Obesity May Be Bad: Compressed Convolutional Networks for Biomedical Image Segmentation

Editors: Under Review for MIDL 2020

Abstract

Compressed Highly parameterize convolutional neural network (CNN) is essential for practical biomedical image segmentation applications. However, the accuracy of compressed CNN based on current compression methods severely declines. Reducing parameters while keeping enough capacity to store features is crucial to CNN compression. CNN usually doubles its number of channels when down-sampling and halves it when up-sampling. Nevertheless, we found that if convolution layers of the encoder can extract maximum value, making max-pooling only for dimension reduction, in this optimal situation, CNN can remove non-maximum value waiting for eliminating. In this paper, we introduce the Optimum Mimic Backbone (OMB), which can force compressed CNN mimics what the original CNN behaves in optimal situations. OMB was used to redesign U-Net. It maintained, or exceeded in some tasks, the performance of the original U-Net in our experiment while reducing 99.58% parameters. We successfully transplanted OMB to different state-of-the-art variants of U-Net without additional parameter modification. In our experiments, OMB and its compact version got higher IoU scores than other state-of-the-art compression techniques in experiments on four popular, different biomedical image segmentation datasets.

Keywords: Compression, Convolutional Neural Networks, Image Segmentation.

1. Introduction

Using deep CNN for biomedical image segmentation has become a trend. Typically, they implemented variants of fully convolutional networks (FCN) (Long et al., 2015), like (Kamnitsas et al., 2017) or using U-Net (Ronneberger et al., 2015) family (Li et al., 2018) with the same backbone called encoder-decoder. The receptive field of a feature becomes bigger along with a deeper perception of the image when encoding also is known as down-sampling or contraction, and the receptive field of a feature would concentrate along with higher accuracy to recover image when decoding also is known as up-sampling or expansion. These models perform better and better, thanks, in part, to bigger labeled datasets and stronger computation like multiple graphics processing units (GPUs). However, most schools and hospitals are short of computing capability to deal with large-scale datasets by current models. Some researchers are studying the potential to increase intraoperative precision using deep learning (Izady yazdanabadi et al., 2017) who suffering from big and redundant parameters in deep CNNs most. Moreover, biomedical datasets usually contain fewer samples (many images came from a single patient) comparing with life scene images. For tackling overfitting, designed networks should use fewer parameters.

Generally, there are three categories of encoder-decoder backbone CNN compression which are parameter pruning (Lebedev and Lempitsky, 2016), low-rank factorization (Denton et al., 2014), and knowledge distillation (Hinton et al., 2015). Implemented these
Compressed Convolutional Networks for Biomedical Image Segmentation

general methods of the neural network of biomedical image tasks are extensively investigated (Rigamonti et al., 2013). U-Net, as the most widely used architecture for biomedical image segmentation, attracts many researchers to study how to compress and accelerate it for practical applications. Mishra et al. (Mishra et al., 2019) developed CC-Net, an image complexity-guided technique to reconstruct U-Net. Zhou et al. (Zhou et al., 2019) implemented a pruning operator while Weng et al. (Weng et al., 2019) applied the Neural architecture search (NAS) to discover efficient U-Net architectures automatically. Mangalam et al. (Mangalam and Salzamann, 2018) proposed distilled-U-Net, transferred knowledge from big, trained teacher network to small, compressed student network. However, these techniques usually compress U-Net at the expense of decreasing the performance of U-Net, which is not acceptable to biomedical image segmentation applications. Besides, many works aim to throw CNN, a big black box, into another small parameters-adjusting black box. It is not only time-consuming and unstable to search optimal parameters in each task, but also severely damage reliability and practicability.

2. Related Work

The architecture of the encoder-decoder backbone has no change for a long time. Therefore, increasing amounts of features extracted by CNN, like ResNet (He et al., 2016) or improving the use ratio of features like U-Net driving the improvement of CNN. Researchers study features in two aspects, cross-layer or cross-channels:

- Many achievements in blending different level feature cross layers. FCN and U-Net build connections of same level features between encoder and decoder; DenseNet (Huang et al., 2017) achieved connections between each layer but split to several blocks and dropout middle blocks with defaulted probability. As for the U-Net family, UNet++ (Zhou et al., 2018) mashed all information cross layers through skip pathways connected to each layer, and MultiResUNet (Ibtehaz and Rahman, 2020) improved the skip pathways of U-Net with residual connections.

- The wider network contains rich information (Shang et al., 2016). Szegedy et al proposed GoogLeNet (Szegedy et al., 2015), which can exploit diversified expression of features in different channels on the same layer. After that, researchers refocused in a wider network and suggested it is better to be wider as long as carefully crafted (Lee et al., 2019).

However, we found that compression can force CNN to use features better. Our method came from studying in the receptive field, which determines how CNN extract features from the image. The convolution changing the receptive field as:

\[
\begin{align*}
    j_{out} &= j_{in} \times s \\
    r_{out} &= r_{in} + (k - 1) \times j_{in}
\end{align*}
\]

(1)

where \(r_{out}\) denotes the size of receptive field of the output feature map. \(j_{in}\) and \(j_{out}\) represent the jump of input and output feature map, respectively. The \(k\), and \(s\) are kernel size, and stride size of the convolution, respectively.

2
Figure 1: Blue matrices represent the input image. Light and ordinary orange matrices denote features extracted from the image after the first convolution layer and max-pooling layer, respectively. The orange, green, yellow, and gray blocks are corresponding to features in different layers. Red and blue bounding boxes indicate ordinary receptive field and optimal receptive field of a feature, respectively. Top right: A convolution in the first convolution layer. Ordinary blue blocks denote features of input image while deep blue block represents the targeted feature corresponds to a significantly high value of the filter. Top left: First contraction layer. Bottom: Receptive field of a feature of the output of each max-pooling layer. (To save space, a $1 \times 1$ block denote $2 \times 2$ area of pixel in the Bottom and $1 \times 1$ area of the pixel in the Top, respectively.)

Figure 2: Blue boxes correspond to feature maps with numbers of channels on the left, map size beneath. Red boxes denote feature maps of valve layers.
3. Method

The original U-Net is an encoder-decoder network with skip pathways for connecting feature maps from the encoder and the feature maps from the decoder. The recovering feature map with low-level features has proved useful for improving biomedical image segmentation accuracy. U-Net family typically use $3 \times 3$ convolution layer with stride $1 \times 1$, $2 \times 2$ max-pooling layer with stride $2 \times 2$, and $1 \times 1$ zero-padding in each contraction layer. Considering the first contraction layer, the receptive field of a feature from $1 \times 1$ to $3 \times 3$ after the convolution layer, and up to $4 \times 4$ after the max-pooling layer like red bounding boxes of the top left of Figure 1. However, when the output of a convolution is approximate to weight the value of the deep blue block. Then the output feature of the convolution as light orange of the top left of Figure 1 is determined by the targeted feature (deep blue block), no matter the value of other features of its receptive field. We define this situation when convolution successfully extracted the most characteristic feature of the receptive field as an optimal situation. We argue in this situation, the $1 \times 1$ targeted feature is governing the receptive field of the output of $3 \times 3$ convolution. Hence the $3 \times 3$ receptive field can approximate to the $1 \times 1$ region, which means the convolution layer does not change the size of the receptive field of a feature. In this case, only max-pooling layers increase $r_{out}$, which equals to $1/4 \ r_{out}$ of the connected contraction layer. The $2 \times 2$ max-pooling takes $1/4 \ r_{out}$ from the region of max value. In the optimal situation, the $r_{out}$ of a contraction layer is equal to $r_{out}$ of the region of the max value of the connected contraction layer, which we argue are beneficial for data learning.

The size of the receptive field of the output of a layer is the number of pixels the layer looks at the input image, and the number of channels is how many times it scans the input image. Hence, the summation of the receptive field of the output of a layer is:

$$N_{\text{receptive fields}} = r_{out} \times N_{\text{channels}}$$  \hspace{1cm} (2)

Therefore, the $r_{out}$ is recast into:

$$4r_{out} = \frac{N_{\text{receptive fields}}}{1/4 N_{\text{channels}}}$$  \hspace{1cm} (3)

When the $N_{\text{channels}}$ is $1/4$ of its original size, the contraction layer needs 4 times the $r_{out}$, which driving the contraction layer finding a region of the max value that can represent a 4 times bigger area. Inside the contraction layer, this structure forcing the convolution layer to extract the targeted features. As illustrated in Figure 2, the first and second contraction layer of OMB mimic the optimal structure of the original U-Net and has $1/4$ the number of channels and $1/4$ the $N_{\text{receptive fields}}$ of the first and second contraction layer of the original U-Net. In the valve layer, the OMB begins to mimic the optimal structure of the Slim-U-Net (with $1/4$ the $N_{\text{channels}}$ of the original U-Net). Even though OMB aims to use fewer the number of receptive fields, the valve layers are necessary because our method works when the convolution layer can extract the targeted features at each layer.

In detail of OMB, each contraction layer contains two $3 \times 3$ convolutions connected with $2 \times 2$ pool size, stride $2 \times 2$ max-pooling. The bottleneck layer has two $3 \times 3$ convolutions. Each expansion layer including $2 \times 2$ deconvolution, concatenation, and a $3 \times 3$ convolution. The convolution activated by a rectified linear unit (ReLU). The decoder is the opposite of
the encoder. However, the $N_{\text{channels}}$ after concatenation of the second expansion layer is 32, which slightly increases parameters. At the second expansion layer, the $N_{\text{channels}}$ shrink to 16 to concatenate with the feature map extracted from the third contraction layer. After concatenate the $N_{\text{channels}}$ is 32. Based on its corresponding contraction layer, it should connect another 16 channels $3 \times 3$ convolution layer. However, the feature map from the second contraction layer has 32 channels, and we do not want the feature map to be smaller than the feature map concatenated.

4. Experiments

To showed the proposed method is generic, we tested on U-Net, and two state-of-the-art variants of U-Net, Unet++ and MultiResUNet. Both have a five-layer deep encoder and a corresponding five-layer deep decoder, and the starting channel depth is 32. All networks trained in BCE-Dice loss (the combination of binary-cross-entropy and Dice loss):

$$L = -\frac{1}{N} \sum_{k=1}^{N} \left( y_k \log + (1 - y_k) \log (1 - t_k) + \frac{2y_k \cdot t_k}{y_k + t_k} \right)$$

in which $y_k$ predicted by the network, $t_k$ denotes ground truth, $N$ indicates the batch size. BCE-Dice loss is the most common loss function used for the U-Net family in biomedical image segmentation.

We adopt He-Normal initializer, ADAM optimizer with the learning rate 1e-3, momentum 0.8, and batch-normalization after each convolution and before activation. Weight decayed 1e-4 if validation loss has not improved after 4 epochs, and training stopped once validation loss has not improved after 10 epochs. We used 0.1 dropout for UNet and UNet++ but not MultiResUNet based on their open-source codes.

Four popular, open-source datasets are used to compare the performance of U-Net, UNet++, MultiResUNet with and without our backbone:

**T1-Brain:** Provided by Brain tumor dataset version 5 (Cheng et al., 2015), it contains 3064 T1-weighted contrast-enhanced images from 233 patients. This dataset has 3 types of brain tumors: meningioma (708 slices), glioma (1426 slices), and pituitary tumor (930 slices). This dataset is for testing OMB in T1-weighted brain images, and data do not balance.

**T2-Brain:** Provided by LGG (lower grade glioma) MRI segmentation dataset (Buda et al., 2019), it contains brain T2 fluid-attenuated inversion recovery (FLAIR) magnetic resonance images, 523 slices from 110 patients together with manual abnormality segmentation masks. This dataset is for testing OMB in T2-weighted brain images.

**Skin:** Provided by ISIC challenge 2017 part 1 (Codella et al., 2018), it contains 3 classes, 2000 skin lesion images, paired with the expert manual tracing of the lesion boundaries in the form of a binary mask. This dataset is for testing OMB in different body parts.

**Cell:** Provided by ISBI challenge 2012 (Arganda-Carreras et al., 2015), it contains 30 cell images from electron microscopy and corresponding labels. This dataset is for testing OMB in a different type of biomedical image.

We used different pre-processed in different datasets, followed each practical standard. For the T1-Brain train set and T2-Brain dataset, we resized the image to 128,128, while
Compressed Convolutional Networks for Biomedical Image Segmentation

resized the Skin dataset to 192,256. The Cell dataset has no resized. For the T1-Brain train set and the Cell train set, we used the flip upped size from 2451 to 4902 and 30 to 60, respectively. For the T2-Brain train set, we used the zooming and flip upped train size from 523 to 6285. We have not augmented the Skin train set.

In all tables, prefix slim denotes U-Net family with 1/4 the number of channels, distilled, and CC indicated another two compression methods, Distill-U-Net and CC-U-Net. To test the number of channels of OMB is better than other compression methods, we have test two versions of OMB. OMB with prefix small represents compact OMB, which has halved the number of channels of OMB and has similar parameters with another two compression methods. We use three standards to evaluate the performance of each method: parameters, Intersection over Union (IoU), and time. The overall reduction is \( R(\%) = 1 - \frac{\text{compressed}}{\text{base}} \) and shown together with the parameters of OMB networks. The base networks are networks without prefix OMB.

Table 1: Segmentation results of U-Net with different backbones.

<table>
<thead>
<tr>
<th>Method</th>
<th>Params</th>
<th>Segmentation dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>T1-Brain (%)</td>
</tr>
<tr>
<td>U-Net</td>
<td>34,514,113</td>
<td>65.13±2.60</td>
</tr>
<tr>
<td>Slim-U-Net</td>
<td>2,158,897</td>
<td>66.13±1.22</td>
</tr>
<tr>
<td>Distilled-U-Net</td>
<td>33,991</td>
<td>43.75±3.81</td>
</tr>
<tr>
<td>CC-U-Net</td>
<td>35,625</td>
<td>50.26±3.01</td>
</tr>
<tr>
<td>Small-OMB-U-Net</td>
<td>37,313</td>
<td>59.16±2.78</td>
</tr>
<tr>
<td>OMB-U-Net</td>
<td>148,353(99.58%)</td>
<td>65.39±2.62</td>
</tr>
</tbody>
</table>

The Table 1 showed the OMB and the Small-OMB got similar IoU scores with the U-Net with original backbone and Slim-backbone. And the other two compression methods, Distilled-backbone and CC-backbone showed the problem of lacking capacity in two brain datasets. We argue because they reduce the number of parameters without improvement of the original U-Net. The OMB has fewer feature maps in some layer compared with the other two compressed methods, but the results in four datasets, giving added weight to our assumption that OMB can extract targeted features from images. The comparison with the Small-OMB and the other two compression methods indicates it is driving by the structure of the OMB. According to (3), the structure of the OMB can force itself to find the representative area and the targeted feature. Hence, the OMB has relatively less redundant features and uses parameters more efficiently compared with the original backbone.

The Table 2 showed OMB performs better in U-Net++ than in MultiResUNet. We selected these two state-of-the-art architectures because they have different characters. The U-Net++ increases the number of skip pathways, which means it improves the original U-Net by expanding the capacity. As for the MultiResUNet, it replaces the skip pathways with residual connections, which may conflict with the OMB. In the Conclusions part, we give more details about this problem. The other two compression methods in two state-of-the-art architectures showed better results than in original U-Net, indicating that they lack capacity when implemented in the original U-Net and eased by two state-of-the-art architectures.

The T2-Brain dataset has 1/4 the amount of data of the T1-Brain dataset. In the T1-Brain dataset, the Small-OMB performed worst in MultiResUNet and got 87.1% of the highest IoU score. In the T2-Brain dataset, the Small-OMB performed worst in the
Table 2: Segmentation results of U-Net++ and MultiResUNet with different backbones.

<table>
<thead>
<tr>
<th>Method</th>
<th>Params</th>
<th>T1-Brain (%)</th>
<th>T2-Brain (%)</th>
<th>Skin (%)</th>
<th>Cell (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-Net++</td>
<td>9,042,561</td>
<td>65.88±1.26</td>
<td>75.10±1.63</td>
<td>81.54±1.04</td>
<td>92.03±0.38</td>
</tr>
<tr>
<td>Slim-U-Net++</td>
<td>2,262,081</td>
<td>65.98±1.71</td>
<td>75.65±0.85</td>
<td>82.01±0.53</td>
<td>91.70±0.35</td>
</tr>
<tr>
<td>Distilled-U-Net++</td>
<td>35,661</td>
<td>45.95±6.01</td>
<td>59.41±8.13</td>
<td>77.37±3.27</td>
<td>90.62±0.30</td>
</tr>
<tr>
<td>CC-U-Net++</td>
<td>71,927</td>
<td>59.91±3.77</td>
<td>69.66±3.96</td>
<td>77.92±1.14</td>
<td>91.42±0.28</td>
</tr>
<tr>
<td>Small-OMB-U-Net++</td>
<td>69,633</td>
<td>60.96±5.37</td>
<td>71.87±2.48</td>
<td>80.09±1.67</td>
<td>91.62±0.18</td>
</tr>
<tr>
<td>OMB-U-Net++</td>
<td>277,249(96.96%)</td>
<td>66.56±2.28</td>
<td>75.85±1.04</td>
<td>81.95±0.81</td>
<td>92.27±0.29</td>
</tr>
<tr>
<td>MultiResUNet</td>
<td>7,238,228</td>
<td>66.46±2.12</td>
<td>77.12±0.32</td>
<td>80.74±0.94</td>
<td>85.97±3.58</td>
</tr>
<tr>
<td>Slim-MultiResUNet</td>
<td>1,801,561</td>
<td>65.63±1.55</td>
<td>76.40±0.73</td>
<td>78.73±1.74</td>
<td>88.82±2.39</td>
</tr>
<tr>
<td>Distilled-MultiResUNet</td>
<td>27,061</td>
<td>51.49±4.35</td>
<td>60.45±2.92</td>
<td>75.83±2.32</td>
<td>85.90±4.07</td>
</tr>
<tr>
<td>CC-MultiResUNet</td>
<td>31,911</td>
<td>48.69±4.40</td>
<td>62.28±3.14</td>
<td>71.44±2.10</td>
<td>85.48±5.38</td>
</tr>
<tr>
<td>Small-OMB-MultiResUNet</td>
<td>38,741</td>
<td>53.90±5.88</td>
<td>71.54±1.82</td>
<td>75.78±2.12</td>
<td>88.57±2.89</td>
</tr>
<tr>
<td>OMB-MultiResUNet</td>
<td>155,735(97.85%)</td>
<td>65.11±1.90</td>
<td>76.23±0.91</td>
<td>79.24±1.58</td>
<td>89.61±1.92</td>
</tr>
</tbody>
</table>

Original UNet but got 94.7% of the highest IoU score. Besides, the original backbone got the highest IoU score in each dataset when the Small-OMB performs worst. We argue the OMB alleviates the overfitting caused by highly parameterized U-Net trained in small size sample tasks, but the Small-OMB has not enough capacity.

Table 3: Result of training time per epoch.

<table>
<thead>
<tr>
<th>Method</th>
<th>Segmentation dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T1-Brain (%)</td>
</tr>
<tr>
<td>U-Net</td>
<td>1225ms</td>
</tr>
<tr>
<td>OMB-U-Net</td>
<td>900ms</td>
</tr>
<tr>
<td>U-Net++</td>
<td>919ms</td>
</tr>
<tr>
<td>OMB-U-Net++</td>
<td>613ms</td>
</tr>
<tr>
<td>MultiResUNet</td>
<td>1532ms</td>
</tr>
<tr>
<td>OMB-MultiResUNet</td>
<td>706ms</td>
</tr>
</tbody>
</table>

We trained in Keras with NVIDIA Tesla K80 GPU. The batch size of the Cell dataset is 2, and the other three datasets are 32. The results of training time per epoch are showed in the Table 3. Because of fewer parameters comparing to the original backbone, the U-Net family with OMB uses less time per epoch.

We also tested the OMB in deeper architectures. In the fifth and sixth contraction layers, the OMB mimics the optimal structure of the U-Net with 1/8 the number of channels. In Table 4, the prefix L and LL denote one and two layers deeper, respectively. Because the original U-Net family in deeper architectures did not learn anything in many tasks, we compare OMB with Slim backbone.

In contrast to the original U-Net, the Slim backbone and the OMB continue to improve. Except the OMB-U-Net++ got the highest IoU score and improved 2.5% over U-Net++ in the T1-Brain dataset. The L-Slim-U-Net got the best performance in the Skin and the Cell dataset with an improvement of 2.2% and 0.3%, respectively. And the L-OMB-U-Net++ improve 1.4% over U-Net++ in the T2-Brain dataset. The result in Table 4 showed the OMB is suitable for a network in deeper layers. Although in the last two layers of the encoder of the LL-OMB have 1/64 and 1/16 the size of the feature map of the original
backbone and the Slim-backbone, respectively, the results of our experiments support our assumption in Figure 1 that the OMB can find the area of maximum value.

5. Conclusions

For practical biomedical image segmentation applications, we designed the OMB, which has achieved outstanding results in our experiments. The U-Net family with OMB maintains the performance of the original backbone based on the IoU score while reducing more than 95% parameters. However, further researches are needed:

- The MultiResUNet is an improvement of the ResNet which based on the residual function $F(x) := H(x) - x$ where the $H(x)$ denote the desired underlying mapping. They found that it would be easier for CNN to learn $F(x)+x$. The shortcut connection of ResNet is a $1 \times 1$ convolution, like the convolution layers of OMB, which do not change the receptive field. The character of the shortcut connection implies that OMB can address vanishing/exploding gradients and explain why MultiResUNet with OMB performs weaker than UNet++ with OMB. We need more experiments for testifying the relationship between the OMB and the vanishing/exploding gradients problems.

- This paper showed that the redundant parameters impaired the performance of the original U-Net. In our experiments, the best performances of 3 datasets are achieved by OMB in deeper architectures. Further research can use the OMB to reconstruct other state-of-the-art variants of the original U-Net.

- Except for biomedical image segmentation, researchers can test the OMB on life scene image tasks. Besides, using the concept of the OMB to reconstruct other CNN architectures is worth expecting.

References

Ignacio Arganda-Carreras, Srinivas C Turaga, Daniel R Berger, Dan Cireșan, Alessandro Giusti, Luca M Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi,
Compressed Convolutional Networks for Biomedical Image Segmentation


