Detailed Dense Inference with Convolutional Neural Networks via Discrete Wavelet Transform

Lingni Ma[1], Jörg Stückler[2], Tao Wu[1] and Daniel Cremers[1]

Abstract— Dense pixelwise prediction such as semantic segmentation is an up-to-date challenge for deep convolutional neural networks (CNNs). Many state-of-the-art approaches either tackle the loss of high-resolution information due to pooling in the encoder stage, or use dilated convolutions or high-resolution lanes to maintain detailed feature maps and predictions. Motivated by the structural analogy between multi-resolution wavelet analysis and the pooling/unpooling layers of CNNs, we introduce discrete wavelet transform (DWT) into the CNN encoder-decoder architecture and propose WCNN. The high-frequency wavelet coefficients are computed at encoder, which are later used at the decoder to unpooled jointly with coarse-resolution feature maps through the inverse DWT. The DWT/iDWT is further used to develop two wavelet pyramids to capture the global context, where the multi-resolution DWT is applied to successively reduce the spatial resolution and increase the receptive field. Experiment with the Cityscape dataset, the proposed WCNNs are computationally efficient and yield improvements the accuracy for high-resolution dense pixelwise prediction.

I. INTRODUCTION

Dense pixelwise prediction tasks such as semantic segmentation, optical flow or depth estimation remain up-to-date challenges in computer vision. They find rapidly rising interests for applications such as autonomous driving, robotic vision and image scene understanding. Succeeded by its remarkable success in image recognition [1], deep convolutional neural networks (CNNs) have achieved state-of-the-art performances in dense prediction tasks such as semantic segmentation [2]–[4] or single-image depth estimation [5].

Many dense prediction tasks consist of two concurrent goals: classification and localization. Classification is well tackled by an end-to-end trainable CNN architecture, e.g. VGGNet [6] or ResNet [7], which typically stacks multiple layers of successive convolution, nonlinear activation, and pooling. A typical pooling step, which performs either a subsampling or some strided averaging on an input volume, is favorable for the invariance of prediction results to small spatial translations in the input data as well as for the boost of computational efficiency via dimension reduction. Its downside, however, is the loss of resolution in output feature maps, which renders high-quality pixelwise prediction challenging.

Several remedies for such a dilemma have been proposed in the literatures. As suggested in [8], [9], one may mirror

the encoder network by a decoder network. Each upsampling (or unpooling) layer in the decoder network is introduced in symmetry to a corresponding pooling layer in the encoder network, and then followed by trainable convolutional layers. Alternatively, one may use dilated (also known as atrous) convolutions in a CNN encoder as proposed in [10]–[12]. This enables the CNN to expand the receptive fields of pixels as convolutional layers stack up without losing resolution in the feature maps, however, at the cost of significant computational time and memory. Another alternative is to combine a CNN low-resolution classifier with a conditional random field (CRF) [13], [14], either as a stand-alone post-processing step [11], [12] or combined with a CNN in an end-to-end trainable architecture [15], [16]. The latter also comes with an increased demand in run-time for training and inference.

Motivated by close analogy between pooling (resp. unpooling) in an encoder-decoder CNN and decomposition (resp. reconstruction) in multi-resolution wavelet analysis, this paper proposes a new class of CNNs with wavelet unpooling and wavelet pyramid. We name the network WCNN. The first contribution with WCNN is to achieve unpooling with the inverse discrete wavelet transform (iDWT). To this end, DWT is applied at the encoder to decompose feature maps into frequency bands. The high frequency components are skip-connected to the decoder to perform iDWT jointly with the coarse-resolution feature maps. The wavelet unpooling does not require any additional parameters over baseline CNNs, where the overhead only comes from the memory to cache frequency coefficients from encoder. The second contribution of WCNN are two wavelet-based pyramid variants to bridge the standard encoder and decoder. The wavelet pyramids obtain global context from a receptive field of the entire image by exploiting multi-resolution DWT/iDWT. The experiments over the dataset Cityscape show that the proposed WCNN yields systematically improvements in dense prediction accuracy.

II. RELATED WORK

Many challenging tasks in computer vision such as single image depth prediction or semantic image segmentation require models for dense prediction, since they either involve regressing quantities pixelwise or classifying the pixels. Most current state-of-the-art methods for dense prediction tasks are based on end-to-end trainable deep learning architectures. Early methods segment the image into regions such as superpixels in a bottom-up fashion. Predictions for the regions are determined based on deep neural network features [17]–

---

[1] Lingni Ma, Tao Wu and Daniel Cremers are with the Computer Vision and Artificial Intelligence Group at the Computer Science Department, Technical University of Munich, ({lingni,tao.wu,cremers}@in.tum.de)
[2] Jörg Stückler (joerg.stueckler@tuebingen.mpg.de) is with Max Planck Institute for Intelligent Systems.

[19]. The use of image-based bottom-up regions supports adherence of the dense predictions to the boundaries in the image.

Aim at end-to-end CNNs, Long et al. [20] propose a fully connected convolutional (FCN) architecture for semantic image segmentation which successively convolves and pools feature maps of an increasing number of feature channels. FCNs employ the transposed convolution to learn the up-sampling of coarse feature maps. To obtain segmentation, feature maps of the intermediate resolutions are concatenated and further processed by transposed convolutions. Since the introduction of FCNs, many variants for dense prediction are proposed. Hariharan et al. [21] classify pixels based on feature vectors that are extracted at corresponding locations across all feature maps in a CNN. This way, the method combines features across all layers available in the network, capturing high-resolution detail as well as context in large receptive fields. However, this approach becomes inefficient in deep architectures with many wide layers. Noh et al. [8] and Dosovitsky et al. [22] propose encoder-decoder CNN architectures which successively unpool and convolve the lowest resolution feature map of the encoder back to a high output resolution. Since the feature maps in the encoder lose spatial information through pooling, Noh et al. [8] exploit the memorized unpooling [27] to upscale coarse feature maps at the decoder stage, where the pooling locations are used to unpool accordingly. The FCN of Laina et al. [5] uses the deep residual network [7] as an encoder, where most pooling layers are replaced by stride-two convolution. For upscaling, the upprojection block is developed as an efficient implementation of upconvolution. The principle of upconvolution is developed by [28], which first unpools a feature map by putting activations to one entry of a 2×2 block and then filter the sparse feature map with convolution. Details in the predictions of such encoder-decoder FCNs can be improved by feeding the feature maps in each scale of the encoder to the corresponding scale of the decoder (skip connections, e.g. [22]). In RefineNet [3], the decoder feature maps are successively refined using multi-resolution fusion with their higher resolution counterparts in the encoder. In this paper, we also reincorporate the high-frequency information that is discarded during pooling to successively refine feature maps in the decoder.

Some FCN architectures use dilated convolutions in order to increase receptive field without pooling and maintain high-resolution of the feature maps [10]–[12]. These dilated CNNs trade high-resolution output with the high memory consumption, which quickly become a bottleneck for training with large batch size for encoder-decoder CNNs. The full-resolution residual network (FRRN) by [4] is an alternative model which keeps features in a high-resolution lane and at the same time, extracts low-resolution higher-order features in an encoder-decoder architecture. The high-resolution features are successively refined from residuals computed through the encoder-decoder lane. While the model is highly demanding in memory and training time, it achieves high-resolution predictions that well adhere to seg-ment boundaries. [23] take inspiration from Laplace image decompositions for their network design. They successively refine the high-frequency parts of the score maps in order to improve predictions at segment boundaries. Structured prediction approaches integrate inference in CRFs with deep neural networks in end-to-end trainable models [15], [16], [24], [25]. While the models are capable of recovering high-resolution predictions, inference and learning typically requires tedious iterative procedures. In contrast to those approaches, we aim to provide detailed predictions in a swift and direct forward pass. Recently, the pyramid scene parsing network (PSPNet) from [2] extracts global context features using a pyramid pooling module, which shows the benefit of aggregation global information for dense predictions. The pyramid design in PSPNet relies multiple average pooling layers with heuristic window size. In this work, we also propose a more efficient pyramid pooling stage based on multi-resolution DWT.

III. WCNN ENCODER-DECODER ARCHITECTURES

Recently, CNNs have demonstrated impressive performance on many dense pixelwise prediction tasks, including image semantic segmentation, optical flow estimation, and depth regression. CNNs extract image features through successive layers of convolution and non-linear activation. In encoder architectures, as the stack of layers gets deeper, the dimension of the feature vectors increases while the spatial resolution is reduced. For dense prediction tasks, CNNs with encoder-decoder architecture are widely applied in which the feature maps of the encoder are successively unpooled and deconvolved. Research on architectures for the encoder part is relatively mature, e.g., the state-of-the-art CNNs such as VGGNet [6] and ResNet [7] are commonly used in various applications. In contrast, the design of the decoder has not yet converged to a universally accepted solution. While it is easy to reduce spatial dimension by either pooling or strided convolution, recovering a detailed prediction from a coarse and high-dimensional feature space is less straight-forward. In this paper, we make an analogy between CNN encoder-decoders to the multi-resolution wavelet transform (see Figure 1). We match the pooling operations of the CNN encoder with the multilevel forward transformation of a signal by a wavelet. The decoder performs the corresponding inverse wavelet transform for unpooling. The analogy is straight-forward: the wavelet transform successively filters the signal into frequency subbands while reducing the spatial resolution. The inverse wavelet transform successively composes the frequency subband back to full resolution. While the encoder and the decoder transform between different domains (e.g. image-to-semantic segmentation vs. image-to-image in wavelet transforms), we find that wavelet unpooling provides an elegant mechanism to transmit high-frequency information from the image domain to the semantic segmentation. It also imposes a strong architectural regularization, as the feature dimensions between the encoder and the decoder need to match through the wavelet coefficients.
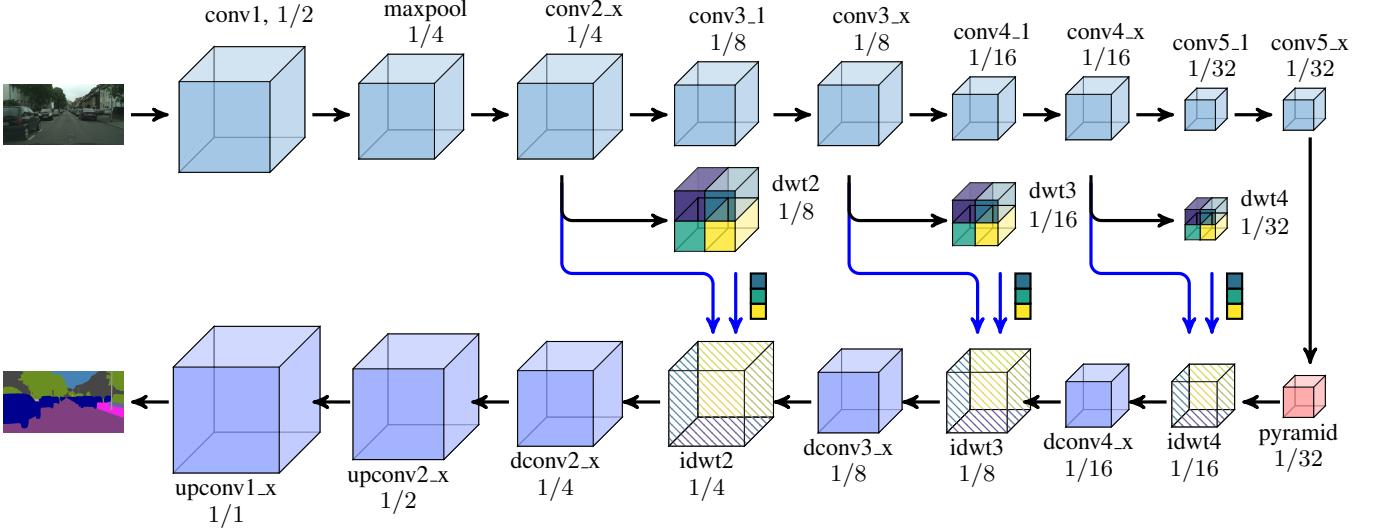
Figure 1: The encoder-decoder architecture of the proposed WCNN, where the data flow is indicated by black arrows and shortcuts are by blue arrows. Assume the input resolution is 1, the output resolution of each building block is denoted by $1/x$. WCNN employs ResNet [7] for the encoder, which reduces the input resolution by a factor of 32 via stride-two convolutional layers, except for one maxpool layer after conv1. To restore the input resolution, WCNN inserts three DWT layer after conv2, conv3 and conv4. The high frequencies from DWT layers are used in the decoder to perform unpooling by the iDWT layers. To extract global context, WCNN introduces two pyramid variants to bridge the encoder and decoder, which also exploits DWT/iDWT layers (see details in Figure 2).

A. Discrete Wavelet Transform

We briefly introduce main concepts of DWT (see [26] for a comprehensive introduction). The multi-resolution wavelet transform provides localized time-frequency analysis of signals and images. Consider a 2D input data $X \in \mathbb{R}^{2M \times 2N}$, $\phi \in \mathbb{R}^2$ and $\psi \in \mathbb{R}^2$ as 1D low-pass and high-pass filters, respectively. Denote the indexed array element by x_{ij}, the single-level DWT is defined as follows,

$$
\begin{aligned}
y_{kl}^{ll} &= \sum_l \sum_k x_{2i+k,2j+l} \phi_k \phi_l, \\
y_{kl}^{lh} &= \sum_l \sum_k x_{2i+k,2j+l} \phi_k \psi_l, \\
y_{kl}^{hl} &= \sum_l \sum_k x_{2i+k,2j+l} \psi_k \phi_l, \\
y_{kl}^{hh} &= \sum_l \sum_k x_{2i+k,2j+l} \psi_k \psi_l.
\end{aligned}
\tag{1}
$$

All the convolutions above are performed with stride 2, yielding a subsampling of factor 2 along each spatial dimension. Let the low-low frequency component $Y^{ll} := \{y_{kl}^{ll}\}$, the low-high frequency component $Y^{lh} := \{y_{kl}^{lh}\}$, the high-low frequency component $Y^{hl} := \{y_{i,j}^{hl}\}$, and the high-high frequency component $Y^{hh} := \{y_{i,j}^{hh}\}$. The DWT results in $\{Y^{ll}, Y^{lh}, Y^{hl}, Y^{hh}\} \in \mathbb{R}^{M \times N}$. Conversely, supplied with the wavelet coefficients, and provided that $\{\phi, \psi\}$ and $\{\widetilde{\phi}, \widetilde{\psi}\}$ are bi-orthogonal wavelet filters, the original input X can be reconstructed by the inverse DWT as

$$
\begin{aligned}
x_{ij} = \sum_l \sum_k \Big(& y_{kl}^{ll} \widetilde{\phi}_{i-2k} \widetilde{\phi}_{j-2l} + y_{kl}^{lh} \widetilde{\phi}_{i-2k} \widetilde{\psi}_{j-2l} \\
& + y_{kl}^{hl} \widetilde{\psi}_{i-2k} \widetilde{\phi}_{j-2l} + y_{kl}^{hh} \widetilde{\psi}_{i-2k} \widetilde{\psi}_{j-2l} \Big).
\end{aligned}
\tag{2}
$$

A cascaded wavelet decomposition successively performs Equation (1) on low-low frequency coefficients $\{(\cdot)^{ll}\}$ from fine to coarse resolution, while the reconstruction works reversely from coarse to fine resolution. In this sense, decomposition-reconstruction in multi-resolution wavelet analysis is in analogy to the pooling-unpooling steps in an encoder-decoder CNN (e.g., [8]). Moreover, it is worth noting that, while the low-frequency coefficients $\{(\cdot)^{ll}\}$ store local averages of the input data, its high-frequency counterparts, namely $\{(\cdot)^{lh}\}$, $\{(\cdot)^{hl}\}$, and $\{(\cdot)^{hh}\}$ encode local textures which are vital in recovering sharp boundaries. This motivates us to make use of the high-frequency wavelet coefficients to improve the quality of unpooling during the decoder stage and, hence, improve the accuracy of CNN in pixelwise prediction.

Throughout this paper, we extensively use the Haar wavelet for its simplicity and effectiveness to boost the performances of the underlying CNN. In this scenario, the Haar filters used for decomposition, see Equation (1), are given by

$$
\phi = \left(\frac{1}{2}, \frac{1}{2} \right) , \quad \psi = \left(\frac{1}{2}, -\frac{1}{2} \right) .
\tag{3}
$$

The corresponding reconstruction filters in Equation (2) are given by $\widetilde{\phi} = 2\phi$, $\widetilde{\psi} = 2\psi$, and hence the inverse transform reduces to a sum of Kronecker products (denoted with $\otimes$)

$$
\begin{aligned}
X = & Y^{ll} \otimes \widetilde{\phi}^\top \otimes \widetilde{\phi} + Y^{lh} \otimes \widetilde{\phi}^\top \otimes \widetilde{\psi} \\
& + Y^{hl} \otimes \widetilde{\psi}^\top \otimes \widetilde{\phi} + Y^{hh} \otimes \widetilde{\psi}^\top \otimes \widetilde{\psi} .
\end{aligned}
\tag{4}
$$

With CNNs, data at every layer are structured into 4D tensors, i.e., along the dimensions of the batch size, the channel number, the width and the height. To perform

the wavelet transform for CNNs, we apply DWT/iDWT channelwise. Without confusion, the remaining text adopts the shorthand notations $G_h(X)$ for the Haar DWT and $G_h^{-1}(Y^{ll}, Y^{lh}, Y^{hl}, Y^{hh})$ for the corresponding iDWT.

B. Wavelet CNN Encoder-Decoder Architecture

We propose a CNN encoder-decoder that resembles multi-resolution wavelet decomposition and reconstruction by its pooling and unpooling operations. In addition, we introduce two pyramid variants to capture global contextual features based on the wavelet transformation.
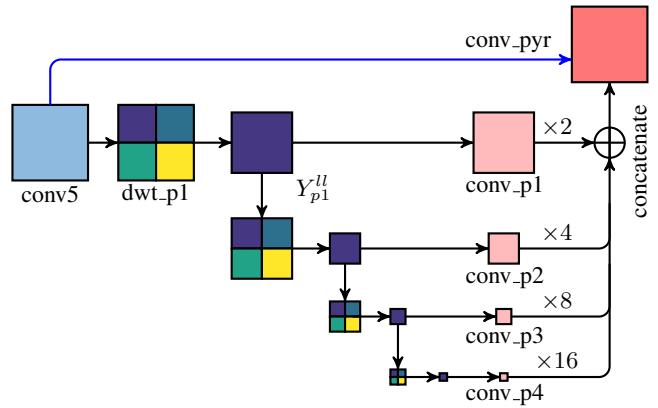
Figure 1 gives an overview of the proposed WCNN architecture. WCNN employs ResNet [7] for the encoder. In ResNet, the input resolution is successively reduced by a factor of 32 via one max-pooling layer and four stride-two convolutional layers, *i.e.,*conv1, conv3_1, conv4_1 and conv5_1. In order to restore the input resolution with the decoder, WCNN inserts three DWT layer after conv2, conv3 and conv4 to decompose the corresponding feature maps into four frequency bands. The high frequencies Y^{lh}, Y^{hl}, Y^{hh} are skip-connected to the decoder to perform unpooling via the iDWT layers, which we will discuss in details with Section III-B.1. We add three convolutional residual block [7] to filter the unpooled feature maps further before the next unpooling stage. As illustrated in Figure 1, the three iDWT layers upsample the output to $1/4$ input resolution. The full-resolution output is obtained with two upconvolutional blocks by transposed convolution. To bridge the encoder and decoder, the contextual pyramid with wavelet transformation is added. Section III-B.2 will detail the pyramid design.

1) Wavelet Unpooling: WCNN achieves the unpooling through iDWT layers. To this end, the DWT layers are added consistently into the encoder to obtain high-frequency components. The idea is straight-forward. At encoder, the DWT layers decompose the feature map into four frequency bands channelwise, where each frequency band is half-resolution of the input. The high-frequency components are skip-connected to the decoder where the spatial resolution needs to be upscaled by a factor of two. Taking the layer idwt_4 in Figure 1 as an example, the input to this layer are four components of spatial resolution $1/32$ to perform iDWT. The pyramid output serve the low-low frequency $\widetilde{Y}^{ll}$, while the output of the dwt4 layer operating on the conv4 provide the three high-frequency components Y^{lh}, Y^{hl}, and Y^{hh}. With iDWT, the spatial resolution is upscaled to $1/16$. The output of layer idwt4 is finalized by adding the $1/16$ resolution direct output of conv4, which is a standard skip connection commonly used by many state-of-the-art encoder-decoder CNNs to improve the upsampling performance. The iDWT layer can thus be described by
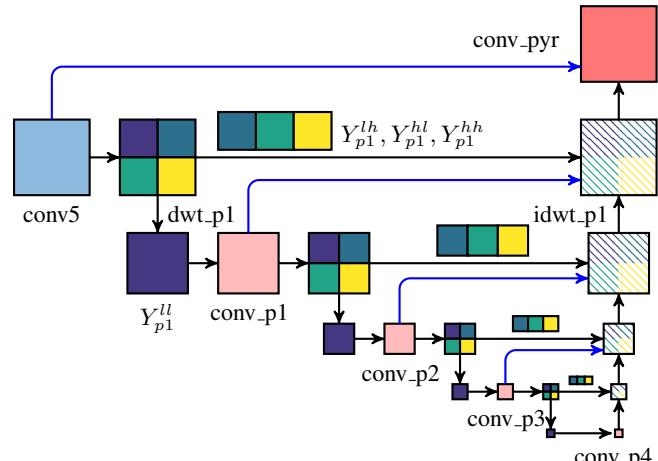
$$G_h^{-1}(\widetilde{Y}^{ll}, Y^{lh}, Y^{hl}, Y^{hh}) + X . \tag{5}$$

We denote this appproach of upscaling the decoder feature map with the wavelet coefficients from the encoder as wavelet unpooling.

Typically, CNNs extract feature with many layers of convolution and nonlinear operations, which transform and

(a) wavelet pyramid variant: low frequency propagation (LFP)

(b) wavelet pyramid variant: full frequency composition (FFC)

Figure 2: The proposed wavelet pyramid variants, with the data flow indicated by black arrows and shortcuts by blue arrows. Both pyramids take conv5 as input and produce conv_pyr as output, without changing the spatial resolution. Both pyramids build a multi-resolution wavelet pyramid via successive DWT. The LFP pyramid only utilizes the low-low frequency Y^{ll}, where each Y^{ll} is filtered by further convolutions, bilinear upsampled to the input resolution and concatenated. The FFC pyramid employs the high frequency bands for upscaling via iDWT.

embed the feature space differently layer by layer. The wavelet unpooling aims to maintain the similar frequency structure throughout CNNs. By replacing the low-frequency of the encoder with the corresponding output of the decoder to perform iDWT with the high-frequency bands from the encoder, the wavelet unpooling aims to enforce learning feature maps of invariant frequency structure under layers of filtering. The skip connections of the signals before DWT also support learning such consistency.

In comparison to the other unpooling methods, for example to upsampling by transposed convolution as proposed in [20], wavelet unpooling does not require any parameters for both DWT and iDWT layers. Compare to the memorized unpooling as proposed in [27], or the method to map the low-resolution feature map to the top-left entry of a 2×2 block [28], the wavelet unpooling aims to restore every

entries according to the frequency structure.

2) Wavelet Pyramid: With CNNs that are designed for classification task, the last few layers typically reduce the spatial resolution to 1×1. Such feature maps have the receptive field of the entire input image and therefore capture the global context. Recent works have demonstrated that capturing global context information is also crucial for accurate dense pixelwise prediction [2], [11]. While it is straight-forward to obtain global context with fully-connected layer or with convolutional layers of large filter size, it is difficult to bridge an encoder with drastically reduced spatial resolution to a proper decoder. Most state-of-the-art CNN encoder reduce the spatial resolution by a factor of 32, which produces 7×7 output given 224×224 input dimensions. If the global context is captured by a simple fully-connected layer, learning 7×7 upsampling kernels is challenging.

One solution is to use the dilated convolutions, which increase the perceptive field with the same amount of parameters [11], [12]. Building on the dilated CNNs, the pyramid spatial pooling network PSPNet [2] introduces a pyramid on the feature map with multiple average pooling of different window sizes. Noticeably, the dilated convolutions demand considerably larger amounts of memory to host the data, which quickly becomes the bottleneck for training with large batch size. In this work, we base our network design on non-dilated CNNs and instead construct the pyramids through wavelet transformations. We propose two wavelet pyramids variants, namely the low frequency propagation (LFP) and the full frequency composition (FFC) as shown in Figure 2.

a) Low-Frequency Propagation Wavelet Pyramid: Shown in Figure 2 (a), the LFP pyramid successively performs DWT on the low-low frequency components Y^{ll}. At each pyramid level, the extracted Y^{ll} component is further transformed with a convolutional layer, which is then bilinear upsampled to the same spatial resolution as the pyramid input, i.e., conv5. We then concatenate these the upsampled feature maps to aggregate the global context that are captured at different scale. This concatenated feature map is combined with the skip-connected conv5 by an elementwise addition, which sis then filtered with a 1×1 convolutional layer to match the channel dimension of the decoder.

With LFP, a multi-resolution wavelet pyramid is constructed, where only the low-low frequency bands of each level are used. The LFP pyramid resembles the pyramid proposed by the PSPNet [2]. In particular, with the Haar wavelet, the low-low frequency is equivalent to the average pooling by a 2×2 window. However, the difference is the PSPNet design average pooling with a multiple heuristic window size, whereas LFP pyramid is strictly performed accordingly to frequency decomposition. Despite the Haar wavelet is used in this work, the LFP pyramid can be easily generalized with other wavelet base functions.

b) Full-Frequency Composition Wavelet Pyramid: The LFP pyramid only uses the low-low frequency bands. In order to make full use of the frequency decomposition, the FFC pyramid is developed. Shown in Figure 2 (b), the FFC pyramid amounts to a small encoder-decoder with wavelet unpooling. Start from the input conv5, DWT is performed to obtain the four frequency bands. The low-low frequency band Y^{ll} is filtered by an additional convolutional layer and the high frequency bands Y^{lh}, Y^{hl}, Y^{hh} are cached for unpooling. The filtered low-low frequency is then further decomposed by DWT into the finer level and the same operation repeats until the finest feature map is obtained. To upscale from the finest level, we again adopt the wavelet unpooling as described by Equation (5). To this end, the iDWT is first performed using the cached high frequency bands, and then the output is further fused with the skip connection. The wavelet unpooling successively restore the spatial resolution to the same as the input to the pyramid. Finally, we skip connect conv5 with the wavelet output by an elementwise addition, and project the global context with a 1×1 convolution to bridge the following decoder. It can be seen that, the FFC pyramid mimic the encoder-decoder design, which naturally reduces the spatial resolution and restore it in the consistent manner with the remaining network.

IV. EVALUATIONS

In this section, we evaluate the proposed WCNN method for the task of semantic image segmentation. To this end, we use the Cityscape benchmark dataset [29] which contains 2,975 training, 500 validation and 1,525 test images that are captured in 50 different cities from a driving car. All the images are densely annotated into 30 commonly observed objects classes occurring in urban street scenes from which 19 classes are used for evaluation. The Cityscape benchmark provides all the images with the same high resolution of 2048×1024. The ground truth for the test images is not publicly available and evaluations on the test set are submitted online[1] for fair comparison.

A. WCNN Configurations

Table I presents the network configurations of the proposed WCNN. We take the state-of-the-art ResNet101 [7] for the encoder. The ResNet101 uses stride-two convolution to reduce spatial resolution. To implement WCNN, we preserve the stride-two convolution layers and insert three DWT layers dwt2, dwt3, dwt4 into the decoder conv2_x, conv3_x, conv4_x, respectively to obtain the frequency bands. At each upscaling stage at the decoder, the corresponding frequency bands are used, then followed by several residual blocks before the next upscaling stage. The last two upscaling stages are implemented as upconvolution, where transposed convolution is first applied to scale up the resolution by a factor of two, then residual blocks are used to further filter the intermediate output. In WCNN, we reply heavily on the residual blocks proposed in ResNet [7], where each block is a stack of three convolutional layers with the second layer of 3×3 for feature extraction and the first and third layers as 1×1 convolutions for feature projection.

---

[1]http://www.cityscapes-dataset.com

Table I: The layer configurations of the proposed WCNN (see Figure 1). The encoder is based on ResNet101 [7]. The resblock is the residual block from ResNet, where $(x, y) \times z$ denotes stacking z blocks of $[(1 \times 1, x), (3 \times 3, x), (1 \times 1, y)]$ convolutional layers. For upconvolution, the transposed convolution is first used to upscale the input by a factor of two, followed by residual blocks. We denote the stride-two operations with s2, and elementwise addition with ⊞. The dimension of the layer output assumes the spatial resolution of input image is normalized to 1, and the second entry denotes the depth of the feature maps.

layer	operation	input	dimension
conv1	$(7 \times 7, 64)$, s2	RGB	1/2, 64
maxpool	(2×2), s2	conv1	1/4, 64
conv2_x	resblock $(64, 256) \times 3$	maxpool	1/4, 256
dwt2	G_h	conv2_x	1/8, 256
conv3_1	resblock $(128, 512)$, s2	conv2_x	1/8, 512
conv3_x	resblock $(128, 512) \times 3$	conv3_1	1/8, 512
dwt3	G_h	conv3_x	1/16, 512
conv4_1	resblock $(256, 1024)$, s2	conv3_x	1/16, 1024
conv4_x	resblock $(256, 1024) \times 22$	conv4_1	1/16, 1024
dwt4	G_h	conv4_x	1/32, 1024
conv5_1	resblock $(512, 2048)$, s2	conv4_x	1/32, 2048
conv5_x	resblock $(512, 2048) \times 2$	conv5_1	1/32, 2048
pyramid		conv5x	1/32, 1024
idwt4	G_h^{-1}	$\begin{cases} \text{pyramid} \\ Y_4^{lh}, Y_4^{hl}, Y_4^{hh} \end{cases}$	1/16, 1024
dconv4_x	resblock $(256, 512) \times 3$	idwt4 ⊞ conv4_x	1/16, 512
idwt3	G_h^{-1}	$\begin{cases} \text{dconv4\_x} \\ Y_3^{lh}, Y_3^{hl}, Y_3^{hh} \end{cases}$	1/8, 512
dconv3_x	resblock $(128, 256) \times 3$	idwt3 ⊞ conv3_x	1/8, 256
idwt2	G_h^{-1}	$\begin{cases} \text{dconv3\_x} \\ Y_2^{lh}, Y_2^{hl}, Y_2^{hh} \end{cases}$	1/4, 256
dconv2_x	resblock $(64, 128) \times 3$	idwt2 ⊞ conv2_x	1/4, 128
upconv2_x	upconv $(64, 64) \times 3$	dconv2_x	1/2, 64
upconv1_x	upconv $(64, 64) \times 2$	upconv2_x	1/1, 64

Table II: The configurations of the proposed LFP and FFC pyramids (see Figure 2). Assuming conv5 has a resolution of $16 \times 32, 2048$, both LFP and FFC pyramids have four levels. For simplicity, the outer two levels are presented in the table, whereas the inner two levels repeats the same patterns. The operator $\star a$ denotes bilinear upsample by a factor of a and the operator ⊞ denotes elementwise addition.

layer	operation	input	dimension
	LFP-pyramid		
dwt_p1	G_h	conv5	$8 \times 16, 2048$
conv_p1	$(1 \times 1, 512)$	Y_{p1}^{ll}	$8 \times 16, 512$
dwt_p2	G_h	Y_{p1}^{ll}	$4 \times 8, 512$
conv_p2	$(1 \times 1, 512)$	Y_{p2}^{ll}	$4 \times 8, 512$
⋮	⋮	⋮	⋮
concat	concatenation	$\begin{cases} Y_{p1}^{ll} \star 2, Y_{p2}^{ll} \star 4 \\ Y_{p3}^{ll} \star 8, Y_{p4}^{ll} \star 16 \end{cases}$	$16 \times 32, 2048$
conv_pyr	$(1 \times 1, 1024)$	concat ⊞ conv5	$16 \times 32, 1024$
	FFP-pyramid		
dwt_p1	G_h	conv5	$8 \times 16, 2048$
conv_p1	$(1 \times 1, 2048)$	Y_{p1}^{ll}	$8 \times 16, 2048$
dwt_p2	G_h	conv_p1	$4 \times 8, 2048$
conv_p2	$(1 \times 1, 2048)$	Y_{p2}^{ll}	$4 \times 8, 2048$
⋮	⋮	⋮	⋮
idwt_p2	G_h^{-1}	$\begin{cases} \text{conv\_p2} ⊞ \text{idwt\_p3} \\ Y_2^{lh}, Y_2^{hl}, Y_2^{hh} \end{cases}$	$8 \times 16, 2048$
idwt_p1	G_h^{-1}	$\begin{cases} \text{conv\_p1} ⊞ \text{idwt\_p2} \\ Y_1^{lh}, Y_1^{hl}, Y_1^{hh} \end{cases}$	$16 \times 32, 2048$
conv_pyr	$(1 \times 1, 1024)$	idwt_p1 ⊞ conv5	$16 \times 32, 1024$

In this work, we develop CNNs for high-resolution predictions. An input image of 512×1024 yields conv5_x to have the spatial resolution of 16×32. Therefore, we design both LFP and FFC pyramids to have four levels of DWT, which produce the four levels of frequency components of 8×16, 4×8, 2×4 and 1×2, respectively. The finest pyramid level thus has the receptive field of the entire input. The details of the LFP and FFC pyramids are given in Table II.

To evaluate the proposed network, the baseline CNN is designed to have minimum difference with WCNN. Taking the WCNN configuration in Table I, the baseline model 1) removes all DWT layers at encoder 2) replaces the pyramid by one $1 \times 1, 1024$ convolutional layer, and 3) replaces the iDWT layers by transposed convolution to upscale the feature map by a factor of 2. The rest layers are the same with WCNN. In the following experiment, we compare the baseline model, the baseline model with LFP and FFC pyramid, the WCNN with LFP and FFC pyramids.

B. Implementation Details

We have implemented all our methods based on the TensorFlow [30] machine learning framework. For network training, we initialize the parameters of the encoder layers from pretrained ResNet model on ImageNet and initialize the convolutional kernels on the decoder with He [31] initialization. We run the training with batch size of four on the Nvidia Titan X GPU. For both training, we minimize the cross-entropy loss using the Stochastic Gradient Descent (SGD) solver with Momentum of 0.9. The initial learning rate is set to 0.001 and decrease with a factor of 0.9 every 10 epoch. We train the network until convergences. For cityscapes, all the variants of our experiments converges around 60K iterations. Following [4], we apply bootstrapping loss minimization for Cityscapes benchmark in order to speed up the training and boost the segmentation accuracy. For all Cityscapes experiments, we fix the threshold of bootstrapping to the top 8192 most difficult pixels per images.

To train all the variants of the baseline and our model, we fix the input to the network to quarter resolution of the original dataset, i.e., 512×1024. For evaluation on the validation dataset, we upsample the output logits bilinear to half of the resolution (to match the network input resolution) and compute the intersection-over-union (IoU) score for each class and on average. We also experiment with test time data augmentation, where we randomly scale the input images and feed them through the network before fuse the score.
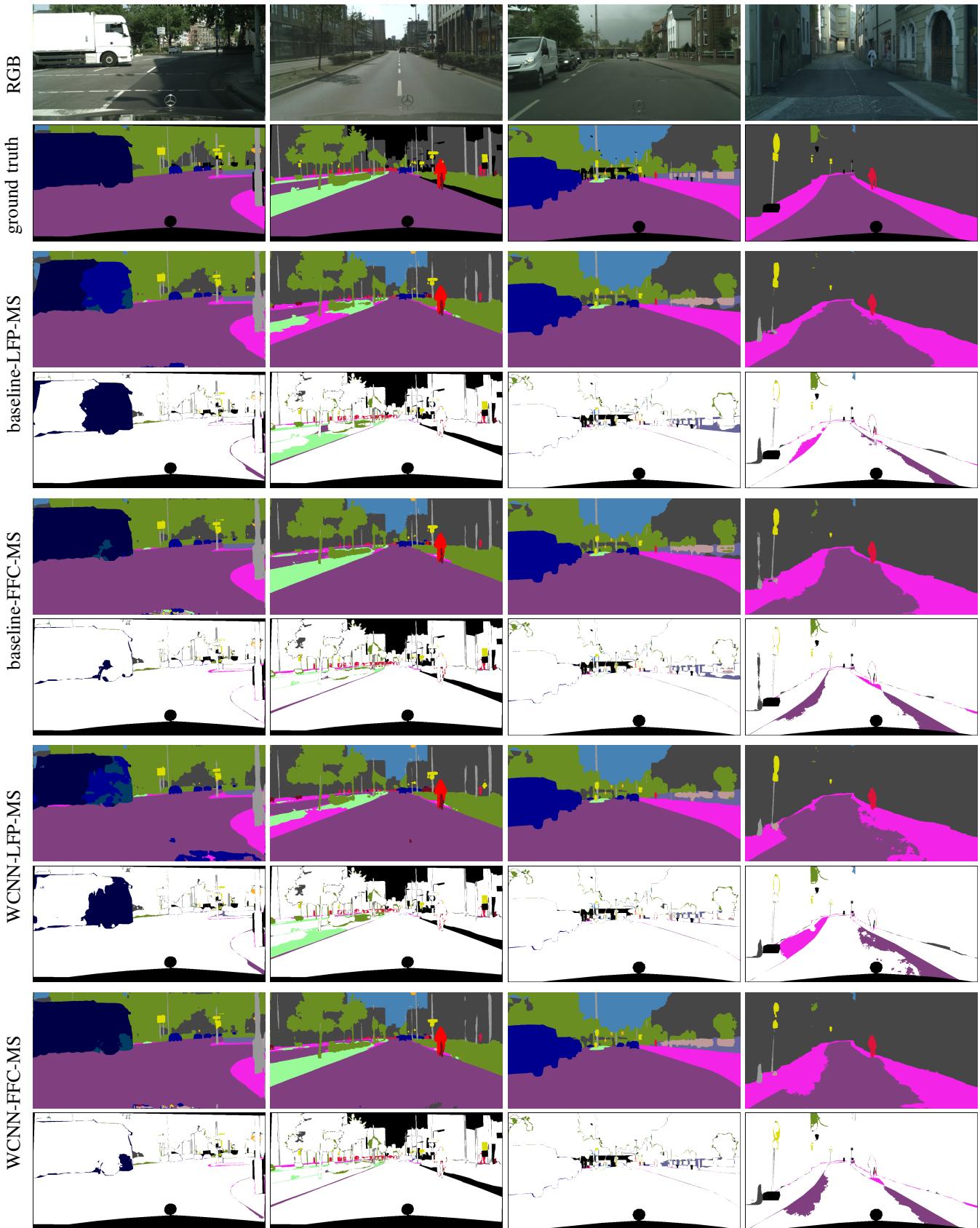
Figure 3: Qualitative exemplary semantic segmentation results on the Cityspaces dataset. From top to bottom: RGB image, ground-truth segmentation, baseline-LFP-MS, baseline-FFC-MS, WCNN-LFP-MS, WCNN-FFC-MS. The semantic color coding is given in Table III.

Table III: Cityscapes 19-class semantic segmentation IoU scores on *val* set. All test results are obtained by comparing to half resolution ground-truth labeling, which is the resolution of input images into our networks. The second part of the table report the performance with multi-scale test time data augmentation, indicated by the MS suffix.

method	road	sidewalk	building	wall	fence	pole	traffic	traffic light	vegetarian	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	avg
frequency	37.7	5.4	21.9	0.7	0.8	1.5	0.2	0.7	17.2	0.8	3.4	1.3	0.2	6.6	0.3	0.4	0.1	0.1	0.7	
baseline	98.8	88.8	96.0	51.5	61.6	62.0	66.6	76.5	96.0	70.1	97.1	85.8	66.4	97.0	81.4	85.4	59.0	53.8	84.6	69.2
baseline-LFP	98.6	90.1	95.5	62.6	62.6	61.3	65.7	76.0	95.9	69.3	97.4	85.4	63.6	97.1	80.1	88.4	73.8	61.2	85.1	71.2
baseline-FFC	98.6	89.6	95.3	63.4	62.0	61.3	67.8	74.4	96.1	64.6	97.3	85.9	63.0	96.9	85.5	89.4	73.6	58.5	84.5	70.7
WCNN-LFP	98.6	89.8	95.7	63.0	65.8	61.5	67.8	76.2	96.3	69.4	97.4	85.8	67.4	97.2	82.0	88.9	69.9	59.9	84.9	71.6
WCNN-FFC	98.7	90.5	95.6	64.8	64.6	63.2	67.8	77.3	96.1	71.0	97.3	86.1	65.3	97.0	82.7	88.7	77.6	57.7	85.1	71.9
baseline-MS	**99.0**	90.6	**96.7**	48.0	61.2	68.2	72.9	80.2	96.3	**72.5**	97.7	89.1	70.3	97.6	76.6	82.2	48.9	60.7	84.9	71.4
baseline-LFP-MS	98.7	92.2	96.5	54.0	65.5	68.9	71.2	79.0	96.1	64.7	97.6	88.1	64.3	**97.8**	71.2	87.3	71.8	**68.5**	85.7	73.3
baseline-FFC-MS	98.7	91.7	96.4	64.6	65.0	67.4	**74.3**	79.7	**96.7**	68.9	**98.0**	88.8	68.9	97.5	**88.3**	**90.6**	**79.3**	60.9	**85.8**	74.7
WCNN-LFP-MS	98.8	**92.4**	96.2	61.2	**68.0**	68.5	71.2	79.8	96.3	64.8	97.5	88.4	**70.1**	**97.8**	77.8	89.3	61.6	74.1	87.1	73.9
WCNN-FFC-MS	98.8	92.2	96.6	**68.6**	64.8	**69.1**	73.9	**81.6**	**96.7**	72.4	97.8	**89.3**	68.9	97.5	87.3	90.5	73.3	58.0	85.3	**75.2**

Table IV: IoU scores for the Cityscapes 19-class and category semantic segmentation on the *test* set (benchmark). All test results are obtained by testing on half resolution and comparing to full resolution groundtruth labeling through upsampling.

method	class mIoU	category mIoU
FRRN [4]	71.8	**88.9**
WCNN-FFC	70.9	86.1
WCNN-FFC-MS	**73.7**	88.3

C. Cityscapes

We evaluate segmentation accuracy using the commonly used evaluation metric of IoU. Table III gives the class-wise IoU and the mean IoU over the 19 classes. It can be seen that adding LFP and FFC pyramids to the baseline network already significantly improves the segmentation performance over the baseline. The FFC pyramid consistently outperforms the LFP pyramid. With WCNN we gain another increase in mean IoU of up to 1.2 over the corresponding baseline. With multi-scale test time augmentation, the accuracy of each model is increased, but the similar rank is observed among the different methods. Our variants strongly benefit, while the combination of wavelet unpooling and FFC wavelet pyramid achieves best increase in performance towards the baseline (6.0 mIoU). These results demonstrate that wavelet unpooling as well as the FFC wavelet pyramid improve the dense prediction of the baseline model. The qualitative comparisons are shown in Figure 3. It can be seen that the WCNN approach recovers fine-detailed structures such as fences, poles or traffic signs with higher accuracy than the baselines.

Table IV compares our method with the current state-of-the-art method FRRN [4] on the same input resolution (2x subsampling) on the Cityscapes benchmark. It can be seen that our method WCNN-FFC-MS outperforms FRRN by 1.9 mean IoU over the 19-classes while it is worse (0.6 mIoU) on the category level. Notably, WCNN is much less memory demanding than FRRN.

V. CONCLUSION

This paper introduce WCNN, a novel encoder-decoder CNN architecture for dense pixelwise prediction. The key innovation is to exploits the discrete wavelet transform (DWT) and inverse DWT to design the unpooling operation. In the proposed network, the high-frequency coefficients extracted by DWT at the encoder stage are cached and later combined with coarse-resolution feature maps at the decoder to perform accurate upsampling and hence, ultimate pixelwise prediction. Further, two wavelet pyramid variants are introduced, i.e., the low frequency propagation (LFP) pyramid and the full frequency composition (FFC) pyramid. Both pyramid extract the global context from the encoder output with multi-resolution wavelet decomposition. Shown in experiment, WCNN outperforms the variant baseline CNNs and achieve the state-of-the-art semantic segmentation performance on the Cityscapes dataset.

In the future work, we will evaluate WCNNs for different dense pixelwise prediction tasks, e.g., depth estimation and optical flow estimation. We will also perform ablation study of the wavelet pyramid to evaluate different pyramid configuration. It is also interesting to extend the WCNN for different wavelet base functions or ultimately learn the optimal base functions with CNNs.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 1097–1105, 2012.

[2] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *CVPR*, July 2017.

[4] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *International Conference on 3D Vision (3DV)*, pp. 239–248, 2016.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[8] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 1520–1528, 2015.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv:1511.00561*, 2016.

[10] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations (ICLR)*, 2016.

[11] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *arXiv:1606.00915*, 2016.

[12] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834–848, 2018.

[13] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[14] P. Krähenbühl and V. Koltun, "Parameter learning and convergent inference for dense random fields," in *International Conference on Machine Learning (ICML)*, pp. 513–521, 2013.

[15] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *International Conference on Computer Vision (ICCV)*, pp. 1529–1537, 2015.

[16] G. Lin, C. Shen, A. van den Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3194–3203, 2016.

[17] J. Yan, Y. Yu, X. Zhu, Z. Lei, and S. Z. Li, "Object detection by labeling superpixels," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[18] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1915–1929, Aug 2013.

[19] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5162–5170, 2015.

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Nov. 2015.

[21] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[22] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[23] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *European Conference on Computer Vision (ECCV)*, pp. 519–534, 2016.

[24] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1377–1385, 2015.

[25] S. Chandra and I. Kokkinos, "Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs," in *European Conference on Computer Vision (ECCV)*, pp. 402–418, 2016.

[26] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 3rd ed., 2009.

[27] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ICCV '11, (Washington, DC, USA), pp. 2018–2025, IEEE Computer Society, 2011.

[28] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox, "Learning to generate chairs with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1538–1546, 2015.

[29] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223, 2016.

[30] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.