コンピュータビジョンのためのインセプション・アーキテクチャの再考

クリスチャン・セゲディ　ヴィンセント・ヴァンフーク
セルゲイ・イオフェ　ジョン・シュレン
Google Inc.　vanhoucke@google.com　sioffe@google.com　shlens@google.com
szegedy@google.com

Zbigniew Wojna
ユニバーシティ・カレッジ・ロンドン
zbigniewwojna@gmail.com

アブストラクト

畳み込みネットワークは、多種多様なタスクに対応する最先端のコンピュータビジョンソリューションのほとんどの核となっています。2014年以降、非常に深い畳み込みネットワークが主流になり始め、様々なベンチマークで大きな成果を上げています。モデルサイズと計算コストの増加は、ほとんどのタスクですぐに品質の向上につながる傾向がありますが（学習用に十分なラベル付きデータが提供されている限り）、計算効率とパラメータ数の少なさは、モバイルビジョンやビッグデータシナリオなどの様々なユースケースを可能にする要因となっています。ここでは、適切に因数分解された畳み込みと積極的な正則化によって、追加された計算を可能な限り効率的に利用することを目的とした、ネットワークの拡張方法を探求しています。我々の手法をILSVRC
2012の分類チャレンジ検証セットでベンチマークしたところ、1フレームの評価でトップ1 に 21.2%、トップ5 に 5.6%の誤差が生じ、推論ごとに50億回の乗算加算を行う計算コストと2500万個以下のパラメータを使用するネットワークを用いた場合には、現状よりも大幅に改善されました。また，4 つの
モデルのアンサンブルとマルチクロップ評価では，検証セットで3.5%トップ5エラー，17.3%トップ1エラー，公式テストセットで3.6%トップ5エラーを報告しています.

1.はじめに

2012年のImageNetコンペティション[16]でKrizhevskyら[9]が優勝して以来，彼らのネットワーク "AlexNet
"は，物体検出[5]，セグメンテーション[12]，人間の姿勢推定[22]，ビデオ分類[8]，物体追跡[23]，超解像[3]など，より多様なコンピュータビジョンのタスクに応用されている.

これらの成功を受けて、より高性能な畳み込みニューラルネットワークの研究が進められました。2014年からは、より深く、より広いネットワークを利用することで、ネットワーク・アーキテクチャーの品質が大幅に向上しました。VGGNet [18]とGoogLeNet [20]は，2014年のILSVRC
[16]の分類チャレンジでも，ほぼ同様に高い性能を示しました．興味深いのは，分類性能の向上が，さまざまな応用分野での大幅な品質向上につながる傾向があることだ．これは，深層畳み込みアルゴリズムのアーキテクチャを改善することで，学習した高品質な視覚的特徴にますます依存する他のほとんどのコンピュータビジョンタスクの性能向上に利用できることを意味している．また，ネットワークの品質が向上したことで，AlexNet
の特徴が手作業で作成されたソリューションに太刀打ちできなかった場合に，畳み込みネットワークの新たな応用分野が生まれました（例：検出における提案の生成）[4].

VGGNet
[18]は、アーキテクチャがシンプルであるという魅力的な特徴を持っていますが、ネットワークの評価には多くの計算が必要になるという高いコストがかかります。一方、GoogLeNet

[20]のInceptionアーキテクチャは、メモリや計算量に厳しい制約があっても、十分な性能を発揮できるように設計されています。例えば，GoogleNetでは約700万個のパラメータを使用しており，前身のAlexNetでは6,000万個のパラメータを使用していたのに対し，9 倍の削減となっています．さらに、VGGNetはAlexNetの約3倍のパラメータを使用しています。

また、Inceptionの計算コストは、VGGNetやその上位機種に比べてはるかに低くなっています[6]。このため、Inceptionネットワークは、大量のデータを低コストで処理する必要があるビッグデータ・シナリオ[17]、[13]や、モバイル・ビジョンのようにメモリや計算能力が本質的に制限されているシナリオでの利用が可能になりました。このような問題の一部を軽減するには、メモリ使用量の削減に特化したソリューションを適用したり[2]、[15]、計算トリックを用いて特定の処理の実行を最適化したりすることが可能です[10]。しかし，これらの方法は複雑さを増します．さらに、これらの手法をInceptionのアーキテクチャの最適化にも適用することで、効率性の差が再び拡大する可能性があります。

しかし、Inceptionのアーキテクチャは複雑であるため、ネットワークの変更が難しくなっています。アーキテクチャーを単純にスケールアップすると、せっかく得られた計算能力の大部分がすぐに失われてしまいます。また、[20]では、GoogLeNetアーキテクチャの様々な設計上の決定につながる要因について、明確な説明がなされていません。そのため、効率性を維持しながら新しいユースケースに適応させることが非常に困難になっています。例えば、Inceptionスタイルのモデルの容量を増やす必要があると考えた場合、フィルターバンクのサイズを2倍にするだけで、計算コストとパラメータの数が4倍になってしまいます。これは、多くの実用的なシナリオにおいて、特に関連する利益がわずかな場合には、法外なことであることがわかります。この論文では、まず、畳み込みネットワークを効率的にスケールアップするために有効な、いくつかの一般的な原理と最適化のアイデアを説明しています。この原理はInception型ネットワークに限定されたものではないが、Inception型のビルディングブロックの一般的な構造は、これらの制約を自然に取り入れるのに十分な柔軟性を持っているため、その文脈では観察が容易である。これは、Inceptionモジュールが次元削減と並列構造を多用しているためで、構造変更が近隣のコンポーネントに与える影響を緩和することができます。ただし、モデルの品質を高く保つためには、いくつかの指針を守る必要があるため、慎重に行う必要があります。

2.デザインの基本方針

ここでは、畳み込みネットワークのさまざまなアーキテクチャの選択に関する大規模な実験に基づいて、いくつかの設計原則を説明します。現時点では、これらの原則の有用性は推測の域を出ず、その妥当性を評価するためには、さらに実験的な証拠が必要です。しかし、これらの原則から大きく外れると、ネットワークの品質が低下する傾向があり、そのような逸脱が発見された場合には、アーキテクチャの改善が行われました。

1. 表現上のボトルネックを避ける（特にネットワークの初期）。フィードフォワードネットワークは、入力層から分類器または回帰器までの非周期的なグラフで表現することができます。これにより、情報の流れの方向性が明確になります。入力と出力を分離するどのような切り口でも、切り口を通過する情報量にアクセスできます。極端な圧縮によるボトルネックは避けるべきである。一般的に、表現サイズは入力から出力に向かって緩やかに減少し、最終的にはタスクに使用される表現に到達するべきであるとされています。理論的には、表現の二次元性だけで情報量を評価することはできない。二次元性は、相関構造などの重要な要素を排除してしまうからだ。

2. 高次元の表現は、ネットワーク内で局所的に処理することが容易である。畳み込みネットワークのタイルあたりのアクティベーションを増やすことで、より多くの特徴を分離することができます。その結果、ネットワークの学習速度が向上します。

3. 空間的な集約は、より低次元の埋め込みに対しても、表現力を損なうことなく行うことができます。例えば、より広範な（例えば 3 ×
3を実行する前に、入力表現の次元を下げてから空間的集約を行っても、深刻な逆効果にはならない。その理由は、隣接するユニット間に強い相関関係があるため、出力が空間的なアグレガシオンのコンテキストで使用される場合、次元の再縮小時に情報の損失が非常に少なくなるからであると考えています。これらの信号は簡単に圧縮できるはずなので、次元の縮小はより速い学習を促進することになります。

4. ネットワークの幅と深さのバランスをとる。ネットワークの性能を最適化するには、ステージごとのフィルター数とネットワークの深さのバランスをとる必要があります。ネットワークの幅と深さの両方を増やすことで、より高品質なネットワークを構築することができます。しかし、限られた計算量の中で最適な改善を行うには、両方を並行して増やすことが必要です。そのため、ネットワークの深さと幅の間でバランスよく計算量を配分する必要があります。

これらの原則は理にかなっているかもしれませんが、これを使ってネットワークの質を向上させるのは容易ではありません。曖昧な状況でのみ判断して使用することが大切です。

3. 大きなフィルタサイズでのコンボリューションの因数分解

GoogLeNetネットワーク[20]の元々の利益の多くは、Linらによる「Network in network」アーキテクチャのように、次元再生産を非常に寛大に使用したことから生じています。これは、計算効率の良い方法で畳み込みを行う特殊なケースと考えることができます。例えば、1×1 の畳み込み層の後に、3×3 の畳み込み層がある場合を考えてみましょう。視覚ネットワークでは、近くにある活性化の出力は高い相関があると予想されます。そのため、集約する前に活性度を下げることで、同じような表現の局所表現が得られることが期待できます。

ここでは、さまざまな設定で畳み込みを因数分解する他の方法を探り、特に解の計算効率を高めることを目的としています。Inceptionのネットワークは完全な畳み込み式であるため、各重みは活性化ごとに1つの乗算に対応します。したがって、計算コストの削減は、パラメータの数の削減につながります。つまり、適切な因数分解を行えば、より多くのパラメータを分離することができ、その結果、高速な学習が可能になります。また，計算量とメモリを節約することで，1台のコンピュータで各モデルのレプリカを学習する能力を維持しながら，ネットワークのフィルタバンクのサイズを大きくすることができます.

3.1.より小さな畳み込みへの因数分解

より大きな空間フィルターを用いたコンボリューション（例えば 5×5や 7×7)のコンボリューションは、計算量が不均衡になる傾向があります。例えば，m 個のフィルタを持つグリッド上に n 個のフィルタを持つ 5 × 5 コンボリューションは，同じ数のフィルタを持つ 3 × 3 コンボリューションと比較して，25/9 = 2.78 倍の計算量となる．もちろん、5×5 のフィルターは、前の層のより遠くにあるユニットの活性化の信号間の依存性を捉えることができるので、フィルターの幾何学的なサイズを小さくすることは、表現力を大きく犠牲にすることになります。しかし、5×5 の畳み込みを、同じ入力サイズと出力の深さで、より少ないパラーメータの多層ネットワークで置き換えることができるかどうかを問うことができます。5×5 畳み込みの計算グラフを拡大してみると、各出力は、5×5 のタイルを入力上にスライドさせた小さな完全連結ネットワークのように見えます（図1参照）。ここでは視覚ネットワークを構築しているので、翻訳不変性を利用して、完全連結コンポーネントを2層の畳み込み構造に置き換えるのが自然だと思われます。第1層は3×3 の畳み込みで、第2層は第1層の3×3 の出力グリッドの上にある完全連結層です（図1参照）。この小さなネットワークを入力活性化グリッドにスライドさせると、5×5 の畳み込みを 2層の3×3 の畳み込みに置き換えることになります（図4と5を比較）。

この設定では、隣接するタイル間で重みを共有することで、パラメータ数を明らかに減らしています。予想される計算コストの削減を分析するために，典型的な状況に適用されるいくつかのシミュレーションの仮定を行います．n = αmと仮定することができます。つまり、アクティベーション/ユニットの数を一定のαファクターで変化させたいということです。5×5 の畳み込みは集約されているので、αは通常1よりわずかに大きくなります（GoogLeNetの場合は約1.5）。5×5 層を2層に置き換えた場合、この拡張を2つのステップで行うのが妥当だと思われます。$\sqrt{\alpha}$の増加となります。推定値を簡単にするために $\alpha = 1$(拡張なし)を選択すると、このネットワークは、隣接するタイル間の活性化を再利用する2つの3×3 の畳み込み層で表現することができます。この方法では、最終的に純 $\frac{9+9}{25}$の削減となり、28%の相対的な利益を得ることができました。各パラメータは、各ユニットの活性化を計算する際にちょうど1回だけ使用されるため、パラメータ数についてもまったく同じように削減できます。しかし、この設定では2つの一般的な疑問があります。この置き換えによって，表現力が失われることはないのか？我々の主な目的が計算の線形部分を因数分解することであるならば，線形活性化を第1層にとどめておくことが望ましいのではないか？私たちはいくつかの制御実験を行いましたが（例えば図2を参照）、線形活性化を使用すると、因数分解のすべての段階で、整流された線形ユニットを使用するよりも常に劣っていました。これは、出力活性化を一括正規化[7]することで、ネットワークが学習できるバリエーションの空間が広がったことによるものです。次元削減コンポーネントに線形活性化を使用しても、同様の効果が見られます。

3.2.非対称コンボリューションへの空間的因数分解

以上の結果から、3 × 3 a 以上のフィルタを持つ畳み込みは、常に3 × 3
畳み込み層のシーケンスに縮小できるため、一般的には有用ではないと考えられます。しかし、これをもっと小さな、たとえば2×2 の畳み込みに因数分解すべきではないか、という疑問があります。例えば、3×1 の畳み込みの後に1×3 の
畳み込みを行うと、3×3 の
畳み込みと同じ受容野を持つ2層のネットワークをスライドさせることになります（図3参照）。入力フィルターと出力フィルターの数が同じであれば、同じ数の出力フィルターでも2層の方が33%安くなります。これに比べて，3×3
コンボリューションを2×2 コンボリューションに因数分解すると，計算量が11％しか減りません.
理論的には、n × n の畳み込みを、1 × n の畳み込みと n × 1 の畳み込みに置き換えることができ、n
が大きくなるにつれて計算コストが劇的に削減されることになります（図 6
参照）。実際には、この因数分解を採用すると、初期の層ではうまくいかないが、中程度のグリッドサイズ（m×m の
特徴マップで、m は 12から20の範囲）では非常に良い結果が得られることがわかった。その場合、1×7 回の
畳み込みと7×1 回の畳み込みを行うことで、非常に良い結果が得られます。

4.補助的な分類法の有用性

20]は，非常に深いネットワークの収束性を向上させるために，補助的な分類器の概念を導入しました．当初の動機は、有用な勾配を下層に押し付けてすぐに使えるようにし、非常に深いネットワークの消失勾配問題に対処して学習中の収束性を改善することでした。また、Lee et
al[11]は、補助的な分類器は、より安定した学習と収束を促進すると主張しています。興味深いことに、学習の初期段階では、補助分類器は収束性の向上につながらないことがわかりました。学習の終わり近くになると、補助枝を持つネットワークは、補助枝を持たないネットワークの精度を追い越し始め、わずかに高いプラトーに達します。

また、[20]では、ネットワークの異なる段階で2つのサイドヘッドを使用しています。下側の補助枝を削除しても、ネットワークの最終的な品質には何の悪影響もありませんでした。先ほどのパラグラフの観察結果と合わせて考えると、これらのブランチが低レベルの特徴を進化させるのに役立つという[20]のオリジナルの仮説は、ほとんど見当違いであることがわかります。その代わりに、補助的なクラシファイヤーがレギュラライザーとして機能していることを主張します。このことは、側枝がバッチ正規化されていたり[7]、ドロップアウト層を持っていたりすると、ネットワークの主分類器の性能が向上するという事実によって裏付けられます。このことは、バッチ正規化が正則化として働くという推測を弱く裏付ける証拠にもなります。

5.効率的なグリッドサイズの縮小

従来、畳み込みネットワークでは、特徴マップのグリッドサイズを小さくするために、何らかのプーリング操作を行っていました。表現上のボトルネックを回避するために、最大または平均プーリングを適用する前に、ネットワークフィルタの活性化次元を拡張します。例えば、k 個の
フィルタを持つd×dグリッドからスタートして、2k個のフィルタを持つグリッドに到達したい場合は $\frac{d}{2}\times$
$\frac{d}{2}$例えば，k個のフィルタを持つd×d個のグリッドから，2k 個のフィルタを持つグリッドに到達したい場合，まず2k 個の
フィルタでstride-1 con
volutionを計算し，さらにプーリングステップを適用する必要があります．つまり，全体の計算コストは，より大きなグリッドでの
$2d^2k^2$演算を用いた大きなグリッドでのコンボリューションが全体の計算コストの大半を占めています。畳み込みとプーリングを同時に行うことで、計算コストを4分の1に削減することができます。
$2\left(\frac{d}{2}\right)^2 k^2$計算コストを1/4に減らすことができます。しかし、これは表現上のボトルネックとなり、表現の全体的な次元が
$\left(\frac{d}{2}\right)^2 k$その結果、ネットワークの表現力が低下します（図9参照）。このような方法ではなく、表現上のボトルネックを取り除きつつ、計算コストをさらに削減する別の方法を提案しています。図10 参照）。2 つの並列ストライド 2 ブロックを使うことができます。 Pと C.
Pはプーリング層（平均プーリングまたは最大プーリング）で、両方ともストライド2のフィルターバンクを図10のように連結したものです。

6.インセプション-V3

ここでは、上記の点をつなぎ合わせて、ILSVRC
2012の分類ベンチマークで性能を向上させた新しいアーキテクチャを提案します。我々のネットワークのレイアウトを表1に示します。従来の7×7 の畳み込みを、3.1節で述べたのと同じ考え方に基づいて、3×3 の畳み込み
に因数分解しています。ネットワークのインセプション部分では、従来のインセプションモジュールを35×35に3 つ
配置し、それぞれ288 個の
フィルターを使用しています。これを、5章で説明したグリッドリダクション技術を用いて、17×17 のグリッドに768 個のフィルターを配置しました。この後、図5に示すように、因数分解されたインセプション・モジュールの5 つの
インスタンスが続きます。これを、図10のグリッドリダクション技術を用いて、8×8×1280グリッドに縮小しました。最も粗い8×8レベルでは、図6に示すように2つのインセプション・モジュールがあり、各タイルの出力フィルタ・バンク・サイズは2048であることがわかります。インセプション・モジュール内のフィルター・バンクのサイズを含むネットワークの詳細な構造は、本稿のtarファイルに含まれるmodel.txt に記載されて
いる補足資料に記載されています。しかし、第2章の原則を守っていれば、ネットワークの品質は比較的安定していることが確認できました。このネットワークは42層の深さがありますが、計算コストはGoogLeNetに比べて2.5 倍
程度であり、VGGNetよりもはるかに効率的です。

7.ラベルスムージングによるモデルの正則化

ここでは、学習中のラベルドロップアウトの限界効果を推定することで、分類器層を正則化するメカニズムを提案する。

各学習例に対して xこのモデルは、各ラベルの確率を計算します。 $k \in \{1 \ldots K\}: p(k|x) = \frac{exp(z_k)}{\sum_{i=1}^{k} exp(z_i)}$.ここで z_iは，logits または非正規化 log-probabilities である． この学習例のラベルに対するグランドトゥルース分布
$q(k|x)$を正規化したものを考えます． $\sum_k q(k|x) = 1$．簡潔にするために，以下の依存性を省略します。 pと qの例への依存性を省略します。 x．例の損失をクロスエントロピーと定義します。 $l = -\sum_{k=1}^{K} \log\big(p(k)\big)q(k)$.これを最小化することは，ラベルの期待対数尤度を最大化することと等価であり，ここでラベルはグランドトゥルース分布
$q(k)$.クロスエントロピー損失は，対数に対して微分可能です．
z_kに対して微分可能であるため，深層モデルの勾配学習に用いることができる．勾配は
は、かなり単純な形です。 $\frac{\partial l}{\partial z_k} = p(k) - q(k)$これは、-1と1 の間の境界線である。

単一のグランドトゥルース・ラベルがある場合を考えてみましょう。 yの場合、次のようになります。 $q(y) = 1$そして $q(k) = 0$すべての $k \neq y$.この場合，クロスエントロピーを最小化することは，正しいラベルの対数尤度を最大化することと等価である．ラベル y を持つ特定の実例 x について、対数尤度が最大になるのは $q(k) = \delta_{k,y}$ここで $\delta_{k,y}$は Dirac delta で、次の場合は 1、それ以外は 0 となる。 $k = y$の場合は1、それ以外の場合は 0 になる。この最大値は、有限の
z_kでは達成できないが、以下の場合には近づくことができる。 $z_y \gg z_k$すべての場合 $k \neq y$-
つまり、グランドトゥルースのラベルに対応するロジットが他のすべてのロジットよりもはるかに大きい場合です。しかし、これには2つの問題があります。第一に，オーバーフィッティングになる可能性があります．モデルが，各学習例に対してグランドトゥルースラベルに完全な確率を割り当てるように学習した場合，一般化することは保証されません．第二に、最大のロジットと他のすべてのロジットとの差が大きくなることを促し、これが制限された勾配と相まって
$\frac{\partial l}{\partial z_k}$と組み合わせることで、モデルの適応能力を低下させます。直感的に言えば、これはモデルが自分の予測に自信を持ちすぎてしまうために起こります。
我々は，モデルの自信をなくすように促すメカニズムを提案する．訓練ラベルの対数尤度を最大化することが目的であれば，このようなことは望ましくないかもしれないが，モデルを正則化し，適応性を高めることができる．方法はとても簡単です。ラベルの分布を考える $u(k)$の分布を考えます． xと平滑化パラメータ ϵ.真のラベルを持つ学習例に対して yの場合，ラベル分布を $q(k|x) = \delta_{k,y}$を

$$q'^{(k|x)} = (1 - \epsilon)\delta_{k,y} + \epsilon u(k)$$

これは、元のグランドトゥルース分布と固定分布の混合物である $q(k|x)$と固定分布 $u(k)$を混合したもので，重みは $1 - \epsilon$ と ϵである。これは，次のようにして得られるラベル kの分布と見なすことができます。まず、このラベルをグランドトゥ

ルースラベル $k = y$を設定し，次に，確率的に ϵに置き換える。 kという分布から抽出したサンプルで $u(k)$.ここでは，ラベルの事前分布を次のように用いることを提案する． $u(k)$.今回の実験では，一様分布 $u(k) = 1/K$

$$q'^{(k)} = (1-\epsilon)\delta_{k,y} + \frac{\epsilon}{K}$$

私たちは、このようなグランドトゥルースのラベル分布の変化を *ラベル平滑化正則化*（LSR）と呼んでいます。
LSR は、最大のロジットが他のロジットよりもはるかに大きくなることを防ぐという望ましい目標を達成していることに注意してください。実際、もしそうなってしまうと、1 つの $q(k)$が 1 に近づき、他のすべてが 0 に近づくことになります。$q'(k)$とは異なり $q(k) = \delta_{k,y}$とは異なり、すべての $q'(k)$は正の下界を持っているからです。
LSRの別の解釈は、クロスエントロピーを考慮することで得られます。

$$H(q', p) = -\sum_{k=1}^{K} \log p(k)\, q'^{(k)} = (1-\epsilon)H(q,p) + \epsilon H(u,p)$$

したがって、LSR は、1 つのクロスエントロピー損失を 2 つのクロスエントロピー損失に置き換えることに相当します。$H(q,p)$を一対の損失 $H(q,p)$そして $H(u,p)$.2 つ目の損失は，予測されたラベル分布 pからの予測ラベル分布の逸脱をペナルティとして与える。 uの偏差を，相対的な重み $\frac{\epsilon}{1-\epsilon}$.この偏差は，KL ダイバージェンスで同等に表現できることに注意してください． $H(u,p) = D_{KL}(u \parallel p) + H(u)$と $H(u)$は固定です。u が一様分布の場合
$H(u,p)$は、予測された分布pが一様分布にどれだけ似ているかの尺度であり、これも測定可能である（ただし

負のエントロピーによる $-H(p)$しかし、私たちはこの方法を試していません。

を用いた ImageNet の実験では $K = 1000$クラスを使った実験では $u(k) = 1/1000$としました。 $\epsilon = 0.1$.ILSVRC 2012では，トップ1エラーとトップ5エラーの両方で，絶対値で約0.2%の一貫した改善が見られた（表3を参照）．

8.トレーニング方法

TensorFlow [1] 分散型機械学習システムを用いて，50 個のレプリカを NVidia Kepler GPU 上で動作させ，バッチサイズ 32 で 100 回のエポックを行い，確率的勾配を用いてネットワークを学習しました．初期の実験ではmomentum [19]を0.9のディケイで使用していましたが，最良のモデルはRMSProp [21]を0.9 のディケイで使用して達成されました．$\epsilon = 1.0$.学習率は0.045 で，2エポックごとに0.94の指数関数的なレートでディケイした．また，学習を安定させるためには，閾値2.0の勾配クリッピング[14]が有効であることがわかった．モデルの評価は，時間をかけて計算されたパラメータの走行平均を用いて行われる．

9.低解像度入力時のパフォーマンス

ビジョンネットワークの典型的な使用例は、例えばMultibox [4]のように、検出の後に分類することです。これには、単一の物体を含む画像の比較的小さなパッチを、何らかのコンテキストで分析することが含まれます。課題は、パッチの中央部分が何らかのオブジェクトに対応しているかどうかを判断し、対応している場合はそのオブジェクトのクラスを決定することです。しかし、物体は比較的小さく、低解像度であることが多い。このため、低解像度の入力をどのように適切に処理するかという問題があります。

一般的には、高解像度の受容野を採用したモデルは、認識性能が大幅に向上する傾向にあります。しかし、ここで重要なのは、第1層の受容野の解像度が上がったことによる効果と、モデルの容量や計算量が大きくなったことによる効果を区別することです。モデルを調整せずに入力の解像度を変えるだけであれば、より難しい課題を解決するために、計算量がはるかに少ないモデルを使うことになってしまいます。もちろん、このような解決策は、計算量が少ないために、すでに失敗しているのは当然です。正確な評価を行うためには、モデルは曖昧なヒントを分析して、細かい部分を「幻覚」で見ることができるようにする必要があります。これには計算コストがかかります。そこで問題となるのは、計算量を一定に保った場合、入力解像度を上げることでどれだけの効果が得られるかということです。一定の労力を確保するための簡単な方法として、低解像度の入力の場合、最初の2つの層のストライドを減らすか、ネットワークの最初のプーリング層を単純に取り除くことができる。

そのために、以下の3つの実験を行いました。

1. ストライド2で299×299 の受容野を持ち、第1層の後に最大のプーリングが発生する。
2. ストライド1で151×151 の受容野、第1層以降は最大のプーリング。
3. ストライド1で79×79 の受容野を持ち、第1層の後にプーリング**が発生しない。**

この3つのネットワークの計算コストはほとんど同じです。3つ目のネットワークの方がわずかに安いですが，プーリング層のコストはわずかで，（ネットワークの総コストの1%
以内）です．いずれの場合も、ネットワークは収束するまで学習され、ImageNet ILSVRC 2012 classifica-tion benchmarkの検証セットでその品質が測定されました。その結果を表2に示します。低解像度のネットワークは学習に時間がかかりますが，最終的な結果の品質は高解像度のものに非常に近いものになります．

しかし、単純に入力解像度に応じてネットワークサイズを縮小した場合、ネットワークの性能はより低下します。しかし、これは16倍も安いモデルをより難しいタスクで比較していることになるので、不公平な比較となります。

また、表2のこれらの結果から、R-
CNN[5]のように、小さい物体には専用の高コスト低解像度ネットワークを使用することを検討してみてはいかがでしょうか。

10.実験結果と比較

表3は、セクション6で述べた提案アーキテクチャ（Inception- v2）の認識性能に関する実験結果です。Inception-v2の各行は、ハイライトされた新しい変更とそれ以前のすべての変更を含む累積的な変更の結果を示しています。ラベルスムージング」は、第7章で説明した方法です。Fac-torized 7 × 7」は、最初の7 × 7 畳み込み層を3 × 3 畳み込み
層の列に因数分解する変更を含む。BN-
auxiliaryは、畳み込みだけでなく、補助分類器の完全連結層もバッチノーマライズしたバージョンを指します。表3の最後の行のモデルをInception-v3と呼び、マルチクロップとアンサンブルの設定でその性能を評価します。

今回の評価では，[16]で提案されているように，ILSVRC-
2012検証セットの48238個の非ブラックリスト例を用いています．50000個のex-
amplesも同様に評価しましたが、結果はトップ5エラーで約0.1%、トップ1エラーで約0.2%悪化しました。本論文の次のバージョンでは、テストセットでのアンサンブル結果を検証する予定ですが、春にBN-
Inceptionを最後に評価した時点では、[7]は、テストセットと検証セットの誤差が非常によく相関する傾向があることを示しています。

11.結論

我々は、畳み込みネットワークをスケールアップするためのいくつかの設計原則を提示し、Inceptionアーキテクチャの文脈でそれらを研究しました。これにより、単純なモノリシック・アーキテクチャに比べて計算コストが比較的低い、高性能なビジョン・ネットワークを実現することができます。Inception-v2の最高品質バージョンは、ILSVR
2012の分類における**シングルクロップ**評価で、21.2%のトップ1エラーと5.6%の
トップ5エラーを達成し、新たな技術水準を確立しました。これは、Ioffe et al
[7]に記載されているネットワークと比較して、計算コストの増加が比較的緩やか（2.5 倍）
であるにもかかわらず達成されています。我々のモデルは、Heら[6]の結果を上回り、トップ5（トップ1）の誤差をそれぞれ25%（14%）削減し
ました。このように，パラメータ数の削減と，バッチ正規化された補助分類器とラベル平滑化による正則化を組み合わせることで，比較的小さいサイズの学習セットで高品質なネットワークを学習することができます．

リファレンス

[1] M.Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen,
C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghe-mawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia,
R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane,´。
R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster,
J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker,
V. Vanhoucke, V. Vasudevan, F. Viegas,´ O. Vinyals, P. War-den, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng.Tensor-Flow: Large-scale machine learning on heterogeneous sys-tes, 2015.ソフトウェアはtensorflow.orgから入手可能です。7

[2] W.チェン、J.T.ウィルソン、S.タイリー、K.Q.ウェインバーガー、そして

Y. チェンハッシュトリックでニューラルネットワークを圧縮する。In *Proceedings of The 32nd International Conference on Machine Learning*, 2015.1

[3] C.Dong, C. C. Loy, K. He, and X. Tang.Learning a deep convolutional network for image super-resolution.In *Com-Computer Vision-ECCV 2014*, pages 184-199.Springer, 2014.1

[4] D.Erhan, C. Szegedy, A. Toshev, and D. Anguelov.深層ニューラルネットワークを用いたスケーラブルな物体検出. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Confer-ence on*, pages 2155-2162.IEEE, 2014.1, 7

[5] R.Girshick, J. Donahue, T. Darrell, and J. Malik.Rich fea-ture hierarchies for accurate object detection and semantic segmentation.In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR*), 2014.1, 7

[6] K.He, X. Zhang, S. Ren, and J. Sun.Delving deep into rectifiers:Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*, 2015.1, 8

[7] S.Ioffe and C. Szegedy.Batch Normalization:内部の共変量シフトを減らすことでディープネットワークのトレーニングを加速する。In *Proceedings of The 32nd International Conference on Ma-chine Learning*, pages 448-456, 2015.3, 5, 8

[8] A.A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei.Con-Volutional Neural Networkを用いた大規模なビデオ分類。In *Computer Vision and Pat-tern Recognition (CVPR), 2014 IEEE Conference on*, pages 1725-1732.IEEE, 2014.1

[9] A.Krizhevsky, I. Sutskever, and G. E. Hinton.深層畳み込みニューラルネットワークを用いたイメージャー分類In *Advances in neural information processing systems*, pages 1097-1105, 2012.1

[10] A.ラヴィン。畳み込みニューラルネットワークの高速アルゴリズム. *arXiv preprint arXiv:1509.09308*, 2015.1

[11] C.-Y.Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu.*arXiv preprint arXiv:1409.5185*, 2014.5

[12] J.J.Long, E.Shelhamer, and T.Darrell.Fully convolutional networks for semantic segmentation.In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recogni-tion*, pages 3431-3440, 2015.1

[13] Y. Movshovitz-Attias, Q. Yu, M. C. Stumpe, V. Shet,

S. Arnoud, and L. Yatziv.ストリートビューの店頭を細かく分類するためのオントロジー的監督。In *Proceed-ings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1693-1702, 2015.1

[14] R.R. Pascanu, T. Mikolov, and Y. Bengio.このように、本論文では様々な問題点を解決しています。7

[15] D.D. C. Psichogios and L. H. Ungar.Svd-net: ネットワーク構造を自動的に選択するアルゴリズム. *IEEE transac-tions on neural networks/a publication of the IEEE Neural Networks Council*, 5(3):513-515, 1993.1

[16] O.ラサコフスキー、J.Deng、H.Su、J.Krause、S.Satheesh。

S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge.2014.1, 8

[17] F.Schroff, D. Kalenichenko, and J. Philbin.Facenet: A uni-fied embedding for face recognition and clustering. *arXiv preprint arXiv:1503.03832*, 2015.1

[18] K.Simonyan and A. Zisserman.Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.1, 8

[19] I.Sutskever, J. Martens, G. Dahl, and G. Hinton.On the importance of initialization and momentum in deep learning.*30th International Conference on Ma-chine Learning (ICML-13*), volume 28, pages 1139-1147.JMLR Workshop and Conference Proceedings, May 2013.7

[20] C. Szegedy, W.Liu, Y.Jia, P.Sermanet, S.Reed,

D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich.Going deeper with convolutions.In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1-9, 2015.1, 2, 4, 5, 8

[21] T.Tieleman and G. Hinton.勾配を最近の大きさの連続した平均値で割る。COURSERA: Neural Networks for Machine Learning, 4, 2012.Accessed: 2015-11-05.7

[22] A.Toshev and C. Szegedy.Deeppose: Human pose estima-tion via deep neural networks.In *Computer Vision and Pat-tern Recognition (CVPR), 2014 IEEE Conference on*, pages 1653-1660.IEEE, 2014.1

[23] N.WangとD.-Y.Yeung.深いコンパクトな画像表現を学習することで，視覚的な追跡が可能になる．In *Advances in Neural Information Processing* Systems, pages 809-817, 2013.1