを機械的に変えて矩形領域をサンプリングし，その内部のコンテンツのカテゴリー予測と物体のボックス位置の予測を同時に実行する．前者はクラス分類を行うCNNで，後者は回帰を行う（出力層の4ユニットからボックスの座標を出力）CNNである．両者はともにCNNの最終プーリング層までを共有し，それ以降の全結合層だけが異なる．効率を考え，上述の方法で入力画像に対しプーリング層の出力を一度求めておき，それを使い回す．

その後R-CNN（Region-based CNN）の方法，つまりボックスの候補（region proposal）をCNNに入力するやり方がメジャーになり，今も主流の方法である．この方法は，まず何らかの方法でボックスの候補を多数（〜数千）つくり出し，それらを一つずつCNNに入力し，（取り出した特徴量をSVMで分類することで）カテゴリー予測を行い，その結果でボックス候補を取捨選択し検出結果とする方法である．CNNそのものはAlexNetなどのモデルで，出力層を対象とするカテゴリー数（＋何も物体がない「背景クラス」）に対応するもので置き換え，検出用の学習データを用いてfine-tuneしたものである．ボックス位置をCNNの特徴を元に回帰し，上の方法で検出されたボックスの位置精度を向上させる後処理を加える．

最初のR-CNNでは，つくり出した多数のボックス候補の一つ一つをCNNに入力する設計で，候補の数だけCNNの順伝播計算を1からやる必要があり，効率が悪かった．Overfeat同様，入力画像にCNNを適用したときの最終畳込み部分の出力を最初に計算し，以降これを使い回すようにすることで効率が改善された（SPP-Net, Fast R-CNN [Girshick 15]）．

R-CNNはボックス候補（region proposal）の生成とは独立しており，つまりさまざまな方法を利用できる．CNNに基づかないselective search [Uijlings 13] やCNNから取り出した特徴を用いるMultiBox [Szegedy 14, Erhan 14] が代表的である．ただし計算量が多く，物体検出の計算量のボトルネックはこのボックス候補をつくり出す部分になっていた．最近，このボックス候補の生成をR-CNNでのカテゴリー分類と同じ特徴（畳込み層の出力マップ）を使ってニューラルネットで行う方法（region proposal network）が提案されFaster R-CNN [Ren 15] と名付けられた．その結果，ボックス候補の生成を含むトータルの時間をずっと小さく（毎秒5フレーム），しかも従来方法より高い精度で行えるようになっている．

6. CNNの理解と画像合成

ここまで述べてきたように，CNNは画像認識のタスクのほとんどに適用され，いずれもほぼ例外なくCNNを使わなければ決して望めない高い性能を示す．その一方で，なぜそのように高い性能を示すのかの理解はそれほど進んでいない．そういった理解を目指した研究が多数行われているが，そのほとんどが理論的なものではなく実験的な分析である．以下にそういった研究のいくつかを紹介する．ILSVRCのclassificationのタスクを学習済みのAlexNetやVGGNetが分析対象となる．

Zeilerらは，各層の各ユニットでどういった特徴が取り出されているかを調べている [Zeiler 14]．それぞれのユニットが最も活性化する入力を画像データセット中から選び出し，層ごとユニットごとに表示してみせた．下位層のユニットでは単純な幾何学パターンや色の特徴を，上位の層ほどより複雑な形状や，あるいは概念に選択的に反応している様が示された．それまでも共有されていたCNNの働きについての直感的な理解が，改めて確かなものとされた．またこの研究では，逆畳込み（deconvolution）を用いた特徴抽出の可視化が提案されている．

Simonyanらは，AlexNetの出力層の中から一つのユニットを選択し，その出力を最大化する入力画像を計算して見せた [Simonyan 14a]．計算はSGDなど数値最適化によって行う．その際に用いる初期値（ランダムでよい）や，SGDのハイパーパラメータ次第で結果は変動するものの，生成される画像には，選んだユニットに対応するカテゴリーの特徴が何らかの形で浮かび上がるものが多数見られる．ただし，物体のかなり完全な姿形が再現されるカテゴリーがある一方で，一部の特徴しか現れないものや，特徴がどこに現れているのか解釈が難しい場合もある．そのように断片的にではあるが，CNNが学習の結果，何を見ているかを示唆するものとなっている．

Mahendranらは，AlexNetの最終層だけでなく，中間層の出力から入力画像を復元する方法を提案している [Mahendran 14]．特定の画像をCNNに入力して得られる各層の出力を記録しておき，逆に，ある特定の層の出力だけを与え，それを忠実に再現するような入力画像を計算する．上のSimonyanらの研究では，入力画像をより自然なものとするため，画像濃淡のL_2ノルムを目的関数に正則化項として加えていた．Mahendranらは，画像のトータルバリエーションを正則化項として加

図 4　Gatys らの方法 [Gatys 15] による画像合成.
（a）入力画像.（b）ゴッホの絵画（右下囲み）のスタイルを再現した画像.（c）ピカソの絵画（右下囲み）のスタイルを再現した画像

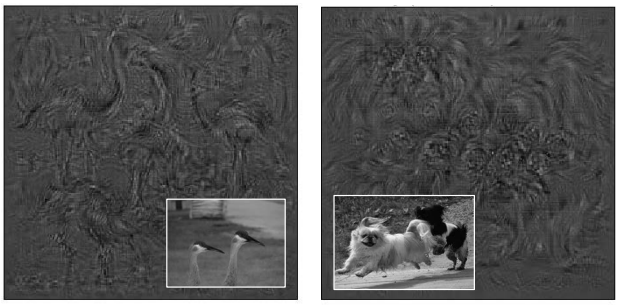

図 5　AlexNet の出力ユニットの一つを最大化する入力画像の一例（[Simonyan 14a] の結果の再現）.
左は“crane”（鶴），右は“Japanese spaniel”をそれぞれ対象に選んだ場合．囲みの写真は典型的な画像例

図 6　入力画像（input）と，それを VGGNet-16 に入力したときの第 2 プール層（全 21 層中の第 6 層），第 4 プール層（同第 14 層），第 5 プール層（同第 18 層）の出力を元に，入力画像を逆算したもの（[Mahendran 14] の結果の再現）

え，より自然な画像を得ようとした．結果は，AlexNet の第 4 畳込み層付近までは，層出力から入力画像をほぼ完全に再現できるが，それより上位の層では徐々に再現精度が低下する．全結合層では，最初に入力した画像そのものというよりも，そこから推定されるカテゴリーを元に再現したような画像になる．図 6 は，以上の結果を VGGNet-16 を用いて再現したものである．入力画素数と層の出力数が逆転する第 4 プール層付近までは，入力画像をほぼ正確に再現できるが，出力数がより少なくなる以降の層では再現は正確でなくなることがわかる．

Szegedy らは，CNN（AlexNet）が正解を返す画像に対し，これにわずかな改変を加えると CNN を騙せる，つまり誤ったカテゴリーを答えとして返すように誘導できることを示した [Szegedy 13]．具体的には，入力画像 I に別の画像 I' を加算して新たな画像 $I+I'$ をつくり，これを CNN に入力したとき，間違えさせたいカテゴリー（正しいカテゴリーでない）の出力が大きくなるよう，上に述べた Simonyan らと同様のやり方で最適化によって I' を決定する．その際，I' の大きさが小さくなるよう正則化すると，人目には I と $I+I'$ の違いがほとんどわからず，したがって両方ともに同じカテゴリーと判断されるにもかかわらず，CNN は $I+I'$ を恣意的に選んだカテゴリーと誤認識する（図 7）．しかもその際の尤度はほぼ 100％にまで近づけることができる．関連して Nguyen らは，学習済みの CNN を別なやり方で騙す画像― CNN は 100％近い尤度（確信度）であるカテゴリーと答えるが，人間にはまずそうは見えない画像―を生成できることを示した [Nguyen 15].

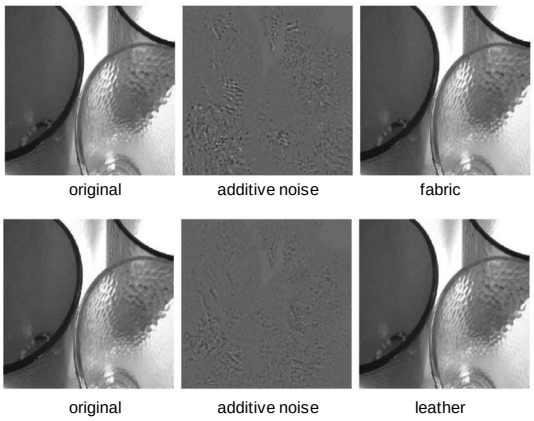

図 7　マテリアル認識を例にした [Szegedy 13] の再現.
CNN は左の画像は正しく“glass”カテゴリーと認識できるが，これに中央のわずかなノイズを加算した右の画像は，上は“fabric”，下は“leather”と誤認識する

以上の研究では，画像の生成は CNN の理解を目的としたものであったが，CNN を用いた画像の合成そのものを目的とする研究が，最近盛んに行われるようになっている．CNN の中間層の出力（集合）を目標値に，入力画像を逆算する上述の方法を使って，画像と絵画のペアを入力に，画像を改変し絵画の描画スタイルを反映

させる方法が発表されている [Gatys 15]（図4）．一方，CNN の入出力の向きを逆さまにしたフィードフォワードネットを使って画像を生成する方法がいくつか提案され [Dosovitskiy 15]，特に対立（adversarial）ネットワークを用いた学習に基づく方法が，クオリティの高い画像の合成を可能にしつつある [Radford 15]．また CNN ではないが，リカレントニューラルネット（RNN）を用いる方法 [Gregor 15] もある．CNN の使われ方はいささか異なるが，シーンを疎な視点から撮影した画像を元に，その間の視点からの画像を合成する方法 [Flynn 15] も提案されており，画像合成の今後のいっそうの発展が予期される．

謝 辞

図示した画像は大関 誠君，劉 星君が行った実験によって得られたものである．また本稿に述べた知見のいくつかは JST，CREST の支援を受けて行った研究によって得た．

◇ 参 考 文 献 ◇

[Cimpoi 15] Cimpoi, M., Maji, S., Kokkinos, I. and Vedaldi, A.: Deep filter banks for texture recognition, description, and segmentation, arXiv preprint arXiv:1507.02620（2015）

[Coates 11] Coates, A. and Ng, A.: Selecting receptive fields in deep networks, *Advances in Neural Information Processing Systems,* pp. 2528-2536（2011）

[Cybenko 89] Cybenko, G.: Approximations by superpositions of sigmoidal functions, *Mathematics of Control, Signals, and Systems,* Vol. 2, No. 4, pp. 303-314（1989）

[Desjardins 15] Desjardins, G., Simonyan, K., and Pascanu, R.: Natural neural networks, *Advances in Neural Information Processing Systems,* pp. 2062-2070（2015）

[Donahue 14] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E. and Darrell, T.: DeCAF: A deep convolutional activation feature for generic visual recognition, *Proc. ICML*（2014）

[Dosovitskiy 15] Dosovitskiy, A., Tobias Springenberg, J. and Brox, T.: Learning to generate chairs with convolutional neural networks, *Proc. CVPR*（2015）

[Duchi 11] Duchi, J., Hazan, E. and Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization, *J. Machine Learning Research,* Vol. 12, pp. 2121-2159（2011）

[Eigen 14] Eigen, D., Rolfe, J., Fergus, R. and LeCun, Y.: Understanding deep architectures using a recursive convolutional network, *Proc. ICLR*（2014）

[Erhan 14] Erhan, D., Szegedy, C., and Toshev, A.: Scalable object detection using deep neural networks, *Proc. CVPR,* pp. 2155-2162（2014）

[Flynn 15] Flynn, J., Neulander, I., Philbin, J. and Snavely, N.: Deep-stereo: Learning to predict new views from the world's imagery, arXiv preprint arXiv:1506.06825（2015）

[Gatys15] Gatys, L. A., Ecker, A. S. and Bethge, M.: Aneural algorithm of artistic style, arXiv preprint arXiv: 1508.06576（2015）

[Girshick 15] Girshick, R.: Fast R-CNN, arXiv preprint arXiv: 1504.08083（2015）

[Glorot 10] Glorot, X. and Bengio, Y.: Understanding the difficulty of training deep feed forward neural networks, *Proc. AISTATS*（2010）

[Graham 14] Graham, B.: Fractional max-pooling, arXiv preprint arXiv:1412.6071（2014）

[Gregor 15] Gregor, K., Danihelka, I., Graves, A. and Wierstra, D.: DRAW:A recurrent neural network for image generation, *Proc. ICML*（2015）

[He 14] He, K., Zhang, X., Ren, S. and Sun, J.: Spatial pyramid poolingin deep convolutional networks for visual recognition, *Proc. ECCV*（2014）

[He 15a] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual learning for image recognition, arXiv preprint arXiv: 1512.03385（2015）

[He 15b] He, K., Zhang, X., Ren, S. and Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, *Proc. ICCV*（2015）

[Hinton 15] Hinton, G., Vinyals, O. and Dean, J.: Distilling the knowledge in a neural network, arXiv preprint arXiv: 1503.02531（2015）

[Hornik 89] Hornik, K., Stinchcombe, M. and White, H.: Multilayer feed forward networks are universal approximators, *Neural Networks,* Vol. 2, No. 5, pp. 359-366（1989）

[Ioffe 15] Ioffe, S. and Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift, *Proc. ICML*（2015）

[Jaderberg 14] Jaderberg, M., Vedaldi, A. and Zisserman, A.: Speeding up convolutional neural networks with low rank expansions, *Proc. BMVC*（2014）

[Jarret 09] Jarret,K.,Kavukcuoglu,K.,and LeCun,Y.: What is the best multi-stage architecture for object recognition, *Proc. ICCV*（2009）

[Jia 12] Jia,Y., Huang, C. and Darrell,T.: Beyond spatial pyramids: Receptive field learning for pooled image features, *Proc. CVPR*（2012）

[Jia 14] Jia, Y., Shelhamer, E., Donahue, J. and Karayev, S.: Caffe: Convolutional architecture for fast feature embedding, *Proc. ACM Int. Conf. on Multimedia,* pp. 675-678（2014）

[Johnson 15] Johnson, J., Krishna, R., Stark, M. and Li, L.: Image retrieval using scene graphs, *Proc. CVPR,* pp. 3668-3678（2015）

[Karpathy15] Karpathy, A. and Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions, *Proc. CVPR*（2015）

[Kingma 14] Kingma, D. and Ba, J.: Adam:Amethod for stochastic optimization, arXiv preprint arXiv:1412.6980（2014）

[Kong14] Kong, S., Jiang, Z. and Yang, Q.: Collaborative receptive field learning, arXiv preprint arXiv:1402.0170（2014）

[Krizhevsky 12] Krizhevsky, A., Sutskever, I. and Hinton, G.: Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems,* pp. 1097-1105（2012）

[Krizhevsky 14] Krizhevsky, A.: One weird trick for parallelizing convolutional neural networks, arXiv preprint arXiv: 1404.5997（2014）

[Lee 15] Lee, C., Xie, S., Gallagher, P., Zhang, Z. and Tu, Z.: Deeply-supervised nets, *Proc. AISTATS*（2015）

[Liang 15] Liang, M. and Hu, X.: Recurrent convolutional neural network for object recognition, *Proc. CVPR,* pp. 3367-3375（2015）

[Lin 14] Lin, M., Chen, Q. and Yan, S.: Network in network, *Proc. ICLR*（2014）

[Long 15] Long, J., Shelhamer, E. and Darrell, T.: Fully convolutional networks for semantic segmentation, *Proc. CVPR,* pp. 3431- 3440（2015）

[Mahendran 14] Mahendran, A. and Vedaldi, A.: Understanding deep image representations by inverting them, arXiv preprint arXiv:1412.0035（2014）

[Mathieu 13] Mathieu, M., Henaff, M. and LeCun, Y.: Fast training of convolutional networks through FFTs, arXiv preprint arXiv:1312.5851（2013）

[Montufar 14] Montufar, G., Pascanu, R. and Cho, K.: On the number of linear regions of deep neural networks, *Advances in*

Neural Information Processing Systems, pp. 2924-2932（2014）
[Neyshabur 15] Neyshabur, B., Salakhutdinov, R. and Srebro, N.: Path-SGD: Path-normalized optimization in deep neural networks, *Advances in Neural Information Processing Systems,* pp. 2413-2421（2015）
[Nguyen 15] Nguyen, A., Yosinski, J. and Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, *Proc. CVPR*（2015）
[Oquab 14] Oquab, M., Bottou, L. and Laptev, I.: Learning and transferring mid-level image representations using convolutional neural networks, *Proc. CVPR,* pp. 1717-1724（2014）
[Radford 15] Radford, A., Metz, L. and Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv preprint arXiv:1511.06434（2015）
[Razavian 14] Razavian, A., Azizpour, H. and Sullivan, J.: CNN features off-the-shelf: An astounding baseline for recognition, *Proc. CVPR,* pp. 512-519（2014）
[Ren 15] Ren, S., He, K., Girshick, R. and Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks, *Advances in Neural Information Processing Systems,* pp. 91-99（2015）
[Rippel 15] Rippel, O., Snoek, J. and Adams, R.: Spectral representations for convolutional neural networks, *Advances in Neural Information Processing Systems,* pp. 2440-2448（2015）
[Russakovsky 14] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. and Fei-Fei, L.: ImageNet large scale visual recognition challenge, arXiv preprint arXiv: 1409.0575（2014）
[Sermanet 13] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. and LeCun, Y.: Overfeat: integrated recognition, Localization and detection using convolutional networks, arXiv preprint arXiv:1312.6229（2013）
[Shimodaira 00] Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function, *J. Statistical Planning and Inference,* Vol. 9, No. 2, pp. 227-244（2000）
[Simonyan 14a] Simonyan, K.,Vedaldi, A. and Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps, *Proc. ICLR Workshop*（2014）
[Simonyan 14b] Simonyan, K. and Zisserman, A.: Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556（2014）
[Springenberg 14] Springenberg, J., Dosovitskiy, A. and Brox, T.: Striving for simplicity: The all convolutional net, arXiv preprint arXiv:1412.6806（2014）
[Srivastava 14] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting, *J. Machine Learning Research,* Vol. 15, No. 1, pp. 1929-1958（2014）
[Sun 15] Sun, Z., Ozay, M. and Okatani, T.: Design of kernels in convolutional neural networks for image classification, arXiv preprint arXiv:1511.09231（2015）
[Szegedy 13] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R.: Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199（2013）
[Szegedy 14] Szegedy, C., Reed, S., Erhan, D. and Anguelov, D.: Scalable, High-quality object detection, arXiv preprint arXiv:1412.1441（2014）
[Szegedy 15] Szegedy, C., Liu, W., Jia, Y., Sermanet, P. and Reed, S.: Going deeper with convolutions, *Proc. CVPR*（2015）
[Taigman 14] Taigman, Y., Yang, M. and Ranzato, M.: Deepface: Closing the gap to human-level performance in face verification, *Proc. CVPR,* pp. 1701-1708（2014）
[Toshev 14] Toshev, A. and Szegedy, C.: Deeppose: Human pose estimation via deep neural networks, *Proc. CVPR,* pp. 1653-1660（2014）
[Uijlings 13] Uijlings, J., Sande, van de K. and Gevers, T.: Selective search for object recognition, *Int. J. Computer Vision,* Vol. 104, No. 2, pp. 154-171（2013）
[Vasilache 14] Vasilache, N., Johnson, J., Mathieu, M. and Chintala, S.: Fast convolutional nets with fbfft: A GPU performance evaluation, arXiv preprint arXiv:1412.7580（2014）
[Wolf 94] Wolf, R. and Platt, J.: Postal address block location using a convolutional locator network, *Advances in Neural Information Processing Systems*（1994）
[Xiao 14] Xiao, J., Ehinger, K., Hays, J. and Torralba, A.: Sun database: Exploring a large collection of scene categories, *Int. J. Computer Vision,* pp. 1-20（2014）
[Zeiler 14] Zeiler, and Fergus, R.: Visualizing and understanding convolutional networks, *Proc. ECCV*（2014）

2016 年 1 月 18 日　受理

著　者　紹　介

岡谷　貴之

1999 年東京大学大学院工学系研究科博士課程修了（計数工学）．同年東北大学大学院情報科学研究科助手，その後講師，助教授を経て，2013 年に教授，現在に至る．コンピュータビジョンの研究に従事し，多視点幾何から物体認識までの幅広い応用に関心をもつ．電子情報通信学会，情報処理学会，IEEE などの各会員．