PDANet : Pyramid Dual Attention Network for Scene Segmenttion

Guoqiang Liu
ID:21821160
Zhejiang University
lgqfhwy@zju.edu.cn

Abstract

Many of the state-of-art models for semantic Segmenttion are based on self-attention mechanism and multi-scale features fusion. Inspired by [3, 14], we proposed a Pyramid Dual Attention Networks (PDANet) that is combined with pyramid features and dual attention. This network could capture global context information by multi-scale features and capturing rich contextual dependencies based on the self-attention mechanism. In pyramid dual attention module, we turned a feature map to four different pyramid scles features, then each scale feature was processed by self-attention mechanism. In self-attention mechanism, we using dual attention module [3], combined with position attention module and channel attention module. The postion attention module could capturing global features at all positions and channel attention module aggregate each channel features at all channels. The result showed that we achieve state-of-the-art segmentation performance on ade20k [16].

1. Introduction

Scene segmentation is the task of splitting a scene into its various object components. It is a fundamental task in computer vision. Semantic image segmentation predicts dense labels for all pixels in the image, and is regarded as a very important task that could help machine understand scene and objects. It has been actively studied in many recent papers and is also critical for various challenging applications such as autonomous driving, virtual reality, and image editing.

Recently, convolutional networks are driving advances in semantic segmentation. Many state-of-the-art semantic segmentation frameworks based on the fully convolutional network (FCN) [7] have made remarkable progress. Due to the fixed geometric structures, they are inherently limited to local receptive fields and short-range contextual information. Many people realized that long-range dependencies could capture useful contextual information to benefit visual understanding problems.

To address the scene segmentation task, in this paper, we propose a framework called Pyramid Dual Attention Network. Inspired by [3, 14], we intergrate pyramid features and dual attention, which is illustrated in Fig. 1. The details of the dual attention is similar to [3], which is showed in Fig. 2. As showed in Fig. 1, we improved the pyramid pooling module, which proposed in [14], add dual attention module which proposed in [3] into each pyramid scale. The pyramid dual attention module fuses features under four different pyramid scales, and each feature was fed into dual attention module. Dual attention module contains position attention module and channel attention module. It introduces a self-attention mechanism to capture visual features dependencies in the spatial and channel dimensions respectively. Our model with pyramid pooling and attention mechanism just aim to adaptively integrate similar features at different scales, position and channel.

Our main contributions can be summarized as followed:

- Inspired by [14, 3], we propose a Pyramid Dual Attention Network (PDAN) with pyramid pooling and self-attention mechanism to enhance the scene segmentation accuracy.
- We achieve state-of-art results on the popular benchmarks in ade20k[16].
- We implement all the framework where all crucial implementation details are included.

2. Related Work

The last years have seen a renewal of interest on semantic segmentation. With the development of deeplearning, many people recently using deeplearning for semantic segmentation. FCN [7] is the first approach to adopt fully convolution network for semantic segmentation. In FCN [7], theirs key insight is to build "fully convolution" networks that take input of arbitrary size and produce correspondingly-sized output with efficient inference and learning. It is the first work

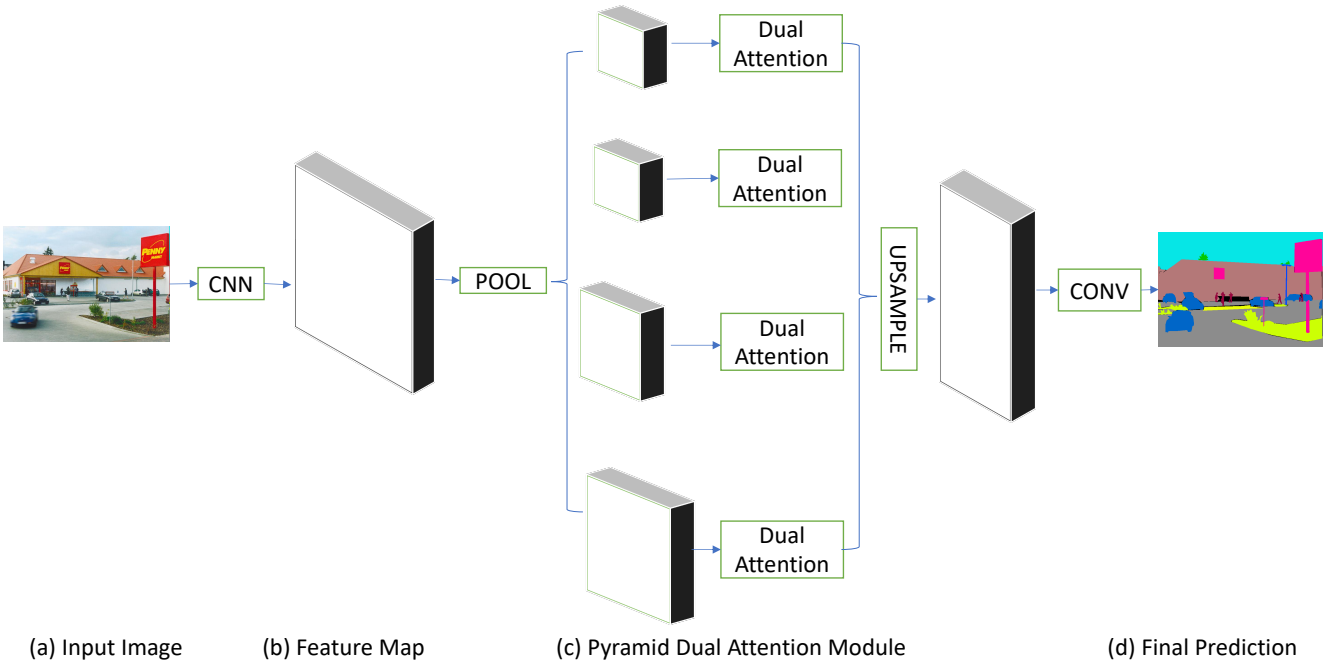


Figure 1. An overview of the Pyramid Dual Attention Network.Given an input image in (a), we first use CNN to get the feature map of the last convolutional layer, the final feature map is $1/8$ of the input image, as shown in (b). then a pyramid dual attention module shown in (c) is applied to harvest different sub-region representations. In Pyramid Dual Attention Module, we first use 4-level pyramid, the pooling kernels cover the whole, half of, and small portions of the image. In each pyramid levels, each features was processed by dual attention module (position attention module and channel attention module), which is shown in Fig. 2. After processed by dual attention module, they was upsampling to similar size then we concatenate the similar size features to form the final feature representation, which carries both local and global context information in (c). Final, the representation is fed into a convolution layer to get the final per-pixel prefiction.

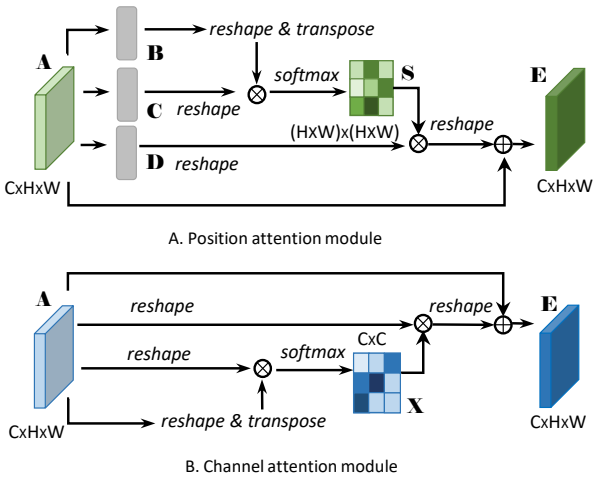
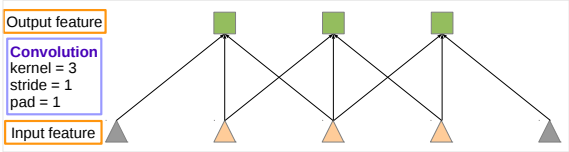


Figure 2. The details of Dual Attention Module, which contains Position Attention Module and Channel Attention Module are illustrated in (A) and (B) [3].

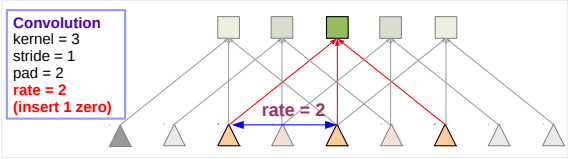
to train FCNs end-to-end number for pixelwise prediction and number from supervised pretraining. It combines layers of the feature hierarchy and refines the spatial precision of the output. FCN [7] combine the final prediction layer with lower layers with finer strides to address that network output is dissatisfyingly coarse. In details, they first divide the output stride in half by predicting from a 16 pixel stride layer. They add a $1/times1$ convolution layer on the top of the pool4 to produce additional class predictions. They fuse this output with the predictions computed on top of conv7 (convolutionalized fc7) at stride 32 by adding a $2/times$ upsampling layer and summing both predictions. Finally, the stride 16 predictions are upsampling back to the image. It is called net FCN-16s. Others such FCN-8s are similar.

To capture long-range dependencies, Chen et al. [2] proposed atrous spatial pyramid pooling module with multi-scale dilation convolutions for contextual information aggregation. As Deeplab [2] said, atrous convolution allows us to explicitly control the resolution at which feature responses are computed with Deep convolutional Neural Networks. It also allow us to effectively enlarge the field of view of filters to incorporate larger context without increasing the number of parameters or the amount of computation.

Atrous convolution advocated by Deeplab [2] originally developed for the efficient computation of the undecimated



(a) Sparse feature extraction



(b) Dense feature extraction

Figure 3. Illustration of atrous convolution in 1-D. (a) Sparse feature extraction with standard convolution on a low resolution input feature map. (b) Dense feature extraction with atrous convolution with rate $r = 2$, applied on a high resolution input feature map.

wavelet transform in the "algorithm 'a trous" scheme of [4] and used before in the Deep CNN context by [8]. This algorithm allows us to compute the responses of any layer at any desirable resolution.

Considering one-dimensioanl signals first, the output $y[i]$ of atrous convolution of 1-D input signal $x[i]$ with a filter $w[k]$ of length K is defined as:

$$y[i] = \sum_{k=1}^{K} x[i + r \cdot k]w[k]. \tag{1}$$

The *rate* parameter r corresponds to the stride with which we sample the input signal. Standard convolution is a special case for rate $r = 1$. See Fig. 3 for illustration.

What's more, deeplab [2] also proposed ASPP (Atrous Spatial Pyramid Pooling), They have experimental with two approaches to handling scale variability in semantic segmentation. The first approach is extract DCNN feature maps from multiple rescaled versions of the original image using parallel DCNN branches that share the same parameters. The second approach is use multiple parallel atrous convolutional layers with different sampling rates. The features extracted from each sampling rate are further processed in separate branches and fused to generate the final result.

To overcome many limitations of short-range CRFs, deeplab [2] intergrated the fully connected CRF model of [6]. The model employs the energy function

$$E(\boldsymbol{x}) = \sum_{i} \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j) \tag{2}$$

where $\boldsymbol{x}$ is the label assignment for pixels. Deeplab [2] use as unary potential $\theta_i(x_i) = -\log P(x_i)$, where $P(x_i)$ is the label assignment probability at pixel i as computed by a DCNN. The pairwise potential has a form that allows for

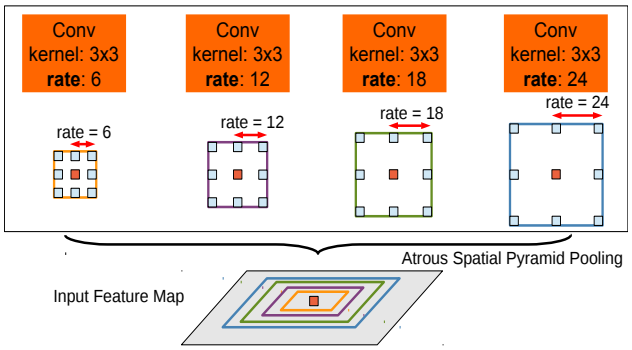


Figure 4. Atrous Spatial Pyramid Pooling (ASPP). To classify the center pixel (orange), ASPP exploits multi-scale features by employing multiple parallel filters with different rates. The effective Field-Of-Views are shown in different colors.

efficient inference while using a fully-connected graph, i.e. when connecting all pairs of image pixels, i, j. In particular, as in [6], deeplab [2] use the following expression:

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \left[w_1 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\alpha^2} - \frac{||I_i - I_j||^2}{2\sigma_\beta^2} \right) + w_2 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\gamma^2} \right) \right] \tag{3}$$

where $\mu(x_i, x_j) = 1 \text{if} x_i \neq x_j$, and zero otherwise, which, as in the Potts model, means that only nodes with distinct labels are penalized.

SegNet [1] is a deep convolution encoder-decoder architecture for image segmentation. It consists of an encoder network, a corresponding decoder network followed by a pixel-wise classification layer. In SegNet [1], each encoder layer has a corresponding decoder layer and hence the decoder network has similar layers with encoder. The final decoder output is fed to a multi-class soft-max classifier to produce class probabilities for each pixel independently. Each encoder in the SegNet [1] encoder network performs convolution with a filter bank to produce a set of feature maps. These are then batch normalized [5]. Then an element-wise rectified-linear non-linearity (ReLU) $max(0, x)$ is applied. Following that, max-pooling with a 2×2 window and stride 2 (non-overlapping window) is performed and the resulting output is sub-sampled by a factor of 2. The appropriate decoder in the decoder network upsamples its input feature maps using the memorized max-pooling indices from the corresponding encoder feature maps.

Yu et al. [11] presented a module used dilated convolutions to systematically aggregate multi-scale contextual information without losing resolution. The architecture is based on the fact that dialted convolutions support exponential expansion of the receptive field without loss of resolution or coverage. 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 [4, 10]. 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

$$(F * k)(p) = \sum_{s+t=p} F(s)k(t). \tag{4}$$

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

$$(F *_l k)(p) = \sum_{s+lt=p} F(s)k(t). \tag{5}$$

In the equation, we refer to $*_l$ as a dilated convolution or an l-dilated convolution. The familiar discrete convolution $*$ is simply the 1-dilated convolution. Yu et al. [11] showed that the dilated convolution operator is particular suited to dense prediction due to its ability to expand the receptive field without losing resolution or coverage. They 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.

3. Pyramid Dual Attention Network

In this section, we first present a general framework of our netwerk and then introduce in detail the pyramid pooling module and two attention modules. Finally we describe how to aggregate them together for further refinement.

3.1. Overview

Given an input image in Fig. 1(a), we first use CNN to get the feature map of the last convolutional layer, the final feature map is $1/8$ of the input image, as shown in Fig. 1(b). then a pyramid dual attention module shown in Fig. 1(c) is applied to harvest different sub-region representations. In Pyramid Dual Attention Module, we first use 4-level pyramid, the pooling kernels cover the whole, half of, and small portions of the image. In each pyramid levels, each features was processed by dual attention module (position attention module and channel attention position), which is shown in Fig. 2. After processed by dual attention module, they was upsampling to similar size then we concatenate the similar size features to form the final feature representation, which carries both local and global context information in Fig. 1(c). Final, the representation is fed into a convolution layer to get the final per-pixel prefiction. The Pyramid Dual Attention Module actually consists of pyramid pooling module, position attention module and channel attention module. We will introduce their details separately.

3.2. Pyramid Pooling Module

The pyramid pooling module fuses features under four different pyramid scales. The pyramid level seperates the feature map into different sub-regions and forms pooled representation for different locations. The output of different levels in the pyramid pooling module contains the feature map with varied sizes. Each feature in different level was sent to dual attention module, which don't change the size of feature.Then we directly upsample the low-dimension feature maps via bilinear interpolation. Finally, different levels of features are concatenated as the final pyramid pooling global feature.

3.3. Position Attention Module [3]

In order to model rich contextual relationships over local features, we introduce a position attention module. The position attention module encodes a wider range of contextual information into local features, thus enhancing their representation capability. Next, we elaborate the process to adaptively aggregate spatial contexts.

As illustrated in Figure.2(A), given a local feature $\mathbf{A} \in \mathbb{R}^{C\times H\times W}$, we first feed it into a convolution layers to generate two new feature maps $\mathbf{B}$ and $\mathbf{C}$, respectively, where $\{\mathbf{B}, \mathbf{C}\} \in \mathbb{R}^{C\times H\times W}$. Then we reshape them to $\mathbb{R}^{C\times N}$, where $N = H \times W$ is the number of pixels. After that we perform a matrix multiplication between the transpose of $\mathbf{C}$ and $\mathbf{B}$, and apply a softmax layer to calculate the spatial attention map $\mathbf{S} \in \mathbb{R}^{N\times N}$:

$$s_{ji} = \frac{exp(B_i \cdot C_j)}{\sum_{i=1}^{N} exp(B_i \cdot C_j)} \tag{6}$$

where s_{ji} measures the i^{th} position's impact on j^{th} position. The more similar feature representations of the two position contributes to greater correlation between them.

Meanwhile, we feed feature $\mathbf{A}$ into a convolution layer to generate a new feature map $\mathbf{D} \in \mathbb{R}^{C\times H\times W}$ and reshape it to $\mathbb{R}^{C\times N}$. Then we perform a matrix multiplication between $\mathbf{D}$ and the transpose of $\mathbf{S}$ and reshape the result to $\mathbb{R}^{C\times H\times W}$. Finally, we multiply it by a scale parameter α and perform a element-wise sum operation with the features $\mathbf{A}$ to obtain the final output $\mathbf{E} \in \mathbb{R}^{C\times H\times W}$ as follows:

$$E_j = \alpha \sum_{i=1}^{N} (s_{ji} D_i) + A_j \tag{7}$$

where α is initialized as 0 and gradually learns to assign more weight [13]. It can be inferred from Equation 7 that the resulting feature $\mathbf{E}$ at each position is a weighted sum of the features across all positions and original features. Therefore, it has a global contextual view and selectively aggregates contexts according to the spatial attention map. The similar semantic features achieve mutual gains, thus imporving intra-class compact and semantic consistency.

Networks	backbone	batch size	imgMaxSize	Pixel accuracy	Mean IoU
FCN [7]	dilated-Resnet18	8	1000	75.61%	0.3305
PSP [14]	dilated-Resnet18	8	1000	77.78%	0.3628
PDANet(ours)	dilated-Resnet18	8	1000	78.44%	0.3768

Table 1. Segmentation results on ADE20K validation set with imgMaxSize is 1000.

Networks	backbone	batch size	imgMaxSize	Pixel accuracy	Mean IoU
FCN [7]	dilated-Resnet18	8	600	74.70%	0.3141
PSP [14]	dilated-Resnet18	8	600	77.59%	0.3575
DANet [3]	dialted-Resnet18	8	600	77.18%	0.3470
PDANet(ours)	dilated-Resnet18	8	600	77.86%	0.3666

Table 2. Segmentation results on ADE20K validation set with imgMaxSize is 600.

3.4. Channel Attention Module [3]

The structure of channel attention module is illustrated in Figure.2(B). Different from the position attention module, we directly calculate the channel attention map $\mathbf{X} \in \mathbb{R}^{C\times C}$ from the original features $\mathbf{A} \in \mathbb{R}^{C\times H\times W}$. Specifically, we reshape $\mathbf{A}$ to $\mathbb{R}^{C\times N}$, and then perform a matrix multiplication between $\mathbf{A}$ and the transpose of $\mathbf{A}$. Finally, we apply a softmax layer to obtain the channel attention map $\mathbf{X} \in \mathbb{R}^{C\times C}$:

$$x_{ji} = \frac{exp(A_i \cdot A_j)}{\sum_{i=1}^{C} exp(A_i \cdot A_j)} \tag{8}$$

where x_{ji} measures the i^{th} channel's impact on the j^{th} channel. In addition, we perform a matrix multiplication between the transpose of $\mathbf{X}$ and $\mathbf{A}$ and reshape their result to $\mathbb{R}^{C\times H\times W}$. Then we multiply the result by a scale parameter β and perform an element-wise sum operation with $\mathbf{A}$ to obtain the final output $\mathbf{E} \in \mathbb{R}^{C\times H\times W}$:

$$E_j = \beta \sum_{i=1}^{C} (x_{ji} A_i) + A_j \tag{9}$$

where β gradually learns a weight from 0. The Equation 9 shows that the final feature of each channel is a weighted sum of the features of all channels and original features, which models the long-range semantic dependencies between feature maps. It helps to boost feature discriminability.

4. Experiments

To evaluate the proposed method, we carry out comprehensive experiments on ADE20k dataset [16]. Experimental results demonstrate that PDANet achives state-of-art performance on ADE20k.

4.1. Datasets and Evaluation Metrics

ADE20K [16] is a scene parsing benchmark containing dense labels of 150 stuff/object categories. There are 20210 images in the training set, 2000 images in the validation set, and 3000 images in the testing set. All the images are exhaustively annotated with objects. Results are reported in two metrics commonly used for semantic segmentation [7]:

- **Pixel accuracy** indicates the proportion of correctly classified pixels;
- **Mean IoU** indicates the intersection-over-union between the predicted and ground-truth pixels, averaged over all the classes.

4.2. Implementation Details

We implement our method based on open source pytorch program : CSAILVision/semantic-segmentation-pytorch [15]. The SGD is used for training. For semantic segmentation, the initial learning rate is 1e-2 for ADE20K. Following prior works [2, 12], we employ a poly learning rate policy where the initial learning rate is multiplied by $1 - (\frac{iter}{max\_iter})^{power}$ with power = 0.9. We use the momentum of 0.9 and a weight decay of 0.0001. We train these network models using the pytorch [9] on NVIDIA Titan Xp GPUs.

4.3. Experiments on ADE20K

We conduct experiments on the ADE20K [16] dataset to verify the generalization of our proposed network. Comparisons with previous state-of-art methods are reported in Table. 1 and Table. 2. Results show that our model achieves 0.3768 in Mean IoU and 78.44% in Pixel Accuracy with image max size is 1000 and it achieves 0.3666 in Mean IoU and 77.86% in Pixel Accuracy with image max size is 600, which outperforms other methods by a large margin.

5. Conclusion and future work

In this paper, we have presented a Pyramid Dual Attention Network (PDANet) for semantic segmentation, which adaptively integrates pyramid pooling features and self-attention mechanism. Our network achieves outstanding performance on ADE20K.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[2] 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. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[3] Jun Fu, Jing Liu, Haijie Tian, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. *arXiv preprint arXiv:1809.02983*, 2018.

[4] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297. Springer, 1990.

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[6] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[7] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[8] George Papandreou, Iasonas Kokkinos, and Pierre-André Savalle. Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 390–399, 2015.

[9] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. *PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration*, 6, 2017.

[10] Mark J Shensa. The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Transactions on signal processing*, 40(10):2464–2482, 1992.

[11] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[12] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7151–7160, 2018.

[13] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[14] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[15] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[16] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019.