# Pixel-Level Prediction:

# From Geometry to Semantics

Fisher Yu

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Computer Science
Adviser: Thomas A. Funkhouser

April 2018

# Abstract

Pixel-level prediction generalizes a wide range of computer vision tasks including semantic image segmentation and dense depth prediction. They are fundamental for image recognition, receiving continual attention from the community. However, although they share common traits that may admit a general solution, they are usually studied in isolation because of different domain characteristics. This thesis aims to study the essential problems behind those tasks and shed light on a general framework.

This thesis starts with an algorithm that can predict plausible depth from almost identical images based on geometric optimization. The motion between those images is called "Accidental Motion". The analysis of accidental motion shows that motion optimization has special convexity properties. It leads to a reconstruction pipeline that can produce a plausible dense depth map for the reference image, which is shown to enable depth based camera effects.

The second part then studies learning pixel representation to predict semantic properties based on the single reference image. Previous works usually use learned upsampling to recover the pixel-level information. This work proposes to use Dilated Convolution to transform the classification networks such that high-resolution prediction is achieved without upsampling. Dilated Convolution can also render an exponential increase in receptive field, which is ideal for learning global context. A context module is proposed based on this property that can improve the network performance significantly and consistently. Dilation is still a standard component in the state-of-the-art method for semantic image segmentation.

The further study of dilated residual networks shows that same high-resolution prediction can also improve image classification results. This indicates no essential network architecture difference exists between image classification and segmentation. Further inspection of class activation maps and layer responses uncover peculiar gridding patterns and their cause. This finding leads to new designs of convolutional networks that can remove the gridding artifacts and produce activations with better spatial consistency. The new networks can improve the performance of both image classification and semantic segmentation.

The presented method and results may inspire new research in building a unified framework for image recognition of geometry and semantics.

# Acknowledgements

I would like to express my sincere gratitude to my Ph.D. advisor, Thomas Funkhouser, for his continuous support for my growth. He provides me invaluable opportunities and instructions to find my directions and initiate my incessant journey of exploring the research world. His knowledge, patience and philosophy have been the greatest encouragement for me to go through difficult times.

I am also honored to work with Vladlen Koltun. Under his mentorship, we started the investigation into dilated convolutions, which is the main topic of two of the chapters in this thesis. His pursuit of rigorousness, simplicity and succinctness became a beacon for my scientific quest. Thanks to David Gallup, Steve Seitz and other great researchers in the same group while I did my summer internship with them. This great experience led to the formation of the first chapter of this thesis.

I am also in debt to all my research collaborators and labmates during my study at Princeton University. Special thanks to Jianxiong Xiao, who was a great exemplar for young researchers. I would also like to thank Adam Finkelstein, who never ceased bringing fun to the research life, especially during the challenging moments.

Finally, I want to dedicate this thesis to my parents and all my friends, for their continuous love, support and encouragement, which are essential gradients for Ph.D. study.

“It is all a matter of clear thinking.”

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivations and Problems

Pixel-level prediction defines a common set of computer vision tasks. They require a function from a multi-channel input with spatial dimension to a structured output map corresponding to the input spatial locations. The input may be a single RGB image or include additional channels such as depth maps and other frames in the same temporal sequence. The output is either category classification or regressed value at each pixel location. It can be the semantic meaning of each pixel such as object category, or a perception property determined by the object identity and 3D relation, such as boundary prediction. The task can also be an inference of pure 3D geometry such as depth estimation. Applications of dense prediction include semantic segmentation, depth prediction, etc. Some examples are shown in Figure 1.1. Although these problems are usually studied in isolation due to domain-specific characteristics, it is important to have a unified framework for these tasks to understand the essential problems behind those tasks and facilitate the deployment of the solutions to real-world applications. This thesis aims to shed light on a general solution to pixel-level prediction problems.

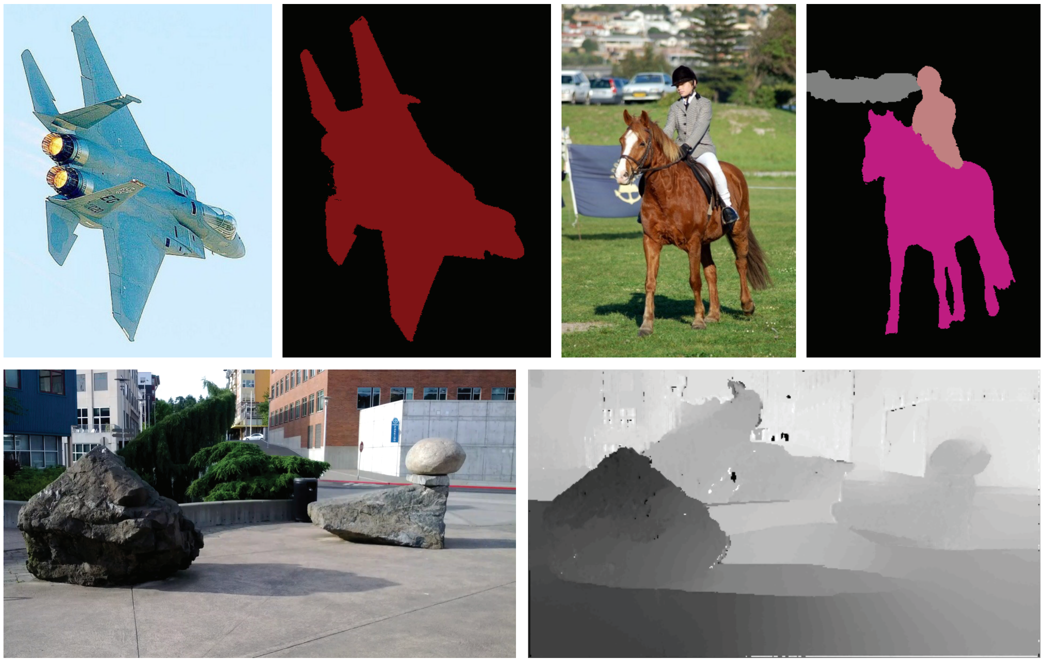


Figure 1.1: Examples of pixel-level prediction problems. The first row shows semantic image segmentation and the second dense depth prediction.

We consider the pixel-level prediction problems together because they share some common challenges. First, a single pixel doesn't contain the desired information, so the prediction will have to rely on the local context. Second, the global context may be necessary for inference, but it is hard to capture due to its vast and complicated space. Third, the pixel values are unstructured and reside in a high-dimensional space, in which not all the points are meaningful. A projection is needed to map the pixel values to a feature space with lower dimensions, in which proximity is correlated to semantic or structural relations so that we can map divisions of the space to target categories or values. Forth, most of the predictions are supposed to have spatial consistency. This constraint can further improve the prediction results when applied properly.

In the thesis, we investigate three directions to address the challenges. For the ill-defined depth estimation problem, we can obtain more information from the data acquisition process. For the general semantic inference problems, we can use Dilated

Convolution to learn high-resolution 2D representations for the input images and aggregate semantic information from the whole image.

First, we investigate single-image dense depth prediction with 3D geometric methods [88]. Our observation stems from the image capturing process. When an image is taken by a cellphone, which is the most popular camera on Flickr [1], the user will inevitably shake their hands slightly during the seconds of efforts to hold the phone steady towards the target scene. This process creates a series of images with small baselines besides the final picture. We call the motion between those images "Accidental Motion". Although it is mathematically possible to recover camera motion from those almost identical images, the existing SfM methods usually fail at such small baselines due to high condition numbers of linear equations for the initial algebraic solutions. A typical Structure from Motion (SfM) pipeline relies on the algebraic solutions to provide a good initialization for non-linear bundle adjustment. Our analysis shows that bundle adjustment on small baselines has some special convexity properties. For example, when the camera poses are fixed, the 3D point triangulation is convex. It indicates a random initialization may lead to good bundle adjustment results in our experiments. SfM only gives sparse 3D points, while a dense reconstruction can be more useful. However, because the small baselines cause noisy pixel-wise depth estimation, to resolve this problem, we borrow the idea of the densely connected conditional random field and use it on the multi-view stereo. The experiments show that dense connections between pixels can provide more robust regularization. The whole pipeline can recover plausible depth maps that enable cameras effects based on 3D geometry.

Second, we study learning high-level feature representations for each pixel [89]. We find that human eyes can easily spot the erroneous pixel depth based on high-level semantics and context. This motivates me to investigate the semantic pixel-level representation, which can potentially improve the depth estimation. Learning image representation has enjoyed great success in recent years, but it has to rely on large-scale image classification dataset for bootstrapping and fine-tuning to adapt to a domain with fewer images. Before our work, the common approach for semantic image segmentation is to upsample the representation learned from image classification. The drawback is of losing the detailed structure and causing blurred 2D predictions. Instead, we propose to use dilated convolutions to retain the high-resolution feature maps within the network. The new network does not require learning upsampling for semantic segmentation. The visual results look more pleasing, and our experiments show that the new network can improve the quantitative evaluation results significantly. We also find that dilated convolutions can also achieve exponential increase of receptive fields with a linear number of layers. This is an ideal property for aggregating contextual information from the whole image. Based on this property, we build a new context module to learn spatial consistency and context aggregation with our front-end network. This module can produce visually spatially consistent segmentation and improve the quantitative evaluation results.

Finally, we try to understand the connection between image classification and semantic segmentation [90]. Up to now, we have been using different network structures for image classification and semantic segmentation, because segmentation requires high-resolution feature representations and classification only needs image-level representation. To understand the gap, we study the effects of high spatial resolution for image classification. This controlled study can be naturally conducted with the help of dilated convolutions. Surprisingly, we find that the dilated networks can outperform their counterparts significantly, although they have the same number of parameters and layers. Further inspection of the class activations shows mysterious gridding patterns that only exist on the high-resolution activation maps. Those patterns also appear in semantic segmentation models, even though they are trained with spatial consistent supervision. Our visualization shows the gridding artifacts are unavoidable due to the combination of learned differentiation filters and dilated convolutions. Hence, we add degridding layers on top of dilated residual networks. The new class activation maps are smooth and different objects on the same image can be delineated. The new networks also improve the quantitative results of image classification and segmentation.

## 1.2 Contributions

This dissertation makes the following contributions.

First, we analyze the optimization difficulty of Structure from Motion when the image baselines are order-of-magnitude smaller than scene structure. Because the small baselines cause high uncertainty in depth estimation, we also propose to use dense conditional random field to improve the spatial consistency of the pixel-level depth estimation. Our study leads to a practical pipeline to predict pixel-level depth from a collection of images captured when a user tries to hold the camera still. This pipeline can further produce camera effects that are not possible without predicting scene geometry.

Second, we also present a comprehensive investigation into dilated convolutions for understanding the pixel-level information at the semantic level. Based on our investigation, we find that dilated convolution is suitable for building deep convolutional networks for semantic segmentation for both preserving spatial resolution and aggregating context.

Third, we further show that high-resolution feature maps for semantic image segmentation can also improve the network performance in image classification. Based on the feature map visualization, we propose several changes to the dilated residual networks, which improve the network performance both qualitatively and quantitatively.

# Chapter 2

# 3D Reconstruction from Accidental Motion

## 2.1 Introduction

When a person captures a still photo by hand, it usually takes several seconds between pointing the camera to the scene and pressing the shutter button. During this time, while one intends to hold the camera still, there is inevitable motion due to hand shaking or heart beating, especially when a lightweight camera like a smartphone, is used. We call this type of motion *accidental motion*. If a camera were to capture a short video before and/or after the capture of a still, would it be possible to use the baseline (translation) from accidental motion for 3D reconstruction? We demonstrate in this paper that indeed 3D reconstruction can be achieved, and that the resulting reconstruction can be used for a variety of applications.

In this chapter, we investigate the properties of accidental motion and find a method to reconstruct 3D information of the image sequences. There are two main challenges to this problem. First, the commonly used Structure from Motion (SfM) approaches assume that a good two-view reconstruction can be obtained with algebraic methods, which in turn depend on adequate baseline between overlapping views. In accidental motion, the maximum viewing angle for a 3D point is usually less than 0.2 degrees, where algebraic methods are very unstable. Second, the depth uncertainty is very large due to small baseline and, therefore, the previous multiview stereo methods can produce serious artifacts.

To address these issues, we find that we can use multiple images together to do SfM directly. Due to accidental motion, we use inverse depth relative to a reference view to parameterize the 3D points, which helps regularize bundle adjustment. We find that random depth and identical camera poses are good initialization for bundle adjustment with all the images. We also find that many images can help reduce uncertainty.

Given camera poses, the depth estimation of most of the pixels is noisy and has high uncertainty. Because the depth signal is weak and noisy, we find that the popular first-order CRF is not very effective in regularizing depth, and can often result in an oversmoothed depth map, as shown in Figure 2.1. We propose to use long range connections, and we show that direct connections between a pixel and its bigger neighborhood can improve the dense reconstruction in our case.

We have conducted a user study that yields empirical evidence that there is several-millimeter translation throughout the capture of a still photo. Under reasonable conditions (3-meter depth, focal length of 2000 pixels, and localization standard deviation of 1 pixel), a baseline of 3 mm over 100 frames (a few seconds at 30 fps) is enough to estimate depth with a standard deviation of 150 mm, which is low enough uncertainty for many applications.

We test our algorithm on a variety of scenes captured by a variety of users. The proposed method can indeed produce high quality depth maps, and these depth maps are good enough for RGB-D photography applications, such as synthetic aperture (focus change) and parallax effects.

## 2.2 Previous Work

We follow the common pipeline to build dense 3D models from a collection of images. We first do SfM to estimate the viewing parameters of each image and then use them to do multiview stereo to get dense reconstruction. A wealth of previous work has studied this two problems, and we mention some of them here to show the difference of our system.

**Structure from motion** has been actively studied for a long time and we have got a good understanding of the geometric properties of estimating sparse structure and camera poses [27]. Bundle adjustment is commonly used to obtain the optimal estimates [85]. Nonlinear least squares is used to measure the projection errors because of its nice error modeling properties. But it is usually difficult to optimize the nonlinear cost function and a good initialization is critical. [75] presents a successful way to do incremental bundle adjustment, which relies on two-view reconstruction. However, when the motion is very small as in our case, the two-view reconstruction is ill conditioned and therefore it can't provide reliable initialization. Discrete optimization [15] is also proposed to initialize structure and camera parameters. But the optimization itself is a hard problem and there is a tradeoff between accuracy and complexity. To work around the nonlinearity of the cost functions, some other error measures are also proposed. [36, 73, 37] propose to use $L_\infty$ norm instead of $L_2$ to measure the reprojection error because the resulting cost function is convex. But $L_\infty$ is not robust to outliers, which are unavoidable in most of the applications. We will show that even in our case, where the feature matching is supposed to be easier than the general case due to little view point and illumination change, we still need to deal with outliers in feature matching. Robustifing the cost function can help improve the reconstruction result.

Instead of doing bundle adjustment with multiple images, some works [81, 11, 77, 83, 57] propose factorization methods to do multiview SfM directly. Potentially, those methods should be used as initialization for bundle adjustment. However, in our experiments, we find that in presence of feature localization noise and outliers, these methods are unstable and our proposed initialization is the most effective.

Several works [56, 58, 10] study the ambiguity properties of structure from small motion and propose some algorithms. But the analysis of the bundle adjustment is mainly for two-view case. In this chapter, we will present analysis for the multiview case and show that with the assumption of small motion, tasks such as estimating point depth can be easier to solve. Although several researchers [57, 10] have proposed methods to reconstruct sparse structure, to our knowledge, our method is the first to deal successfully with outliers and to work in practice. A recent work [54] proposes to use a similar initialization approach to ours to initialize a tracking system. But their goal is not to find a 3D structure and we find that random depth initialization works better than their proposed constant depth initialization.

**Multiview stereo** When the camera motion is very small, the view change is very small. We aim to estimate depth for each pixel in the reference view instead of a complete 3D model. Therefore stereo methods are more relevant here. Even if SfM can provide perfect camera parameters, the photo-consistency measurement at each pixel can still be noisy due to various reasons such as image noise and the aperture problem. Various methods have been proposed to solve this problem by smoothing or regularizing the depth estimation. The Conditional Random Field (CRF) framework is one of the most successful methods [6, 31]. A probabilistic model is used to associate adjacent pixels to encourage them to have similar depth values. Second order Markov Random Field (MRF) is also proposed [38, 87, 40] to avoid the fronto-parallel bias. However, those methods can only connect adjacent pixels, although they are global methods. In our experiments, we find that the low order connection can't regularize our depth effectively. Therefore, we propose to connect pixels over even longer ranges. The inference is made possible by the recent development of high dimensional Gaussian Filtering [2] and the mean field method [41]. We will show that this method can effectively regularize noisy depth maps estimated from weak data terms. Some local methods [63] based on cost-volume filtering have also been proposed to solve the stereo problem. Our method bears a similarity to the filtering methods, but our method is based on a global formulation, which usually performs better than local methods, as evaluated on Middlebury benchmark [69].

## 2.3 Structure from Motion

Given feature correspondences between images, we use bundle adjustment to get the 3D structure and camera poses of these images. It is well known that the cost function of bundle adjustment is nonlinear and it is easy to get stuck in a local minimum that is far away from the global minimum. It is hard to even solve part of the problem [28]. Incremental bundle adjustment based on two view reconstruction is often used to get a good initialization. Surprisingly, we find experimentally that in the small motion case, identical camera poses and random point depth are good initialization for the cost function. What's more, because the view change is small, we can parameterize the 3D point position as depth in the reference view, which also contributes to the success of bundle adjustment. The small motion assumption also makes the analysis of the cost function in bundle adjustment easier. In this section, we first analyze the cost function of bundle adjustment with the assumption of small motion (both rotation and translation). Although the bundle adjustment is still a complicated optimization problem under this assumption, we can show that it has some nice properties. When the camera poses are fixed, it is convex to get the depth of a feature relative to a reference view. Also, it is convex to optimize the rotation for the points at infinity when an approximation is used. We will present our method after proofs of the properties. In Section 2.5, we demonstrate that our method is effective in reasonably restricted environments.

### 2.3.1 Definitions

Assume we have an image sequence with $N_c$ images and $N_p$ points in 3D, where every point is visible to all the images. Let the camera of the first image be the reference view, and the $i$-th camera is related to it by a relative rotation matrix $R_i$ followed by relative translation $\mathbf{T}_i = [T_i^x, T_i^y, T_i^z]^T$. Assume $P_j$ is the position of the $j$-th point in the coordinate system of the reference camera. Its position in the coordinate system of the $i$-th camera is $\mathbf{R}_i\mathbf{P}_j + \mathbf{T}_i$.

Let $\boldsymbol{\Theta} = [\theta_i^x, \theta_i^y, \theta_i^y]$ be the rotation angles of the $i$-th camera. With the assumption of small angles, $\mathbf{R}_i$ can be approximated by

$$\mathbf{R}_i = \begin{bmatrix} 1 & -\theta_i^z & \theta_i^y \\ \theta_i^z & 1 & -\theta_i^x \\ -\theta_i^y & \theta_i^x & 1 \end{bmatrix}. \tag{2.1}$$

To make the resulting optimization easier, we parameterize each 3D point by its inverse depth. so we have $\mathbf{P}_j = \frac{1}{w_j}[x_j, y_j, 1]^T$, where $(x_j, y_j)$ is the projection of $\mathbf{P}_j$ in the reference image. The projection of $\mathbf{P}_j$ on the $i$-th image is $\mathbf{p}_{ij} = [p_{ij}^x, p_{ij}^y]^T$. Let $\pi : \mathcal{R}^3 \rightarrow \mathcal{R}^2$ be the projection function, that is, $\pi([x, y, z]^T) = [x/z, y/z]^T$.

### 2.3.2 Analysis

We use the $L_2$ norm to measure the reprojection error because it has nice statistical interpretation and can be robustified [85].

Based on the above definitions, we can define the cost function of bundle adjustment in the retina plane as

$$\begin{aligned} F &= \sum_{i=1}^{N_c}\sum_{j=1}^{N_p} ||p_{ij} - \pi(R_iP_j + T_i)||^2, \\ &= \sum_{i=1}^{N_c}\sum_{j=1}^{N_p} (\frac{e^x_{ij} + f^x_{ij}w_j}{c_{ij} + d_{ij}w_j})^2 + (\frac{e^y_{ij} + f^y_{ij}w_j}{c_{ij} + d_{ij}w_j})^2, \end{aligned} \tag{2.2}$$

where

$$\begin{aligned} a^x_{ij} &= x_j - \theta^z_i y_j + \theta^y_i, \\ b^x_{ij} &= T^x_i, \\ a^y_{ij} &= y_j - \theta^x_i + \theta^z_i x_j, \\ b^y_{ij} &= T^y_i, \\ c_{ij} &= -\theta^y_i x_j + \theta^x_i y_j + 1, \\ d_{ij} &= T^z_i, \\ e^x_{ij} &= p^x_{ij} c_{ij} - a^x_{ij}, \\ f^x_{ij} &= p^x_{ij} d_{ij} - b^x_{ij}, \\ e^y_{ij} &= p^y_{ij} c_{ij} - a^y_{ij}, \\ f^y_{ij} &= p^y_{ij} d_{ij} - b^y_{ij}. \end{aligned} \tag{2.3}$$

**Depth Estimation** Assume that the correct camera poses are given and fixed. The depth estimation is to find the depth of a point minimizing

$$F_i(w_j) = \sum_{i=1}^{N_c} f^x_j(w_j) + f^y_j(w_j), \tag{2.4}$$

where $f^x_j(w_j) = (\frac{e^x_{ij}+f^x_{ij}w_j}{c_{ij}+d_{ij}w_j})^2$ and $f^y_j(w_j) = (\frac{e^y_{ij}+f^y_{ij}w_j}{c_{ij}+d_{ij}w_j})^2$. We will prove estimating the depth is easier in the context of small motion.

First, consider the general form of $f_j^x$ and $f_j^y$, $f(x) = (\frac{x-a}{x-b})^2$, where $a$ and $b$ are the zero and pole of the function, respectively. When $a > b$, the function is convex in $(b, \frac{3a}{2} - \frac{b}{2})$. When $a < b$, the function is convex in $(\frac{3a}{2} - \frac{b}{2}, b)$.

Assume that $f_j^x(\bar{w}_j^x) = 0$, that is, $\bar{w}_j^x = -\frac{e_{ij}}{f_{ij}}$. Because $c_{ij} \approx 1$ and $|d_{ij}| \ll \frac{1}{w_j}$, $w_j \ll |\frac{c_{ij}}{d_{ij}}|$. So $f_j^x(w_j)$ is convex in $(0, |\frac{c_{ij}}{2d_{ij}}|)$, so is $f_j^y(w_j)$. Hence, $F(w_j)$ is convex in $(0, \min_i |\frac{c_{ij}}{2d_{ij}}|)$. Since $|\frac{c_{ij}}{2d_{ij}}|$ is supposed to be far greater than reasonable values of $w_j$, we can easily optimize $w_j$ for the reprojection error in Equation 2.4. Also, note that if there is noise in the detection $\mathbf{p}_{ij}$, it doesn't change $c_{ij}$ and $d_{ij}$, and hence the convex interval $(0, \min_i |\frac{c_{ij}}{2d_{ij}}|)$ of $F_i(w_j)$. What's more, the convexity analysis of the cost function doesn't depend on the approximation of the rotation matrix. It is an exact property of depth estimation with small motion.

**Points at Infinity** If the points are approximately at infinity, the cost function in Equation 2.2 can be approximated by

$$F \approx \sum_{i=1}^{N_c} \sum_{j=1}^{N_p} (e_{ij}^x)^2 + (e_{ij}^y)^2. \tag{2.5}$$

It is a convex function of the camera rotation angles on the domain around 0.

**Depth Uncertainty** Consider a rectified stereo pair separated by a baseline $b$, observing a point at inverse depth $w$. The relationship between disparity and depth is given by $w = \frac{d}{fb}$, where $d$ is the disparity and $f$ is the focal length. Ignoring quantization errors and mismatches, we can obtain the inverse depth estimation at any single pixel, namely

$$\mathrm{Var}[\hat{w}] = \mathbf{E}[(\frac{d+\epsilon}{fb} - \frac{d}{fb})^2] = \frac{\mathrm{Var}[\epsilon]}{f^2 b^2}, \tag{2.6}$$

where $\epsilon$ is the feature localization error. Unlike analyzing the variance of depth, we don't have to take first-order approximation here. Similarly, assuming that we have $n$ observations of the point and they have the same variance, we can get the variance of the combined estimation $\hat{w} = \frac{1}{n}\sum_{i=1}^{n}\hat{w}_i$:

$$\begin{aligned} \mathrm{Var}[\hat{w}] &= \frac{1}{n^2 f^2 b^2}\mathbf{E}[(\sum_{i=1}^{n}\epsilon_i)^2] \\ &= \frac{1}{f^2 b^2}(\frac{1}{n} + \rho(1 - \frac{1}{n}))\mathrm{Var}[\epsilon], \end{aligned} \tag{2.7}$$

where $\mathrm{Cov}[\epsilon_i, \epsilon_j] = \rho\mathrm{Var}[\epsilon]$ for all $i, j$ between 1 and $n$ and $i \neq j$, and $\mathrm{Var}[\epsilon_i] = \mathrm{Var}[\epsilon]$. This indicates that if the feature detection errors are independent, the standard deviation of the inverse depth estimation decrease linearly with $\sqrt{n}$. However, if the feature detection errors are fully correlated, multiple observations don't help reduce uncertainty. Similar conclusion can be drawn for depth [21].

### 2.3.3 Initialization

A good initialization is crucial to finding good minima of reprojection errors. Because of the results in Section, 2.3.2 we conjecture that a random initialization for structure may give good results. Given a sequence of images, we select a reference view and initialize all the camera poses with zero rotation and translation. As mentioned above, the points are parameterized by inverse depth. The projections of the 3D points are proposed by feature tracking across the images. First, corner features [71] are detected in the reference image. Then, instead of tracking the corners in the image sequence order, we track all the corners from the reference image to each of the other images with Kanade-Lucas-Tomasi (KLT) [53, 80] feature tracker. This can effectively reduce the accumulative localization error of feature tracking. To remove the tracking outliers, we require that all the features can be tracked to all the non-reference images and the maximum color gradient difference per pixel between the two patches should be under a threshold. KLT method can provide subpixel accuracy, and this is critical when the camera motion and therefore the feature movement are small.

### 2.3.4 Optimization

We optimize the cost function of bundle adjustment in Equation 2.2 with Ceres Solver [3]. Robustifiers are optionally used in the cost function. The camera of the reference view is fixed at the coordinate origin. Usually, the outliers can be neglected after the feature tracking and selection in initialization. However, we find cases where robustifiers can improve the reconstruction results. On the other hand, after each optimization, we remove the points with negative depth and optimize again with the remaining points.

## 2.4 Dense Reconstruction

After getting structure from motion, we want to densely reconstruct the 3D scene by estimating the depth of the images. Because all the input images capture the scene from a similar viewpoint, we can only get a 3D structure seen from the common viewpoint. Therefore, we aim to get a depth map of a reference view as the 3D reconstruction output. Because the depth signal at each pixel tends to be noisy in our case, we adopt plane sweeping together with the CRF framework [31] to solve a smooth depth map.

One distinct attribute of multiview stereo from small baseline images is that the confidence of depth minima is low in general instead of just in textureless areas. Therefore, the details can be easily smoothed out, as shown in Figure 2.1. To preserve the details while smoothing the depth map, we propose to use long range connection

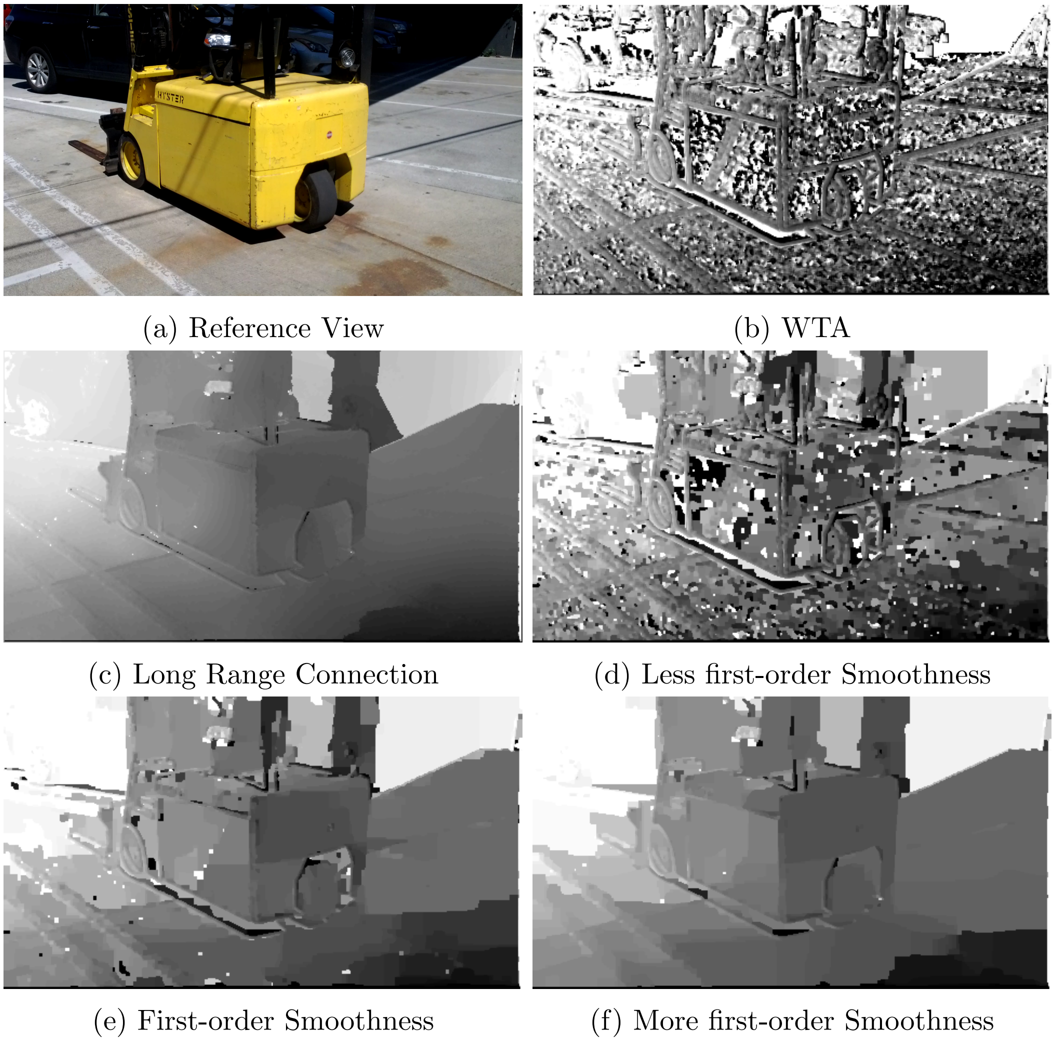

(a) Reference View
(b) WTA
(c) Long Range Connection
(d) Less first-order Smoothness
(e) First-order Smoothness
(f) More first-order Smoothness

Figure 2.1: Comparison of first-order and long range connection. (b) shows the data term. (c) is the result optimized based on the long range connected model. (d) to (f) shows the graph cut solution of the first-order smoothness with increasing regularization. Because the data term is very noisy, first-order regularization always oversmooth the estimated depth to reduce noise.

between pixels in the CRF energy function, which can pass information to a pixel effectively.

### 2.4.1 Formulation

The input is a set of images. Let $\mathcal{I}$ be the index set of the pixels in a reference view I, and $\mathrm{I}(i), i \in \mathcal{I}$, is the color of the $i$-th pixel. The goal is to determine a dense depth map, D, of the reference view. Let L map each pixel index $i \in \mathcal{I}$ to a 2D location in the image. Let P be the photo-consistency function such that $\mathrm{P}(i, d)$ is the photo-consistency score of the $i$-th pixel at distance $d$.

The energy we intend to minimize is

$$E(\mathrm{D}) = E_p(\mathrm{D}) + \alpha E_s(\mathrm{D}). \tag{2.8}$$

$E_p$ is the standard photo-consistency term of the form

$$E_p(\mathrm{D}) = \sum_{i \in \mathcal{I}} \mathrm{P}(i, \mathrm{D}(i)), \tag{2.9}$$

which can be obtained by plane sweeping algorithm [13].

$E_s$ is the smoothness term to regularize the depth estimation. It often represents first-order or second-order CRF model to connect and pass information between adjacent pixels. However, we find that those adjacent connected model can't effectively regularize the noisy data term. Hence, we propose to connect pixels with longer range so that the photo-consistency measurement can be effectively aggregated from an area to a pixel in it.

To build a connection between pixels that are not adjacent, we introduce the function $\mathrm{C}(i, j, \mathrm{I}, \mathrm{L}, \mathrm{D})$, which gives a score for the depth assignment of the $i$-th and the $j$-th pixels based on the color intensities and their locations in the reference images. So $E_s$ is the long range connection term of the form

$$E_s(\mathrm{D}) = \sum_{i \in \mathcal{I}, j \in \mathcal{I}, i \neq j} \mathrm{C}(i, j, \mathrm{I}, \mathrm{L}, \mathrm{D}), \tag{2.10}$$

and

$$\mathrm{C}(i, j, \mathrm{I}, \mathrm{L}, \mathrm{D}) \;=\; \rho_c(\mathrm{D}(i), \mathrm{D}(j)) \times \exp(-\frac{||\mathrm{I}(i) - \mathrm{I}(j)||^2}{\theta_c} - \frac{||L(i) - L(j)||^2}{\theta_p}), \tag{2.11}$$

where $\rho_c$ is robust measurement of depth difference, and $\theta_c$ and $\theta_p$ are parameters to control the connection strength and range. We choose $\rho_c$ to be the truncated linear function, i.e., $\rho_c = \min(t, |\mathrm{D}(i) - \mathrm{D}(j)|)$, where $t$ is a threshold. The purpose of $E_c$ is to connect pixels within an area with similar colors such that they can have consistent depth, since they are more likely to belong to the same object.

### 2.4.2 Optimization

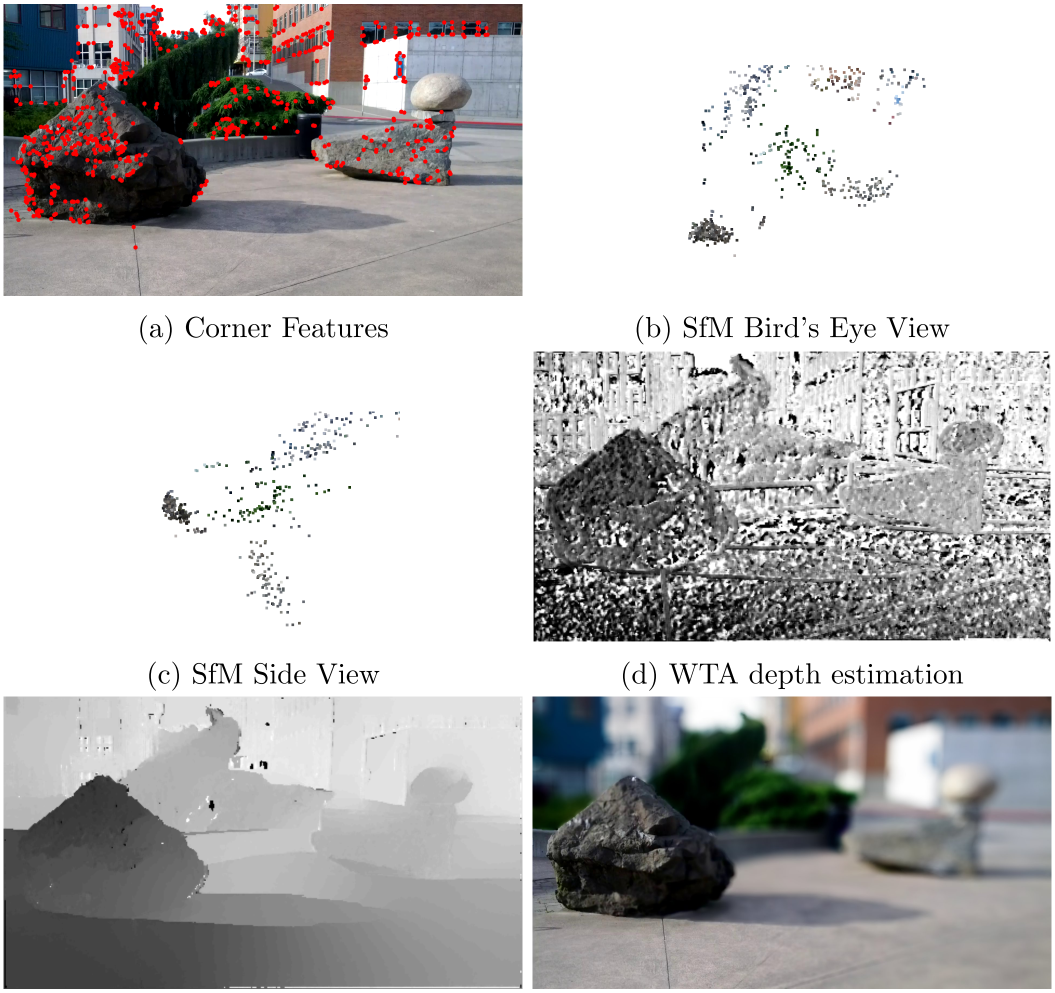
We use the mean field method with an efficient implementation proposed in [41] to optimize Equation 2.8. It can solve the dense CRF model and give a smooth depth map efficiently.

## 2.5 Experiments

We evaluate our methods on both synthetic and real data. The real data is collected by a smartphone camera, and it is captured in the video mode at 24 frames per second. To make our system practical to real world applications, we limit the number of images to 100, which is about a 4-second video. The camera intrinsic parameters are calculated from the factory specification of the phone and the image distortion is not accounted for. Better results are expected when a better camera is used.

### 2.5.1 User Behavior

We have conducted a user study to determine the magnitude of accidental translational motion during still photography. To measure camera motion, we asked users to capture videos of a calibration pattern at a distance of roughly 0.5 meters. Users were instructed to hold the camera steady, as if they were capturing a photograph, for a duration of 5 seconds. We evaluated 9 participants and two cameras: a Google Nexus 4 smartphone, and a Canon PowerShot S95 point-and-shoot. The results are shown



(a) Corner Features

(b) SfM Bird's Eye View

(c) SfM Side View

(d) WTA depth estimation

(e) Smoothed depth estimation

(f) Synthetic aperture

Figure 2.2: Pipeline of our system. (a) We select the first image in a sequence as the reference view. Corner features (Red dots) are extracted in the reference view and tracked to the other images. (b) and (c) show the SfM result with initialization of random structure and identical camera poses. (d) WTA of the photo-consistency at each pixel (e) The smoothed depth estimation based on our proposed energy function with long range connections. (f) Given the depth map, we can refocus on part of the image.

in Figure 2.3. From this study, we observe that after 3 seconds, the camera centers exhibit a standard deviation of 3.9 mm, which yields sufficient baseline to obtain a good reconstruction under reasonable conditions. For example, for a scene depth of 3 meters, 100 frames of video, feature localization (or disparity) standard deviation of 1 pixel, and a focal length of 2000 pixels, we would expect a depth standard deviation of 0.115 meters, assuming measurements are uncorrelated. We have asked several users

| Google Nexus 4 (smartphone) | | | | |
|---|---|---|---|---|
| | | Translation | | |
| All | Translation | stdev. (mm) after | | |
| users | speed (mm/s) | 1s | 2s | 3s |
| Mean | 18.07 | 2.18 | 3.35 | 3.81 |
| Stdev. | 6.67 | 1.11 | 1.99 | 2.31 |

| Canon PowerShot S95 (point-and-shoot) | | | | |
|---|---|---|---|---|
| | | Translation | | |
| All | Translation | stdev. (mm) after | | |
| users | speed (mm/s) | 1s | 2s | 3s |
| Mean | 9.23 | 1.71 | 3.02 | 3.99 |
| Stdev. | 2.10 | 0.65 | 1.23 | 1.88 |

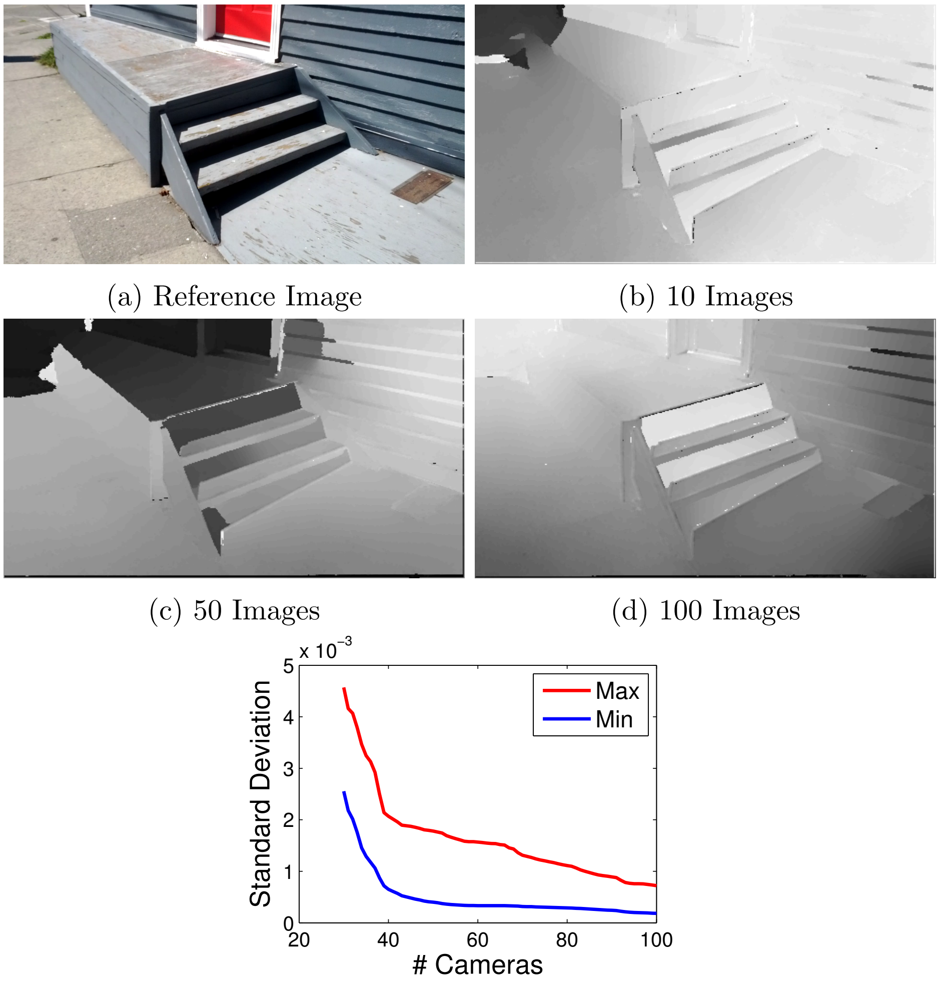
Figure 2.3: Camera translation statistics obtained from a user study of 9 participants. Users were asked to record video of a calibration pattern and hold the camera steady, as if they were capturing a photograph. Although the smartphone moves faster than the point-and-shoot (perhaps due to the weight and form), both cameras exhibit similar standard deviation of translation (camera centers).

to capture a 4-second video of a natural scene using a Galaxy Nexus smartphone, and our algorithm generates similar results shown in this chapter.

### 2.5.2 Structure from Motion

We follow the method described in Section 2.3 in our experiments. An image sequence of a video is taken as the input. The first image of a sequence is selected as the reference view. When we remove the feature tracking outliers by average pixel difference in a patch, we usually use 6 as the threshold for a 8-bit encoded gray image. All the 3D positions of the feature points are parametrized by their inverse depth relative to the reference view. Before the bundle adjustment, all the cameras have zero rotation and translation, and all the points have uniformly random depth between 2 and 4 meters.

**SfM Results** The bundle adjustment results are shown in Figure 2.2b and 2.2c. It demonstrates that our simple initialization method is effective in the small motion



(a) Reference Image

(b) 10 Images

(c) 50 Images

(d) 100 Images

(e)

Figure 2.4: Change of smoothed depth maps with different number of images. Darker color indicates closer depth. More images decrease the uncertainty of the reconstruction and also reduce the influence of outliers. (b) to (d) show the change the structure with their smoothed depth map. (e) shows the change of depth estimation uncertainty with number of cameras in this example. The Y axis shows the standard deviation of the inverse depth. The maximum and minimum of the uncertainty continue decreasing with addition of images. The depth uncertainty is measured with camera poses fixed. Please note darker means closer.

scenario. Since we don't have to do two-view reconstruction for each pair of images or solve hard optimization problem [15], SfM is very fast. With about 1000 points and 100 cameras, it usually takes several seconds on a modern desktop.

Feature tracking outliers are inevitable, but in most of the cases, they don't affect the result. However, when there are too many outliers, a robustifier can be used as in the general structure from motion problem. We observe that when the robustifier is not necessary, the SfM results look better without it.

**Multiple Images** To understand how the multiple images help the reconstruction, we can first look at the depth estimation uncertainty. As shown in Figure 2.4e, the depth uncertainty of the 3D points decreases with more input images. It shows that in the case of KLT tracked features, more images can help reduce the tracking noise.

To understand how different numbers of images change final structure, we did bundle adjustment with different number of images while fixing the detected features and their matching. Since the camera intrinsic paramters are known and the 3D points are reconstructed up to scale, we first normalize the inverse depth values to have the same mean and variance. One of the results is shown in Figure 2.5. The blue curve shows the structure error measured by sum of squared difference between the models reconstructed by certain number of images and all the images. The green curve shows the baseline between a camera and the reference camera in the model reconstructed by all the images. only the points with middle 90% depth ranking are considered in normalization to reduce the effects of outliers. As we can observe in Figure 2.5a, there is a big error jump between 60 and 70 cameras. Since the baseline doesn't significantly increase, the error change may be because of the matching outliers. If robustifier is added to the cost function, there is no sudden error change, as shown in Figure 2.5b.

Figure 2.4 shows the evolution of the structure in 2D with depth maps. When more images are used, the structure gradually becomes better. In some cases, we observe that the structure is already good enough when 50 images are used, although more images can decrease the point position uncertainty.

Therefore, more images can help reduce the reconstructed depth uncertainty and the effects of possible outliers.

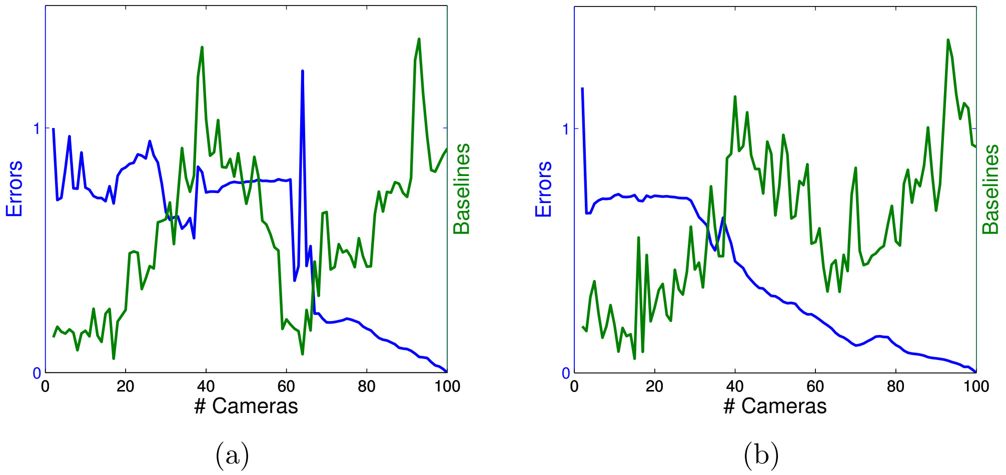


Figure 2.5: How multiple images help the reconstructed result. (a) shows that there is a sudden change in the reconstruction error. (b) shows that this sudden change is due to matching outliers and in general, multiple images can help reduce the effect of outliers.

**Points at Infinity** In Section 2.3, we mentioned that optimizing with points at infinity is equivalent to convex optimization of rotation. In practice, as the distant points are approximately at infinity, they play an important role in resolving the ambiguities between camera rotation and translation. If we remove the distant points, the bundle adjustment can easily get stuck in a local minimum. Even if we initialize the bundle adjustment with a good structure, the bundle adjustment can still distort the structure due to feature noise.

### 2.5.3 Dense Reconstruction

After getting camera poses from SfM, we can do a dense reconstruction using the method in Section 2.4. We will show the role of each term in the energy function in Equation 2.8 and argue that the terms are necessary to get plausible depth map.

**Data Term** If we only optimize $E_a$ of the energy function in Equation 2.8, we will get the noisy depth map that optimize the photo consistency at each pixel, which is winner-take-all (WTA), as shown in Figure 2.2d. We observe that the planes, such as the ground and the wall, present consistent depth values in general, though the values are noisy.

**First-Order Smoothness** Figure 2.1d-f show regularized depth maps with first-order smoothness. We observe that although some areas of the depth map are still noisy, part of it is already oversmoothed. When the regularization is weak, the estimated depth is still noisy. When the noise is reduced to a good level, the estimated depth is oversmoothed and an object is reconstructed to several layers. This motivates us to seek long connection between pixels to pass the information more effectively.

**Long Range Connection** Instead of only connecting adjacent pixels for smoothness, we connect pixels with longer range. This can effectively accumulate the information from a selected neighborhood. Inspired by the recent works of joint segmentation and stereo estimation, we first smooth the reference image with mean shift before using its color to compute the pixel connection weight in Equation 2.11. For an image of size 480 by 270, we normally choose $\theta_c$ from 20 to 30 and $\theta_p$ from 5 to 9. Greater $\theta_p$ should be used for higher resolution image. Because of the efficient implementation of mean field inference, the running time doesn't change with the values of $\theta_c$ and $\theta_p$. The connection threshold $t$ used in Equation 2.11 is chosen to be a fixed percentage of the total label number, which is 15% in our system. Because the truncated linear function can be implemented as two convolutions of 1D box filtering, the running time is linear to the number of depth labels. The results are shown in Figure 2.2e.

### 2.5.4 Points at Infinity

The points at infinity play an important role in resolving the ambiguity of the camera poses. Figure 2.6 shows that what may happen when the points in the background are removed. As mentioned in Section 2.3.2, we observe the structure in the foreground is slanted.

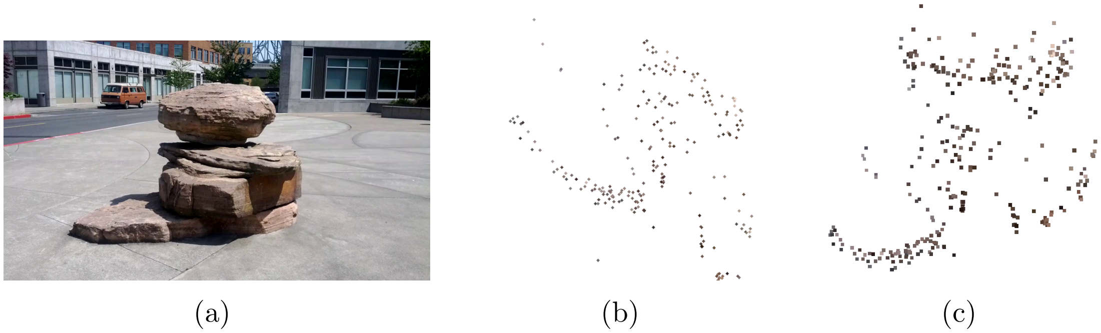


Figure 2.6: SfM results of only foreground points. The first half of the feature points are removed ranked by depth in descending order. The bundle adjustment is initialized by the structure reconstructed with all the feature points. (a) shows the original image. (b) shows the model reconstructed with only the foreground points. (c) shows the model with all the points. (b) and (c) view the model from a similar view point. The planes in (b) are slanted.

### 2.5.5 Feature Matching Outliers

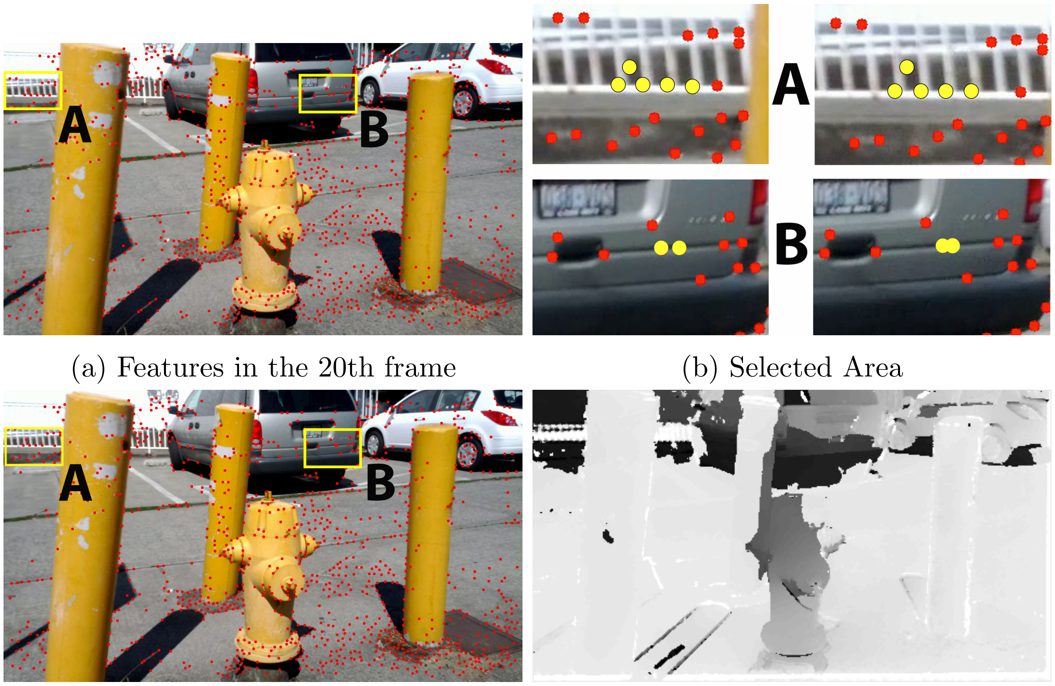
Although the camera motion is small and it is easier to track the features compared to general camera motion, feature tracking outliers are unavoidable. Figure 2.7 shows one example with so many of outliers that robustifier [85] is necessary in the bundle adjustment, as mentioned in Section 2.3 in the paper. Figure 2.8c shows that we can still get good reconstruction results with robustifier.

### 2.5.6 More Results

More reconstruction results are shown in Figure 2.8 and 2.9. To show the sparse 3D structure more clearly, we also show them in the accompanied video. The view angle distribution of 3D points are shown in Figure 2.8d. For each point, we calculate its view angles between the reference view and the other views. The cumulative distribution of 25%, median and 75% percentiles of the view angles are shown. It shows that the baselines are small for all the scenes. As measured in the user study, the hand motion is generally several millimeters. In our experiments, we find that as long as the conditions mentioned in Section 2.1 are met, a good structure can always be obtained.

### 2.5.7 Application

The reconstructed depth map can facilitate a lot of applications that are nearly impossible with a single color image. For example, we can use the 3D information to simulate different aperture effects or synthesize new views. To test our depth map is good enough for such applications, we can do refocus of the reference image. As



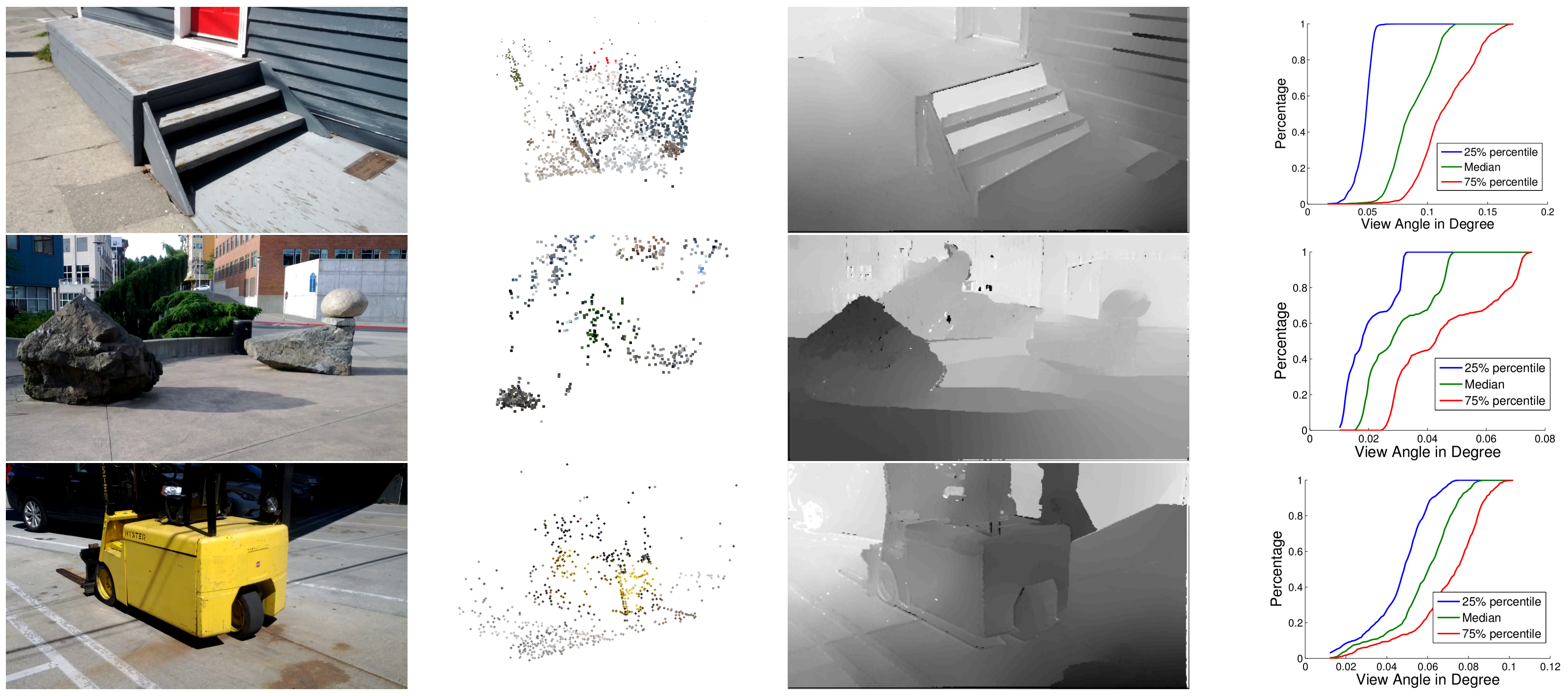
(a) Features in the 20th frame

(b) Selected Area

(c) Features in the 60th frame

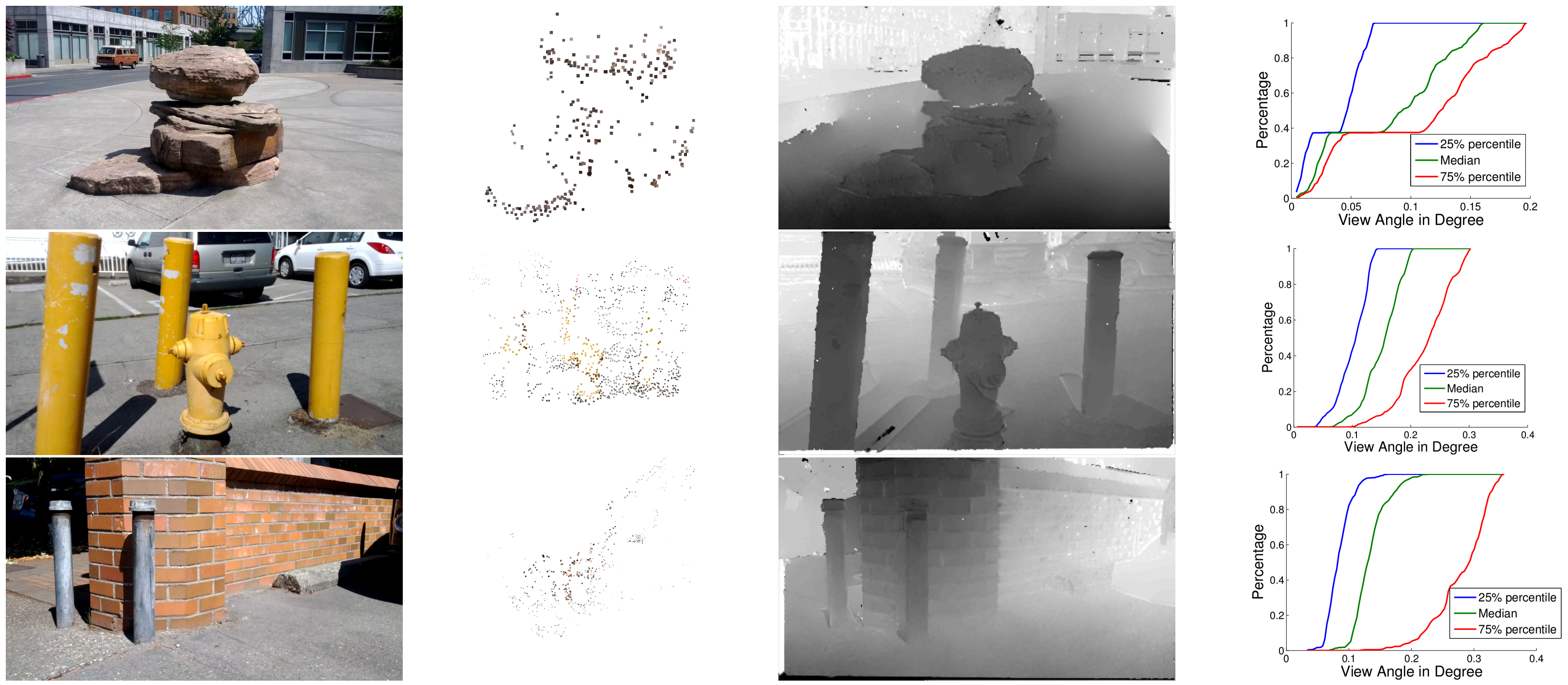
(d) Depth Map

Figure 2.7: When feature tracking outliers are present, we can use robustifier [85] in bundle adjustment. In (a) and (c), two frames in the sequence are shown with features plotted in red. The yellow rectangles highlight the tracking outliers highlighted in yellow in (b). (d) Without robustifier, the resulted depthmap doesn't show the real structure. With the help of robustifier, we can still get a good depth map as in Figure 2.8c.



(a) Reference View (b) SfM Bird's Eye View (c) Depth Map (d) View Angle Distribution

Figure 2.8: Reconstruction results at each stage. First, an image is selected as the reference view for this sequence. Then, we get the sparse structure and camera poses by bundle adjustment with our proposed initialization. As shown in (b), although the distant points have big uncertainty, the foreground points recover the shape very well. A clearer view of the structures are shown in the supplementary video. Given the camera poses, we can do dense reconstruction by calculating photoconsistency at each pixel and the final depth map regularized by the CRF model with long range connection is shown in (c). (d) shows the distribution of the view angles of the 3D points.



(a) Reference View (b) SfM Bird's Eye View (c) Depth Map (d) View Angle Distribution

Figure 2.9: More results following Figure 2.8

shown in Figure 2.2f, the generated depth map can clearly show the depth change of the objects in the scene.

## 2.6 Conclusion

We propose the first practical system to reconstruct 3D structure from small motion image sequences. We discover that in the case of small motion, random point depth relative to a reference view and identical camera poses are good initialization for the bundle adjustment cost function, even in presence of outliers. Although the reconstructed 3D points at the background have very high uncertainty, the foreground points clearly show the 3D structure. We provide some analysis of the cost function and find some of its nice properties with the assumption of small motion. Further, based on the noisy nature of the photo consistency measurement, we propose to use long range connection between pixels to regularize the depth map, and the resulted depth map looks much better than only using connections between adjacent pixels. We also demonstrate that the resulting depth map has enough quality to make perceptually plausible refocused images.

The presented algorithm still makes many discernible mistakes. For example, in Fig 2.8c, we can feel the depth map is wrong at some places because of discontinuity on the ground. We can realize it because our brain first infer semantics of those regions, i.e., planar ground. Then we understand pixels on the ground should have continuous depth. This semantic information can also be applied to the algorithms as a smoothness prior. However, the framework discussed in this chapter only utilizes 3D geometric method together with features extracted from small patches. It does not make inference about the semantic meaning of each pixel. To better address this problem, in the next chapter, we discuss convolutional networks that can effectively group the pixels based on their semantics.

# Chapter 3

# Multi-Scale Context Aggregation By Dilated Convolutions

## 3.1 Introduction

Many natural problems in computer vision are instances of dense prediction. The goal is to compute a discrete or continuous label for each pixel in the image. A prominent example is semantic segmentation, which calls for classifying each pixel into one of a given set of categories [30, 72, 39, 41]. Semantic segmentation is challenging because it requires combining pixel-level accuracy with multi-scale contextual reasoning [30, 19].

Significant accuracy gains in semantic segmentation have recently been obtained through the use of convolutional networks [47] trained by backpropagation [67]. Specifically, [52] showed that convolutional network architectures that had originally been developed for image classification can be successfully repurposed for dense prediction. These reporposed networks substantially outperform the prior state of the art on challenging semantic segmentation benchmarks. This prompts new questions motivated by the structural differences between image classification and dense prediction. Which aspects of the repurposed networks are truly necessary and which reduce accuracy when operated densely? Can dedicated modules designed specifically for dense prediction improve accuracy further?

Modern image classification networks integrate multi-scale contextual information via successive pooling and subsampling layers that reduce resolution until a global prediction is obtained [42, 74]. In contrast, dense prediction calls for multi-scale contextual reasoning in combination with full-resolution output. Recent work has studied two approaches to dealing with the conflicting demands of multi-scale reasoning and full-resolution dense prediction. One approach involves repeated up-convolutions that aim to recover lost resolution while carrying over the global perspective from downsampled layers [55, 18]. This leaves open the question of whether severe intermediate downsampling was truly necessary. Another approach involves providing multiple rescaled versions of the image as input to the network and combining the predictions obtained for these multiple inputs [17, 48, 9]. Again, it is not clear whether separate analysis of rescaled input images is truly necessary.

In this work, we develop a convolutional network module that aggregates multi-scale contextual information without losing resolution or analyzing rescaled images. The module can be plugged into existing architectures at any resolution. Unlike pyramid-shaped architectures carried over from image classification, the presented context module is designed specifically for dense prediction. It is a rectangular prism of convolutional layers, with no pooling or subsampling. The module is based on dilated convolutions, which support exponential expansion of the receptive field without loss of resolution or coverage.

As part of this work, we also re-examine the performance of repurposed image classification networks on semantic segmentation. The performance of the core prediction modules can be unintentionally obscured by increasingly elaborate systems that involve structured prediction, multi-column architectures, multiple training datasets, and other augmentations. We therefore examine the leading adaptations of deep image classification networks in a controlled setting and remove vestigial components that hinder dense prediction performance. The result is an initial prediction module that is both simpler and more accurate than prior adaptations.

Using the simplified prediction module, we evaluate the presented context network through controlled experiments on the Pascal VOC 2012 dataset [16]. The experiments demonstrate that plugging the context module into existing semantic segmentation architectures reliably increases their accuracy.

## 3.2 Dilated Convolutions

Let $F : \mathbb{Z}^2 \rightarrow \mathbb{R}$ be a discrete function. Let $\Omega_r = [-r, r]^2 \cap \mathbb{Z}^2$ and let $k : \Omega_r \rightarrow \mathbb{R}$ be a discrete filter of size $(2r+1)^2$. The discrete convolution operator $*$ can be defined as

$$(F * k)(\mathbf{p}) = \sum_{\mathbf{s}+\mathbf{t}=\mathbf{p}} F(\mathbf{s})\, k(\mathbf{t}). \tag{3.1}$$

We now generalize this operator. Let $l$ be a dilation factor and let $*_l$ be defined as

$$(F *_l k)(\mathbf{p}) = \sum_{\mathbf{s}+l\mathbf{t}=\mathbf{p}} F(\mathbf{s})\, k(\mathbf{t}). \tag{3.2}$$

We will refer to $*_l$ as a dilated convolution or an $l$-dilated convolution. The familiar discrete convolution $*$ is simply the 1-dilated convolution.

The dilated convolution operator has been referred to in the past as “convolution with a dilated filter”. It plays a key role in the *algorithme à trous*, an algorithm for wavelet decomposition [32, 70].[1] We use the term “dilated convolution” instead of “convolution with a dilated filter” to clarify that no “dilated filter” is constructed or represented. The convolution operator itself is modified to use the filter parameters

[1] Some recent work mistakenly referred to the dilated convolution operator itself as the *algorithme à trous*. This is incorrect. The *algorithme à trous* applies a filter at multiple scales to produce a signal decomposition. The algorithm uses dilated convolutions, but is not equivalent to the dilated convolution operator itself.

in a different way. The dilated convolution operator can apply the same filter at different ranges using different dilation factors. Our definition reflects the proper implementation of the dilated convolution operator, which does not involve construction of dilated filters.

In recent work on convolutional networks for semantic segmentation, [52] analyzed filter dilation but chose not to use it. [7] used dilation to simplify the architecture of [52]. In contrast, we develop a new convolutional network architecture that systematically uses dilated convolutions for multi-scale context aggregation.

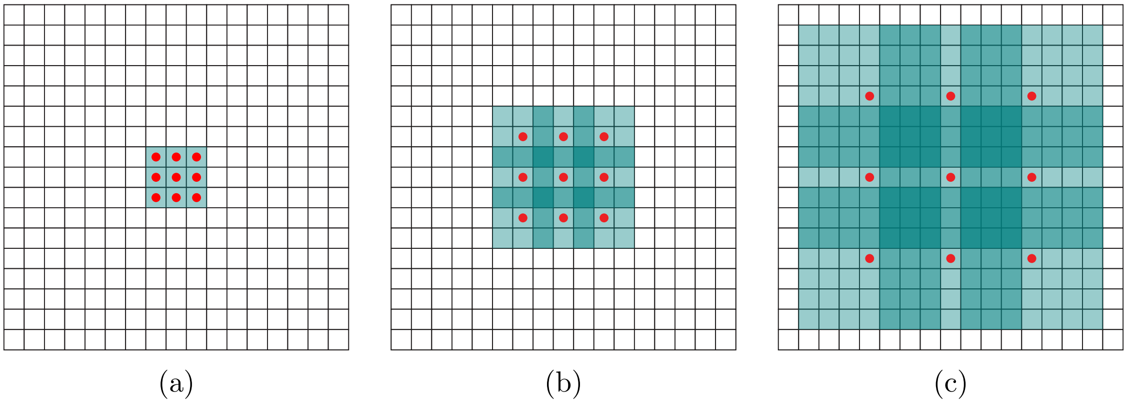
Our architecture is motivated by the fact that dilated convolutions support exponentially expanding receptive fields without losing resolution or coverage. Let $F_0, F_1, \ldots, F_{n-1} : \mathbb{Z}^2 \rightarrow \mathbb{R}$ be discrete functions and let $k_0, k_1, \ldots, k_{n-2} : \Omega_1 \rightarrow \mathbb{R}$ be discrete $3 \times 3$ filters. Consider applying the filters with exponentially increasing dilation:

$$F_{i+1} = F_i *_{2^i} k_i \quad \text{for} \quad i = 0, 1, \ldots, n-2. \tag{3.3}$$

Define the receptive field of an element $\mathbf{p}$ in $F_{i+1}$ as the set of elements in $F_0$ that modify the value of $F_{i+1}(\mathbf{p})$. Let the size of the receptive field of $\mathbf{p}$ in $F_{i+1}$ be the number of these elements. It is easy to see that the size of the receptive field of each element in $F_{i+1}$ is $(2^{i+2} - 1) \times (2^{i+2} - 1)$. The receptive field is a square of exponentially increasing size. This is illustrated in Figure 3.1.

## 3.3 Multi-Scale Context Aggregation

The context module is designed to increase the performance of dense prediction architectures by aggregating multi-scale contextual information. The module takes $C$ feature maps as input and produces $C$ feature maps as output. The input and output



(a) (b) (c)

Figure 3.1: Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage. (a) $F_1$ is produced from $F_0$ by a 1-dilated convolution; each element in $F_1$ has a receptive field of 3×3. (b) $F_2$ is produced from $F_1$ by a 2-dilated convolution; each element in $F_2$ has a receptive field of 7×7. (c) $F_3$ is produced from $F_2$ by a 4-dilated convolution; each element in $F_3$ has a receptive field of 15×15. The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly.

have the same form, thus the module can be plugged into existing dense prediction architectures.

We begin by describing a basic form of the context module. In this basic form, each layer has $C$ channels. The representation in each layer is the same and could be used to directly obtain a dense per-class prediction, although the feature maps are not normalized and no loss is defined inside the module. Intuitively, the module can increase the accuracy of the feature maps by passing them through multiple layers that expose contextual information.

The basic context module has 7 layers that apply 3×3 convolutions with different dilation factors. The dilations are 1, 1, 2, 4, 8, 16, and 1. Each convolution operates on all layers: strictly speaking, these are $3\times3\times C$ convolutions with dilation in the first two dimensions. Each of these convolutions is followed by a pointwise truncation $\max(\cdot, 0)$. A final layer performs $1\times1\times C$ convolutions and produces the output of the module. The architecture is summarized in Table 3.1. Note that the front-end module that provides the input to the context network in our experiments produces

| 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 | $2C$ | $2C$ | $4C$ | $8C$ | $16C$ | $32C$ | $32C$ | $C$ |

Table 3.1: Context network architecture. The network processes $C$ feature maps by aggregating contextual information at progressively increasing scales without losing resolution.

feature maps at $64 \times 64$ resolution. We therefore stop the exponential expansion of the receptive field after layer 6.

Our initial attempts to train the context module failed to yield an improvement in prediction accuracy. Experiments revealed that standard initialization procedures do not readily support the training of the module. Convolutional networks are commonly initialized using samples from random distributions [23, 42, 74]. However, we found that random initialization schemes were not effective for the context module. We found an alternative initialization with clear semantics to be much more effective:

$$k^b(\mathbf{t}, a) = 1_{[\mathbf{t}=0]} 1_{[a=b]}, \tag{3.4}$$

where $a$ is the index of the input feature map and $b$ is the index of the output map. This is a form of identity initialization, which has recently been advocated for recurrent networks [46]. This initialization sets all filters such that each layer simply passes the input directly to the next. A natural concern is that this initialization could put the network in a mode where backpropagation cannot significantly improve the default behavior of simply passing information through. However, experiments indicate that this is not the case. Backpropagation reliably harvests the contextual information provided by the network to increase the accuracy of the processed maps.

This completes the presentation of the basic context network. Our experiments show that even this basic module can increase dense prediction accuracy both quantitatively and qualitatively. This is particularly notable given the small number of parameters in the network: $\approx 64C^2$ parameters in total.

We have also trained a larger context network that uses a larger number of feature maps in the deeper layers. The number of maps in the large network is summarized in Table 3.1. We generalize the initialization scheme to account for the difference in the number of feature maps in different layers. Let $c_i$ and $c_{i+1}$ be the number of feature maps in two consecutive layers. Assume that $C$ divides both $c_i$ and $c_{i+1}$. The initialization is

$$k^b(\mathbf{t}, a) = \begin{cases} \dfrac{C}{c_{i+1}} & \mathbf{t} = 0 \text{ and } \left\lfloor \dfrac{aC}{c_i} \right\rfloor = \left\lfloor \dfrac{bC}{c_{i+1}} \right\rfloor \\ \varepsilon & \text{otherwise} \end{cases} \tag{3.5}$$

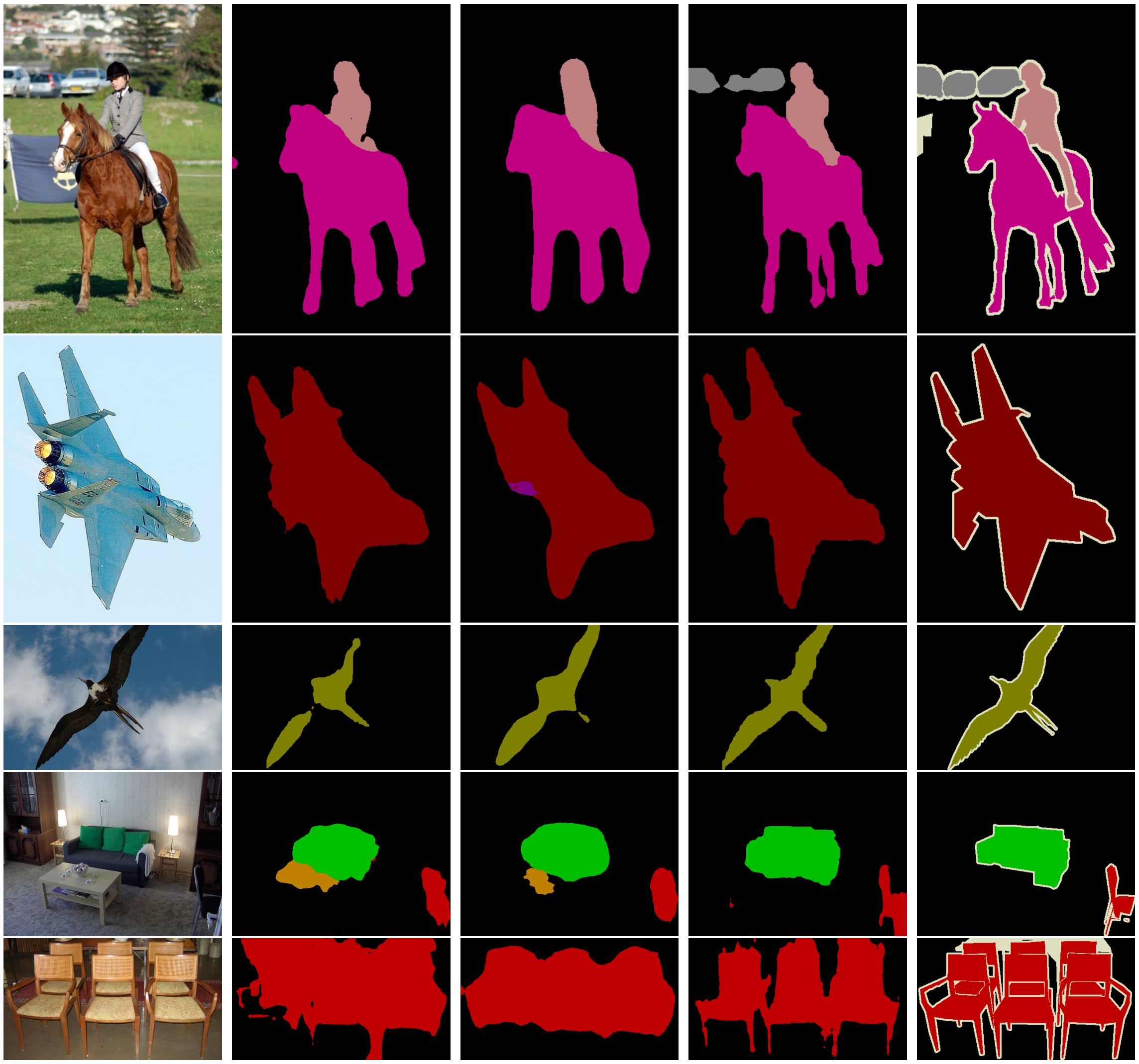
Here $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and $\sigma \ll C/c_{i+1}$. The use of random noise breaks ties among feature maps with a common predecessor.

## 3.4 Front End

We implemented and trained a front-end prediction module that takes a color image as input and produces $C = 21$ feature maps as output. The front-end module follows the work of [52] and [7], but was implemented separately. We adapted the VGG-16 network [74] for dense prediction and removed the last two pooling and striding layers. Specifically, each of these pooling and striding layers was removed and convolutions in all subsequent layers were dilated by a factor of 2 for each pooling layer that was ablated. Thus convolutions in the final layers, which follow both ablated pooling layers, are dilated by a factor of 4. This enables initialization with the parameters of the original classification network, but produces higher-resolution output. The front-

| | 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** |

Table 3.2: Our front-end prediction module is simpler and more accurate than prior models. This table reports accuracy on the VOC-2012 test set.



(a) Image (b) FCN-8s (c) DeepLab (d) Our front end (e) Ground truth

Figure 3.2: Semantic segmentations produced by different adaptations of the VGG-16 classification network. From left to right: (a) input image, (b) prediction by FCN-8s [52], (c) prediction by DeepLab [7], (d) prediction by our simplified front-end module, (e) ground truth.

end module takes padded images as input and produces feature maps at resolution $64 \times 64$. We use reflection padding: the buffer zone is filled by reflecting the image about each edge.

Our front-end module is obtained by removing vestiges of the classification network that are counter-productive for dense prediction. Most significantly, we remove the last two pooling and striding layers entirely, whereas Long et al. kept them and

Chen et al. replaced striding by dilation but kept the pooling layers. We found that simplifying the network by removing the pooling layers made it more accurate. We also remove the padding of the intermediate feature maps. Intermediate padding was used in the original classification network, but is neither necessary nor justified in dense prediction.

This simplified prediction module was trained on the Pascal VOC 2012 training set, augmented by the annotations created by [25]. We did not use images from the VOC-2012 validation set for training and therefore only used a subset of the annotations of [25]. Training was performed by stochastic gradient descent (SGD) with mini-batch size 14, learning rate $10^{-3}$, and momentum 0.9. The network was trained for 60K iterations.

We now compare the accuracy of our front-end module to the FCN-8s design of [52] and the DeepLab network of [7]. For FCN-8s and DeepLab, we evaluate the public models trained by the original authors on VOC-2012. Segmentations produced by the different models on images from the VOC-2012 dataset are shown in Figure 3.2. The accuracy of the models on the VOC-2012 test set is reported in Table 3.2.

Our front-end prediction module is both simpler and more accurate than the prior models. Specifically, our simplified model outperforms both FCN-8s and the DeepLab network by more than 5 percentage points on the test set. Interestingly, our simplified front-end module outperforms the leaderboard accuracy of DeepLab+CRF on the test set by more than a percentage point (67.6% vs. 66.4%) without using a CRF.

## 3.5 Experiments

Our implementation is based on the Caffe library [35]. Our implementation of dilated convolutions is now part of the stanfard Caffe distribution.

For fair comparison with recent high-performing systems, we trained a front-end module that has the same structure as described in Section 3.4, but is trained on additional images from the Microsoft COCO dataset [50]. We used all images in Microsoft COCO with at least one object from the VOC-2012 categories. Annotated objects from other categories were treated as background.

Training was performed in two stages. In the first stage, we trained on VOC-2012 images and Microsoft COCO images together. Training was performed by SGD with mini-batch size 14 and momentum 0.9. 100K iterations were performed with a learning rate of $10^{-3}$ and 40K subsequent iterations were performed with a learning rate of $10^{-4}$. In the second stage, we fine-tuned the network on VOC-2012 images only. Fine-tuning was performed for 50K iterations with a learning rate of $10^{-5}$. Images from the VOC-2012 validation set were not used for training.

The front-end module trained by this procedure achieves 69.8% mean IoU on the VOC-2012 validation set and 71.3% mean IoU on the test set. Note that this level of accuracy is achieved by the front-end alone, without the context module or structured prediction. We again attribute this high accuracy in part to the removal of vestigial components originally developed for image classification rather than dense prediction.

**Controlled evaluation of context aggregation.** We now perform controlled experiments to evaluate the utility of the context network presented in Section 3.3. We begin by plugging each of the two context modules (Basic and Large) into the front end. Since the receptive field of the context network is $67 \times 67$, we pad the input feature maps by a buffer of width 33. Zero padding and reflection padding yielded similar results in our experiments. The context module accepts feature maps from the front end as input and is given this input during training. Joint training of the context module and the front-end module did not yield a significant improvement in our experiments. The learning rate was set to $10^{-3}$. Training was initialized as described in Section 3.3.

Table 3.3 shows the effect of adding the context module to three different architectures for semantic segmentation. The first architecture (top) is the front end described in Section 3.4. It performs semantic segmentation without structured prediction, akin to the original work of [52]. The second architecture (Table 3.3, middle) uses the dense CRF to perform structured prediction, akin to the system of [7]. We use the implementation of [41] and train the CRF parameters by grid search on the validation set. The third architecture (Table 3.3, bottom) uses the CRF-RNN for structured prediction [93]. We use the implementation of [93] and train the CRF-RNN in each condition.

The experimental results demonstrate that the context module improves accuracy in each of the three configurations. The basic context module increases accuracy in each configuration. The large context module increases accuracy by a larger margin. The experiments indicate that the context module and structured prediction are synergisic: the context module increases accuracy with or without subsequent structured prediction. Qualitative results are shown in Figure 3.3.

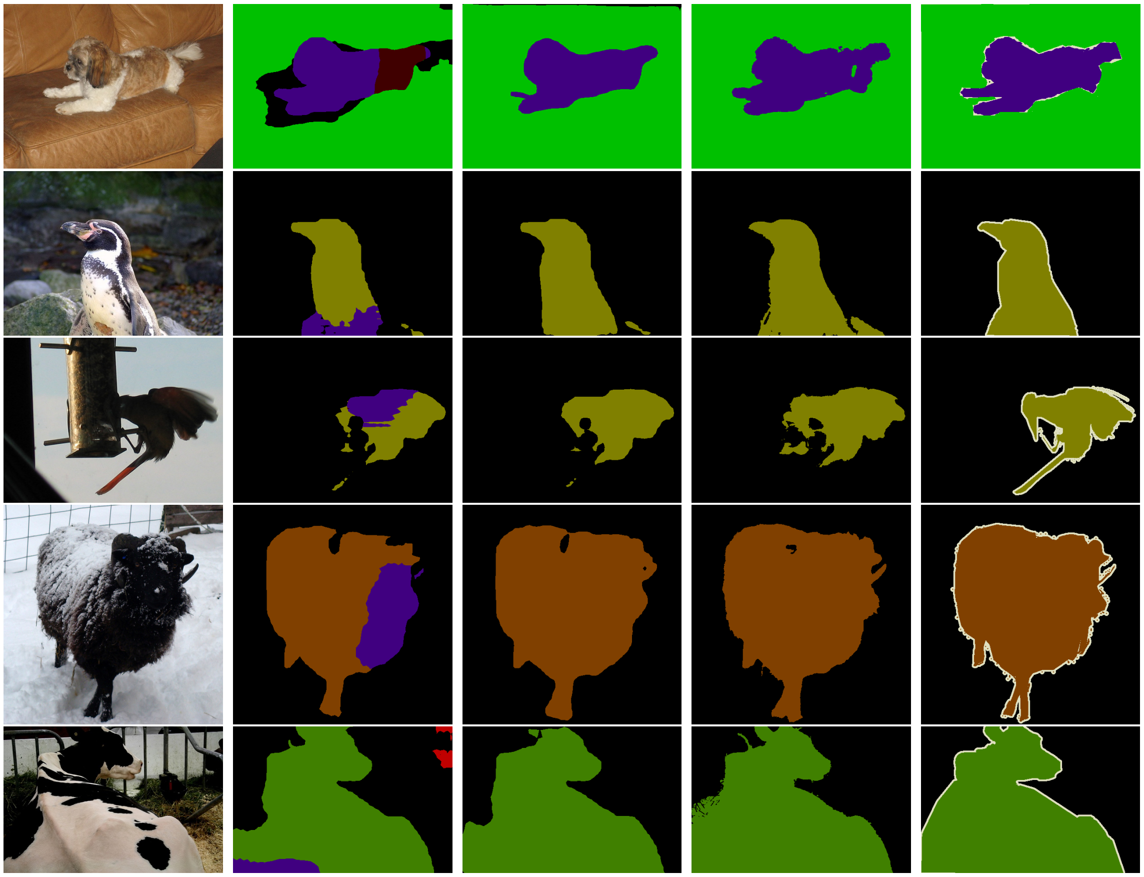
**Evaluation on the test set.** We now perform an evaluation on the test set by submitting our results to the Pascal VOC 2012 evaluation server. The results are reported in Table 3.4. We use the large context module for these experiments. As the results demonstrate, the context module yields a significant boost in accuracy over the front end. The context module alone, without subsequent structured prediction, outperforms DeepLab-CRF-COCO-LargeFOV [7]. The context module with the dense CRF, using the original implementation of [41], performs on par with the very recent CRF-RNN [93]. The context module in combination with the CRF-RNN further increases accuracy over the performance of the CRF-RNN.

|  | 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** |

Table 3.3: Controlled evaluation of the effect of the context module on the accuracy of three different architectures for semantic segmentation. Experiments performed on the VOC-2012 validation set. Validation images were not used for training. Top: adding the context module to a semantic segmentation front end with no structured prediction [52]. The basic context module increases accuracy, the large module increases it by a larger margin. Middle: the context module increases accuracy when plugged into a front-end + dense CRF configuration [7]. Bottom: the context module increases accuracy when plugged into a front-end + CRF-RNN configuration [93].

| | 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** |

Table 3.4: Evaluation on the VOC-2012 test set. ‘DeepLab++’ stands for DeepLab-CRF-COCO-LargeFOV and ‘DeepLab-MSc++’ stands for DeepLab-MSc-CRF-LargeFOV-COCO-CrossJoint [7]. ‘CRF-RNN’ is the system of [93]. ‘Context’ refers to the large context module plugged into our front end. The context network yields very high accuracy, ourperforming the DeepLab++ architecture without performing structured prediction. Combining the context network with the CRF-RNN structured prediction module increases the accuracy of the CRF-RNN system.



(a) Image (b) Front end (c) +Context (d) +CRF-RNN (e) Ground truth

Figure 3.3: Semantic segmentations produced by different models. From left to right: (a) input image, (b) prediction by the front-end module, (c) prediction by the large context network plugged into the front end, (d) prediction by the front end + context module + CRF-RNN, (e) ground truth.

## 3.6 Urban Scene Understanding

In this section, we report experiments on three datasets for urban scene understanding: the CamVid dataset [5], the KITTI dataset [20], and the new Cityscapes dataset [14]. As the accuracy measure we use the mean IoU [16]. We only train our model on the training set, even when a validation set is available. The results reported in this section do not use conditional random fields or other forms of structured prediction. They were obtained with convolutional networks that combine a front-end module and a context module, akin to the "Front + Basic" network evaluated in Table 3.3. The trained models can be found at `https://github.com/fyu/dilation`.

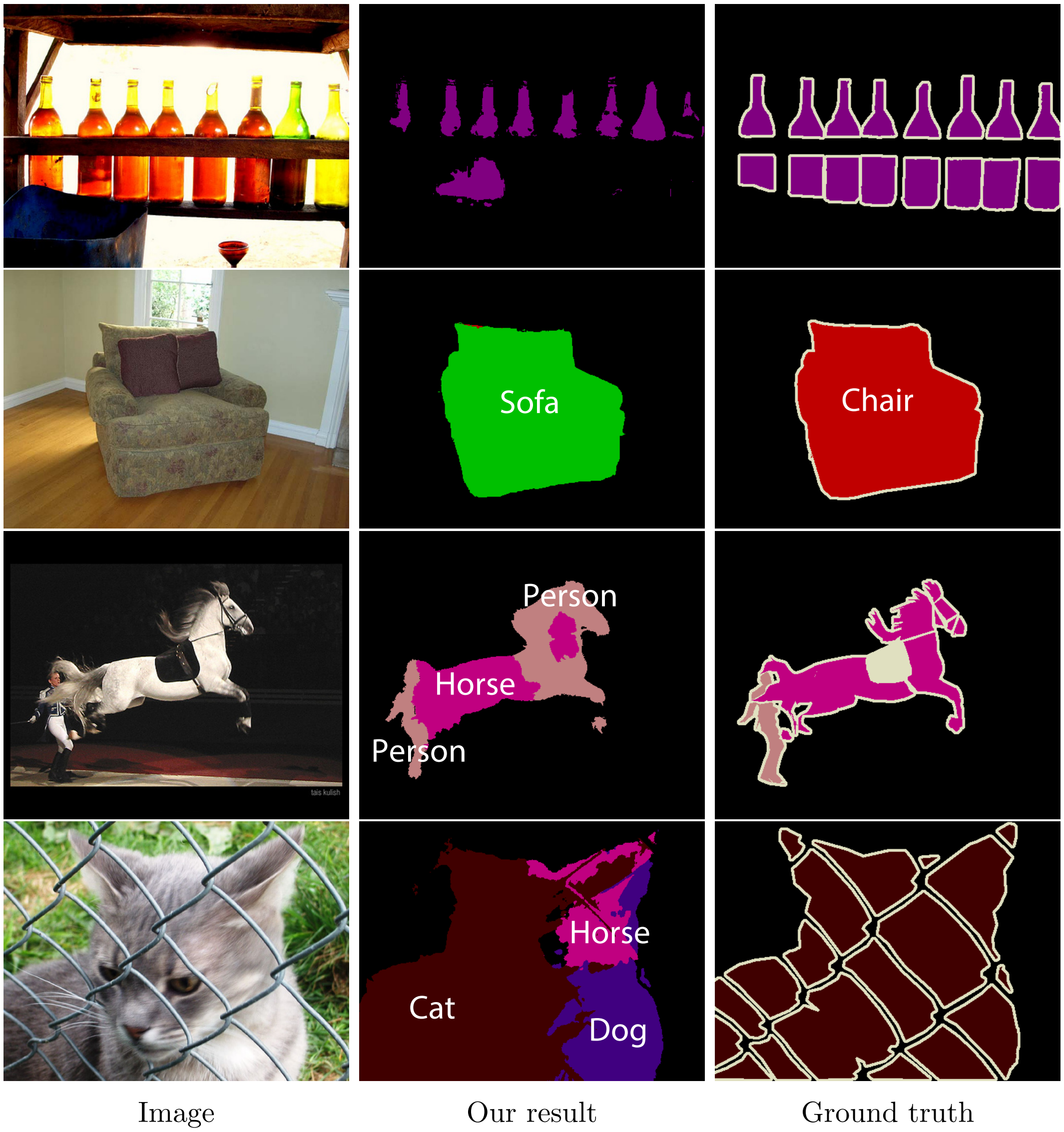


Figure 3.4: Failure cases from the VOC-2012 validation set. The most accurate architecture we trained (Context + CRF-RNN) performs poorly on these images.

We now summarize the training procedure used for training the front-end module. This procedure applies to all datasets. Training is performed with stochastic gradient descent. Each mini-batch contains 8 crops from randomly sampled images. Each crop is of size 628×628 and is randomly sampled from a padded image. Images are padded using reflection padding. No padding is used in the intermediate layers. The learning rate is $10^{-4}$ and momentum is set to 0.99. The number of iterations depends on the number of images in the dataset and is reported for each dataset below.

The context modules used for these datasets are all derived from the "Basic" network, using the terminology of Table 3.1. The number of channels in each layer is the number of predicted classes $C$. (For example, $C = 19$ for the Cityscapes dataset.) Each layer in the context module is padded such that the input and response maps have the same size. The number of layers in the context module depends on the resolution of the images in the dataset. Joint training of the complete model, composed of the front-end and the context module, is summarized below for each dataset.

### 3.6.1 CamVid

We use the split of [76], which partitions the dataset into 367 training images, 100 validation images, and 233 test images. 11 semantic classes are used. The images are downsampled to $640\times480$.

The context module has 8 layers, akin to the model used for the Pascal VOC dataset. The overall training procedure is as follows. First, the front-end module is trained for 20K iterations. Then the complete model (front-end + context) is jointly trained by sampling crops of size $852\times852$ with batch size 1. The learning rate for joint training is set to $10^{-5}$ and the momentum is set to 0.9.

Results on the CamVid test set are reported in Table 3.5. We refer to our complete convolutional network (front-end + context) as Dilation8, since the context module has 8 layers. Our model outperforms the prior work. This model was used as the unary classifier in the recent work of [43].

| | 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** |

Table 3.5: Semantic segmentation results on the CamVid dataset. Our model (Dilation8) is compared to ALE [44], SuperParsing [79], Liu and He [51], SegNet [4], and the DeepLab-LargeFOV model [7]. Our model outperforms the prior work.

### 3.6.2 KITTI

We use the training and validation split of [64]: 100 training images and 46 test images. The images were all collected from the KITTI visual odometry/SLAM dataset. The image resolution is $1226 \times 370$. Since the vertical resolution is small compared to the other datasets, we remove Layer 6 in Table 3.1. The resulting context module has 7 layers. The complete network (front-end + context) is referred to as Dilation7.

The front-end is trained for 10K iterations. Next, the front-end and the context module are trained jointly. For joint training, the crop size is 900×900 and momentum is set to 0.99, while the other parameters are the same as the ones used for the CamVid dataset. Joint training is performed for 20K iterations.

The results are shown in Table 3.6. As the table demonstrates, our model outperforms the prior work.

|  | 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** |

Table 3.6: Semantic segmentation results on the KITTI dataset. We compare our results to [64] and to the DeepLab-LargeFOV model [7]. Our network (Dilation7) yields higher accuracy than the prior work.

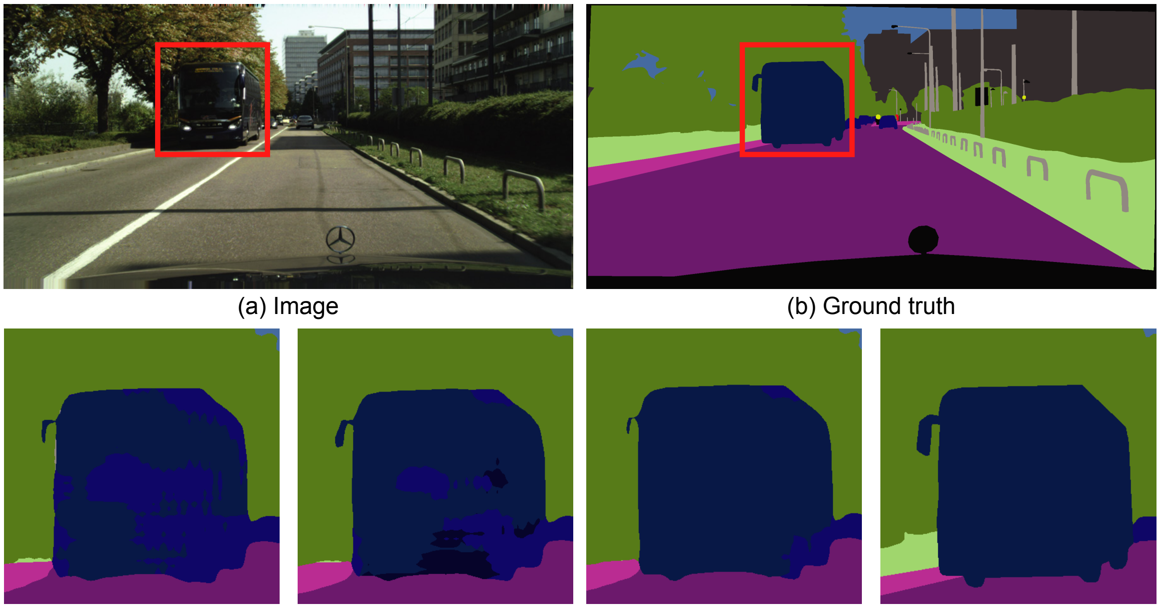
### 3.6.3 Cityscapes

The Cityscapes dataset contains 2975 training images, 500 validation images, and 1525 test images [14]. Due to the high image resolution (2048 × 1024), we add two layers to the context network after Layer 6 in Table 3.1. These two layers have dilation 32 and 64, respectively. The total number of layers in the context module is 10 and we refer to the complete model (front-end + context) as Dilation10.

The Dilation10 network was trained in three stages. First, the front-end prediction module was trained for 40K iterations. Second, the context module was trained for 24K iterations on whole (uncropped) images, with learning rate $10^{-4}$, momentum 0.99, and batch size 100. Third, the complete model (front-end + context) was jointly trained for 60K iterations on halves of images (input size 1396×1396, including padding), with learning rate $10^{-5}$, momentum 0.99, and batch size 1.

Figure 3.5 visualizes the effect of the training stages on the performance of the model. Quantitative results are given in Tables 3.7 and 3.8.

The performance of Dilation10 was compared to prior work on the Cityscapes dataset by [14]. In their evaluation, Dilation10 outperformed all prior models [14]. Dilation10 was also used as the unary classifier in the recent work of [43], which used structured prediction to increase accuracy further.


(a) Image

(b) Ground truth

(c) Front end

(d) +Context

(e) +Joint

(f) Ground truth

Figure 3.5: Results produced by the Dilation10 model after different training stages. (a) Input image. (b) Ground truth segmentation. (c) Segmentation produced by the model after the first stage of training (front-end only). (d) Segmentation produced after the second stage, which trains the context module. (e) Segmentation produced after the third stage, in which both modules are trained jointly.

| 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 |

Table 3.7: Per-class and mean class-level IoU achieved by our model (Dilation10) on the Cityscapes dataset.

## 3.7 Conclusion

We have examined convolutional network architectures for dense prediction. Since the model must produce high-resolution output, we believe that high-resolution operation throughout the network is both feasible and desirable. Our work shows that the dilated convolution operator is particularly suited to dense prediction due to its

| 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 |

Table 3.8: Per-category and mean category-level IoU on the Cityscapes dataset.

ability to expand the receptive field without losing resolution or coverage. We have utilized dilated convolutions to design a new network structure that reliably increases accuracy when plugged into existing semantic segmentation systems. As part of this work, we have also shown that the accuracy of existing convolutional networks for semantic segmentation can be increased by removing vestigial components that had been developed for image classification. State-of-the-art systems for semantic segmentation leave significant room for future advances. Failure cases of our most accurate configuration are shown in Figure 3.4.

We believe that the presented work is a step towards dedicated architectures for dense prediction that are not constrained by image classification precursors. Since the paper [89] was initially published, the idea of dilated convolutions have been widely accepted and adopted in state-of-the-art semantic image segmentation networks [65, 92].

In this chapter, we study dilated convolutions based on VGG network. New generations of deep convolutional network architecture may keep improving the performance of dilated networks. However, the newly proposed network elements may not work well directly with dilation. In the next chapter, we will study the relation between dilation and residual connections and investigate the connection between image classification and segmentation.

# Chapter 4

# Dilated Residual Networks

## 4.1 Introduction

Convolutional networks were originally developed for classifying hand-written digits [47]. More recently, convolutional network architectures have evolved to classify much more complex images [42, 74, 78, 29]. Yet a central aspect of network architecture has remained largely in place. Convolutional networks for image classification progressively reduce resolution until the image is represented by tiny feature maps that retain little spatial information ($7 \times 7$ is typical).

While convolutional networks have done well, the almost complete elimination of spatial acuity may be preventing these models from achieving even higher accuracy, for example by preserving the contribution of small and thin objects that may be important for correctly understanding the image. Such preservation may not have been important in the context of hand-written digit classification, in which a single object dominated the image, but may help in the analysis of complex natural scenes where multiple objects and their relative configurations must be taken into account.

Furthermore, image classification is rarely a convolutional network's raison d'être. Image classification is most often a proxy task that is used to pretrain a model before it is transferred to other applications that involve more detailed scene understanding [22, 52]. In such tasks, severe loss of spatial acuity is a significant handicap. Existing techniques compensate for the lost resolution by introducing up-convolutions [52, 55], skip connections [26], and other post-hoc measures.

Must convolutional networks crush the image in order to classify it? In this chapter, we show that this is not necessary, or even desirable. Starting with the residual network architecture, the current state of the art for image classification [29], we increase the resolution of the network's output by replacing a subset of interior subsampling layers by dilation [89]. We show that dilated residual networks (DRNs) yield improved image classification performance. Specifically, DRNs yield higher accuracy in ImageNet classification than their non-dilated counterparts, with no increase in depth or model complexity.

The output resolution of a DRN on typical ImageNet input is 28×28, comparable to small thumbnails that convey the structure of the image when examined by a human [82]. While it may not be clear a priori that average pooling can properly handle such high-resolution output, we show that it can, yielding a notable accuracy gain. We then study gridding artifacts introduced by dilation, propose a scheme for removing these artifacts, and show that such 'degridding' further improves the accuracy of DRNs.

We also show that DRNs yield improved accuracy on downstream applications such as weakly-supervised object localization and semantic segmentation. With a remarkably simple approach, involving no fine-tuning at all, we obtain state-of-the-art top-1 accuracy in weakly-supervised localization on ImageNet. We also study the performance of DRNs on semantic segmentation and show, for example, that a 42-layer DRN outperforms a ResNet-101 baseline on the Cityscapes dataset by more than 4 percentage points, despite lower depth by a factor of 2.4.

## 4.2 Related Work

**Image classification** has received a lot of attention because techniques developed for image classification can often be used to also improve performance on more detailed scene understanding tasks. In the past, researchers have manually designed feature descriptors for image classification [61, 34]. Such descriptors could be spatially organized to represent the structure of the image [24, 45].

The re-emergence of convolutional networks as the primary feature extractor in computer vision has streamlined and unified this area [47, 42, 78, 74, 29]. While the design of convolutional networks has evolved significantly, a core structural principle has remained largely in place: the image is progressively downsampled until almost no spatial resolution is left. Typically, the images are downsampled by a factor of 32 between the input layer and the output of the final convolutional layer. In this chapter, we revisit this design choice and propose to retain higher spatial resolution all the way through to the output of the final convolutional layer, such that considerably higher-resolution activation maps are pooled for the final prediction.

To aggregate the signal from a high-resolution convolutional layer, we use global average pooling [49, 95, 29]. We show that backpropagation can effectively handle global average pooling over much bigger feature maps than previously thought, yielding high accuracy in the final predictor as well as informative high-resolution activations.

**Weakly-supervised localization** concerns localization of objects in images given only image-level labels during training. Since image-level labels are much easier to obtain than finer-grained annotations [68, 96], weakly-supervised localization leverages image-level supervision for object-level image understanding. This problem has received significant attention [59, 60, 62, 94, 95, 12]. Many approaches [59, 62, 12, 60] aggregate information from trained feature maps to identify potential object locations. Zhou et al. [94] analyze the responses of feature maps and localize image regions that activate these responses. A later work by Zhou et al. [95] proposes to remove layers from trained image classification models in order to get to higher-resolution feature maps, then fine-tuning such ablated models. This approach requires additional training and, as the authors show, weakens classification performance. In contrast, we develop an image classification model that produces informative high-resolution activation maps directly, with no sacrifice in classification accuracy and no need for post-hoc fine-tuning.

## 4.3 Dilated Residual Networks

Our key idea is to preserve spatial resolution in convolutional networks for image classification. Although progressive downsampling has been very successful in classifying digits or iconic views of objects, the loss of spatial information may be harmful for classifying natural images and can significantly hamper transfer to other tasks that involve spatially detailed image understanding. Natural images often feature many objects whose identities and relative configurations are important for understanding the scene. The classification task becomes difficult when a key object is not spatially dominant – for example, when the labeled object is thin (e.g., a tripod) or when there is a big background object such as a mountain. In these cases, the background response may suppress the signal from the object of interest. What's worse, if the object's signal is lost due to downsampling, there is little hope to recover it during training. However, if we retain high spatial resolution throughout the model and provide output signals that densely cover the input field, backpropagation can learn to preserve important information about smaller and less salient objects.

The starting point of our construction is the set of network architectures presented by He et al. [29]. Each of these architectures consists of five groups of convolutional layers. The first layer in each group performs downsampling by striding: that is, the convolutional filter is only evaluated at even rows and columns. Let each group of layers be denoted by $\mathcal{G}^\ell$, for $\ell = 1, \ldots, 5$. Denote the $i^{\text{th}}$ layer in group $\ell$ by $\mathcal{G}_i^\ell$. For simplicity of exposition, consider an idealized model in which each layer consists of a single feature map: the extension to multiple feature maps is straightforward. Let $f_i^\ell$ be the filter associated with layer $\mathcal{G}_i^\ell$. In the original model, the output of $\mathcal{G}_i^\ell$ is

$$(\mathcal{G}_i^\ell * f_i^\ell)(\mathbf{p}) = \sum_{\mathbf{a}+\mathbf{b}=\mathbf{p}} \mathcal{G}_i^\ell(\mathbf{a})\, f_i^\ell(\mathbf{b}), \tag{4.1}$$

where the domain of $\mathbf{p}$ is the feature map in $\mathcal{G}_i^\ell$. This is followed by a nonlinearity, which does not affect the presented construction.

A naive approach to increasing resolution in higher layers of the network would be to simply remove subsampling (striding) from some of the interior layers. This does increase downstream resolution, but has a detrimental side effect that negates the benefits: removing subsampling correspondingly reduces the receptive field in subsequent layers. Thus removing striding such that the resolution of the output layer is increased by a factor of 4 also reduces the receptive field of each output unit by a factor of 4. This severely reduces the amount of context that can inform the prediction produced by each unit. Since contextual information is important in disambiguating local cues [19], such reduction in receptive field is an unacceptable price to pay for higher resolution. For this reason, we use dilated convolutions [89] to increase the receptive field of the higher layers, compensating for the reduction in receptive field induced by removing subsampling. The effect is that units in the dilated layers have the same receptive field as corresponding units in the original model.

We focus on the two final groups of convolutional layers: $\mathcal{G}^4$ and $\mathcal{G}^5$. In the original ResNet, the first layer in each group ($\mathcal{G}_1^4$ and $\mathcal{G}_1^5$) is strided: the convolution is evaluated at even rows and columns, which reduces the output resolution of these layers by a factor of 2 in each dimension. The first step in the conversion to DRN is to remove the striding in both $\mathcal{G}_1^4$ and $\mathcal{G}_1^5$. Note that the receptive field of each unit in $\mathcal{G}_1^4$ remains unaffected: we just doubled the output resolution of $\mathcal{G}_1^4$ without affecting the receptive field of its units. However, subsequent layers are all affected: their receptive fields have been reduced by a factor of 2 in each dimension. We therefore replace the convolution operators in those layers by 2-dilated convolutions [89]:

$$(\mathcal{G}_i^4 *_2 f_i^4)(\mathbf{p}) = \sum_{\mathbf{a}+2\mathbf{b}=\mathbf{p}} \mathcal{G}_i^4(\mathbf{a})\, f_i^4(\mathbf{b}) \tag{4.2}$$

for all $i \geq 2$. The same transformation is applied to $\mathcal{G}_1^5$:

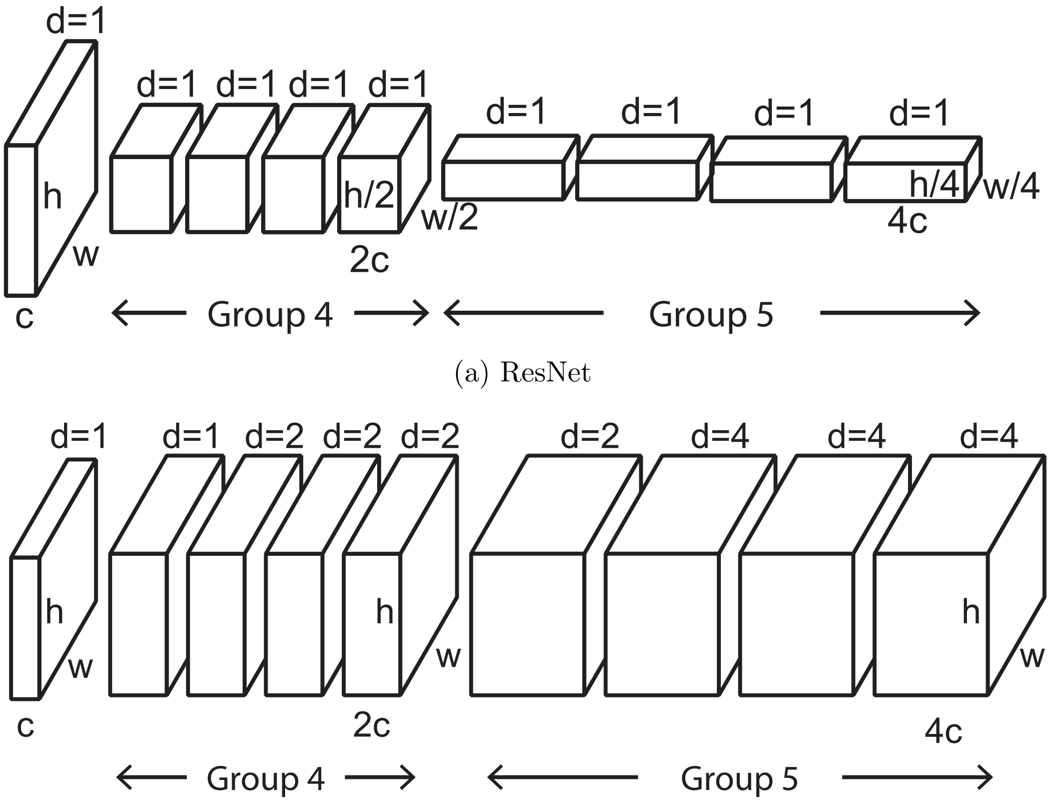
$$(\mathcal{G}_1^5 *_2 f_1^5)(\mathbf{p}) = \sum_{\mathbf{a}+2\mathbf{b}=\mathbf{p}} \mathcal{G}_1^5(\mathbf{a})\, f_1^5(\mathbf{b}). \tag{4.3}$$

Subsequent layers in $\mathcal{G}^5$ follow two striding layers that have been eliminated. The elimination of striding has reduced their receptive fields by a factor of 4 in each dimension. Their convolutions need to be dilated by a factor of 4 to compensate for the loss:

$$(\mathcal{G}_i^5 *_4 f_i^5)(\mathbf{p}) = \sum_{\mathbf{a}+4\mathbf{b}=\mathbf{p}} \mathcal{G}_i^5(\mathbf{a})\, f_i^5(\mathbf{b}) \tag{4.4}$$

for all $i \geq 2$. Finally, as in the original architecture, $\mathcal{G}^5$ is followed by global average pooling, which reduces the output feature maps to a vector, and a $1\times1$ convolution that maps this vector to a vector that comprises the prediction scores for all classes. The transformation of a ResNet into a DRN is illustrated in Figure 4.1.

The converted DRN has the same number of layers and parameters as the original ResNet. The key difference is that the original ResNet downsamples the input image by a factor of 32 in each dimension (a thousand-fold reduction in area), while the DRN downsamples the input by a factor of 8. For example, when the input resolution is $224\times224$, the output resolution of $\mathcal{G}^5$ in the original ResNet is $7\times7$, which is not



(a) ResNet

(b) DRN

Figure 4.1: Converting a ResNet into a DRN. The original ResNet is shown in (a), the resulting DRN is shown in (b). Striding in $\mathcal{G}_1^4$ and $\mathcal{G}_1^5$ is removed, bringing the resolution of all layers in $\mathcal{G}^4$ and $\mathcal{G}^5$ to the resolution of $\mathcal{G}^3$. To compensate for the consequent shrinkage of the receptive field, $\mathcal{G}_i^4$ and $\mathcal{G}_1^5$ are dilated by a factor of 2 and $\mathcal{G}_i^5$ are dilated by a factor of 4, for all $i \geq 2$. $c$, $2c$, and $4c$ denote the number of feature maps in a layer, $w$ and $h$ denote feature map resolution, and $d$ is the dilation factor.

sufficient for the spatial structure of the input to be discernable. The output of $\mathcal{G}^5$ in a DRN is $28 \times 28$. Global average pooling therefore takes in $2^4$ times more values, which can help the classifier recognize objects that cover a smaller number of pixels in the input image and take such objects into account in its prediction.

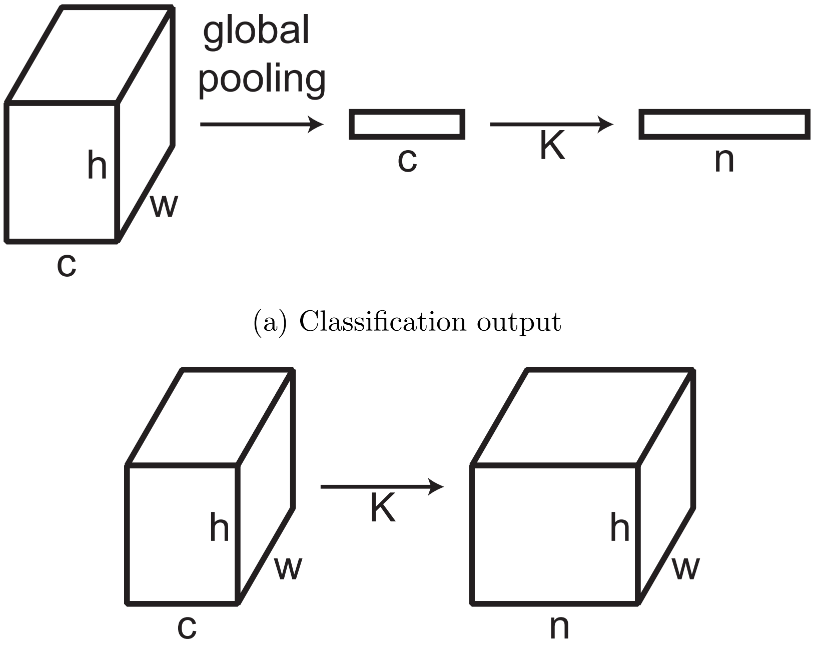
The presented construction could also be applied to earlier groups of layers ($\mathcal{G}^1$, $\mathcal{G}^2$, or $\mathcal{G}^3$), in the limit retaining the full resolution of the input. We chose not to do this because a downsampling factor of 8 is known to preserve most of the information necessary to correctly parse the original image at pixel level [52]. Furthermore, a 28×28 thumbnail, while small, is sufficiently resolved for humans to discern the structure of the scene [82]. Additional increase in resolution has costs and should not be pursued without commensurate gains: when feature map resolution is increased by a factor of 2 in each dimension, the memory consumption of that feature map increases by a factor of 4. Operating at full resolution throughout, with no downsampling at all, is beyond the capabilities of current hardware.

## 4.4 Localization

Given a DRN trained for image classification, we can directly produce dense pixel-level class activation maps without any additional training or parameter tuning. This allows a DRN trained for image classification to be immediately used for object localization and segmentation.

To obtain high-resolution class activation maps, we remove the global average pooling operator. We then connect the final $1 \times 1$ convolution directly to $\mathcal{G}^5$. A softmax is applied to each column in the resulting volume to convert the pixelwise prediction scores to proper probability distributions. This procedure is illustrated in Figure 4.2. The output of the resulting network is a set of activation maps that have the same spatial resolution as $\mathcal{G}^5$ ($28 \times 28$). Each classification category $y$ has a corresponding activation map. For each pixel in this map, the map contains the probability that the object observed at this pixel is of category $y$.

The activation maps produced by our construction serve the same purpose as the results of the procedure of Zhou et al. [95]. However, the procedures are fundamentally different. Zhou et al. worked with convolutional networks that produce drastically downsampled output that is not sufficiently resolved for object localization. For this reason, Zhou et al. had to remove layers from the classification network, introduce parameters that compensate for the ablated layers, and then fine-tune the modified



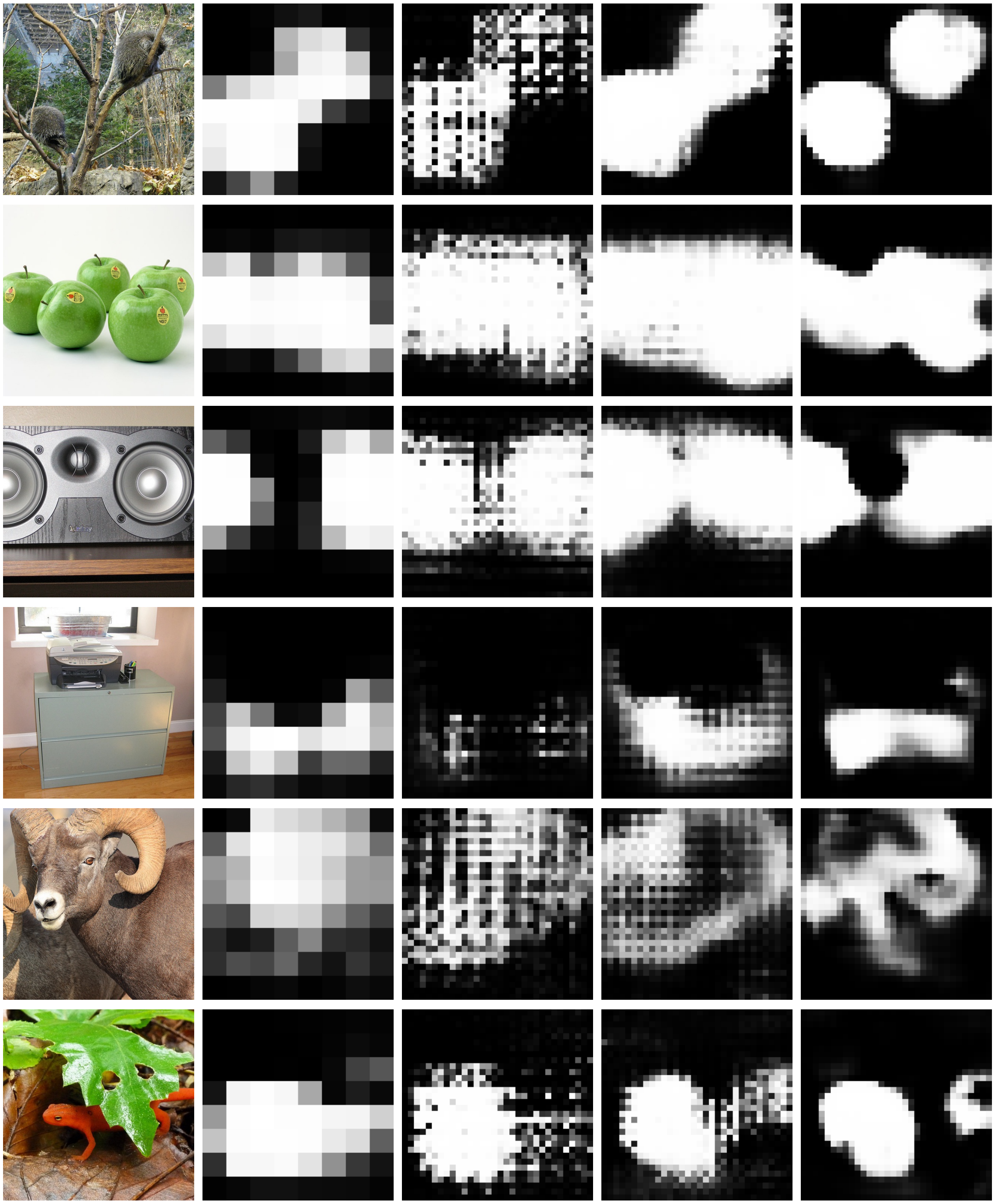
(a) Classification output

(b) Localization output

Figure 4.2: Using a classification network for localization. The output stages of a DRN trained for image classification are shown in (a). Here $K$ is a $1 \times 1$ convolution that maps $c$ channels to $n$. To reconfigure the network for localization, we remove the pooling operator. The result is shown in (b). The reconfigured network produces $n$ activation maps of resolution $w \times h$. No training or parameter tuning is involved.

models to train the new parameters. Even then, the output resolution obtained by Zhou et al. was quite small (14×14) and the classification performance of the modified networks was impaired.

In contrast, the DRN was designed to produce high-resolution output maps and is trained in this configuration from the start. Thus the model trained for image classification already produces high-resolution activation maps. As our experiments will show, DRNs are more accurate than the original ResNets in image classification. Since DRNs produce high-resolution output maps from the start, there is no need to remove layers, add parameters, and retrain the model for localization. The original accurate classification model can be used for localization directly.

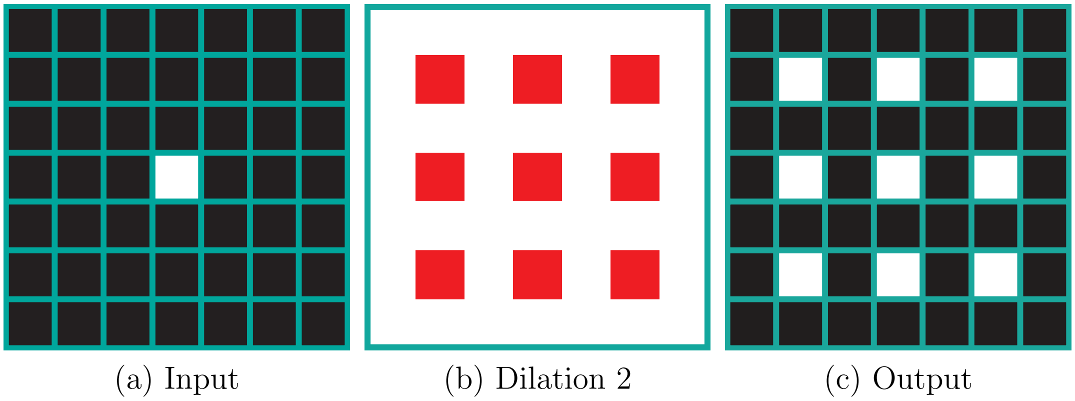


(a) Input (b) ResNet-18 (c) DRN-A-18 (d) DRN-B-26 (e) DRN-C-26

Figure 4.3: Activation maps of ResNet-18 and corresponding DRNs. A DRN constructed from ResNet-18 as described in Section 4.3 is referred to as DRN-A-18. The corresponding DRN produced by the degridding scheme described in Section 4.5 is referred to as DRN-C-26. The DRN-B-26 is an intermediate construction.

## 4.5 Degridding

The use of dilated convolutions can cause gridding artifacts. Such artifacts are shown in Figure 4.3(c) and have also been observed in concurrent work on semantic segmentation [86]. Gridding artifacts occur when a feature map has higher-frequency content than the sampling rate of the dilated convolution. Figure 4.4 shows a didactic example. In Figure 4.4(a), the input feature map has a single active pixel. A 2-dilated convolution (Figure 4.4(b)) induces a corresponding grid pattern in the output (Figure 4.4(c)).

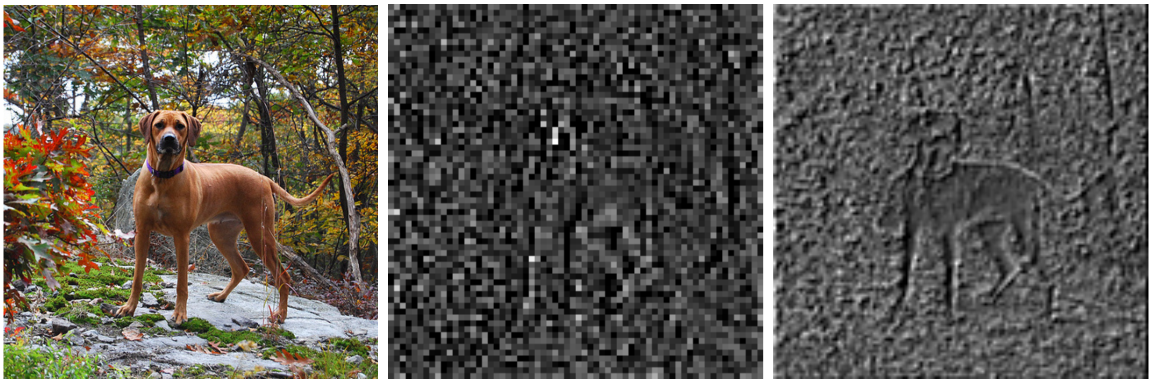


(a) Input (b) Dilation 2 (c) Output

Figure 4.4: A gridding artifact.

In this section, we develop a scheme for removing gridding artifacts from output activation maps produced by DRNs. The scheme is illustrated in Figure 4.6. A DRN constructed as described in Section 4.3 is referred to as DRN-A and is illustrated in Figure 4.6(a). An intermediate stage of the construction described in the present section is referred to as DRN-B and is illustrated in Figure 4.6(b). The final construction is referred to as DRN-C, illustrated in Figure 4.6(c).

**Removing max pooling.** As shown in Figure 4.6(a), DRN-A inherits from the ResNet architecture a max pooling operation after the initial 7×7 convolution. They are labeled as Level 1 and 2 in the row for DRN-18-A in Figure 4.6. We find that this max pooling operation leads to high-amplitude high-frequency activations, as shown in Figure 4.5(b). Such high-frequency activations can be propagated to later layers and ultimately exacerbate gridding artifacts. We thus replace max pooling by convolutional filters, as shown in Figure 4.6(b). The effect of this transformation is shown in Figure 4.5(c). As shown in Figure 4.5(c), the modified network learns to preserve important contour cues after downsampling. Visually, this looks helpful because we can still recognize the objects based on the contours, which can provide more information for the later layers. Even though the max pooling is removed, the network is still invariant to 2D translation thanks to the global average pooling.



(a) Input (b) DRN-A-18 (c) DRN-B-26

Figure 4.5: First stage of degridding, which modifies the early layers of the network. (b) and (c) show input feature maps for the first convolutional layer in level 3 of DRN-A-18 and DRN-B-26. The feature map with the highest average activation is shown.

**Adding layers.** To remove gridding artifacts, we add convolutional layers at the end of the network, with progressively lower dilation. Specifically, after the last 4-dilated layer in DRN-A (Figure 4.6(a)), we add a 2-dilated residual block followed by a 1-dilated block. These become levels 7 and 8 in DRN-B, shown in Figure 4.6(b). This is akin to removing aliasing artifacts using filters with appropriate frequency [84].

**Removing residual connections.** Adding layers with decreasing dilation, as described in the preceding paragraph, does not remove gridding artifacts entirely because of residual connections. The residual connections in levels 7 and 8 of DRN-B can propagate gridding artifacts from level 6. To remove gridding artifacts more effectively, we remove the residual connections in levels 7 and 8. This yields the DRN-C, our proposed construction, illustrated in Figure 4.6(c). Note that the DRN-C has higher depth and capacity than the corresponding DRN-A or the ResNet that had been used as the starting point. However, we will show that the presented degridding scheme has a dramatic effect on accuracy, such that the accuracy gain compensates for the added depth and capacity. For example, experiments will demonstrate that DRN-C-26 has similar image classification accuracy to DRN-A-34 and higher object localization and semantic segmentation accuracy than DRN-A-50.

The activations inside a DRN-C are illustrated in Figure 4.7. This figure shows a feature map from the output of each level in the network. The feature map with the largest average activation magnitude is shown. Although the representation of each image only shows us one perspective of the layer, we can have a comprehensive view of what the level is doing by looking at visualization of multiple images. For example, in Figure 4.7, we can observe that Level 1 is separating low and high frequency components of the input images. The images in the first two rows for Level 1 show smoothed images compared to the inputs while the last row only shows the high frequency component. Level 2 tries to preserve the object contour after downsampling. Replacing max pooling with convolutions in Level 1 and 2 seems to be preserving more information for the higher layers to learn from.

## 4.6 Experiments

### 4.6.1 Image Classification

Training is performed on the ImageNet 2012 training set [68]. The training procedure is similar to He et al. [29]. We use scale and aspect ratio augmentation as in Szegedy et al. [78] and color perturbation as in Krizhevsky et al. [42] and Howard [33]. Training is performed by SGD with momentum 0.9 and weight decay $10^{-4}$. The learning rate

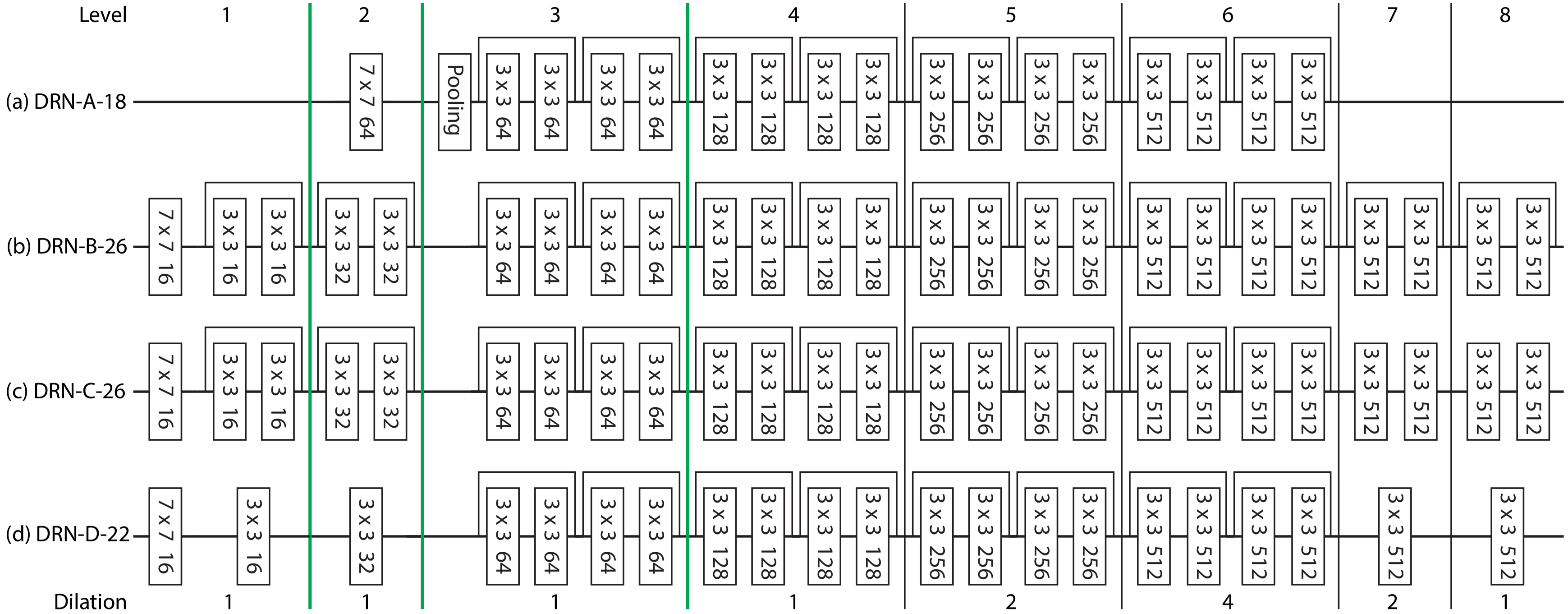


Figure 4.6: Changing the DRN architecture to remove gridding artifacts from the output activation maps. Each rectangle is a Conv-BN-ReLU group and the numbers specify the filter size and the number of channels in that layer. The bold green lines represent downsampling by stride 2. The networks are divided into levels, such that all layers within a given level have the same dilation and spatial resolution. (a) DRN-A dilates the ResNet model directly, as described in Section 4.3. (b) DRN-B replaces an early max pooling layer by residual blocks and adds residual blocks at the end of the network. (c) DRN-C removes residual connections from some of the added blocks. The rationale for each step is described in the text. (d) DRN-D, discussed in Section 4.6.1, is a simplified version of DRN-C with fewer 3 × 3 convolutions.

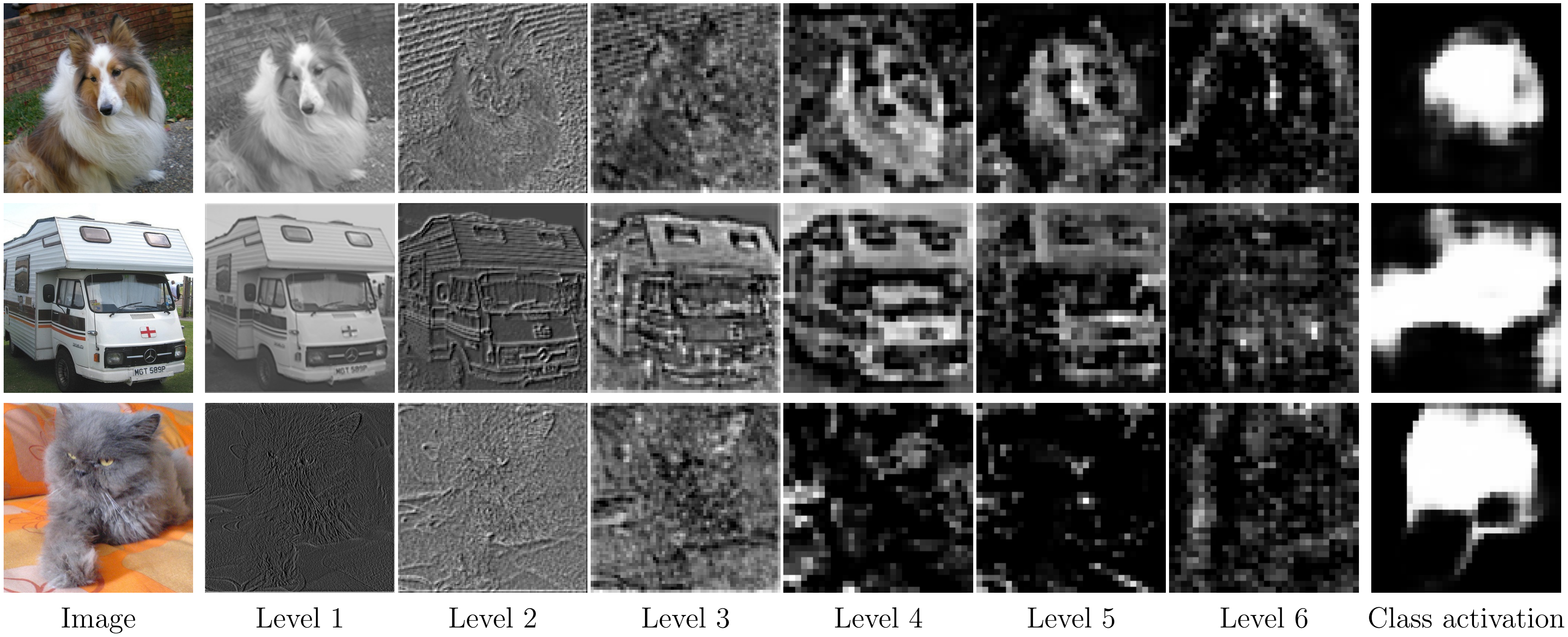


Figure 4.7: Activations inside a trained DRN-C-26. For each level, we show the feature map with the highest average activation magnitude among feature maps in the level's output. The levels are defined in Figure 4.6.

is initially set to $10^{-1}$ and is reduced by a factor of 10 every 30 epochs. Training proceeds for 120 epochs total.

|  | Top-1 |  | Top-5 |  |
|---|---|---|---|---|
| # layers | ResNet | DRN-A | ResNet | DRN-A |
| 18 | 30.43 | **27.97** | 10.76 | **9.54** |
| 34 | 26.73 | **24.81** | 8.74 | **7.54** |
| 50 | 24.01 | **22.94** | 7.02 | **6.57** |

Table 4.1: Image classification accuracy on the ImageNet 2012 validation set. Lower is better. Each DRN outperforms the corresponding ResNet model.

The performance of trained models is evaluated on the ImageNet 2012 validation set. The images are resized so that the shorter side has 256 pixels. We use two evaluation protocols: 1-crop and 10-crop. In the 1-crop protocol, prediction accuracy is measured on the central 224×224 crop. In the 10-crop protocol, prediction accuracy is measured on 10 crops from each image. Specifically, for each image we take the center crop, four corner crops, and flipped versions of these crops. The reported 10-crop accuracy is averaged over these 10 crops.

**ResNet vs. DRN-A.** Table 4.1 reports the accuracy of different models according to both evaluation protocols. Each DRN-A outperforms the corresponding ResNet model, despite having the same depth and capacity. For example, DRN-A-18 and DRN-A-34 outperform ResNet-18 and ResNet-34 in 1-crop top-1 accuracy by 2.43 and 2.92 percentage points, respectively. (A 10.5% error reduction in the case of ResNet-34 $\rightarrow$ DRN-A-34.)

DRN-A-50 outperforms ResNet-50 in 1-crop top-1 accuracy by more than a percentage point. For comparison, the corresponding error reduction achieved by ResNet-152 over ResNet-101 is 0.3 percentage points. (From 22.44 to 22.16 on the center crop.) These results indicate that even the direct transformation of a ResNet into a DRN-A, which does not change the depth or capacity of the model at all, significantly improves classification accuracy.

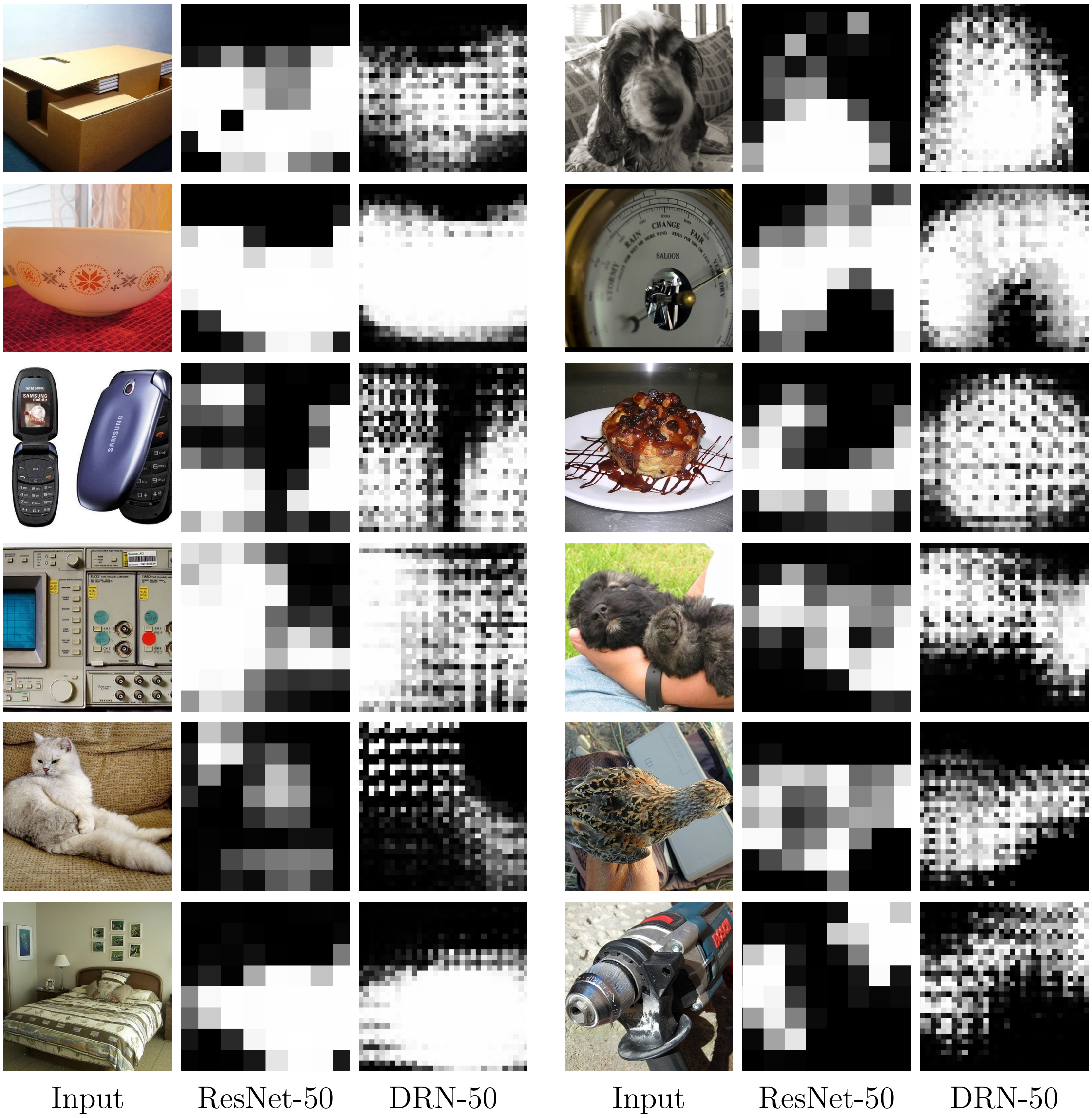


Figure 4.8: Acivation maps produced by ResNet-50 and DRN-50 on images from the ImageNet validation set. For each image, the figure shows activation maps for the predicted class, produced by the procedure described in Section 4.4. Activation maps produced by DRN are much better spatially resolved.

| Model | 1 crop | | 10 crops | | $P$ |
|---|---|---|---|---|---|
| | top-1 | top-5 | top-1 | top-5 | |
| ResNet-18 | 30.43 | 10.76 | 28.22 | 9.42 | 11.7M |
| DRN-A-18 | 28.00 | 9.50 | 25.75 | 8.25 | 11.7M |
| DRN-B-26 | 25.19 | 7.91 | 23.33 | 6.69 | 21.1M |
| DRN-C-26 | 24.86 | 7.55 | 22.93 | 6.39 | 21.1M |
| ResNet-34 | 27.73 | 8.74 | 24.76 | 7.35 | 21.8M |
| DRN-A-34 | 24.81 | 7.54 | 22.64 | 6.34 | 21.8M |
| DRN-C-42 | 22.94 | 6.57 | 21.20 | 5.60 | 31.2M |
| ResNet-50 | 24.01 | 7.02 | 22.24 | 6.08 | 25.6M |
| DRN-A-50 | 22.94 | 6.57 | 21.34 | 5.74 | 25.6M |
| ResNet-101 | 22.44 | 6.21 | 21.08 | 5.35 | 44.5M |

Table 4.2: Image classification accuracy (error rates) on the ImageNet 2012 validation set. Lower is better. $P$ is the number of parameters in each model.

**DRN-A vs. DRN-C.** Table 4.2 also shows that the degridding construction described in Section 4.5 is beneficial. Specifically, each DRN-C significantly outperforms the corresponding DRN-A. Although the degridding procedure increases depth and capacity, the resultant increase in accuracy is so substantial that the transformed DRN matches the accuracy of deeper models. Specifically, DRN-C-26, which is derived from DRN-A-18, matches the accuracy of the deeper DRN-A-34. In turn, DRN-C-42, which is derived from DRN-A-34, matches the accuracy of the deeper DRN-A-50. Comparing the degridded DRN to the original ResNet models, we see that DRN-C-42 approaches the accuracy of ResNet-101, although the latter is deeper by a factor of 2.4.

Some examples are shown in Figure 4.10 to illustrate the difference between DRN and ResNet. ResNet-50 can make the wrong prediction when other objects are more prominent than the labeled ones, while DRN-50 can classify those images correctly. It indicates that it is helpful to keep spatial resolution to preserve object location information, even for image classification.

**DRN-D** After understanding the correct structure of the degridding layers, we further explore how to simplify DRN-C. The proposed structure is shown in Figure 4.6 (d). The main change is to shrink the couples of 3 × 3 convolutions to a single layers. It can improve the efficiency of DRN-C, while still have degridding effects. The accuracies of DRN-D on ImageNet classification is shown in Table 4.3. The full comparison of different DRN models and ResNets is shown in Figure 4.9. It shows that DRN-D can achieve better performance compared to other models with similar number of layers or more parameters.

| Model | 1 crop | | $P$ |
|---|---|---|---|
| | top-1 | top-5 | |
| ResNet-18 | 30.4 | 10.8 | 11.7M |
| DRN-D-22 | 25.8 | 8.2 | 16.4M |
| ResNet-34 | 27.7 | 8.7 | 21.8M |
| DRN-D-38 | 23.8 | 6.9 | 26.5M |
| ResNet-50 | 24.0 | 7.0 | 25.6M |
| DRN-D-54 | 21.2 | 5.9 | 35.8M |
| ResNet-101 | 22.4 | 6.2 | 44.5M |
| DRN-D-105 | 20.6 | 5.5 | 54.8M |
| ResNet-152 | 22.2 | 6.2 | 60.2M |

Table 4.3: Image classification accuracy (error rates) on the ImageNet 2012 validation set. Lower is better. $P$ is the number of parameters in each model.

### 4.6.2 Object Localization

We now evaluate the use of DRNs for weakly-supervised object localization, as described in Section 4.4. As shown in Figure 4.8, class activation maps provided by DRNs are much better spatially resolved than activation maps extracted from the corresponding ResNet.

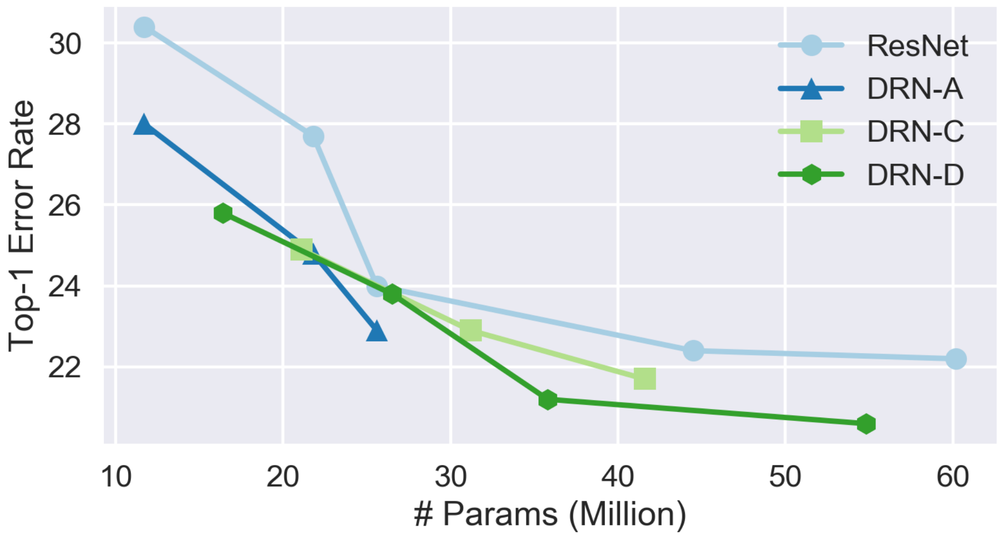


Figure 4.9: Visual comparison of different models regarding model accuracy and paramter efficiency.

We evaluate the utility of the high-resolution activation maps provided by DRNs for weakly-supervised object localization using the ImageNet 2012 validation set. We first predict the image categories based on 10-crop testing. Since the ground truth is in the form of bounding boxes, we need to fit bounding boxes to the activation maps. We predict the object bounding boxes by analyzing the class responses on all the response maps. The general idea is to find tight bounding boxes that cover pixels for which the dominant response indicates the correct object class. Specifically, given C response maps of resolution W×H, let $\mathbf{f}(c, w, h)$ be the response at location $(w, h)$ on the $c^{\text{th}}$ response map. In the ImageNet dataset, $C$ is 1000. We identify the dominant class at each location:

$$\mathbf{g}(w, h) = \left\{c \mid \forall 1 \leq c' \leq \text{C}.\ \mathbf{f}(c, w, h) \geq \mathbf{f}(c', w, h)\right\}.$$

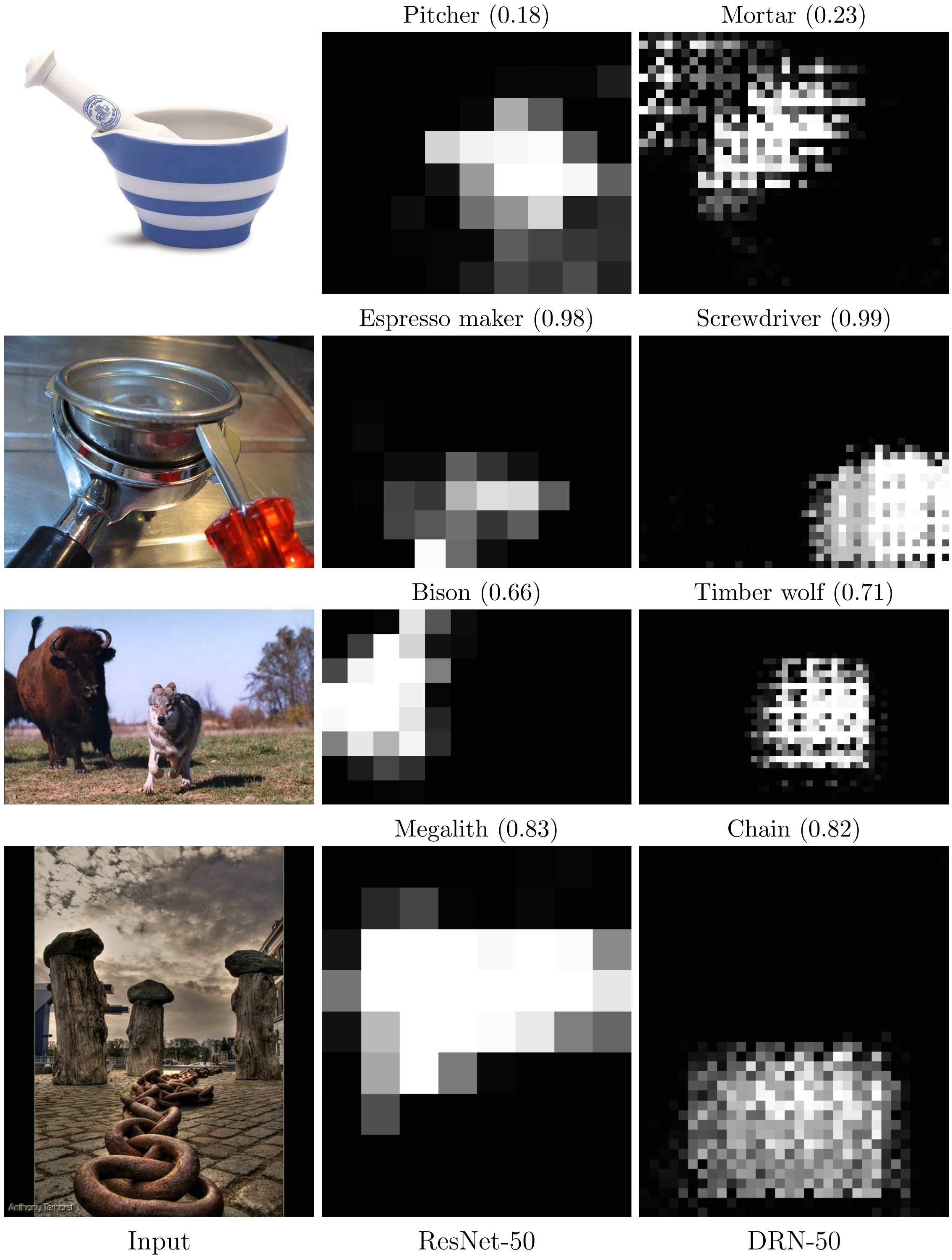


Figure 4.10: Images with small objects classified correctly by DRN-50 but missed by ResNet-50. The small captions above each activation map above are the predicted categories and their probability based on 10-crop testing. Smaller objects in the images can be better recognized by DRN-50.

For each class $c_i$, define the set of valid bounding boxes as

$$\mathcal{B}_i = \big\{((w_1, h_1), (w_2, h_2)) | \forall \mathbf{g}(w, h) = c_i \text{ and } \mathbf{f}(w, h, c_i) > t. \\ w_1 \leq w \leq w_2 \text{ and } h_1 \leq h \leq h_2 \big\},$$

where $t$ is an activation threshold. The minimal bounding box for class $c_i$ is defined as

$$\mathbf{b}_i = \underset{((w_1,h_1),(w_2,h_2)) \in \mathcal{B}_i}{\arg\min} (w_2 - w_1)(h_2 - h_1). \tag{4.5}$$

To evaluate the accuracy of DRNs on weakly-supervised object localization, we simply compute the minimal bounding box $\mathbf{b}_i$ for the predicted class $i$ on each image. In the localization challenge, a predicted bounding box is considered accurate when its IoU with the ground-truth box is greater than 0.5. Table 4.4 reports the results. Note that the classification networks are used for localization directly, with no fine-tuning.

As shown in Table 4.4, DRNs outperform the corresponding ResNet models. (Compare ResNet-18 to DRN-A-18, ResNet-34 to DRN-A-34, and ResNet-50 to DRN-A-50.) This again illustrates the benefits of the basic DRN construction presented in Section 4.3. Furthermore, DRN-C-26 significantly outperforms DRN-A-50, despite having much lower depth. This indicates that that the degridding scheme described in Section 4.5 has particularly significant benefits for applications that require more detailed spatial image analysis. DRN-C-26 also outperforms ResNet-101.

To evaluate the accuracy of DRNs on weakly-supervised object localization, we simply compute the minimal bounding box $\mathbf{b}_i$ for the predicted class $i$ on each image. In the localization challenge, a predicted bounding box is considered accurate when its IoU with the ground-truth box is greater than 0.5. Table 4.4 reports the results. DRNs achieve lower error than corresponding ResNets, and DRN-34 and DRN-50 outperform the state-of-the-art method of Zhou et al. [95] in top-1 accuracy. Note that

| Model | top-1 | top-5 |
|---|---|---|
| ResNet-18 | 61.5 | 59.3 |
| DRN-A-18 | 54.6 | 48.2 |
| DRN-B-26 | 53.8 | 49.3 |
| DRN-C-26 | 52.3 | 47.7 |
| ResNet-34 | 58.7 | 56.4 |
| DRN-A-34 | 55.5 | 50.7 |
| DRN-C-42 | 50.7 | 46.8 |
| ResNet-50 | 55.7 | 52.8 |
| DRN-A-50 | 54.0 | 48.4 |
| ResNet-101 | 54.6 | 51.9 |

Table 4.4: Weakly-supervised object localization error rates on the ImageNet validation set. Lower is better. The degridded DRN-C-26 outperforms DRN-A-50, despite lower depth and classification accuracy. DRN-C-26 also outperforms ResNet-101.

unlike the method of Zhou et al. [95], our results were produced with no additional training. Examples of weakly-supervised localization are shown in Figure 4.11.

The class response maps produced by DRNs can also be used for weakly-supervised segmentation. This is illustrated in Figure 4.12. For this purpose, we simply apply GrabCut [66] to the bounding box $\mathbf{b}_i$ computed as described above for the predicted class $i$. The class activation map is treated as the unary energy. The results suggest that a DRN trained for image classification, with no localization or segmentation supervision, can produce clean object segmentations with no fine-tuning.

### 4.6.3 Semantic Segmentation

We now transfer DRNs to semantic segmentation. High-resolution internal representations are known to be important for this task [52, 89, 14]. Due to the severe downsampling in prior image classification architectures, their transfer to semantic segmentation necessitated post-hoc adaptations such as up-convolutions, skip connections, and post-hoc dilation [52, 8, 55, 89]. In contrast, the high resolution of the

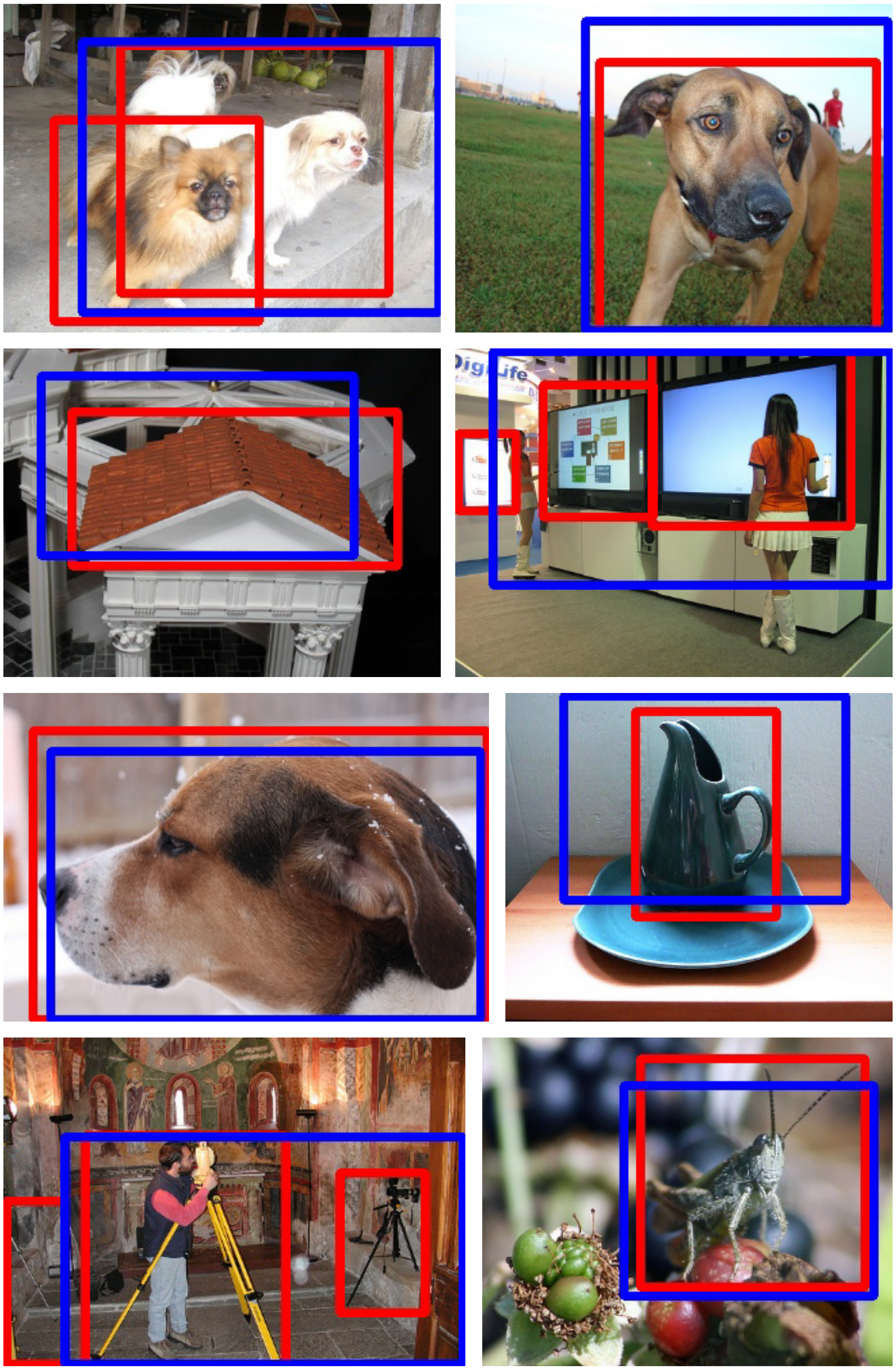


Figure 4.11: Weakly-supervised object localization by DRN-50 on images from the ImageNet validation set. The ground-truth bounding box is shown in red, the bounding box predicted by our approach is shown in blue.

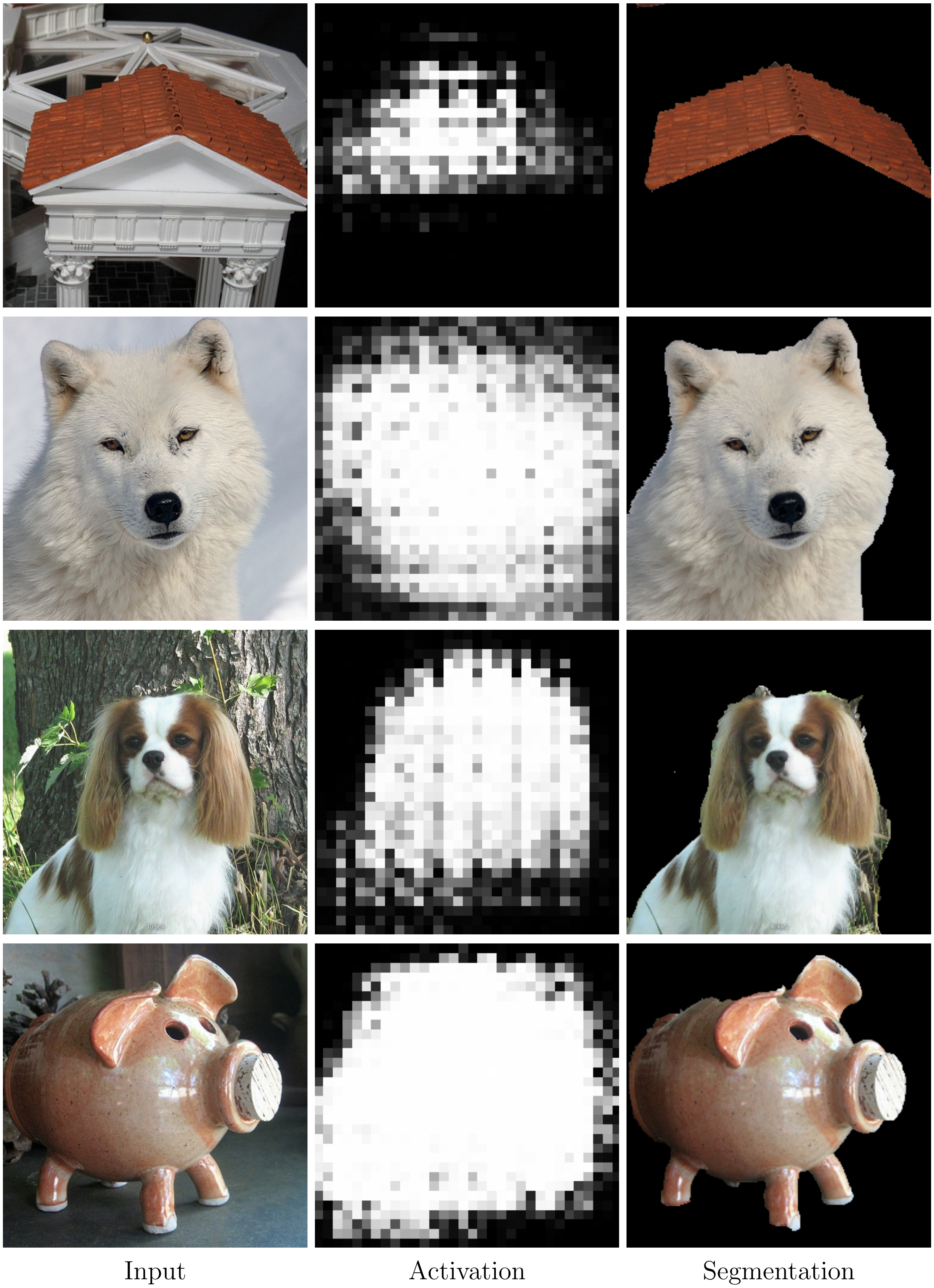
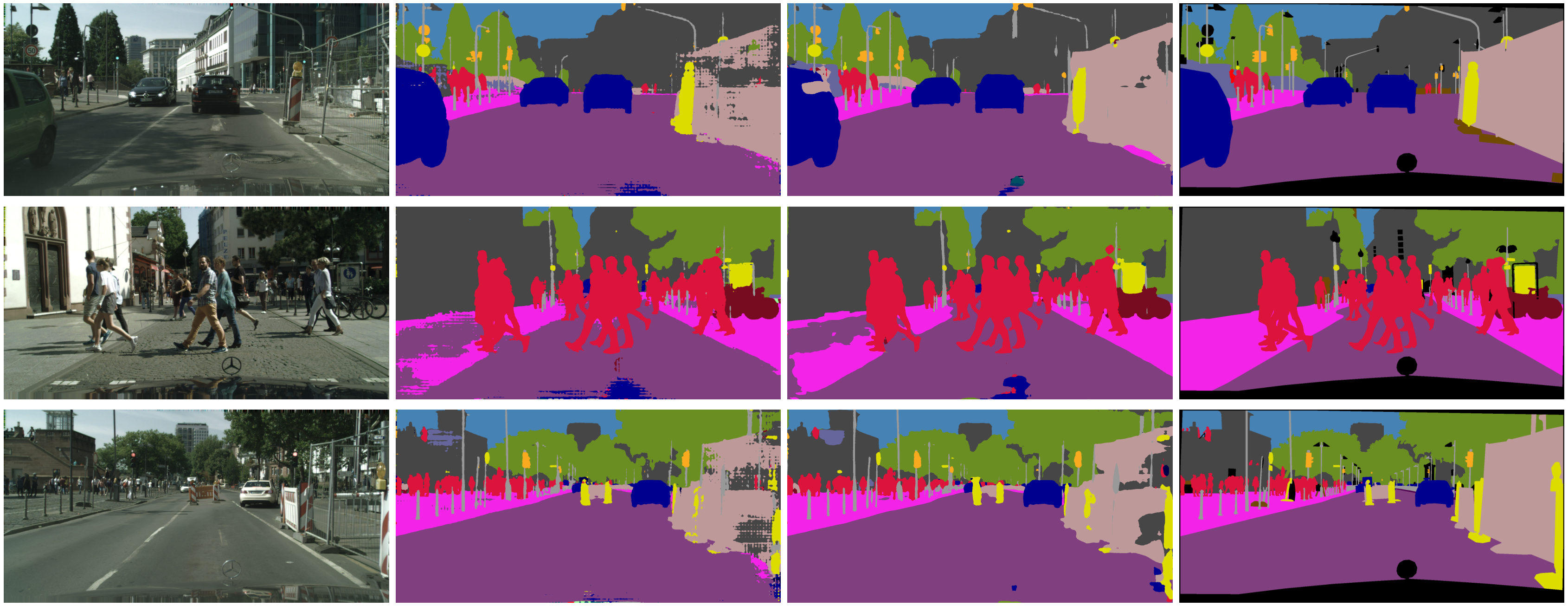


Figure 4.12: Weakly-supervised object segmentation using DRN-50.

|  | Road | Sidewalk | Building | Wall | Fence | Pole | Light | Sign | Vegetation | Terrain | Sky | Person | Rider | Car | Truck | Bus | Train | Motorcycle | Bicycle | mean IoU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DRN-A-50 | 96.9 | 77.4 | 90.3 | 35.8 | 42.8 | 59.0 | **66.8** | 74.5 | 91.6 | 57.0 | 93.4 | 78.7 | 55.3 | 92.1 | 43.2 | 59.5 | 36.2 | **52.0** | **75.2** | 67.3 |
| DRN-C-26 | 97.4 | 80.7 | 90.4 | 36.1 | 47.0 | 56.9 | 63.8 | 73.0 | 91.2 | **57.9** | 93.4 | 77.3 | 53.8 | 92.7 | 45.0 | 70.5 | 48.4 | 44.2 | 72.8 | 68.0 |
| DRN-C-42 | **97.7** | **82.2** | **91.2** | **40.5** | **52.6** | **59.2** | 66.7 | **74.6** | **91.7** | 57.7 | **94.1** | **79.1** | **56.0** | **93.6** | **56.0** | **74.3** | **54.7** | 50.9 | 74.1 | **70.9** |

Table 4.5: Performance of dilated residual networks on the Cityscapes validation set. Higher is better. DRN-C-26 outperforms DRN-A-50, despite lower depth. DRN-C-42 achieves even higher accuracy. For reference, a comparable baseline setup of ResNet-101 was reported to achieve a mean IoU of 66.6.



(a) Input (b) DRN-A-50 (c) DRN-C-26 (d) Ground truth

Figure 4.13: Semantic segmentation on the Cityscapes dataset. The degridded DRN-C-26 produces cleaner results than the deeper DRN-A-50.

output layer in a DRN means that we can transfer a classification-trained DRN to semantic segmentation by simply removing the global pooling layer and operating the network fully-convolutionally [52], without any additional structural changes. The predictions synthesized by the output layer are upsampled to full resolution using bilinear interpolation, which does not involve any parameters.

We evaluate this capability using the Cityscapes dataset [14]. We use the standard Cityscapes training and validation sets. To understand the properties of the models themselves, we only use image cropping and mirroring for training. We do not use any other data augmentation and do not append additional modules to the network. The results are reported in Table 4.5.

All presented models outperform a comparable baseline setup of ResNet-101, which was reported to achieve a mean IoU of 66.6 [8]. For example, DRN-C-26 outperforms the ResNet-101 baseline by more than a percentage point, despite having 4 times lower depth. The DRN-C-42 model outperforms the ResNet-101 baseline by more than 4 percentage points, despite 2.4 times lower depth. We also test the state-of-the-art training method on Cityscapes following PSPNet [91]. DRN-D-105 get 76.2% mIoU on Cityscapes testing set, which is the state-of-the-art results without using additional context modules on this challenging dataset.

Comparing different DRN models, we see that both DRN-C-26 and DRN-C-42 outperform DRN-A-50, suggesting that the degridding construction presented in Section 4.5 is particularly beneficial for dense prediction tasks. A qualitative comparison between DRN-A-50 and DRN-C-26 is shown in Figure 4.13. As the images show, the predictions of DRN-A-50 are marred by gridding artifacts even though the model was trained with dense pixel-level supervision. In contrast, the predictions of DRN-C-26 are not only more accurate, but also visibly cleaner.

## 4.7 Conclusion

We have presented an approach to designing image classification networks. Rather than progressively reducing the resolution of the internal representations until the spatial structure of the scene is no longer discernable, we keep high spatial resolution all the way through the final output layers. We have shown that this design increases image classification accuracy, outperforming state-of-the-art models on the ImageNet dataset. We have further shown that the presented image classification networks produce informative output activations, which can be used directly for weakly-supervised object localization, without any fine-tuning. Experiments also demonstrated that the presented design supports direct transfer to dense fully-convolutional operation, providing state-of-the-art performance without post-hoc reconfiguration. The results suggest that the presented approach can usefully inform the design of convolutional networks for complex natural images.

# Chapter 5

# Conclusion

## 5.1 Key Problems and Contributions

Pixel-wise prediction is a generalization of computer vision tasks. They range from estimating low-level geometry to understanding high-level semantics in images. Even though most of the problems are studied in isolation, they share some common insights. They all require a rich image representation that can connect the pixels to the semantics. The representation also has to be invariant to scale and incorporate context information. We also show that videos can also help obtain plausible estimations without a training process, even though the motion between video frames are very small. This thesis observes the connections among the pixel-level prediction tasks and tries to build a basic framework for them.

We first investigate depth prediction based on images captured by mobile application. We observe that there are numerous other frames around the target image that can provide auxiliary geometry information. However, the small baselines between the additional frames prohibit the traditional methods to work reliably. We look into the structure from motion formulation and analyze the convex properties of different motion and structure variables. We find that some properties such as camera rotation and view angles of points on the infinite plane are easier to optimize. However, the estimated 3D structures usually have high uncertainty. We propose to use densely connected conditional random field to regularize the dense depth estimation and it is shown to work better than locally connected constraints. The analysis leads to a pipeline that can produce plausible depth estimation for different computation photography effects.

To build high-level semantic image representation, we study the usage of dilated convolutions in the convolutional networks. Dilated convolutions have two prominent properties, which make them suitable for constructing dense prediction networks. First, we can increase the output resolution of an image classification networks by replacing the strides with dilations without changing the number of parameters and connections between original activations. Second, by exponentially increasing the dilations through layers, we are able to increase the receptive fields of different layers exponentially. This can help aggregate context information from a large extent on the images. Our experiments show that the dilated networks can outperform alternative designs using skip connections for up-sampling. We also find that our context module composed of convolutional layers with exponentially increased dilations can further improve the results significantly.

Although dilated convolutions can transform an existing image classification networks to have higher resolution output without adding additional parameters, it is questionable that the transformation is necessary. Therefore, we investigate the difference between image classification and segmentation by studying the role of high-resolution layers enabled by dilation convolutions in image classification. To conduct the study based on state-of-the-art networks, we compare dilated residual networks to the original ResNets with the same number of parameters and layers. Interestingly, we find that the dilated residual networks always perform better than their counterparts with the same number of parameters and layers. This suggests that the spatial resolution of layers also plays an important role in network capacity. To understand the layer activations arising from the high spatial resolution, we visualize the class activations without retraining the networks. We find that there are gridding artifacts in the feature maps. Its impact on semantic image segmentation motivates our further study into removing the artifacts. Our layer visualization shows that the gridding artifacts is due to compound effects of the discontinuity in the layer responses and dilated convolutions. Hence, we proposed changes to the directly dilated networks. The new networks can produce smooth class activation maps and improve the performance of the classification networks. What's more, the new networks also produce better results in semantic image classification.

This thesis studies geometry, semantic and context cues for pixel-level prediction problems. The ideas have inspired new developments in these fields. This work alludes to several future directions.

## 5.2 Future Works

### 5.2.1 Connections between Geometry and Semantics

Staring at an image of a natural scene, we can write a long essay to describe the 3D relations between the objects, because we can recognize grouping of pixels and their occlusion orders. The same cues can also help the computers to infer better 3D information from either a single image or a collection of images with accidental motion. The model will have to recognize the high-level information to infer geometry relations and hopefully aid semantic image understanding with geometry knowledge. We hope to combine the 3D geometry knowledge with semantic information in a single learning system.

### 5.2.2 Unified Framework for Image Recognition

In this thesis, we understand that there is no essential difference between image classification, which only requires one single prediction for the whole image, and pixel-wise prediction problems. This motivates us to explore in two directions. First, image recognition also includes other problems such as object detection and boundary prediction. Current works are designing different models for different problems, although they share some common requirements such as semantics and contexts. This practice also hinders the application of convolutional networks in real-world scenarios. We hope to design a more general framework that can achieve state-of-the-art performance on all these problems so that it can provide a base representation for different perception tasks. Second, such unified framework also has to be efficient regarding parameters and computation so that it can be applied to different domains without incurring overhead. We hope this thesis work can inspire future discovery of such unified network.

# Bibliography

[1] Top Devices of 2017 on Flickr . `https://blog.flickr.net/en/2017/12/07/top-devices-of-2017/`, 2016.

[2] Andrew Adams, Jongmin Baek, and Myers Abraham Davis. Fast high-dimensional filtering using the permutohedral lattice. *CGF*, 2010.

[3] Sameer Agarwal and Keir Mierle. *Ceres Solver: Tutorial & Reference*. Google Inc.

[4] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv:1505.07293*, 2015.

[5] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2), 2009.

[6] Neill D.F. Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *ECCV*, 2008.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *ICLR*, 2015.

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[9] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L. Yuille. Attention to scale: Scale-aware semantic image segmentation. *arXiv:1511.03339*, 2015.

[10] Alessandro Chiuso, Roger Brockett, and Stefano Soatto. Optimal structure from motion: Local ambiguities and global estimates. *IJCV*, 2000.

[11] Stéphane Christy and Radu Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *TPAMI*, 1996.

[12] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *PAMI*, 2016.

[13] R.T. Collins. A space-sweep approach to true multi-image matching. In *CVPR*, 1996.

[14] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[15] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *CVPR*, 2011.

[16] Mark Everingham, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The Pascal visual object classes (VOC) challenge. *IJCV*, 88(2), 2010.

[17] Clément Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *PAMI*, 35(8), 2013.

[18] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazrba, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Learning optical flow with convolutional neural networks. In *ICCV*, 2015.

[19] Carolina Galleguillos and Serge J. Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6), 2010.

[20] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 32(11), 2013.

[21] C. Geyer, T. Templeton, M. Meingast, and S.S. Sastry. The recursive multi-frame planar parallax algorithm. In *3DPVT*, 2006.

[22] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *PAMI*, 38(1), 2016.

[23] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[24] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Efficient learning with sets of features. *JMLR*, 8, 2007.

[25] Bharath Hariharan, Pablo Arbelaez, Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[26] Bharath Hariharan, Pablo Andrés Arbeláez, Ross B. Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015.

[27] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Second edition, 2004.

[28] Richard Hartley, Fredrik Kahl, Carl Olsson, and Yongduek Seo. Verifying global minima for $L_2$ minimization problems in multiple view geometry. *IJCV*, 2013.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[30] Xuming He, Richard S. Zemel, and Miguel Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *CVPR*, 2004.

[31] Carlos Hernández and George Vogiatzis. Shape from photographs: A multi-view stereo pipeline. In *Computer Vision*, Studies in Computational Intelligence. 2010.

[32] M. Holschneider, R. Kronland-Martinet, J. Morlet, and Ph. Tchamitchian. 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.

[33] Andrew G Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013.

[34] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *PAMI*, 34(9), 2012.

[35] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proc. ACM Multimedia*, 2014.

[36] Fredrik Kahl. Multiple view geometry and the $L_8$-norm. In *ICCV*, 2005.

[37] Fredrik Kahl, Sameer Agarwal, ManmohanKrishna Chandraker, David Kriegman, and Serge Belongie. Practical global optimization for multiview geometry. *IJCV*, 2008.

[38] P. Kohli, L. Ladicky, and P. H S Torr. Robust higher order potentials for enforcing label consistency. In *CVPR*, 2008.

[39] Pushmeet Kohli, Lubor Ladicky, and Philip H. S. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 82(3), 2009.

[40] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order mrfs. In *CVPR*, 2009.

[41] Philipp Krähenbühl and Vladlen Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *NIPS*, 2011.

[42] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[43] Abhijit Kundu, Vibhav Vineet, and Vladlen Koltun. Feature space optimization for semantic video segmentation. In *CVPR*, 2016.

[44] Lubor Ladicky, Christopher Russell, Pushmeet Kohli, and Philip H. S. Torr. Associative hierarchical CRFs for object class image segmentation. In *ICCV*, 2009.

[45] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[46] Quoc V. Le, Navdeep Jaitly, and Geoffrey E. Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv:1504.00941*, 2015.

[47] Yann LeCun, Bernhard Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne Hubbard, and Lawrence D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 1989.

[48] Guosheng Lin, Chunhua Shen, Ian Reid, and Anton van dan Hengel. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv:1504.01013*, 2015.

[49] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv:1312.4400*, 2013.

[50] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[51] Buyu Liu and Xuming He. Multiclass semantic video segmentation with object-level active inference. In *CVPR*, 2015.

[52] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[53] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981.

[54] Alessandro Mulloni, Gerhard Reitmayr, Daniel Wagner, Raphael Grasset, and Serafin Diaz. User Friendly SLAM Initialization. *ISMAR*, 2013.

[55] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015.

[56] John Oliensis. Computing the camera heading from multiple frames. In *CVPR*, 1998.

[57] John Oliensis. A multi-frame structure-from-motion algorithm under perspective projection. *IJCV*, 1999.

[58] John Oliensis. The least-squares error for structure from infinitesimal motion. *IJCV*, 2005.

[59] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014.

[60] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *CVPR*, 2015.

[61] Florent Perronnin and Christopher R. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.

[62] Pedro H. O. Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015.

[63] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *CVPR*, 2011.

[64] Germán Ros, Sebastian Ramos, Manuel Granados, Amir Bakhtiary, David Vázquez, and Antonio Manuel López. Vision-based offline-online perception paradigm for autonomous driving. In *WACV*, 2015.

[65] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of DNNs. *CoRR*, abs/1712.02616, December 2017.

[66] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3), 2004.

[67] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323, 1986.

[68] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. ImageNet large scale visual recognition challenge. *IJCV*, 115(3), 2015.

[69] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2002.

[70] Mark J. Shensa. The discrete wavelet transform: wedding the à trous and Mallat algorithms. *IEEE Transactions on Signal Processing*, 40(10), 1992.

[71] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

[72] Jamie Shotton, John M. Winn, Carsten Rother, and Antonio Criminisi. TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1), 2009.

[73] K. Sim and R. Hartley. Recovering camera motion using linfty minimization. In *CVPR*, 2006.

[74] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[75] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *IJCV*, 2008.

[76] Paul Sturgess, Karteek Alahari, Lubor Ladicky, and Philip H. S. Torr. Combining appearance and structure from motion features for road scene understanding. In *BMVC*, 2009.

[77] Peter Sturm and Bill Triggs. A factorization based algorithm for multi-image projective structure and motion. In *ECCV*. 1996.

[78] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[79] Joseph Tighe and Svetlana Lazebnik. Superparsing – scalable nonparametric image parsing with superpixels. *IJCV*, 101(2), 2013.

[80] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. 1991.

[81] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 1992.

[82] Antonio Torralba, Robert Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *PAMI*, 30(11), 2008.

[83] Bill Triggs. Factorization methods for projective structure and motion. CVPR, 1996.

[84] Bill Triggs. Empirical filter estimation for subpixel interpolation and matching. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 550–557. IEEE, 2001.

[85] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustmenta modern synthesis. In *Vision algorithms: theory and practice*. 2000.

[86] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. *arXiv preprint arXiv:1702.08502*, 2017.

[87] O.J. Woodford, P. H S Torr, I.D. Reid, and A.W. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *CVPR*, 2008.

[88] Fisher Yu and David Gallup. 3d reconstruction from accidental motion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[89] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[90] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[91] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*, 2016.

[92] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.

[93] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.

[94] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene CNNs. In *ICLR*, 2015.

[95] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

[96] Bolei Zhou, Àgata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using Places database. In *NIPS*, 2014.