U-net and U-net++ Using Stacked Dilated Convolutions for Instance Segmentation at Various Scales

Nolan Vaughn
University of Michigan
Ann Arbor, MI USA
ndvaughn@umich.edu

Abstract

This paper investigates the performance of multiple models with respect to the task of multi-scale segmentation. We first compare an adapted Unet model, dilatedUnet, consisting of stacked dilated convolutions as encoders and decoders to traditional methods of instance segmentation. Furthermore, we propose the network dU-net++ a variation of the Unet++ model proposed by Z. Zhou et al. which uses stacked dilated convolutions as an encoder and decoder. In our studies we have found that dialatedUnet outperforms traditional segmentation models with comparable complexity and while dU-net++ shows promising results, the overall model has some serious performance issues which must be addressed.

1. Introduction

Automated object extraction is becoming increasingly important in many fields of study as a way to increase the amount of information gained from a single image. In medical imagery, current methods of instance segmentation allow for automating the process of tumor detection, and in satellite imagery, quick object segmentation can be useful for many applications such as defense and emergency preparedness.

Over all areas of object segmentation, a very important requirement is being able to detect objects at a variety of scales, while still being able to segment true examples with high precision. For example, in satellite imaging a single image could contain positive examples of large deserts and also positive examples of "small" buildings (small compared to the landscape). In addition, for medical imaging the segmentation of lesions and abnormalities require a much higher precision than what is required in natural images in order to predict a correct diagnosis.

To address the concern for multiscale detection, we will compare the performance of the SDU-net model proposed by Wang et al. [1] to a vanilla U-net architecture with respect to high scale variant classes. We go further by trying to improve the precision on highly detailed instances with the introduction of dU-net++, a model adapted from Z. Zhou et al which incorporates the encoder/decoder blocks introduced in SDU-net.

1.1. Previous Work

While vanilla encoding and decoding architectures such as U-net have paved the way for high level segmentation, there are many shortcomings which limit performance. For example, U-net convolutions have a very limited receptive field reducing the model's ability to detect at varying levels. A common method to introduce larger receptive fields, without increasing complexity, is by adding a "padding" between active kernel nodes during convolution. This process is called a dilated convolution, the one-dimensional variation is described by the formula:

$$(x * w)[i] = \sum_{k=1}^{K} x(i + d \cdot k)w(k)$$

where x*w indicated the dilated convulsion of x and w, and d is the dilation factor (so in this example each active kernel weight is separated from the next weight by a factor of d). While effective in increasing the receptive field, Hamaguchi et al [3] showed that aggressive dilation can result in inability to detect small objects due to the sparsity of the kernel's active weights. To mediate this problem Wang et al. [1] introduce SDU-net which included a new encoding block which concatenates several feature maps formed by dilated convolutions of differing dilation factors, this differs from the vanilla encoding/decoding block which consists of a single convolutional layer. The idea is that by introducing convolutions with large dilation factors we increase the receptive field, but by including convolutions with small dilation factors, we still keep a dense kernel and maintain the ability to pick up on small objects.

One part of the vanilla U-net which is unchanged through the move to SDU-net are skip connections. Skip connections allow structure rich encoder layers to be combined with detail rich decoder layers allowing for feature maps to include as much information as possible. In the proposal of U-net++, Z. Zhou et al. [2] propose a new network of skip connections, under the suspicion that concatenating layers which have large semantic differences

poses a harder task for learning. U-net++ gradually introduces lower level feature maps to the each layer of the skip connections, meaning that during decoding, the decoding output layer is concatenated with a feature map with a semantic level much closer to itself, allowing for an easier learning task. Zhou was able to show that with the addition of a dense network of skip connections, U-net++ was able to perform to a higher level of precision over medical imaging data.

Our proposed network dU-net++ combines the idea's of Wang and Zhou by implement U-net++ with encoding/decoding blocks introduced in SDU-net. Our hypothesis is that the dilated blocks will allow for higher ability to detect scale variant objects while the skip connection network will allow us to detect with high accuracy.

2. Methods

To accomplish our main goal we created a U-net model which incorporated the dilated encoding/decoding layers and also incorporated the dense network of skip connections. As a means of comparison, we first designed a model with dilated encodings and a generic skip connection pathway as well as a "vanilla" U-net which uses no dilated convolutions and generic skip connections.

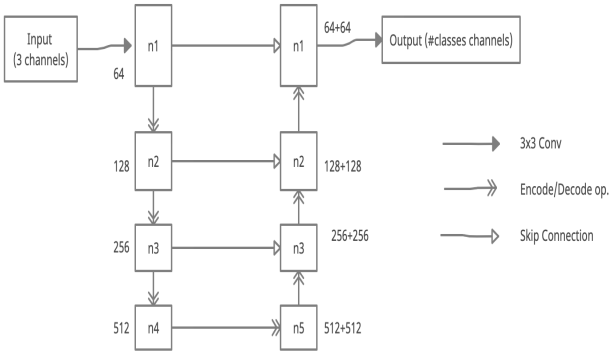

Figure 1: The above diagram is the generic U-net architecture which consists of encoding on the left hand side and decoding on the right hand side. The number of channels at each stage is shown next to the block. Each convolution is a 3x3 kernel. The input dimensions are 256x256 and each encode/decode scales by a factor of 2.

Figure 1 shows the generic architecture of a U-net model. It is important to note that each following iteration starts with the same backbone, and that the generic encode and decode operations are 3x3 convolutions with no dilation factor.

2.1. Dilated encode/decode

In figure 2 we show the encode/decode block introduced by Wang et al. in [1]. Our Dilated model, dilatedUnet, substitutes these encoding and decoding operations into the vanilla U-net design. We note that n_in and n_out are the input and output channels of an operation and are defined in Figure 1. We can see in figure 2 that each operation has

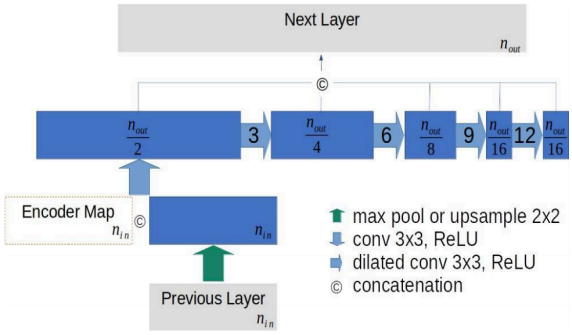

Figure 2: This figure taken from [2] shows the encode operation adapted into SDU-net, we see that the encoder map is connected to decoding layers and that the green arrow is where we down/up sample, all other convolutions should be preserving dimensionality.

several dilation factors allowing us to increase the receptive field. The concatenation applied before passing to the next layer ensures that we do not lose localized info and are still able to detect small objects.

2.2. Skip Connection Network

Figure 3 shows how we implement the skip connection layer in dU-net++. We note that this is slightly adapted from [2], this is because of the limited hardware, in specific we remove one layer of encoding.

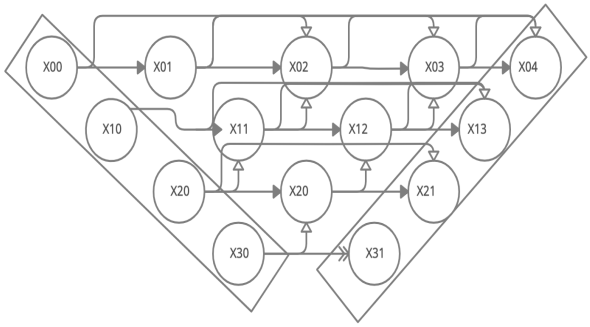

Figure 3: In this image each circle enclosed in a rectangle represent the base encoding/decoding blocks and are all connected implicitly by an encoding/decoding operation. Each arrow represents a concatenation, and each internal node is a convolutional layer with a 3x3 kernel, which does not change channel size.

The network of skip connections allows us to detect more fine detail and implement a more trainable model. The equation below provides a detailing for each node's output.

$$x^{i,j} = \begin{cases} H(x^{i-1,j}) \; if \; j = 0 \\ H\left(\left[\left[x^{i,k}\right]_{k=0}^{j-1}, U(x^{i+1,j-1})\right]\right) \; otherwise \end{cases}$$

H(.) represents a 3x3 convolution, [.] represents concatenation over the channel dimension and U(.) represents an up-sampling of factor 2. We note that all top

layer nodes produce a full feature map and can be used to output predictions. In [2] each of these layers are inputted into the Loss function in order to implement deep supervision. This was excluded from our final model due to hardware limitations. dU-net++ uses this skip connection network and implements U-net with dilated operations as the backbone (everything inside the rectangles in Figure 3).

3. Experiments

In the following sections we will summarize the results from our experiments.

Dataset: The dataset we used to evaluate each model was the Ade20k challenge set. The set consists of 20,000 train images as well as 2000 validation images. During experimentation we were limited by hardware constrains which only allowed us to use around 15,000 training examples. There is a total of 150 object classes of the dataset, as well as the non-object class, giving us 151 classifications.

Baseline Models: Both dilatedUnet and dU-net++ are compared to the vanilla U-net described in section 2. This was the model we implemented in class for problem set 6. We chose U-net as the baseline because it is a common performance baseline for segmentation models.

Model Evaluation: To evaluate the model we chose to use class mean IoU. The Intersection over Union (IoU) measures the size of the intersection of the ground truth blob of class i with the predicted blob of class i, and divides it over the size of the union of these two blobs. The class mean IoU averages IoUs over each class i. In addition we chose Cross Entropy loss as our loss function as it is a common loss function for multiclass segmentation.

Hyperparameters: We experimented with learning rates ranging from 2e-4 to 2e-3, and found that the best learning rate over all models was 3e-4 (although this did not differ much). We chose to evaluate over a batch size of 16 for 10 epochs, however, this decision was mainly made on hardware and time constraints, and we noticed that after 10 epochs, validation IoU was still increasing. We also note, due to the bulkiness of dU-net++, we were only able to train for 2 epochs.

Method of Evaluation: We had two main methods of experimentation, the first method trains segmentation over all 151 classes. The second method trains only over the most common 15 classes. This was done since we were not able to train on the entire Ade20k dataset, but tested on the entire testing dataset, this meant that we could be seeing a disproportionate amount of one class during training as compared to testing. We recorded the average loss and average class mean IoU after every 2 epochs and recorded the best on table 1.

3.1 Results

We have summarized the results in table 1 and have included some images from each testing phase below. We see that dilatedUnet outperforms vanilla unet at every test while dU-net++ falls short at each iteration, however, this is likely due to not being able to train over the same amount of epochs.

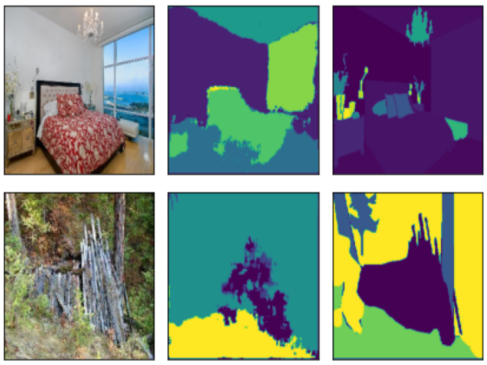

Figure 4: The raw, predicted and ground truth image from vanilla U-net over 151 classes.

All Classes	Loss after 10	IoU

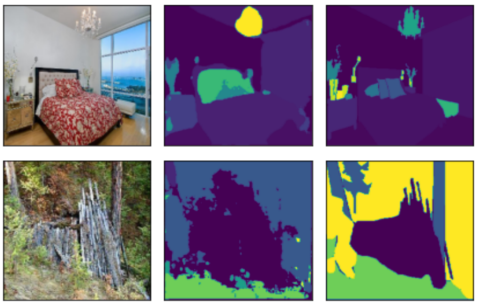

Figure 5: raw, predicted, ground truth mages from dilatedU-net over 151 classes.

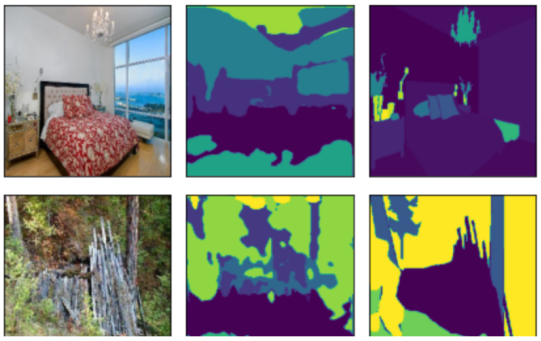

Figure 6: raw, predicted, ground truth mages from dU-net++ over 151 classes

	epochs	
U-net	2.086 ± 0.156	0.1570
dilatedU-net	1.422 ± 0.198	0.3663
dU-net++	3.085 ± 0.223	0.1220
Top 15 classes		
U-net	1.057 ± 0.112	0.4236
dilatedU-net	0.739 ± 0.089	0.5215
dU-net++	0.985 ± 0.125	0.3823

4. Conclusion

As seen from our results, dilatedU-net performs best on a dataset with high scale variance, and while dU-net++ looks promising, the results were inconclusive. Going forward, it is important to look at the amount of trainable parameters each model has. Not only is this a good way to evaluate model complexity, but it is also important to guarantee models have similar parameters when comparing them, so we know that one model is not performing better due to higher a structure.

Furthermore, it is impossible to summarize our results without talking about the huge computational cost associated to dU-net++. The cost comes from a very high number of channels computed at each internal layer in the skip node network. One possible work around is to change the convolutions in the network to decrease the channel numbers in a way similar to how encoding and decoding work for SDU-net, such that each decoding layer is concatenated with no more channels than the corresponding layer in vanilla U-net. This would cap the number of channels at each layer of the net with some constant chosen by the programmer. Hopefully with this change we could see dU-net++ emerge as a viable network.

References

[1] Shuhang Wang and Szu-Yeu Hu and Eugene Cheah and Xiaohong Wang and Jingchao Wang and Lei Chen and Masoud Baikpour and Arinc Ozturk and Qian Li and Shinn-Huey Chou: U-Net Using Stacked Dilated Convolutions for Medical Image Segmentation arvix 2017

[2] Zongwei Zhou and Md Mahfuzur Rahman Siddiquee and Nima Tajbakhsh and Jianming Liang: UNet++: A Nested U-Net Architecture for Medical Image Segmentation arvix 2018

[3] Hamaguchi, R., Fujita, A., Nemoto, K., Imaizumi, T., Hikosaka, S: Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1442–1450. IEEE, Lake Tahoe, NV (2018)