

Multi-Scale Context Aggregation By Dilated Convolutions

- Fisher Yu (Princeton University)
- Vladlen Koltun (Intel Labs)

Abstract

セマンティックセグメンテーションの最先端のモデルは、当初は画像分類のために設計された畳み込みネットワークの適応に基づいています。しかし、セマンティックセグメンテーションのような密集した予測問題は、画像分類とは構造的に異なる。この研究では、密集予測用に特別に設計された新しい畳み込みネットワークモジュールを開発します。提示されたモジュールは、拡張された畳み込みを使用して、解像度を失うことなくマルチスケールの文脈情報を系統的に集約します。このアーキテクチャは、拡張された畳み込みが、解像度またはカバレッジの損失なしに受容野の指數関数的拡張をサポートするという事実に基づいている。提示されたコンテキストモジュールが最先端のセマンティックセグメンテーションシステムの精度を向上させることを示す。さらに、画像分類ネットワークの高密度予測への適応を検討し、適応ネットワークを簡素化すると精度が向上する可能性があることを示します。

1 Introduction

コンピュータビジョンの多くの自然問題は、密集した予測の例です。目標は、画像の各ピクセルごとに離散または連続ラベルを作成することです。顕著な例は、セマンティックセグメンテーションであり、各ピクセルを所定のカテゴリのセットの1つに分類する必要がある (Heら、2004; Shottonら、2009; Kohliら、2009; Krähenbühl&Koltun、2011)。セマンティックセグメンテーションは、ピクセルレベルの精度とマルチスケールコンテキストの推論を組み合わせる必要があるため、困難です (Heら、2004; Galleguillos&Belongie、2010)。

セマンティックセグメンテーションにおける重要な精度向上は、バックプロパゲーション (Rumelhart et al.、1986) によって訓練された畳み込みネットワーク (LeCunら、1989) の使用によって最近得られている。具体的には、Longら (2015年) は、もともと画像分類のために開発された畳み込みネットワークアーキテクチャが密集予測のためにうまく再利用できることを示しました。これらの再編成されたネットワークは、セマンティックセグメンテーションベンチマークに挑戦するまでの先行技術水準を大幅に上回ります。これは、画像分類と高密度予測との間の構造的差異によって動機付けられる新しい質問を促す。再利用されるネットワークのどの側面が本当に必要であり、密集して運用されると精度が低下するのですか？密な予測のために特別に設計された専用モジュールで精度をさらに向上できますか？

現代の画像分類ネットワークは、グローバルな予測が得られるまで分解能を低下させる連続的なプールおよびサブサンプリング層を介して、マルチスケールの文脈情報を統合する (Krizhevskyら、2012; Simonyan&Zisserman、2015)。対照的に、高密度予測では、フル解像度の出力と組み合わせたマルチスケールのコンテキスト推論が必要です。最近の研究では、複数スケール推論とフル解像度高密度予測の相反する要求に対処するための2つのアプローチを研究してきた。1つの手法は、ダウンサンプリングされたレイヤーからのグローバルな視点を持ちながら、失われた解像度を回復

することを目的とする繰り返しアップコンボリューションを含む (Noh et al.、2015; Fischer et al.、2015)。これにより、重大な中間ダウンサンプリングが本当に必要かどうかという疑問が残されます。別のアプローチは、ネットワークへの入力として画像の複数のリスケーリングされたバージョンを提供し、これらの複数の入力について得られた予測を組み合わせることを含む (Farabetら、2013; Linら、2015; Chenら、2015b)。再び、再スケーリングされた入力画像の別個の解析が本当に必要かどうかは明らかではない。

この作業では、解像度を失うことなく、再スケーリングされた画像を解析することなく、マルチスケールのコンテキスト情報を集約する畳み込みネットワークモジュールを開発しています。このモジュールは、任意の解像度で既存のアーキテクチャにプラグインすることができます。提示されたコンテキストモジュールは、画像分類から継承されたピラミッド形のアーキテクチャとは異なり、密集した予測のために特別に設計されています。これは、畳み込みレイヤーの直角プリズムで、プールもサブサンプリングもありません。このモジュールは、解像度またはカバレッジを損なうことなく受容野の指數関数的拡張をサポートする拡張された畳み込みに基づいています。

この作業の一環として、セマンティックセグメンテーションのための再利用された画像分類ネットワークの性能を再検討する。コア予測モジュールのパフォーマンスは、構造予測、マルチカラム・アーキテクチャー、複数のトレーニング・データセット、およびその他の機能強化を伴うますます精巧なシステムによって、意図的に不明瞭になる可能性があります。したがって、深い画像分類ネットワークの主要な適応を制御された設定で調べ、密集した予測性能を妨げる残留成分を除去する。その結果、以前の適応よりも単純で正確な初期予測モジュールが得られる。

単純化された予測モジュールを使用して、Pascal VOC 2012データセット (Everingham et al.、2010) の制御された実験によって提示されたコンテキストネットワークを評価する。実験では、コンテキストモジュールを既存のセマンティックセグメンテーションアーキテクチャーに差し込むと、その正確性が確かに高くなることが実証されています。

2 Dilated Convolutions

Let $F : Z^2 \rightarrow R$ の離散関数。 $\text{Let } r = [-r, r] \cap Z^2 \text{ and let } k : \Omega_r \rightarrow R \text{ be discrete サイズ } (2r + 1) \text{ のフィルタ} 2$ 。離散畳み込み演算子*は、次のように定義できます。

$$(F * k)(\mathbf{p}) = \sum_{\mathbf{s} + \mathbf{t} = \mathbf{p}} F(\mathbf{s})k(\mathbf{t}) \quad (1)$$

この演算子を一般化する。 $|l|$ を膨張因子とし、 $*|l|$ を

$$(F *_{|l|} k)(\mathbf{p}) = \sum_{\mathbf{s} + l\mathbf{t} = \mathbf{p}} F(\mathbf{s})k(\mathbf{t}) \quad (2)$$

$*|l|$ は、拡張された畳み込みまたは $|l|$ 拡張された畳み込みと呼ばれます。おなじみのディスクリートコンボルーション*は、単純に1拡張されたコンボリューションです。

拡張された畳み込み演算子は、これまで「拡張フィルタを用いた畳み込み」と呼ばれていました。これは、ウェーブレット分解アルゴリズム (Holschneider et al.、1987; Shensa、1992) において重要な役割を果たす。1「拡張フィルタ」が構築されていないことを明確にするために、「拡張フィルタを用いた畳み込み」ではなく「拡張畳み込み」または表される。コンボリューション演算子自体は、フィルタパラメータを異なる方法で使用するように変更されています。拡張畳み込み演算子は、異なる拡張係数を使用して異なる範囲で同じフィルタを適用できます。我々の定義は、拡張されたフィルタの構築を伴わない、拡張された畳み込み演算子の適切な実装を反映する。

セマンティックセグメンテーションのための畳み込みネットワークに関する最近の研究では、Longら（2015年）はフィルタ拡張を分析したが、それを使用しないことを選択した。Chenら（2015a）はLong et al. のアーキテクチャを単純化するために拡張を使用した。（2015）。対照的に、私たちは、マルチスケールのコンテキストアグリゲーションのために拡張された畳み込みを体系的に使用する新しい畳み込みネットワークアーキテクチャを開発します。

私たちのアーキテクチャは、拡張された畳み込みが、解像度やカバレッジを失うことなく指数関数的に拡大する受容野をサポートするという事実によって動機付けられています。 F_0, F_1, \dots, F_{n-1} : $Z^2 \rightarrow R$ は離散関数であり、 k_0, k_1, \dots, k_{n-2} : $\Omega^2 \rightarrow R$ は離散的な 3×3 のフィルタである。指数関数的に増加するフィルタリングを適用することを検討してください。

$$F_{i+1} = F_i *_{2^i} k_i \quad \text{for } i = 0, 1, \dots, n-2 \quad (3)$$

$F_i + 1$ の要素 p の受容野を、 F_0 の値を変更する要素の集合として定義する $F_i + 1(p)$ の $F_i + 1$ における p の受容野のサイズをこれらの要素の数とする。 $F_i + 1$ の各要素の受容野の大きさは $(2i + 2 - 1) \times (2i + 2 - 1)$ であることは容易に分かります。受容野は、指数関数的に増加する二乗の大きさである。これを図1に示します。

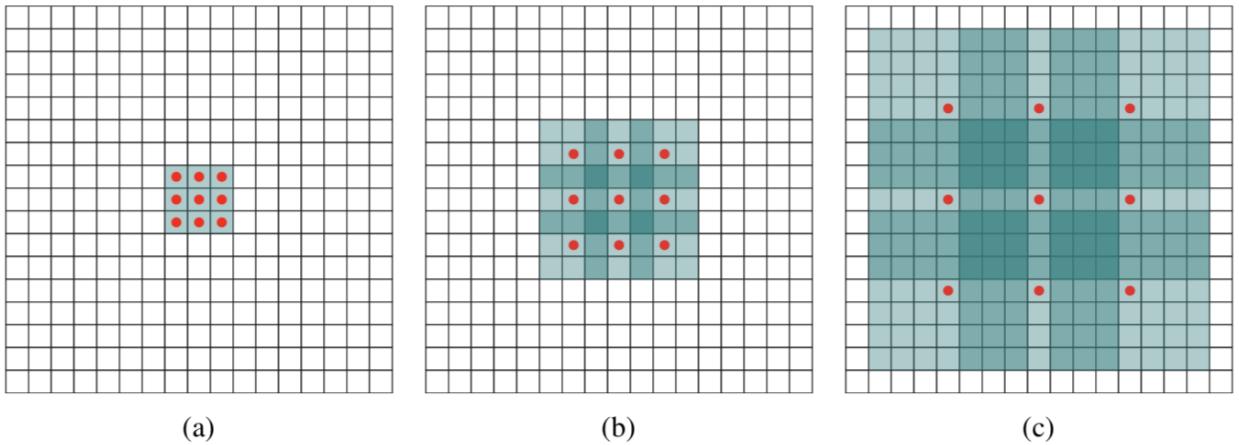


図1：体系的拡張は、分解能またはカバレッジを失うことなく受容野の指数関数的拡張をサポートする。（a） F_1 は、 F_0 から1拡張された畳み込みによって生成される。 F_1 の各要素は 3×3 の受容野を有する。（b） F_2 は、 F_1 から2つの拡張された畳み込みによって生成される。 F_2 の各要素は 7×7 の受容野を持つ。（c） F_3 は、4倍畳み込みによって F_2 から生成される。 F_3 の各要素は 15×15 の受容野を持っています。各層に関連するパラメータの数は同一である。受容野は指数関数的に増加するが、パラメータの数は直線的に増加する。

3 Multi-Scale Context Aggregation

コンテキストモジュールは、マルチスケールのコンテキスト情報を集約することで、密集した予測アーキテクチャのパフォーマンスを向上させるように設計されています。このモジュールは、Cフィーチャマップを入力として受け取り、出力としてCフィーチャマップを生成する。入力と出力は同じ形式なので、モジュールは既存の高密度予測アーキテクチャにプラグインすることができます。

コンテキストモジュールの基本的な形式を記述することから始めます。この基本的な形態では、各層はCチャネルを有する。各層の表現は同じであり、特徴マップが正規化されず、モジュール内に損失が定義されていないが、クラス毎の高密度予測を直接得るために使用することができます。直観的には、モジュールは、コンテキスト情報を公開する複数のレイヤーにそれらを渡すことによって、フィーチャマップの精度を向上させることができます。

基本的なコンテキストモジュールは、異なる拡張係数で 3×3 畳み込みを適用する7つのレイヤーを有する。拡張は1,1,2,4,8,16,1です。各畳み込みはすべてのレイヤーで動作します。厳密に言えば、これらは最初の2つの次元で拡張した $3\times 3\times C$ 畳み込みです。これらの畳み込みのそれぞれには、ポイントワイズの切捨て $\max(\cdot, 0)$ が続きます。最終層は $1\times 1\times C$ の畳み込みを行い、モジュールの出力を生成する。アーキテクチャの概要を表1に示します。実験でコンテキストネットワークへの入力を提供するフロントエンドモジュールは、 64×64 解像度のフィーチャマップを生成します。したがって、我々は、層6の後の受容野の指數関数的な拡張を停止する。

コンテキストモジュールを訓練する当初の試みは、予測精度の改善に失敗した。実験では、標準の初期化手順ではモジュールのトレーニングを容易にサポートできないことが明らかになりました。畳み込みネットワークは、一般に、ランダム分布からのサンプルを用いて初期化される (Glorot & Bengio, 2010; Krizhevskyら, 2012; Simonyan & Zisserman, 2015)。しかし、ランダム初期化方式はコンテキストモジュールには有効ではないことがわかった。はっきりとしたセマンティクスの代替初期化がはるかに効果的であることを発見しました。

$$k^b(t, a) = \mathbf{1}_{[t=0]} \mathbf{1}_{[a=b]} \quad (4)$$

a は入力フィーチャマップのインデックス、 b は出力マップのインデックスです。これは、アイデンティティの初期化の一形態であり、最近、反復的なネットワークのために提唱されている (Le et al.、2015)。この初期化では、各レイヤーが入力を次のレイヤーに直接渡すようにすべてのフィルターを設定します。自然な関心事は、この初期化によって、バックプロパゲーションが単純に情報を渡すというデフォルトの動作を大幅に改善できないモードにネットワークを置くことができるこです。しかし、実験によると、これは当てはまりません。バックプロパゲーションは、処理されたマップの精度を高めるために、ネットワークによって提供される文脈情報を確実に収穫する。

Layer	1	2	3	4	5	6	7	8
Convolution	3×3	3×3	3×3	3×3	3×3	3×3	3×3	1×1
Dilation	1	1	2	4	8	16	1	1
Truncation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Receptive field	3×3	5×5	9×9	17×17	33×33	65×65	67×67	67×67
Output channels								
Basic	C	C	C	C	C	C	C	C
Large	$2C$	$2C$	$4C$	$8C$	$16C$	$32C$	$32C$	C

表1：コンテキストネットワークアーキテクチャ。ネットワークは、解像度を失うことなく徐々に増加するスケールでコンテキスト情報を集約することによって、 C の特徴マップを処理する。

以上で、基本コンテキストネットワークの表示は完了です。我々の実験は、この基本的なモジュールでさえ、定量的および定性的に密集した予測精度を高めることができることを示している。これは、ネットワーク内のパラメータの数が少ない場合に特に顕著です。合計で $\approx 64C^2$ 個のパラメータ。

我々はまた、より深い層で多くの数の特徴マップを使用する、より大きなコンテキストネットワークを訓練した。大規模ネットワークにおけるマップの数は表1に要約されている。我々は、異なるレイヤーにおける特徴マップの数の違いを説明するために初期化スキームを一般化する。 c_i および c_{i+1} を、2つの連続する層内の特徴マップの数とする。 C は c_i と c_{i+1} の両方を分割すると仮定する。初期化は

$$k^b(t, a) = \begin{cases} \frac{C}{c_{i+1}} & t = 0 \text{ and } \lfloor \frac{aC}{c_i} \rfloor = \lfloor \frac{bC}{c_{i+1}} \rfloor \\ \epsilon & \text{otherwise} \end{cases} \quad (5)$$

ここで $\varepsilon \sim N(0, \sigma^2)$ と $\sigma C / c_i + 1$ 。ランダムノイズの使用は、共通の前任者との特徴マップ間の結びつきを破る。

4 Front End

我々は、カラー画像を入力とし、 $C = 21$ の特徴マップを出力として生成するフロントエンド予測モジュールを実装し、訓練した。フロントエンドモジュールは、ロング (Long) らの著作に従う。

(2015) およびChenら (2015a) が、別途実施された。密集した予測のためにVGG-16ネットワーク (Simonyan & Zisserman, 2015) を適用し、最後の2つのプールとストライドレイヤを削除しました。具体的には、これらのプールおよびストライド層のそれぞれを除去し、その後のすべての層における畳み込みを、アブレーションされた各プール層について2倍に膨張させた。したがって、両方のアブレーションされたプーリング層に続く最終層の畳み込みは、4倍に拡張される。これにより、元の分類ネットワークのパラメータで初期化することができるが、より高解像度の出力が得られる。フロントエンドモジュールは、埋め込まれた画像を入力として取り込み、解像度 64×64 でフィーチャマップを生成する。反射パディングを使用します。バッファゾーンは、各エッジについてイメージを反射することで塗りつぶされます。

我々のフロントエンドモジュールは、密集予測に対抗する分類ネットワークの痕跡を除去することによって得られる。最も重要なのは、最後の2つのプールとストライドのレイヤーを完全に削除するのに対し、Longらは、それらを保管し、Chenら。膨張によるストライドに取って代わったが、プール層を維持した。我々は、プール層を除去することによってネットワークを単純化することにより、それをより正確にすることが分かった。また、中間のフィーチャマップのパディングを削除します。中間パディングは元の分類ネットワークで使用されていましたが、密集予測では必要でも正当でもありません。

この単純化された予測モジュールは、Hariharanらによって作成された注釈によって補強された Pascal VOC 2012トレーニングセットで訓練されました。 (2011年)。トレーニングのために VOC-2012検証セットの画像を使用しなかったため、Hariharanらの注釈のサブセットのみを使用しました。 (2011年)。トレーニングは、ミニバッチサイズ14、学習率10-3、および運動量0.9の確率勾配降下 (SGD) によって行った。ネットワークは60K回の訓練を受けました。

フロントエンドモジュールの精度をLongらのFCN-8s設計と比較します。 (2015年) とChenらの DeepLabネットワーク (2015a)。 FCN-8sとDeepLabについては、元の著者がVOC-2012で訓練した公開モデルを評価します。VOC-2012データセットの画像上の異なるモデルによって生成されたセグメンテーションを図2に示します。VOC-2012テストセットのモデルの精度は、表2に報告されています。

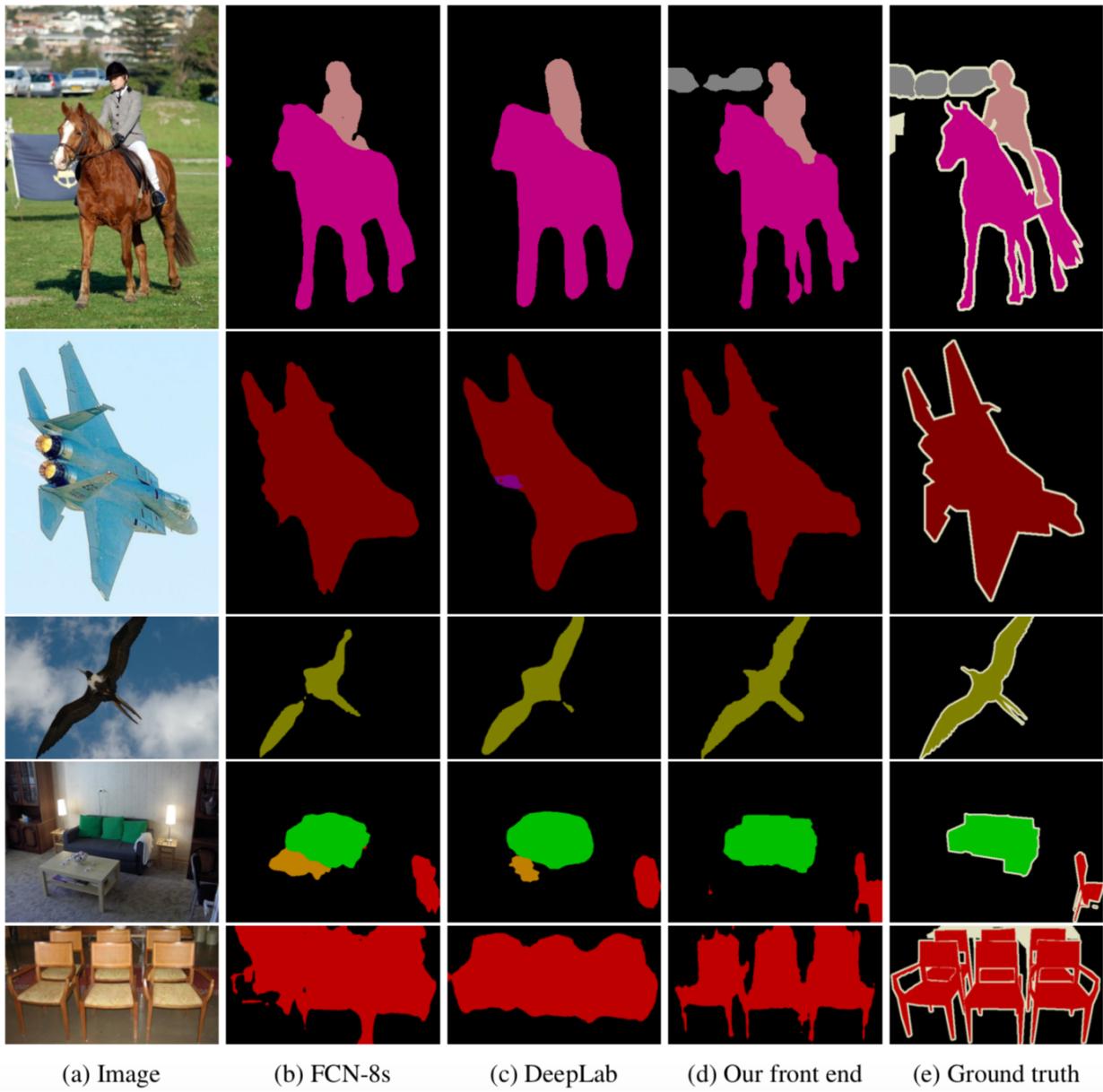


図2：VGG-16分類ネットワークの異なる適応によって生成されたセマンティックセグメンテーション。 (a) 入力画像、 (b) FCN-8sによる予測 (Long et al.、 2015) 、 (c) DeepLabによる予測 (Chen et al.、 2015a) 、 (d) 単純化による予測フロントエンドモジュール、 (e) グランドトゥルース

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
FCN-8s	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
DeepLab	72	31	71.2	53.7	60.5	77	71.9	73.1	25.2	62.6	49.1	68.7	63.3	73.9	73.6	50.8	72.3	42.1	67.9	52.6	62.1
DeepLab-Msc	74.9	34.1	72.6	52.9	61.0	77.9	73.0	73.7	26.4	62.2	49.3	68.4	64.1	74.0	75.0	51.7	72.7	42.5	67.2	55.7	62.9
Our front end	82.2	37.4	72.7	57.1	62.7	82.8	77.8	78.9	28	70	51.6	73.1	72.8	81.5	79.1	56.6	77.1	49.9	75.3	60.9	67.6

表2：当社のフロントエンド予測モジュールは、従来のモデルよりも簡単で正確です。この表は、VOC-2012テストセットの精度を報告します。

当社のフロントエンド予測モジュールは、従来のモデルよりも簡単で正確です。特に、単純化されたモデルはFCN-8とDeepLabネットワークの両方よりもテストセットで5%以上優れています。面白いことに、私たちの単純化されたフロントエンドモジュールは、CRFを使用せずに、テストセット上のDeepLab + CRFのリーダーボード精度を1%ポイント以上（67.6%対66.4%）上回っています。

5 Experiments

私たちの実装は、Caffe図書館 (Jia et al.、2014) に基づいています。拡張された畳み込みの実装は現在、スタンファードCaffeディストリビューションの一部です。

最近の高性能システムとの公正な比較のために、セクション4で説明したのと同じ構造を持つフロントエンドモジュールを訓練しましたが、Microsoft COCOデータセット (Lin et al.、2014) の追加イメージを訓練しました。Microsoft COCOのすべての画像には、VOC-2012カテゴリの少なくとも1つのオブジェクトを使用しました。他のカテゴリの注釈付きオブジェクトは背景として扱われました。

トレーニングは2段階で行われた。第1段階では、VOC-2012画像とMicrosoft COCO画像と一緒に訓練しました。トレーニングは、ミニバッチサイズ14および運動量0.9のSGDによって実施された。100K回の反復を10-3の学習率で実施し、40Kの反復を10-4の学習率で行った。第2段階では、VOC-2012画像のみでネットワークを微調整しました。ファインチューニングは、10-5の学習率で50Kの反復に対して実行されました。VOC-2012検証セットの画像はトレーニングに使用されませんでした。

この手順で訓練されたフロントエンドモジュールは、VOC-2012評価セットで平均IoUを69.8%、テストセットで平均IoUを71.3%達成しました。このレベルの精度は、コンテキストモジュールまたは構造化された予測なしで、フロントエンドだけで達成されることに注意してください。この高い精度は、元々、密集予測よりもむしろ画像分類のためにもともと開発された残留成分の除去に帰結する。

コンテキスト集約の制御された評価。ここでは、3章で提示したコンテキストネットワークの有用性を評価するための制御された実験を行います。まず、2つのコンテキストモジュール（基本と大）をフロントエンドに差し込みます。コンテキストネットワークの受容野は 67×67 であるため、入力フィーチャマップを幅33のバッファで埋める。ゼロパディングと反射パディングは、我々の実験で同様の結果を生じた。コンテキストモジュールはフロントエンドからのフィーチャマップを入力として受け取り、トレーニング中にこの入力を与えられます。コンテキストモジュールとフロントエンドモジュールの共同訓練は、我々の実験において有意な改善をもたらさなかった。学習率は10-3に設定されました。トレーニングはセクション3で説明されているように初期化されました。

表3は、セマンティックセグメンテーションのための3つの異なるアーキテクチャへのコンテキストモジュールの追加の効果を示す。最初のアーキテクチャ（上）は、セクション4で説明したフロントエンドです。Longらのオリジナルの作業と同様に、構造化された予測なしにセミナルセグメンテーションを実行します。（2015）。第2のアーキテクチャ（表3、中央）は、密集したCRFを使用して、Chenらのシステムに類似した構造化予測を実行する。（2015a）。我々は、Krähenbühl & Koltun（2011）の実装を使用し、バリデーションセットのグリッド検索によってCRFパラメータを訓練します。第3のアーキテクチャ（表3、下部）は、構造的予測のためにCRF-RNNを使用する（Zheng et al.、2015）。Zheng et al. の実装を使用します。（2015）、各条件でCRF-RNNを訓練する。

実験結果は、コンテキストモジュールが3つの構成のそれぞれの精度を改善することを示している。基本コンテキストモジュールは、各構成の精度を向上させます。大規模なコンテキストモジュールは、精度を大幅に向上させます。実験では、コンテキストモジュールと構造化予測が相乗的であることが示されています。コンテキストモジュールは、その後の構造化予測の有無にかかわらず精度を向上させます。定性的結果を図3に示す。

テストセットの評価。 テストセットの評価は、Pascal VOC 2012評価サーバに結果を提出して行います。結果を表4に報告する。我々は、これらの実験に大きなコンテキストモジュールを使用する。結果が示すように、コンテキストモジュールは、フロントエンドよりも精度が大幅に向上了します。コンテキストモジュール単独では、その後の構造化予測なしに、DeepLab-CRF-COCO-LargeFOVよりも優れています（Chen et al.、2015a）。Krähenbühl&Koltun（2011）の元の実装を使用して密集したCRFを持つ文章モジュールは、最近のCRF-RNN（Zheng et al.、2015）と同等の性能を発揮します。コンテキストモジュールは、CRF-RNNと組み合わせて、CRF-RNNの性能よりも精度をさらに高める。

6 Conclusion

密な予測のための畳み込みネットワークアーキテクチャを検討した。モデルは高解像度出力を生成する必要があるため、ネットワーク全体の高解像度動作が実現可能であり、望ましいと考えています。我々の研究は、拡張された畳み込み演算子が、分解能またはカバレッジを失うことなく受容野を拡張する能力のために、密集予測に特に適していることを示している。我々は、拡張された畳み込みを利用して、既存のセマンティックセグメンテーションシステムにプラグインするときに正確に精度を向上させる新しいネットワーク構造を設計した。この作業の一環として、我々は、画像分類のために開発された残留成分を除去することによって、セマンティックセグメンテーションのための既存の畳み込みネットワークの精度を高めることもできることを示した。



図3：異なるモデルによって生成されたセマンティックセグメンテーション (a) 入力画像、(b) フロントエンドモジュールによる予測、(c) フロントエンドにプラグインされたラージコンテキストネットワークによる予測、(d) フロントエンド+コンテキストモジュール+CRFによる予測-RNN、(e) 地上真理。

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
Front end	86.3	38.2	76.8	66.8	63.2	87.3	78.7	82	33.7	76.7	53.5	73.7	76	76.6	83	51.9	77.8	44	79.9	66.3	69.8
Front + Basic	86.4	37.6	78.5	66.3	64.1	89.9	79.9	84.9	36.1	79.4	55.8	77.6	81.6	79	83.1	51.2	81.3	43.7	82.3	65.7	71.3
Front + Large	87.3	39.2	80.3	65.6	66.4	90.2	82.6	85.8	34.8	81.9	51.7	79	84.1	80.9	83.2	51.2	83.2	44.7	83.4	65.6	72.1
Front end + CRF	89.2	38.8	80	69.8	63.2	88.8	80	85.2	33.8	80.6	55.5	77.1	80.8	77.3	84.3	53.1	80.4	45	80.7	67.9	71.6
Front + Basic + CRF	89.1	38.7	81.4	67.4	65	91	81	86.7	37.5	81	57	79.6	83.6	79.9	84.6	52.7	83.3	44.3	82.6	67.2	72.7
Front + Large + CRF	89.6	39.9	82.7	66.7	67.5	91.1	83.3	87.4	36	83.3	52.5	80.7	85.7	81.8	84.4	52.6	84.4	45.3	83.7	66.7	73.3
Front end + RNN	88.8	38.1	80.8	69.1	65.6	89.9	79.6	85.7	36.3	83.6	57.3	77.9	83.2	77	84.6	54.7	82.1	46.9	80.9	66.7	72.5
Front + Basic + RNN	89	38.4	82.3	67.9	65.2	91.5	80.4	87.2	38.4	82.1	57.7	79.9	85	79.6	84.5	53.5	84	45	82.8	66.2	73.1
Front + Large + RNN	89.3	39.2	83.6	67.2	69	92.1	83.1	88	38.4	84.8	55.3	81.2	86.7	81.3	84.3	53.6	84.4	45.8	83.8	67	73.9

表3：セマンティックセグメンテーションのための3つの異なるアーキテクチャの精度に対するコンテキストモジュールの効果の制御された評価。VOC-2012検証セットで行われた実験。検証画像はトレーニングに使用されませんでした。Top：構造化された予測のないセマンティックセグメンテーションフロントエンドにコンテキストモジュールを追加する (Long et al.、2015)。基本コンテキストモジュールは精度を向上させ、大きなモジュールは大きなコンテキストマージンでそれを増加させます。Middle：コンテキストモジュールは、フロントエンド+高密度CRF構成 (Chen et al.、2015a) に接続すると精度が向上します。ボトム：フロントエンド+CRF-RNN構成 (Zheng et al.、2015) にプラグインすると、コンテキストモジュールが精度を向上させます。

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
DeepLab++	89.1	38.3	88.1	63.3	69.7	87.1	83.1	85	29.3	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1	72.7
DeepLab-MSc++	89.2	46.7	88.5	63.5	68.4	87.0	81.2	86.3	32.6	80.7	62.4	81.0	81.3	84.3	82.1	56.2	84.6	58.3	76.2	67.2	73.9
CRF-RNN	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
Front end	86.6	37.3	84.9	62.4	67.3	86.2	81.2	82.1	32.6	77.4	58.3	75.9	81	83.6	82.3	54.2	81.5	50.1	77.5	63	71.3
Context	89.1	39.1	86.8	62.6	68.9	88.2	82.6	87.7	33.8	81.2	59.2	81.8	87.2	83.3	83.6	53.6	84.9	53.7	80.5	62.9	73.5
Context + CRF	91.3	39.9	88.9	64.3	69.8	88.9	82.6	89.7	34.7	82.7	59.5	83	88.4	84.2	85	55.3	86.7	54.4	81.9	63.6	74.7
Context + CRF-RNN	91.7	39.6	87.8	63.1	71.8	89.7	82.9	89.8	37.2	84	63	83.3	89	83.8	85.1	56.8	87.6	56	80.2	64.7	75.3

表4：VOC-2012テストセットの評価「DeepLab ++」はDeepLab-CRF-COCO-LargeFOVを表し、「DeepLab-MSc ++」はDeepLab-MSc-CRF-LargeFOV-COCO-クロスジョイントを表します (Chen et al.、2015a)。「CRF-RNN」は、Zhengらのシステムである。(2015)。「コンテキスト」とは、フロントエンドに接続された大きなコンテキストモジュールを指します。コンテキストネットワークは非常に高精度であり、我々は構造化予測を実行せずにDeepLab ++アーキテクチャを実行します。コンテキストネットワークをCRF-RNN構造化予測モジュールと組み合わせることにより、CRF-RNNシステムの精度が向上する。

提示された研究は、画像分類前駆体によって制約されない高密度予測のための専用アーキテクチャへの一歩であると考えられる。新しいデータソースが利用可能になると、将来のアーキテクチャは密接にエンドツーエンドで訓練され、画像分類データセットの事前訓練の必要性がなくなる可能性があります。これにより、アーキテクチャの簡素化と統合が可能になります。具体的には、エンド・ツー・エンドの高密度トレーニングは、生のイメージを入力として受け入れ、完全な解像度で高密度のラベル割り当てを出力として生成する、完全な解像度で動作する、提示されたコンテキストネットワークに似た完全密なアーキテクチャを可能にする。

セマンティックセグメンテーションのための最先端のシステムは、将来の進歩のための大きな余地を残す。最も正確な設定が失敗したケースを図4に示します。このエリアの進捗状況をサポートするために、コードと訓練されたモデルをリリースします。

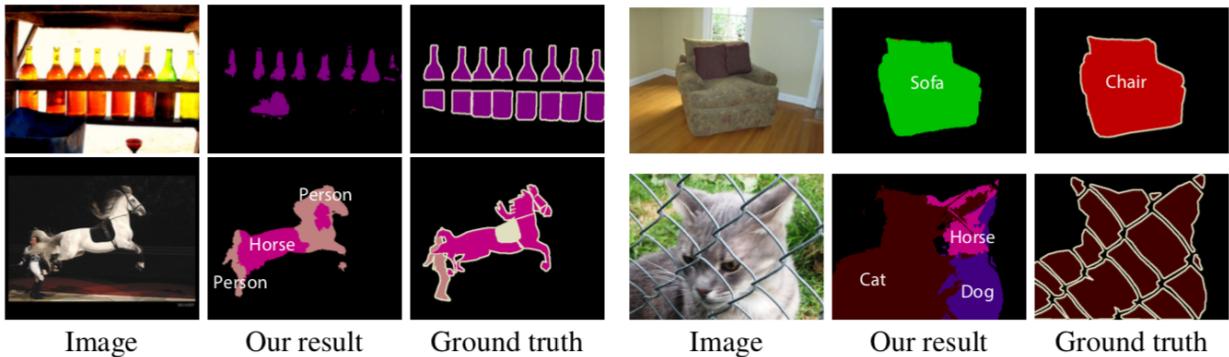


図4：VOC-2012検証セットの障害ケース私たちが訓練した最も正確なアーキテクチャ（Context + CRF-RNN）は、これらの画像ではほとんど機能しません。

Acknowledgements

私たちはVibhav Vineetに校正、実験、および関連する議論に感謝します。 Jonathan LongとCaffeチームに感謝の意を表し、 Caffeライブラリへの実装を急速に取り入れることにも感謝しています。

References

1. Badrinarayanan, Vijay, Handa, Ankur, and Cipolla, Roberto. SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. arXiv:1505.07293, 2015.
2. Brostow, Gabriel J., Fauqueur, Julien, and Cipolla, Roberto. Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters, 30(2), 2009.
3. Chen, Liang-Chieh, Papandreou, George, Kokkinos, Iasonas, Murphy, Kevin, and Yuille, Alan L. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In ICLR, 2015a.
4. Chen, Liang-Chieh, Yang, Yi, Wang, Jiang, Xu, Wei, and Yuille, Alan L. Attention to scale: Scale-aware semantic image segmentation. arXiv:1511.03339, 2015b.
5. Cordts, Marius, Omran, Mohamed, Ramos, Sebastian, Rehfeld, Timo, Enzweiler, Markus, Benenson, Rodrigo, Franke, Uwe, Roth, Stefan, and Schiele, Bernt. The Cityscapes dataset for semantic urban scene understanding. In CVPR, 2016.
6. Everingham, Mark, Gool, Luc J. Van, Williams, Christopher K. I., Winn, John M., and Zisserman, Andrew. The Pascal visual object classes (VOC) challenge. IJCV, 88(2), 2010.
7. Farabet, Clément, Couprie, Camille, Najman, Laurent, and LeCun, Yann. Learning hierarchical features for scene labeling. PAMI, 35(8), 2013.
8. Fischer, Philipp, Dosovitskiy, Alexey, Ilg, Eddy, Häusser, Philip, Hazrba, Caner, Golkov, Vladimir, van der Smagt, Patrick, Cremers, Daniel, and Brox, Thomas. Learning optical flow with convolutional neural networks. In ICCV, 2015.
9. Galleguillos, Carolina and Belongie, Serge J. Context based object categorization: A critical survey. Computer Vision and Image Understanding, 114(6), 2010.
10. Geiger, Andreas, Lenz, Philip, Stiller, Christoph, and Urtasun, Raquel. Vision meets robotics: The KITTI dataset. International Journal of Robotics Research, 32(11), 2013.
11. Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In AISTATS, 2010.
12. Hariharan, Bharath, Arbelaez, Pablo, Bourdev, Lubomir D., Maji, Subhransu, and Malik, Jitendra. Semantic contours from inverse detectors. In ICCV, 2011.

13. He, Xuming, Zemel, Richard S., and Carreira-Perpin  n, Miguel A. Multiscale conditional random fields for image labeling. In CVPR, 2004.
14. Holschneider, M., Kronland-Martinet, R., Morlet, J., and Tchamitchian, Ph. A real-time algorithm for signal analysis with the help of the wavelet transform. In Wavelets: Time-Frequency Methods and Phase Space. Proceedings of the International Conference, 1987.
15. Jia, Yangqing, Shelhamer, Evan, Donahue, Jeff, Karayev, Sergey, Long, Jonathan, Girshick, Ross B., Guadarrama, Sergio, and Darrell, Trevor. Caffe: Convolutional architecture for fast feature embedding. In Proc. ACM Multimedia, 2014.
16. Kohli, Pushmeet, Ladicky, Lubor, and Torr, Philip H. S. Robust higher order potentials for enforcing label consistency. IJCV, 82(3), 2009.
17. Kra  henb  hl, Philipp and Koltun, Vladlen. Efficient inference in fully connected CRFs with Gaussian edge potentials. In NIPS, 2011.
18. Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.
19. Kundu, Abhijit, Vineet, Vibhav, and Koltun, Vladlen. Feature space optimization for semantic video segmentation. In CVPR, 2016.
20. Ladicky, Lubor, Russell, Christopher, Kohli, Pushmeet, and Torr, Philip H. S. Associative hierarchical CRFs for object class image segmentation. In ICCV, 2009.
21. Le, Quoc V., Jaitly, Navdeep, and Hinton, Geoffrey E. A simple way to initialize recurrent networks of rectified linear units. arXiv:1504.00941, 2015.
22. LeCun, Yann, Boser, Bernhard, Denker, John S., Henderson, Donnie, Howard, Richard E., Hubbard, Wayne, and Jackel, Lawrence D. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4), 1989.
23. Lin, Guosheng, Shen, Chunhua, Reid, Ian, and van dan Hengel, Anton. Efficient piecewise training of deep structured models for semantic segmentation. arXiv:1504.01013, 2015.
24. Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Doll  r, Piotr, and Zitnick, C. Lawrence. Microsoft COCO: Common objects in context. In ECCV, 2014.
25. Liu, Buyu and He, Xuming. Multiclass semantic video segmentation with object-level active inference. In CVPR, 2015.
26. Long, Jonathan, Shelhamer, Evan, and Darrell, Trevor. Fully convolutional networks for semantic segmentation. In CVPR, 2015.
27. Noh, Hyeonwoo, Hong, Seunghoon, and Han, Bohyung. Learning deconvolution network for semantic segmentation. In ICCV, 2015.
28. Ros, Germa  n, Ramos, Sebastian, Granados, Manuel, Bakhtiary, Amir, Va  zquez, David, and Lo  pez, Antonio Manuel. Vision-based offline-online perception paradigm for autonomous driving. In WACV, 2015.
29. Rumelhart, David E., Hinton, Geoffrey E., and Williams, Ronald J. Learning representations by back-propagating errors. Nature, 323, 1986.
30. Shensa, Mark J. The discrete wavelet transform: wedding the a` trous and Mallat algorithms. IEEE Transactions on Signal Processing, 40(10), 1992.
31. Shotton, Jamie, Winn, John M., Rother, Carsten, and Criminisi, Antonio. TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. IJCV, 81(1), 2009.
32. Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.
33. Sturgess, Paul, Alahari, Karteek, Ladicky, Lubor, and Torr, Philip H. S. Combining

- appearance and structure from motion features for road scene understanding. In BMVC, 2009.
34. Tighe, Joseph and Lazebnik, Svetlana. Superparsing – scalable nonparametric image parsing with superpixels. IJCV, 101(2), 2013.
 35. Zheng, Shuai, Jayasumana, Sadeep, Romera-Paredes, Bernardino, Vineet, Vibhav, Su, Zhizhong, Du, Dalong, Huang, Chang, and Torr, Philip. Conditional random fields as recurrent neural networks. In ICCV, 2015.

Appendix A Urban Scene Understanding

本稿では、CamVidデータセット (Brostow et al.、2009) 、KITTIデータセット (Geiger et al.、2013) 、新しいCityscapesデータセット (Cordts et al. 2016) 。精度の測度として平均IoUを用いる (Everingham et al.、2010) 。検証セットが利用可能であっても、モデルを訓練セットでトレーニングするだけです。このセクションで報告される結果は、条件付きランダムフィールドまたは他の形式の構造化予測を使用しません。それらはフロントエンドモジュールとコンテキストモジュールを組み合わせた畳み込みネットワークで得られました。表3で評価された "Front + Basic" ネットワークと同様です。トレーニングされたモデルは <https://github.com/fyu/dilation> で見つけることができます。

ここで、フロントエンドモジュールのトレーニングに使用されるトレーニング手順を要約します。この手順はすべてのデータセットに適用されます。トレーニングは確率的な勾配降下で実行される。各ミニバッチにはランダムにサンプリングされた画像からの8作物が含まれています。各作物のサイズは628×628であり、パッディングされた画像からランダムにサンプリングされます。画像は反射パディングを使用してパディングされます。中間層にはパディングは使用されない。学習率は10-4、運動量は0.99に設定されています。反復の回数は、データセット内のイメージの数に依存し、下の各データセットについて報告されます。

A.1 CamVid

私たちは、Sturgess et al. (2009) は、データセットを367のトレーニング画像、100の検証画像、233のテスト画像に分割します。11の意味クラスが使用される。画像は640×480にダウンサンプリングされます。

コンテキストモジュールには、紙の本体のPascal VOCデータセットに使用されるモデルに類似した8つのレイヤがあります。全体のトレーニング手順は次のとおりです。まず、フロントエンドモジュールを20K回繰り返しトレーニングします。次に、完全なモデル（フロントエンド+コンテキスト）は、バッチサイズ1のサイズ852×852の作物をサンプリングすることによって共同訓練される。共同トレーニングの学習率は10-5に設定され、運動量は0.9に設定される。

CamVidテストセットの結果は表5に報告されています。コンテキストモジュールには8つのレイヤがあるため、完全な畳み込みネットワーク（フロントエンド+コンテキスト）をDilation8と呼びます。私たちのモデルは先の研究よりも優れています。このモデルは、Kundu et al. の最近の研究で単項式分類子として用いられた。 (2016) 。

	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Bicyclist	mean IoU
ALE	73.4	70.2	91.1	64.2	24.4	91.1	29.1	31.0	13.6	72.4	28.6	53.6
SuperParsing	70.4	54.8	83.5	43.3	25.4	83.4	11.6	18.3	5.2	57.4	8.9	42.0
Liu and He	66.8	66.6	90.1	62.9	21.4	85.8	28.0	17.8	8.3	63.5	8.5	47.2
SegNet	68.7	52.0	87.0	58.5	13.4	86.2	25.3	17.9	16.0	60.5	24.8	46.4
DeepLab-LFOV	81.5	74.6	89.0	82.2	42.3	92.2	48.4	27.2	14.3	75.4	50.1	61.6
Dilation8	82.6	76.2	89.9	84.0	46.9	92.2	56.3	35.8	23.4	75.3	55.5	65.3

表5：CamVidデータセットのセマンティックセグメンテーション結果我々のモデル（Dilation8）は ALE (Ladicky et al.、2009) 、 SuperParsing (Tighe&Lazebnik、2013) 、 Liu and He (Liu&He、2015) 、 SegNet (Badrinarayananら、2015) DeepLab-LargeFOVモデル (Chen et al.、2015a) が含まれる。私たちのモデルは先の研究よりも優れています。

A.2 KITTI

我々は、Rosらの訓練および検証分割を使用する。（2015年）：100のトレーニング画像と46のテスト画像。画像はすべてKITTI視覚オドメトリ/ SLAMデータセットから収集した。画像の解像度は1226×370です。垂直解像度は他のデータセットに比べて小さいので、表1のレイヤ6を削除します。結果として得られるコンテキストモジュールには7つのレイヤがあります。完全なネットワーク（フロントエンド+コンテキスト）はDilation7と呼ばれます。

フロントエンドは10Kの反復のために訓練されています。次に、フロントエンドとコンテキストモジュールが共同して訓練されます。共同訓練の場合、作物のサイズは900×900であり、運動量は0.99に設定され、他のパラメータはCamVidデータセットに使用されるパラメータと同じです。ジョイントトレーニングは20K回繰り返し実行されます。

結果を表6に示す。表が示すように、我々のモデルは先行研究よりも優れている。

	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Bicyclist	mean IoU
Ros et al.	71.8	69.5	84.4	51.2	4.2	72.4	1.7	32.4	2.6	45.3	3.2	39.9
DeepLab-LFOV	82.8	78.6	82.4	78.0	28.8	91.3	0.0	39.4	29.9	72.4	12.9	54.2
Dilation7	84.6	81.1	83	81.4	41.8	92.9	4.6	47.1	35.2	73.1	26.4	59.2

表6：KITTIデータセットのセマンティックセグメンテーション結果その結果をRosらと比較する。（2015）DeepLab-LargeFOVモデル (Chen et al.、2015a) に適用されます。私たちのネットワーク（Dilation7）は、以前の研究よりも高い精度をもたらします。

A.3 Cityscapes

Cityscapesのデータセットには、2975のトレーニング画像、500の検証画像、1525のテスト画像が含まれています（Cordts et al.、2016）。高い画像解像度（2048×1024）のために、表1のレイヤー6の後に2つのレイヤーをコンテキストネットワークに追加します。これらの2つのレイヤーは、それぞれ32と64の膨張を持ちます。コンテキストモジュールのレイヤーの総数は10であり、完全なモデル（フロントエンド+コンテキスト）はDilation10と呼ばれます。

Dilation10ネットワークは3段階で訓練されました。まず、フロントエンド予測モジュールを40K回繰り返しトレーニングしました。第2に、コンテクストモジュールは、学習率10-4、運動量0.99、およびバッチサイズ100の、全部の（切り抜かれていらない）画像について、24K反復訓練を受けた。第3に、60K反復の完全なモデル（フロントエンド+学習率10-5、運動量0.99、バッチサイズ1で、画像の半分（入力サイズ1396×1396、パディングを含む）に表示されます。

図5は、訓練段階がモデルの性能に与える影響を視覚化しています。定量結果を表7および表8に示す。

Dilation10のパフォーマンスは、Cordts他によるCityscapesデータセットの以前の作業と比較されました。（2016）。彼らの評価では、Dilation10はすべての以前のモデルよりも優れていた（Cordts et al.、2016）。Dilation10はKundu et al. の最近の研究では単項分類子としても使われました。（2016）、構造予測を使用して精度をさらに向上させました。

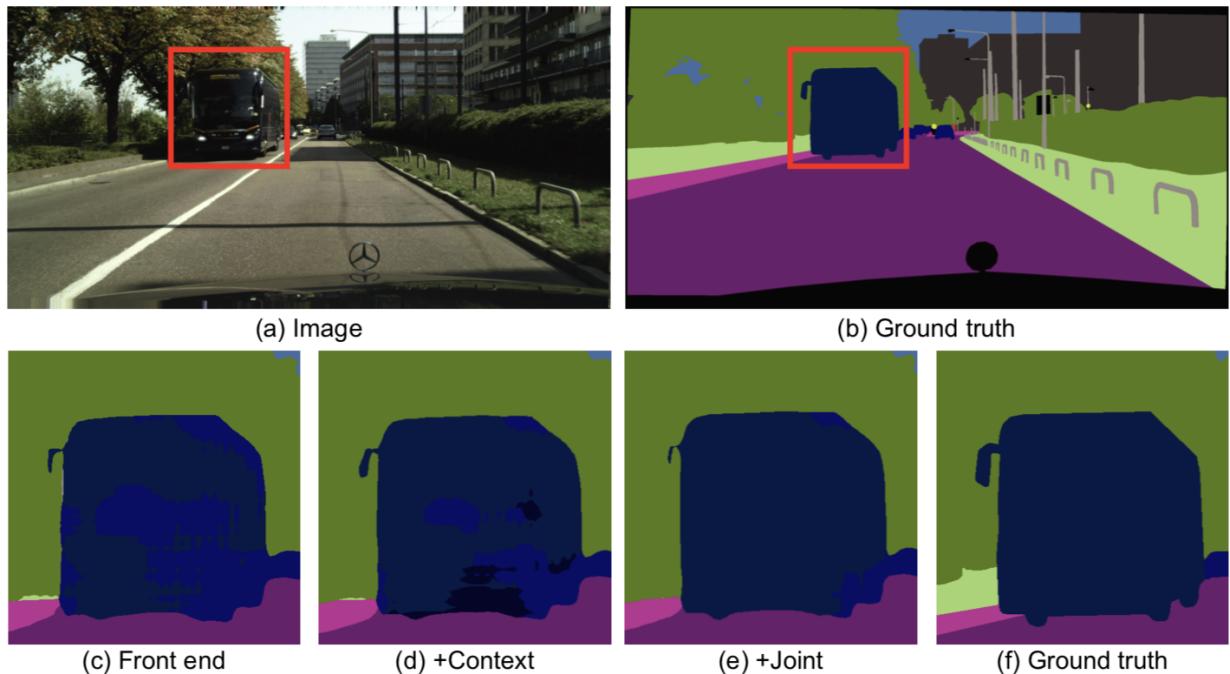


図5：異なるトレーニング段階の後にDilation10モデルによって生成された結果 (a) 入力画像。(b) グラウンドトゥルーセグメンテーション。(c) トレーニングの第1段階後にモデルによって生成されたセグメンテーション（フロントエンドのみ）。(d) コンテクストモジュールを訓練する第2段階後に生成されたセグメンテーション。(e) 第3段階後に生成されたセグメンテーション。両方のモジュールが共同して訓練される。

Road	Sidewalk	Building	Wall	Fence	Pole	Light	Sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	mean IoU
Validation set																			
97.2	79.5	90.4	44.9	52.4	55.1	56.7	69	91	58.7	92.6	75.7	50	92.2	56.2	72.6	54.3	46.2	70.1	68.7
Test set																			
97.6	79.2	89.9	37.3	47.6	53.2	58.6	65.2	91.8	69.4	93.7	78.9	55	93.3	45.5	53.4	47.7	52.2	66	67.1

表7：Cityscapesデータセットのモデル（Dilation10）によって達成されたクラス単位および平均クラスレベルのIoU。

Flat	Nature	Object	Sky	Construction	Human	Vehicle	mean IoU
Validation set							
98.2	91.4	62.3	92.6	90.7	77.6	91	86.3
Test set							
98.3	91.4	60.5	93.7	90.2	79.8	91.8	86.5

表8：Cityscapesデータセット上のカテゴリごとおよびカテゴリレベルIoUの平均値。