Going Deeper with Convolutions

備考

著者

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich

掲載

"Going Deeper with Convolutions", Procs. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)，pp.1--9, 2015.

Abstract

我々は、ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14)において、分類と検出で新しい状態を達成した、コードネームInceptionと呼ばれる深層畳み込みニューラルネットワークアーキテクチャを提案する。このアーキテクチャの最大の特徴は、ネットワーク内のコンピューティングリソースの利用率を向上させたことです。慎重に作られた設計により、計算機予算を一定に保ちながら、ネットワークの深さと幅を増やしました。また、品質を最適化するために、ヘブの原理とマルチスケール処理の直感に基づいてアーキテクチャを決定しました。ILSVRC14に提出した「GoogLeNet」は、22層の深さを持つネットワークで、その品質は分類と検出の文脈で評価されています。

1. はじめに

この3年間で，深層学習や畳み込みネットワークの進歩により，物体の分類・検出能力が劇的に向上しました[10]．心強いニュースは，この進歩のほとんどが，より強力なハードウェア，より大きなデータセット，より大きなモデルの結果だけではなく，主に，新しいアイデア，アルゴリズム，改良されたネットワークアーキテクチャの結果であるということです．例えば、ILSVRC 2014の上位作品で

は、同じコンペティションの分類データセット以外に、検出目的で新たなデータソースは使用されていませんでした。ILSVRC 2014に提出したGoogLeNetは、2年前に優勝したKrizhevskyら[9]のアーキテクチャに比べて、実際には12倍も少ないパラメータしか使用していませんが、その一方で、精度は著しく向上しています。物体検出の分野では，大規模な深層ネットワークを単純に適用することではなく，Girshickら[6]によるR-CNNアルゴリズムのように，深層アーキテクチャと古典的なコンピュータビジョンの相乗効果によって，最大の成果が得られています．

また、モバイルコンピューティングやエンベデッドコンピューティングの普及に伴い、アルゴリズムの効率化、特に電力やメモリの使用量が重要になってきていることも注目すべき点です。本稿で紹介する深層アーキテクチャの設計において、精度の数値に固執するのではなく、この要因を考慮していることは注目に値します。ほとんどの実験において、モデルは推論時に15億回の乗算を行うという計算量を維持するように設計されており、純粋に学術的な好奇心で終わるのではなく、大規模なデータセットであっても合理的なコストで実世界で使用できるようになっています。

本論文では、コンピュータビジョン用の効率的な深層ニューラルネットワークアーキテクチャ、コードネーム「Inception」に焦点を当てます。この名前は、Linら[12]によるNetwork in net-work論文と、有名な「We need to go deep」というインターネットミーム[1]に由来しています。私たちの場合、「深い」という言葉は2つの異なる意味で使われています。まず、「Inceptionモジュール」という形で新しいレベルの組織を導入するという意味と、ネットワークの深さが増すというより直接的な意味があります。一般的に、Inceptionモデルは、Aroraら[2]による理論的研究からインスピレーションと指針を得ながら、[12]を論理的に集大成したものと考えることができます。このアーキテクチャの利点は、ILSVRC 2014の分類と検出の課題で実験的に検証されており、現在の技術水準を大幅に上回っています。

2. 関連作業

LeNet-5 [10]以降、畳み込みニューラルネットワーク（CNN）は、標準的な構造を持っています。すなわち、積み上げられた畳み込み層（オプションとして、コントラストの正規化と最大プール）の後に、1つ以上の完全連結層が続きます。この基本的なデザインのバリエーションは画像分類の分野で広く使われており、MNIST、CIFAR、そして最も注目すべきはImageNet分類チャレンジでこれまでに最高の結果を出している[9, 21]。Imagenetのような大規模なデータセットでは，オーバーフィッティングの問題に対処するためにドロップアウト[7]を使用しながら，層の数[12]と層のサイズ[21, 14]を増やすことが最近の傾向である．

最大プール層を設けると空間情報が失われるという懸念があるものの，[9]と同じ畳み込みネットワーク・アーキテクチャは，ローカリゼーション[9, 14]，物体検出[6, 4, 18, 5]，人間の姿勢推定[19]などにも採用されている．

Serreら[15]は、霊長類の視覚野の神経科学モデルにヒントを得て、サイズの異なる一連の固定ガボール・フィルターを使用し、複数のスケールを処理しました。ここでは，同様の戦略を用いています．しかし、[15]の固定された2層の深層モデルとは異なり、Inceptionアーキテクチャではすべてのフィルタが学習されます。さらに、Inceptionのレイヤーは何度も繰り返され、GoogLeNetモデルの場合、22レイヤーの深層モデルになります。

Network-in-Networkは、Linら[12]によって提案された、ニューラルネットワークの表現力を高めるための手法です。Linらのモデルでは，ネットワークに11の畳み込み層を追加して深さを増しています．我々のアーキテクチャでは、このアプローチを多用しています。最も重要なことは、1x1畳み込みは主に次元削減モジュールとして使用され、ネットワークのサイズを制限する計算上のボトルネックを取り除くことです。これにより、パフォーマンスに大きな影響を与えることなく、ネットワークの深さだけでなく幅も大きくすることができます。

最後に，物体検出の最新技術は，Girshickら[6]によるRegions with Convolutional Neural Networks (R-CNN)法です．R-CNNは、全体的な検出問題を2つのサブ問題に分解します。すなわち、色やテクスチャなどの低レベルの手掛かりを利用して、カテゴリにとらわれない方法でオブジェクトの位置を提案し、CNN分類器を使ってそれらの位置でオブジェクト・カテゴリを識別します。このような2段階のアプローチは、低レベルの手がかりを用いたバウンディングボックス・セグメンテーションの精度と、最先端のCNNの強力な分類能力を活用しています。我々は同様のパイプラインを採用していますが、オブジェクトのバウンディングボックスの再現性を高めるためのマルチボックス[5]予測や、バウンディングボックスの提案をより適切に分類するためのアンサンブルアプローチなど、両方のステージで機能強化を図っています。

3. 動機と高レベルの考慮事項

深層ニューラルネットワークの性能を向上させる最も直接的な方法は、サイズを大きくすることです。これには、深さ（ネットワークレベルの数）と幅（各レベルのユニット数）の両方を増やすことが含まれます。これは、大量のラベル付き学習データがあれば、より高品質なモデルを簡単かつ安全に学習できる方法です。しかし、この単純な方法には2つの大きな欠点があります。

サイズが大きくなると、パラメータの数も多くなり、特に訓練セットのラベル付き例の数が限られている場合は、拡大されたネットワークがオーバーフィッティングしやすくなります。これは、強いラベルを付けたデータセットを入手するには手間と費用がかかり、ImageNetのような（1000クラスのILSVRCサブセットであっても）様々な細かい視覚カテゴリを区別するには、人間の専門的な評価者が必要になることが多いため、図1に示すように、大きな障害となります。


Fig1. ILSVRC 2014の分類チャレンジの1000クラスの中から、2つの異なるクラスを選びました。 これらのクラスを区別するためには、ドメインの知識が再度必要です。

ネットワークのサイズを一律に大きくすると、計算資源の使用量が劇的に増加するという欠点もあります。例えば、ディープビジョンネットワークでは、2つの畳み込み層を連鎖させた場合、そのフィルターの数を一律に増やすと、計算量が2次関数的に増加してしまいます。追加された容量が非効率的に使用された場合（例えば、ほとんどのウェイトがゼロに近い値になってしまった場合）、計算量の多くが無駄になります。計算機の予算は常に有限であるため、性能の質を高めることが主な目的であっても、無差別にサイズを増やすよりも、計算機資源を効率的に配分することが望ましい。

これらの問題を解決するための基本的な方法は、スパース性を導入し、畳み込みの中でも完全連結層をスパース層に置き換えることです。この方法は、生物学的なシステムを模倣するだけでなく、Araoraら[2]の画期的な研究により、より堅固な理論的裏付けを得ることができます。彼らの主な結果は、データセットの確率分布が大規模で非常に疎な深層ニューラルネットワークで表現できる場合、先行する層の活性化の相関統計を分析し、相関性の高い出力を持つニューロンをクラスタリングすることで、最適なネットワークトポロジーを層ごとに構築できるというものです。厳密な数学的証明には非常に強い条件が必要ですが、この記述が、よく知られているヘブの原理（一緒に発火するニューロンは一緒に配線される）と共鳴していることから、その根底にある考えは、実際にはそれほど厳しくない条件でも適用可能であることがわかります。

残念ながら、今日のコンピュータインフラは、非一様な疎なデータ構造上での数値計算に関しては非常に非効率的です。算術演算の回数を100回減らしたとしても、ルックアップやキャッシュミスによるオーバーヘッドが圧倒的に大きいため、スパースな行列への切り替えは利益につながらないかもしれません。このギャップは，CPU や GPU のハードウェアの詳細を利用して非常に高速な密行列の乗算を可能にする，着実に改良され，高度に調整された数値ライブラリを使用することでさらに広がります [16, 9]．また，非一様なスパースモデルでは，より高度なエンジニアリングとコンピューティングインフラが必要となります．現在のビジョン指向の機械学習システムのほとんどは、畳み込みを採用しているという理由だけで、空間ドメインのスパース性を利用しています。しかし、畳み込みは、前の層のパッチへの密な接続の集まりとして実装されています。ConvNetsは、対称性を崩して学習効果を高めるために、[11]以来、特徴次元でランダムで疎な接続テーブルを伝統的に使用してきましたが、並列計算をさらに最適化するために、[9]では完全な接続に戻る傾向がありました。現在のコン

ピュータビジョン用の最先端アーキテクチャは、画一的な構造をしています。フィルターの数が多く、バッチサイズが大きいため、密な計算を効率的に行うことができます。

このため、次の中間的なステップとして、理論で提案されているようにフィルターレベルのスパース性を利用しつつ、現在のハードウェアを利用して密な行列上の計算を利用するアーキテクチャに希望が持てないかという疑問が生じます。疎行列の計算に関する膨大な文献（例えば[3]）によると、疎行列を比較的密な部分行列にクラスタリングすることで、疎行列の乗算において競争力のある性能が得られる傾向があります。近い将来、同様の手法が非一様な深層学習アーキテクチャの自動構築に利用されるようになると考えても、不思議ではありません。

インセプションのアーキテクチャは、高度なネットワークトポロジー構築アルゴリズムの仮想的な出力を評価するケーススタディとしてスタートしました。このアルゴリズムは、視覚ネットワークについて[2]で示唆された疎な構造を近似し、その仮想的な出力を高密度で容易に入手できるコンポーネントでカバーしようとするものです。非常に推測的な試みではありましたが、[12]に基づくリファレンスネットワークと比較して、初期の段階でささやかな利益が観察されました。チューニングを重ねることでその差は広がり、Inceptionは[6]や[5]のベースネットワークとして、特にローカリゼーションや物体検出の文脈で有用であることが証明されました。興味深いことに、当初のアーキテクチャの選択のほとんどは、分離の際に疑問視され、徹底的にテストされましたが、局所的には最適に近いことが判明しました。インセプション・アーキテクチャがコンピュータ・ビジョンで成功を収めたとはいえ、その構築に導いた指導原理に起因するかどうかは、まだ疑問です。それを確かめるには、より詳細な分析と検証が必要です。

4. アーキテクチャの詳細

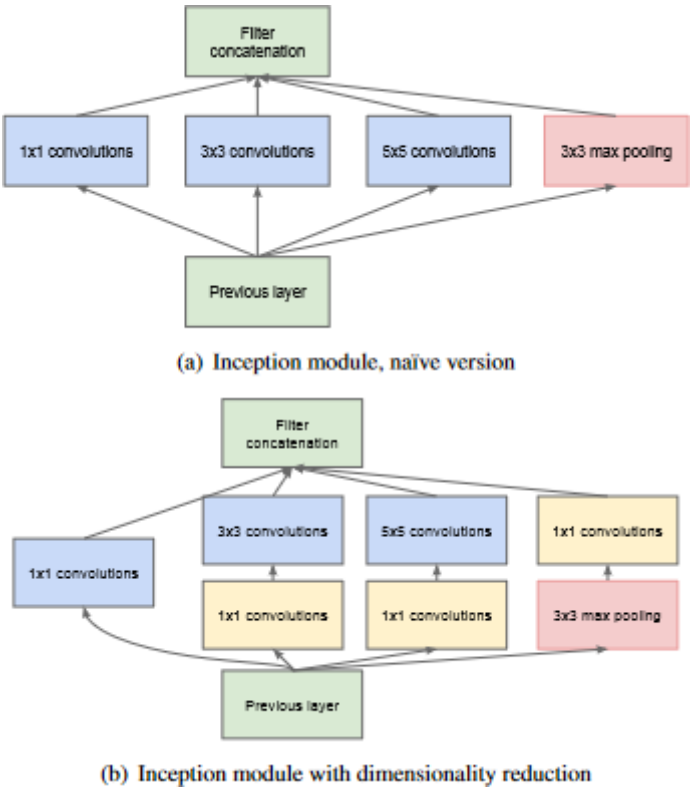

Fig2. インセプションモジュール

Inceptionアーキテクチャの主なアイデアは、畳み込みビジョンネットワークの最適な局所的疎構造が、どのように近似され、容易に入手可能な密なコンポーネントでカバーされるかを考えることです。翻訳不変性を仮定することは、ネットワークが畳み込みビルディングブロックから構築されることを意味することに注意してください。必要なのは、最適な局所構造を見つけ、それを空間的に繰り返すことです。Aroraら[2]は、レイヤーごとに構築する方法を提案しています。この方法では、最後のレイヤーの相関統計を分析し、相関性の高いユニットのグループにクラスタリングします。これらのクラスターが次の層のユニットとなり、前の層のユニットと接続される。ここでは、前の層の各ユニットが入力画像のある領域に対応していると仮定し、これらのユニットをフィルタバンクにグループ化します。下の層（入力に近い層）では、相関関係のあるユニットが局所的な領域に集中します。そのため、[12]で提案されているように、1つの領域に多くのクラスタが集中してしまい、それらは次の層の1x1の畳み込みの層でカバーすることができます。しかし、より大きなパッチ上の畳み込みでカバーできる、より空間的に広がったクラスターの数は少なくなり、より大きな領域上のパッチの数は減っていくことも予想されます。パッチアライメントの問題を回避するために、現在のInceptionアーキテクチャは、フィルタサイズが1x1、3x3、5x5に制限されています。この決定は、必要性というよりも利便性に基づいています。この決定は、必要性よりもむしろ利便性に基づいています。また、提案されているアーキテクチャは、これらすべての層とそれらの出力フィルターバンクを組み合わせて、次のステージの入力となる単一の出力ベクトルにしたものです。さらに、現在の畳み込みネットワークの成功にはプーリング処理が不可欠であることから、各段に別の並列プーリング経路を追加することで、さらに有益な効果が得られると考えられます（図2（a）参照）。

これらの「インセプション・モジュール」を重ねていくと、出力の相関統計が変化していきます。抽象度の高い特徴が上位のレイヤーに取り込まれると、その空間的な集中度は低下すると考えられます。このことから、3x3と5x5の畳み込みの比率は、高レイヤーになるほど高くなるはずです。

上記のモジュールの大きな問題点は、少なくともこのような素朴な形では、適度な数の5x5の畳み込みであっても、多数のフィルターを持つ畳み込み層の上では、法外にコストがかかるということです。この問題は、プーリングユニットが加わると、さらに顕著になります。出力フィルターの数は、前のステージのフィルターの数に等しくなります。プーリング層の出力と畳み込み層の出力を合成すると、ステージごとに出力数が増えるのは避けられない。このアーキテクチャは、最適なスパース構造をカバーできるかもしれませんが、非常に非効率的であり、数ステージで計算量が爆発してしまいます。

これは、Inceptionのアーキテクチャの2つ目のアイデアにつながります。つまり、そうしないと計算量が増えすぎてしまうような場合には、賢明に次元を減らすということです。これは、エンベッディングの成功に基づいています。低次元のエンベッディングであっても、比較的大きな画像パッチに関する多くの情報を含んでいる可能性があります。しかし，エンベッディングは，情報を高密度に圧縮して表現しており，圧縮された情報は処理が困難です．表現は（[2]の条件で要求されているように）ほとんどの場所でスパースに保たれ、信号を一括して集約しなければならない場合にのみ、信号を圧縮する必要があります。つまり、1x1の畳み込みは、高価な3x3や5x5の畳み込みの前にリダクションを計算するために使用されます。また、リダクションとして使用するだけでなく、整流された線形活性化の使用も含まれており、二重の目的で使用されています。最終的な結果は、図2(b)のようになります。

一般的に、インセプション・ネットワークは、上記のタイプのモジュールを重ねて構成されたネットワークであり、時折、グリッドの解像度を半分にするためにストライド2のマックス・プーリング層があります。技術的な理由（学習時のメモリ効率）から、インセプション・モジュールは高次の層でのみ使用し、低次の層は従来の畳み込み方式のままにしておくことが有益だと考えました。これは厳密には必要なことではなく、現在の実装ではインフラ的に非効率な部分があるためです。

このアーキテクチャの有用な点は、各ステージのユニット数を大幅に増やしても、後のステージで計算量が無秩序に増加しないことです。これは、パッチサイズを大きくして高価な畳み込みを行う前に、次元削減を行うことで実現しています。さらに、視覚情報はさまざまなスケールで処理され、次のステージで異なるスケールの特徴を同時に抽出できるように集約されるべきだという実用的な直感に基づいて設計されています。

また、計算機資源の利用効率が向上したことで、計算上の困難に陥ることなく、各ステージの幅とステージ数の両方を増やすことができます。Inceptionのアーキテクチャを利用して、少し劣っていても計算量の少ないバージョンを作ることができます。我々は、利用可能なすべてのノブとレバーを使って、計算資源のバランスを制御することで、Inceptionアーキテクチャを使用していない同様の性能のネットワークよりも3〜10倍高速なネットワークを実現できることを発見しましたが、現時点では慎重な手動設計が必要です。

5. GoogLeNet

"GoogLeNet" という名前は、ILSVRC 2014のコンペティションに提出したInceptionアーキテクチャの特定の形を指しています。また、より深くて広いInceptionネットワークも使用しましたが、このネットワークをアンサンブルに加えても、結果はわずかにしか改善されませんでした。しかし，アンサンブルを追加しても，結果はわずかにしか改善されないようだ．経験的に，ex-actのアーキテクチャ・パラメータの影響は比較的小さいことが示唆されているので，そのネットワークの詳細は省略する．表1は、コンペティションで使用されたInceptionの最も一般的なインスタンスを示している。このネットワーク（異なるイメージパッチサンプリング法で学習）は，アンサンブルの7つのモデルのうち，6つのモデルに使用された．

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

表1. GoogLeNetのインセプション・アーキテクチャーを具現化したもの。

Inceptionモジュールを含むすべての畳み込みは，整流された線形活性化を使用している．このネットワークの受容野のサイズは，RGB色空間で224×224，平均値はゼロである．"3x3 reduce」と「#5x5 reduce」は、3x3および5x5の畳み込みの前に使用されるリダクション層の1x1フィルタの数を表して

います。また、「プール・プロジェクション・レイヤー」の欄には、内蔵された最大プール後のプロジェクション・レイヤーにおける1x1フィルターの数が表示されています。これらのリダクション／プロジェクション層はすべて、整流された線形活性化を使用しています。ネットワークは、パラメータを持つ層のみをカウントすると22層の深さになります（プーリングもカウントすると27層）。ネットワークの構築に使われたレイヤー（独立したビルディングブロック）の数は全体で約100。正確な数は、機械学習のインフラでレイヤーがどのようにカウントされているかによります。分類器の前に平均プーリングを使用することは、[12]に基づいていますが、私たちの実装では線形層を追加しています。線形層は、ネットワークを他のラベルセットに簡単に適応させることができますが、これはほとんど利便性のために使用されており、大きな効果は期待できません。完全連結層から平均プーリングに移行することで、トップ1の精度が約0.6%向上することがわかったが、完全連結層を削除してもドロップアウトの使用は必須であることがわかった。

ネットワークの深度が比較的大きい場合、勾配をすべての層に効果的に伝搬させることができるかどうかが懸念された。この課題では、浅いネットワークが高い性能を発揮していることから、ネットワークの中間層で生成される画像が非常に識別性の高いものであると考えられます。この中間層に補助的な分類器を接続することで、下位の分類器での識別性が期待できます。これにより、正則化を行いながら、消失勾配の問題を解決できると考えられました。これらの分類器は、Inception(4a)と(4d)モジュールの出力の上に置かれた、より小さな畳み込みネットワークの形をしています。学習時には、補助分類器の損失は、ネットワークの総損失に割引の重みを付けて追加されます（補助分類器の損失は0.3で重み付けされました）。推論時には、これらの補助的なネットワークは破棄される。後の制御実験では、補助ネットワークの効果は比較的小さく（約0.5%）、同じ効果を得るためには1つの補助ネットワークしか必要ないことが示された。

補助分類器を含む側の余分なネットワークの正確な構造は、以下の通りです。

- 5x5のフィルターサイズ、ストライド3の平均プーリング層で、(4a)ステージでは4x4x512、(4d)ステージでは4x4x528の出力が得られます。
- 128個のフィルターを用いた1x1の畳み込みにより、次元の縮小と整流された線形活性化が行われます。
- 1,024個のユニットを持つ完全連結層、整流線形活性化
- ドロップアウト層（ドロップアウト率70%）。
- 分類器としてsoftmax lossを用いた線形層（メインの分類器と同じ1000個のクラスを事前に予測するが、推論時には削除される）。

結果として得られたネットワークの概略図を図3に示します。

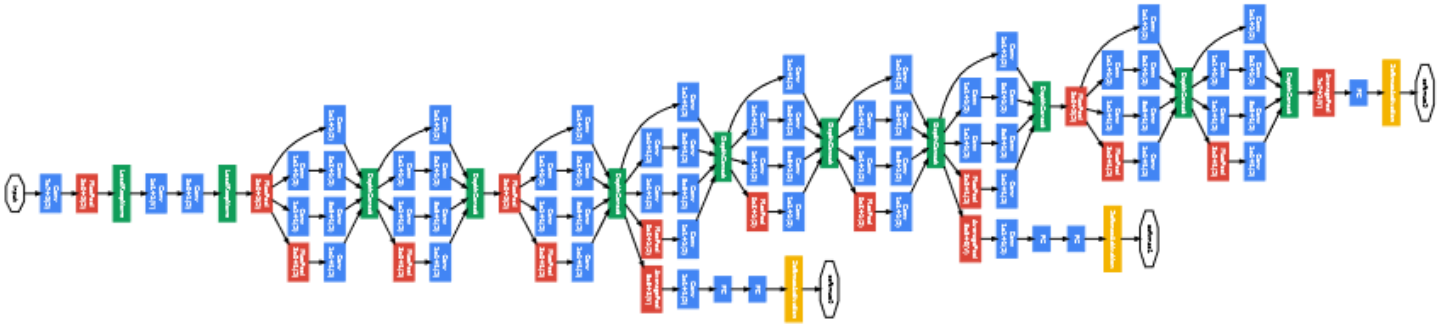

Fig3. あらゆる機能を備えたGoogLeNetネットワーク。

6. トレーニング方法

GoogLeNetネットワークは、DistBelief [4]分散型機械学習システムを用いて、適度なモデルとデータの並列性を用いて学習されました。ここではCPUベースの実装のみを使用していますが、概算ではハイエンドGPUを数台使用してGoogLeNetネットワークを1週間以内に収束させることができると考えられます。学習には、0.9 momen-tum [17]の非同期確率的勾配降下法を用い、固定学習率スケジュール（8エポックごとに学習率を4%ずつ下げる）を採用しました。推論時に使用する最終モデルの作成には，Polyak averaging [13]を用いた.

画像のサンプリング方法は、大会までの間に大幅に変更されており、すでに収束したモデルを別のオプションで学習したり、時にはドロップアウトや学習率などのハイパーパラメータを変更して併用したりしました。そのため、これらのネットワークを訓練する最も効果的な単一の方法について、決定的な指針を与えることは困難です。さらに複雑なのは，[8]にヒントを得て，主に相対的に小さい作物で学習したモデルと，大きい作物で学習したモデルがあることです．しかし、コンテスト後に非常にうまく機能することが確認された処方箋の1つに、サイズが画像領域の8%から100%の間で均等に分布し、アスペクト比が区間 $\left[\frac{3}{4}, \frac{4}{3}\right]$ に制約された、様々なサイズの画像パッチのサンプリングがあります。また，アンドリュー・ハワード（Andrew Howard）[8]のフォトメトリック・ディストーションは，学習データの撮影条件に対するオーバーフィッティングに有効であることがわかりました。

7. ILSVRC2014分類チャレンジの設定と結果

ILSVRC 2014の分類課題では、画像をImagenetの階層にある1000のリーフノードカテゴリーの1つに分類するというタスクがあります。トレーニング用に約120万枚の画像、検証用に5万枚の画像、テスト用に10万枚の画像が用意されています。各画像は、1つのグランドトゥルースカテゴリに関連付けられており、パフォーマンスは、最高スコアの分類器予測に基づいて測定されます。通常，2つの数値が報告されます．1つは，最初に予測されたクラスに対するグラウンドトゥルースの比較で，もう1

つは，最初に予測された5つのクラスに対するグラウンドトゥルースの比較で，トップ5エラーレートです．このチャレンジでは、トップ5エラーレートをランキングに使用しています。

このチャレンジには、外部データを使用せずに参加しました。本稿で紹介した学習手法に加えて、テスト時には、より高い性能を得るための手法を採用したので、以下に紹介する。

1. 同じGoogLeNetモデルの7つのバージョン（ワイドバージョン1つを含む）を独立して学習し，それらを用いてアンサンブル予測を行いました。これらのモデルは，同じ初期化（見落としがあったため，初期重みも同じ）と学習率ポリシーで学習されました。これらのモデルの違いは，サンプリング方法と入力画像の順序がランダムであることだけです．
2. テストでは，Krizhevskyら[9]の手法よりも積極的な切り抜き方を採用しました．具体的には，画像を，短い方の寸法（縦または横）がそれぞれ256，288，320，352となる4つのスケールにリサイズし，これらのリサイズされた画像の左，中央，右の正方形を取ります（ポートレート画像の場合は，上，中央，下の正方形を取ります）。そして、それぞれの正方形について、4つの角と中央の224x224のクロップ、224x224にリサイズされた正方形、およびそれらのミラーリングされたバージョンを取ります。これにより、1枚の画像につき4x3x6x2=144の切り抜きが行われます。同様の手法は，前年の応募作品であるAndrew Howard [8]でも採用されていましたが，我々は経験的に，提案した方式よりも若干性能が悪いことを確認しました．実際のアプリケーションでは、このような積極的なトリミングは必要ないかもしれません。というのも、適切な数のクロップが存在した後は、より多くのクロップの利点は限界に達するからです（後で示すように）。
3. このソフトマックス確率を，複数の作物およびすべての分類器に対して平均化し，最終的な予測値を得ることができます．我々の実験では、作物に対するmax poolingや分類器に対する平均化など、検証データに対する別のアプローチを分析しましたが、単純な平均化よりも劣った性能になってしまいました。

本論文の残りの部分では、最終的な提出物の全体的なパフォーマンスに寄与する複数の要因を分析します。

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Tab2. 分類性能。

我々が最終的に提出した課題では，検証データとテストデータの両方で6.67%のトップ5エラーを獲得し，他の参加者の中で第1位となりました。これは，2012年のSuperVisionと比較して56.5%の相対的な低減，および前年の最優秀アプローチ（Clarifai）と比較して約40%の相対的な低減であり，いずれも分類器の学習に外部データを使用していました。表2は、過去3年間で最も優れた成績を収めたいくつかのアプローチの統計を示しています。

Number of models	Number of Crops	Cost	Top-5 error	compared to base
1	1	1	10.07%	base
1	10	10	9.15%	-0.92%
1	144	144	7.89%	-2.18%
7	1	7	8.09%	-1.98%
7	10	70	7.62%	-2.45%
7	144	1008	6.67%	-3.45%

Tab3. GoogLeNetの分類性能の内訳。

また，表3では，画像を予測する際に使用するモデルの数と作物の数を変化させて，複数のテスト選択の性能を分析し，報告しています。1つのモデルを使用する場合は、検証データ上でトップ1エラーレートが最も低いモデルを選択しました。テストデータの統計量にオーバーフィットしないように、すべての数値は検証データで再計算されています。

8. ILSVRC2014検出チャレンジの設定と結果

ILSVRCの検出タスクは，画像中の物体を囲むバウンディングボックスを，可能性のある200のクラスから作成することである．検出されたオブジェクトが，グランドトゥルースのクラスと一致し，かつ，それらのバウンディングボックスが少なくとも50%（Jaccard indexを使用）オーバーラップしていれば，正解とみなされる．余分な検出は偽陽性とみなされ，ペナルティが課せられます．分類タスクとは異なり，各画像には多くのオブジェクトが含まれている場合もあれば，全く含まれていない場合もあり，そのスケールも様々です．結果は、平均平均精度（mAP）を用いて報告されます。GoogLeNetの検出手法は、[6]のR-CNNに似ていますが、領域分類器としてInceptionモデルを追加しています。さらに、領域提案のステップでは、選択的探索（selective search）[20]とマルチボックス（multi-box）[5]の予測を組み合わせることで、オブジェクトのバウンディングボックスの再現性を高めています。偽陽性の数を減らすために，スーパーピクセルのサイズを2倍にしました．これによ

り、選択的探索アルゴリズムから得られる提案が半分になりました。さらに、マルチボックス[5]から得られた200個の領域提案を追加した結果、[6]で使用された提案の約60%となり、カバー率は92%から93%に向上しました。提案の数を減らしてカバー率を上げたことによる全体的な効果は、単一モデルのケースで平均平均精度が1%向上したことです。最後に、各地域を分類する際に、6つのGoogLeNetsのアンサンブルを使用します。これにより、精度が40%から43.9%に向上しました。なお、R-CNNとは異なり、時間がなかったため、バウンディングボックス回帰は使用しませんでした。

Team	Year	Place	mAP	external data	ensemble	approach
UvA-Euvision	2013	1st	22.6%	none	?	Fisher vectors
Deep Insight	2014	3rd	40.5%	ImageNet 1k	3	CNN
CUHK DeepID-Net	2014	2nd	40.7%	ImageNet 1k	?	CNN
GoogLeNet	2014	1st	43.9%	ImageNet 1k	6	CNN

Tab4. 検知性能の比較。報告されていない値にはクエスチョンマークを付けています。

最初に、トップの検出結果を報告し、検出タスクの第1回目からの進歩を示します。2013年の結果と比較すると、精度はほぼ倍増しています。上位のチームはすべて、畳み込みネットワークを使用しています。表4に公式スコアと、各チームに共通する戦略（外部データの利用、アンサンブルモデル、コンテクストモデル）を報告します。外部データとしては，ILSVRC12の分類データを用いてモデルの事前学習を行い，その後，検出データを用いてモデルを改良することが多い．また、一部のチームは、ローカライゼーションデータの使用についても言及しています。ローカライズタスクのバウンディングボックスの大部分は、検出データセットには含まれていないので、分類を事前学習に使用するのと同じように、このデータを使って一般的なバウンディングボックスのリグレッサーを事前学習することができます。GoogLeNet のエントリーでは、事前トレーニングにローカライゼーション・データを使用していません。

Team	mAP	Contextual model	Bounding box regression
Trimps-Soushen	31.6%	no	?
Berkeley Vision	34.5%	no	yes
UvA-Euvision	35.4%	?	?
CUHK DeepID-Net2	37.7%	no	?
GoogLeNet	38.02%	no	no
Deep Insight	40.2%	yes	yes

Tab5. 検出時のシングルモデルの性能

表5では、単一のモデルのみを使用した結果を比較しています。表5では、単一のモデルのみを使用した結果を比較しています。最も高いパフォーマンスを示したのはDeep Insightのモデルで、驚くべきことに、3つのモデルのアンサンブルでは0.3ポイントしか向上しませんでしたが、GoogLeNetはアンサンブルで非常に高い結果を得ています。

9. 結論

その結果、コンピュータビジョン用のニューラルネットワークを改良するためには、期待される最適な疎な構造を、容易に入手可能な密な構成要素で近似することが有効であるという確かな証拠が得られました。この方法の主な利点は、浅くて狭いアーキテクチャに比べて、わずかな計算量の増加で大きな品質向上が得られることです。

我々の物体検出は、コンテキストを利用せず、バウンディングボックス回帰も行っていないにもかかわらず、競争力がありました。これは、Inceptionアーキテクチャの強みをさらに証明しています。分類と検出の両方において、同様の深さと幅を持つ、より高価な非Inceptionタイプのネットワークでも、同様の品質の結果が得られると予想されます。しかし、我々のアプローチは、スパース・アーキテクチャへの移行が一般的に実現可能で有用なアイデアであることを示す確かな証拠を示している。このことは、[2]に基づいて、より粗く洗練された構造を自動生成することや、Inceptionアーキテクチャの洞察を他のドメインに適用することに向けた将来の研究を示唆しています。