НЕЙРОІНФОРМАТИКА ТА ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

NEUROINFORMATICS AND INTELLIGENT SYSTEMS

UDC 004.93

CONVOLUTIONAL NEURAL NETWORK SCALING METHODS IN SEMANTIC SEGMENTATION

Hmyria I. O. – Post-graduate student of the Department of Software Engineering, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Kravets N. S. – PhD, Associate Professor, Associate Professor of the Department of Software Engineering, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

ABSTRACT

Context. Designing a new architecture is difficult and time-consuming process, that in some cases can be replaced by scaling existing model. In this paper we examine convolutional neural network scaling methods and aiming on the development of the method that allows to scale original network that solves segmentation task into more accurate network.

Objective. The goal of the work is to develop a method of scaling a convolutional neural network, that achieve or outperform existing scaling methods, and to verify its effectiveness in solving semantic segmentation task.

Method. The proposed asymmetric method combines advantages of other methods and provides same high accuracy network in the result as combined method and even outperform other methods. The method is developed to be appliable for convolutional neural networks which follows encoder-decoder architecture designed to solve semantic segmentation task. The method is enhancing feature extraction potential of the encoder part, meanwhile preserving decoder part of architecture. Because of its asymmetric nature, proposed method more efficient, since it results in smaller increase of parameters amount.

Results. The proposed method was implemented on U-net architecture that was applied to solve semantic segmentation task. The evaluation of the method as well as other methods was performed on the semantic dataset. The asymmetric scaling method showed its efficiency outperformed or achieved other scaling methods results, meanwhile it has fewer parameters.

Conclusions. Scaling techniques could be beneficial in cases where some extra computational resources are available. The proposed method was evaluated on the solving semantic segmentation task, on which method showed its efficiency. Even though scaling methods improves original network accuracy they highly increase network requirements, which proposed asymmetric method dedicated to decrease. The prospects for further research may include the optimization process and investigation of tradeoff between accuracy gain and resources requirements, as well as a conducting experiment that includes several different architectures.

KEYWORDS: convolutional neural network, scaling method, asymmetric scaling, semantic segmentation, encoder-decoder, image.

ABBREVIATIONS

CNN is a convolutional neural network;

ELU is a Exponential Linear Unit;

ReLU is a Rectified linear unit;

D-Unet is a U-net scaled in depth;

W-Unet is a U-net scaled in width;

R-Unet is a U-net with scaled input image resolution;

WDR-Unet is a U-net scaled in depth, width and with increased image resolution;

AWDR-Unet is a U-net scaled asymmetrically in depth, width and with increased image resolution.

NOMENCLATURE

$F(X)$is a convolutional neural network;

X is an input space(images);

Y is an output space;

x is an input image;

H,W,C is an input image heigh, width and number of channels respectively;

$R^{H\times W\times C}$ is an three dimensional tensor;

softmax is an output layer;

Dec_i is a deconvolutional block, that can contain several layers;

$Conv_i$ is a convolutional block, that contain convolutional and pooling layers;

W',H',C' is a scaled input image heigh, width, channels;

M_a is a a-th scaling method;

F_o is an origin network;

F_a is a CNN scaled with a-th method;

$accuracy(F(X))$ is a accuracy of the network;

$Params(F(X))$ is a parameters amount;

DOI 10.15588/1607-3274-2024-2-6

$accur_b$ is a network accuracy for the model scaled with method b;

$\max(accur_b)$ is a maximum of acquired model accuracies;

e^x is an exponential function of x;

a is an hyperparameter;

p_i is a is the predicted probability for the true class;

γ is a focusing parameter;

α is a weighing factor;

σ is an activation function;

b_i' is the i-th adjusted bias;

X_{scaled} is a scaled input space;

$conv_{scaled}$ is a scaled convolutional layer;

F' is the adjusted number of filters;

W_i' is the i-th adjusted filter;

F_{scaled} is a scaled amount of layers;

$Conv_s^i$ is a i-th scaled convolutional block.

S is a scaling factor for image parameters;

F_s is a scaled network;

$Deconv$ is a deconvolutional part of network;

X_s is a scaled input images;

P_{origin} is an origin amount of parameters;

P_{scaled} is an scaled amount of parameters.

INTRODUCTION

Thankfully to the rapid development of technologies and the equally rapid growth of computing capabilities of computers and their memory capacity, the wide development and use of approaches based on artificial intelligence and digital image processing became possible.

Many researchers have dedicated their work to developing computer vision and image processing systems to solve complex tasks in life scenarios.

Vision systems are widely used in many aspects of real life. In medicine, computer vision plays an important role in imaging and healthcare applications. In autonomous vehicles – identifying and understanding objects in the environment, helping autonomous vehicles navigate safely. Satellite image analysis for land cover classification, monitoring changes in vegetation, urban areas, and more. In the robotic area such systems help to identify and manipulate objects in their environment.

Often creating a new architecture is not available due to different limitations, but in cases where we have some base model and some extra resources available, we can use a scaling technique.

Our goal in this paper is to examine possible network scaling methods and apply them on the convolutional neural network. Study how different approaches impact network accuracy on solving semantic segmentation tasks. Propose and experimentally verify if there is a reason to scale not the whole network symmetrically, but to scale only part of the network.

The object of study is the process of scaling a convolutional neural network for semantic segmentation.

Creating a new convolutional neural network is difficult, iterative, and time-consuming process. The scaling of existing model could be beneficial in cases when we need to achieve better accuracy and don't strictly limit to computational resources. Scaling refers to the practice of increasing the size and complexity of neural networks to improve their performance. It's reasonable because such techniques provide a pathway to building more powerful and expressive models which can solve complex tasks, leverage vast amounts of data, and push the boundaries of performance.

The subject of study is the scaling methods for convolutional neural network model.

The purpose of the work is to increase the accuracy of the convolutional neural network model by scaling its base architecture.

1 PROBLEM STATEMENT

Formally convolutional neural network can be represented as $F(X)=Y$. An input image $x \in X$ is a three-dimensional tensor, $x \in R^{H \times W \times C}$, where dimensions it's image parameters, such as size and channels.

From the architecture perspective model can be represented as:

$$F(X) = soft\max(Dec_i(...(Conv_2(Conv_1(X)))))\ .$$

Deconvolutional blocks placed in deconvolutional part of the network. Such blocks can consist of different layers, but commonly it consists of convolutional layer, concatenation or skip connection and deconvolutional layer. A convolutional (encode) block is consist of convolutional and pooling layers.

Scaling is the process of increasing the size and complexity of neural network by adjusting number of layers and filters, or other parameters, such as image size (W, H). Let's represent scaling method as M so $M_a(F(X^{W \times H \times C})) = F_a(X^{W' \times H' \times C'})$, where F_a – model with i' amount of layers, received after applying M_a method.

We have a restriction that $accuracy(F_a(X)) > accuracy(F(X))$, that means that method should positively impact on received accuracy.

Another limitation we have its number of parameters, in this work we put that limitation as $Params(F_a) \le 2 * Params(F_o)$. Modeling in such way computational restrictions.

The task is to find such M_b which results in better network accuracy:

© Hmyria I. O., Kravets N. S., 2024
DOI 10.15588/1607-3274-2024-2-6

$$\max(accur_b) > \max(accur_a, ..., accur_{a+n}),$$

and using less or equal number of parameters:

$$Params(F_b) \leq Params(F_a).$$

2 REVIEW OF THE LITERATURE

Convolutional neural networks allow to solve various types of computer vision tasks, such as recognition [1], classification and segmentations. They have huge potential in real world task solving from simple image processing to complex image search engine [2]. Segmentation of an image is one of the indispensable tasks in computer vision. This task is comparatively more complicated than other vision tasks as it needs low-level spatial information. Basically, image segmentation can be of two types: semantic segmentation [3] and instance segmentation [4]. In this article we work only with semantic segmentation.

Image segmentation problems have been approached using several classic pre-deep learning techniques, such as sparsity-based methods [5], k-means clustering [6], Support vector machines [7], Random forests [8], ect. The situation has changed radically with the growth of computing power and the development of machine learning methods. The number of neural networks designed for segmentation increased notably [9] [10]. Methods based on encoder-decoder architectures have become a popular approach to semantic segmentation, particularly U-net [11] found wide usage in different studies [12–14].

There are different techniques to increase CNN's accuracy, such as data-centric and network-based. When data-centric methods propose operations on data, to benefit in result efficiency, network-based, such as scaling, offers to modify the network. Historically the most common way is scaling in depth. We can scale networks in different ways. We can scale up or down the depth of the network [15] which means increase or decrease the number of layers. Usually, it results in a more accurate but heavyweight network.

Also, we can scale networks in another dimension – width [16]. In this case we are not increasing the number of layers, but we are multiplying the amount of filters on each layer.

Besides the methods mentioned above there is image resolution scaling method [17] [18]. We are increasing the image's height and width, allowing the network to learn more features increasing its accuracy.

All those techniques have their own disadvantages, but mostly researchers must balance between accuracy profit and resources requirements. Increasing network depth or width inevitably leads to an increasing number of parameters and computations required to train the network. Meanwhile increasing image resolution leads to lower batch size since increasing image size increases memory usage.

The problem is to find which scaling method brings more profit with the same or about the same computational requirements.

We are going to examine all three methods to inspect its influence on the U-net and its benefits when applying to solve semantic segmentation problems. The main criteria is the result network accuracy, but we also will compare its training speed.

3 MATERIALS AND METHODS

U-net received its name because of its architecture that resembles the letter U. This architecture has an innovative design, containing contracting path, or in other words – encoder, which has the purpose of extracting features from the input image. Following this, an expansive path, or decoder, expanding image to initial size to enable accurate pixel-wise segmentation. In this symmetric architecture information flows seamlessly, preserving spatial information, which is beneficial for accurate segmentation. Base U-net architecture is displayed on Fig. 1.

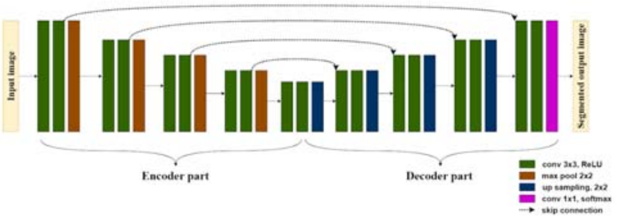

Figure 1 – U-net architecture

The encoder part of the U-Net contains convolutional [19] and pooling layers to systematically reduce the spatial dimensions of the input image. This reduction allows for the extraction of high-level features, creating feature maps. As it moves through consecutive layers, the encoder captures complicated patterns and essential features. During this down sampling process, the U-Net progressively reduces the dimensions of the input image. This down sampling operation involves the use of max-pooling layers, which reduces the information, retaining the most relevant features essential for accurate segmentation.

Unlike the encoder, the decoder section of the U-Net aims to reconstruct the segmented image by expanding the condensed features into the original dimensions. This process is essential to ensure the precise localization of objects within the image and is called up sampling. In this stage, the U-Net uses transposed convolutions, also known as deconvolutions, to reconstruct the segmented image. This method recovers the spatial information lost during the down sampling phase, enabling the network to generate detailed and accurate segmentations.

The basic U-net consists of five blocks in the encoder part and five in decoder. Each encoder block contains consecutive convolutional layers, followed by max-pooling.

DOI 10.15588/1607-3274-2024-2-6

In this article we used U-net as baseline, but made several changes.

In order to reduce the vanishing gradient problem, the activation function was changed from Rectified linear unit (ReLU) to Exponential Linear Unit (ELU). Since ELU mitigates the 'dying ReLU' problem by allowing negative values, which prevents the vanishing gradient issue. And it helps to avoid dead neurons, enhancing the overall robustness of the model during training. ELU activation function could be described as:

$$ELU(x) = f(x) = \begin{cases} x, & x < 0 \\ a(e^{x} - 1), & x \geq 0. \end{cases}$$

Weight initialization technique was used to receive a more robust and stable learning process. As the technique was chosen He Normalization. The advantage of He Normalization lies in its ability to maintain the stability of gradients, allowing the network to train more effectively. By avoiding the vanishing or exploding gradients, it provides a smoother and more consistent learning process. Consequently, this stability leads to enhanced convergence, enabling the model to reach its optimal state efficiently. As a result, the network requires fewer iterations to reach a desired level of accuracy optimizing the training time.

Also we used another loss function called Focal loss [20], a variation of Binary Cross-Entropy, that serves to lower the impact of simpler to learn instances, thereby encouraging the model to concentrate its learning efforts on more complex examples. This specialized loss function demonstrates decent efficiency in scenarios with significant class imbalances. When some classes appear often and some are rarely seen. The desire to use that loss function was mostly derived from the used dataset, as it's highly imbalanced, so in order to increase focus on other classes this loss function was chosen. Focal Loss proposes to focus on hard training examples, downweighing easy to learn examples, using a modular factor, as shown in formula below:

$$FL = \sum_{i=1}^{i=n} \alpha (i - p_i)^{\gamma} \log(p_i) .$$

Here, $\gamma > 0$, but when $\gamma = 1$ this function starts to behave like CrossEntropy loss function. Parameter α usually should be in range [0,1], it can be treated as a hyperparameter, but in our case in order to make this function more data aware modification of inverse class frequency values was used.

In order to receive faster convergence, stability in the learning process, and improved generalization, batch normalization [21] layers were added to the network. They apply a transformation that maintains the mean output close to zero and the output standard deviation close to one, transforming each input in the current mini-batch by subtracting the input mean in the current mini-batch and dividing it by the standard deviation.

Width-wise scaling or expanding Convolutional Neural Networks in width can be beneficial for several reasons. A wider CNN allows the network to capture different features and patterns within the data, improving its accuracy. Besides that, a wider network can better discern finer details in the data due to an enhanced variety of feature maps and activations, which potentially leading to performance improvements, especially in tasks which require specific features extracting. Width-wise scaling refers to adjusting the number of channels increasing number of filters, and can be described as:

$$conv_{scaled}(X) = \sigma(\textstyle\sum_{i=1}^{F'} (X * W_i' + b_i')) .$$

This method can be beneficial, but at the same time wider networks increase computational requirements and need additional memory and processing potential. Besides that, it can lead to overfitting, especially with smaller datasets. As it increases potential for the network to grow overly specialized and less flexible with new or varied data and might compromise its generalization abilities.

The scaling of a CNN depth-wise amplifies its ability to extract complex features and patterns from data, thereby enhancing its representational power. Depth-wise method refers to adjusting number of layers, and can be described as:

$$F_{scaled}(X) = conv_i(..., conv_1(X)) ,$$

where i – is adjusted amount of layer numbers.

By employing deeper architecture, convolutional neural networks (CNNs) efficiently learn hierarchical features, enabling the network to detect compound patterns and characteristics spanning different tiers, thereby amplifying network effectiveness in complex tasks.

This approach also has its disadvantages, like widthwise scaling, its increasing network's depth leading to enlarging complexity, longer time of training and demanding more computational resources. As networks become deeper it increases probability of overfitting, especially on smaller datasets, as they might memorize patterns instead of generalizing from them, decreasing the model's accuracy performing on new data.

Unlike previous methods which are model-based, image resolution scaling is data-based method. The image resolution refers to the amount of detail that an image holds, typically measured in pixels. It is commonly expressed as the width and height of the image in pixels. The resolution determines the clarity and sharpness of the image, with higher resolutions generally providing more detail. Resolution scaling can be described as:

$$X_{scaled} = resize(X^{W \times H \times C}) ,$$

where $X_{scaled} \in R^{W*S \times H*S \times C}$, and S – scaling factor.

DOI 10.15588/1607-3274-2024-2-6

Increasing image resolution can potentially increase network accuracy, since higher image accuracy allows models to learn fine-grained patterns. Commonly in order to increase image resolution we would have to use techniques like interpolation. But since our dataset contains high resolution images it allows us just squeeze images to desired size, which is higher than the original one.

To receive all benefits from the previously described methods we applied all those three methods simultaneously on the same baseline network. But instead of raw multiplying of the baseline network's layer and filters, we construct a modified version of the network displayed on Figure 2.

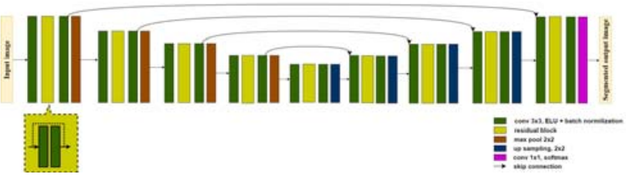

Figure 2 – WDR-Unet architecture

The methods described before applied to whole network, significantly increasing the number of parameters. To use a smaller number of parameters we decided to apply scaling method on U-net architecture asymmetrically, that means applying scaling only on the encoder part of the network. The proposed method can be described as:

$$F_S(X) = Deconv(...,(Conv_S^i(...,Conv_S^1(X_S)))) .$$

Schematic AWDR architecture is displayed on the Figure 3.

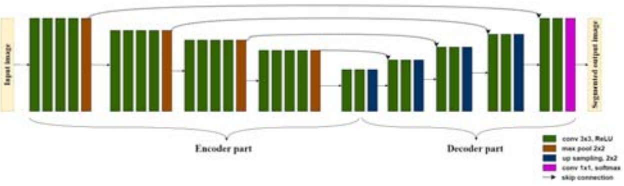

Figure 3 – AWDR-Unet architecture

Encoder part of the network should be scaled using depth-wise and width-wise, besides that input image resolution should be increased. Increasing depth should be performed by adding new convolutional layers in the first half of the network. After that scaling in width should be applied to the encoder part, including new layers.

This concept was inspired by the desire to achieve scaling benefits with lower requirements. Since encoder is responsible for capturing and encoding hierarchical features from the input image, we decided to scale this part. We took a network scaled in all dimensions and modified the decoder part to be the same as in the original baseline. Received network have a smaller number of parameters meanwhile preserves accuracy gain received with combined scaling.

4 EXPERIMENTS

For the training and validation processes was used Cityscapes [22] dataset. It's dataset for semantic understanding of urban street scenes. It provides semantic, instance-wise, and dense pixel annotations for 30 classes grouped into 8 categories: vehicles, humans, constructions, flat surfaces, objects, nature, sky, and void. But for this experiment only semantic information was considered. The dataset consists of around 3475 fine annotated images. Data was captured in 50 cities during several months, daytimes, and good weather conditions. Images were thoroughly selected to have the following features: large number of dynamic objects, varying scene layout, and varying background.

Each image has a size of 2048 x 1024 so we performed image resizing using the nearest neighbor algorithm which provides a sharper result image. To decrease overfitting, we performed data processing which includes random cropping and random flipping.

We applied all five methods described in previous section to baseline. All received networks were trained on the same dataset with the same number of epochs (50).

As the result of width-wise scaling we received architecture that almost did not differ from baseline except for the number of filters in each convolutional layer. The origin amount was multiplied by a defined scale factor. That approach increases the width of the whole network symmetrically. Though the last convolutional layer which has classification purpose remained the same, since the number of classes in the dataset wasn't changed. In the experiment we aim to limit scaling in the way that the number of parameters is increased twice, following established limitation. The received scaling factor was 1.4, since it doubled the parameters amount of the original network.

In depth-wise scaling, to receive deeper network, we extended the baseline architecture with additional layers which were copies of existing layers. Each layer was doubled, so depth of the entire network was doubled. The scaling parameter in this case is two, selected in such a way that the number of parameters in received architecture would not break limitation.

Combined scaling method including all three methods follows previous methods principles, except for different scaling parameters, since in the result of the scaling we need to meet limitations. To increase depth of the network we added several layers, but instead of adding a plain sequence of convolutional layers, we decided to use residual blocks, since the model is quite deep and residual blocks facilitate the stable backpropagation of gradients, reducing the possibility of vanishing or exploding gradients.

5 RESULTS

In the result of the experiment, we received five different networks, which were received after applying scaling methods to the base model.

For each of the method received networks characteristics are shown in Table 1.

© Hmyria I. O., Kravets N. S., 2024
DOI 10.15588/1607-3274-2024-2-6

Table 1 – Scaled networks characteristics

Network	Accuracy, %	Params, M	Speed, ms/step
AWDR-Unet	84.8	55.2	1134
WDR-Unet	84.7	62.8	1585
D-Unet	83.6	62.4	786
W-Unet	82.5	61.3	786
R-Unet	82.9	31.0	910
Baseline	80	31.0	430

As we can see from the table, the best accuracy received AWDR network is 84.8%, approximately the same accuracy as WDR received (84.7%) but has 7.6 M parameters less and faster training speed. Making it a reasonable method of scaling which needs further investigations. In general, methods which combine several methods show better accuracy results (about 1–1.5% better than others), meanwhile have much higher training time.

Increasing model in depth increases the number of parameters by the equation $P_{scaled} = d * P_{origin}$, where d is scaling factor. So, for scaling factor 2 increases number of parameters in twice. For the scaling in width number of parameters can be counted as sum of each layer parameters. The parameters grow is highly depends on the architecture, and in our case increasing each layer's filters number by 1.4 increases overall parameters in twice. Resolution scaling is not increasing parameters amount since it's not affecting architecture. But it has impact on memory usage, we iteratively found that increasing image resolution more than by 1.4 leads to memory overfitting. Using all three methods simultaneously required adjusting scaling factors to meet limitations, so we changed w scaling factor to 1.16, and d to 1.8. Asymmetric method got same scaling factors, but they were applied only to the part of the network.

The computed loss of the five models at each training epoch is shown in Fig. 4.

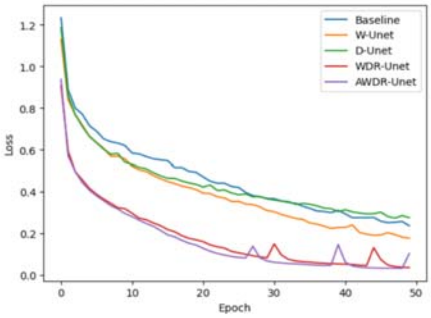

Figure 4 – Training losses of each model

We can observe that WDR and AWDR have lower and approximately equal loss values. From this observation we can confirm that those networks are training better. On the other hand, wider and deeper networks have higher loss values. And decreasing near baseline rate.

On the Fig. 5 we displayed validation loss graphics.

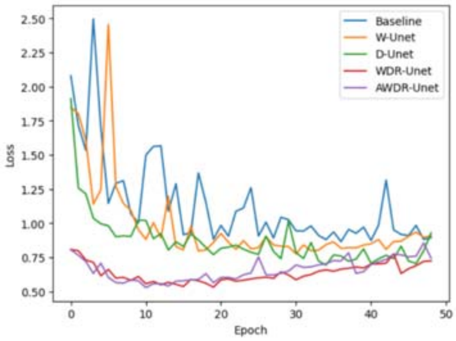

Figure 5 – Validation losses

Here we can see about the same situation as we have with training loss. WDR and AWDR networks have lowest loss values, but here we can see an interesting situation, after the 20-th epoch validation loss value starts to increase. It could be the signal that the model started to learn to perform well on the training data but fails to generalize to new, unseen data. Since those networks are much deeper and complex, they are more vulnerable to overfitting. To fix that in further research we are going to apply more advanced data augmentation techniques to significantly extend the dataset

On Fig. 6 we show accuracy graphic, its display how each model is becoming more accurate with each epoch.

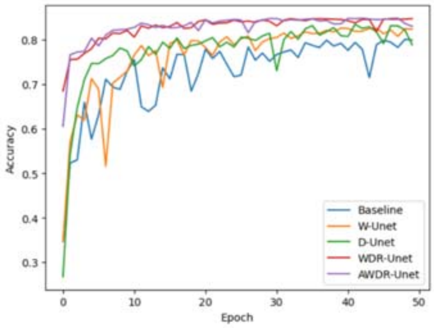

Figure 6 – Accuracy of each model

We can observe from this figure how WDR and AWDR models are converging faster than others, and results in the highest accuracy value. Meanwhile, wider, and deeper networks have second highest values, and have approximately the same values, but deeper networks have slightly better accuracy. Baseline has the lowest accuracy among all networks. Also, it's reasonable to consider using early stopping techniques to stop earlier not perform redundant training and train on lower epochs amount.

6 DISCUSSION

In this study we investigated methods of scaling convolutional neural network and its usage in solving semantic segmentation task. The proposed method allows us to combine benefits from other methods, meanwhile using less parameters, which makes network more accurate.

We conducted a review of literature and explored existing approaches in semantic segmentation task, and existing scaling techniques, which can be applied. Existing

© Hmyria I. O., Kravets N. S., 2024
DOI 10.15588/1607-3274-2024-2-6

scaling methods includes model-based (depth-wise, width-wise) and data-based (resolution scaling). The scaling in depth and width [23] in general improves accuracy, but received accuracy gain with our network and parameters limit reaches about 3%. Image resolution scaling method also allow to achieve accuracy gain, it can reach from 1% to 13% [24] depending on the architecture and dataset, but for our model and parameters limitation its accuracy gain reached 2.9%. Hybrid scaling [25] produces about 4.7% accuracy improvement. The proposed asymmetric method achieves 4.8% accuracy increase, meanwhile produces network with about 12% less parameters.

After it we conducted an experiment with different scaling methods, including model-wise and data-wise scaling. In a result we acquired five architecture modifications with different characteristics. We empirically verified that scaling a CNN is a beneficial approach in cases where computation capabilities are not strictly limited, and some extra resources are available.

Even though scaling in one dimension potentially can lead to accuracy improvements, it’s more efficient to use combined scaling, increasing network architecture in both depth-wise and width-wise ways. Not least important is scaling an input image size. The higher image resolution is the more sophisticated patterns are available for the network to learn. Using proposed method, we received a network architecture that has less parameters while preserving the approximately same high accuracy as existing methods.

The results of the experiment showed that proposed method helps to obtain high-accuracy network, which has high accuracy, but using less parameters than existing methods. However, with benefits from other methods, its also took over the problem of increasing computational requirements and training time, which leaves an open question for further research about tradeoffs between accuracy gain and resources requirements.

CONCLUSIONS

The paper analyzes scaling method for convolutional neural network designed to solve semantic segmentation task.

The scientific novelty of obtained results is the proposed method of asymmetric scaling. This method is appliable to the semantic segmentation models which follows encoder-decoder architecture pattern. It allows to obtain accuracy gain similar to existing methods, but at the same time it uses a smaller number of parameters, that makes it more recourse efficient.

The practical significance of obtained results is that the neural network is trained and validated, that allow method to be used in software development. The experimental results allow to recommend the proposed method for use in practice where semantic segmentation task needs to be solved, it can have potential in the safety area, autonomous driving, and traffic systems.

Prospects for further research are to study the effectiveness of proposed method on other types of networks with different architectures.

ACKNOWLEDGEMENTS

We thank the management of Kharkiv National University of Radioelectronics for the opportunity to conduct scientific research.

REFERENCES

1. Smelyakov K., Chupryna A., Bohomolov O. et al. The Neural Network Models Effectiveness for Face Detection and Face Recognition, *2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Lithuania, 22 April 2021 : proceedings*. Vilnius, IEEE, 2021, pp. 1–7. DOI: 10.1109/estream53087.2021.9431476.
2. Smelyakov K. , Sandrkin D., Ruban I. et al. Search by Image. New Search Engine Service Model, *Problems of Infocommunications. Science and Technology (PIC S&T) : 2018 International Scientific-Practical Conference, Ukraine, 9–12 October 2018 : proceedings*. Kharkiv, IEEE, 2018, pp. 181–186. DOI: 10.1109/infocommst.2018.8632117.
3. Hao S., Zhou Y., Guo Y. A Brief Survey on Semantic Segmentation with Deep Learning, *Neurocomputing*, 2020, Vol. 406, pp. 302–321. DOI: 10.1016/j.neucom.2019.11.118 2020.
4. Hafiz A. M., Bhat G. M. A survey on instance segmentation: state of the art, *International Journal of Multimedia Information Retrieval*, 2020, Vol. 9, No. 3, pp. 171–189. DOI: 10.1007/s13735-020-00195-x.
5. Minaee S., Wang Y. An ADMM Approach to Masked Signal Decomposition Using Subspace Representation, *IEEE Transactions on Image Processing*, 2019, Vol. 28, No. 7, pp. 3192–3204. DOI: 10.1109/tip.2019.2894966.
6. Dhanachandra N., Manglem Khumanthem, Chanu Y. J. Image Segmentation Using K-means Clustering Algorithm and Subtractive Clustering Algorithm, *Procedia Computer Science*, 2015, Vol. 54, pp. 764–771. DOI: 10.1016/j.procs.2015.06.090.
7. Yu Z., Wong H.-S., Wen G. A modified support vector machine and its application to image segmentation, *Image and Vision Computing*, 2011, Vol. 29, No. 1, pp. 29–40. DOI: 10.1016/j.imavis.2010.08.003.
8. Hatami T., Hamghalam M., Reyhani-Galangashi O. et al. A Machine Learning Approach to Brain Tumors Segmentation Using Adaptive Random Forest Algorithm, *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), Iran, 28 February – 1 March 2019 : proceedings*. Tehran, IEEE, 2019, pp. 76–82. DOI: 10.1109/kbei.2019.8735072.
9. Minaee S. , Boykov Y., Porikli F. et al. Image Segmentation Using Deep Learning: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, P. 1. DOI: 10.1109/tpami.2021.3059968.
10. Ulku I., Akagündüz E. A Survey on Deep Learning-based Architectures for Semantic Segmentation on 2D Images, Applied Artificial Intelligence, 2022, pp. 1–45. DOI: 10.1080/08839514.2022.2032924.
11. Ronneberger O., Philipp F., Thomas B. U-Net: Convolutional Networks for Biomedical Image Segmentation, *Lecture Notes in Computer Science*. Cham, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.
12. Huang H., Lin L., Tong R. et al. UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation, *ICASSP 2020 – 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Spain, 4–8 May 2020 : proceedings*. Barcelona, IEEE, 2020, pp. 1055–1059. DOI: 10.1109/icassp40776.2020.9053405.

DOI 10.15588/1607-3274-2024-2-6

13. Cao H., Wang Y., Chen J., et al. Swin-Unet: Unet-Like Pure Transformer for Medical Image Segmentation, *Lecture Notes in Computer Science.* Cham, 2023, pp. 205–218. DOI: 10.1007/978-3-031-25066-8_9.
14. Zhang S., Zhang C. Modified U-Net for plant diseased leaf image segmentation, *Computers and Electronics in Agriculture,* 2023, Vol. 204, P. 107511. DOI: 10.1016/j.compag.2022.107511.
15. Kozal J., Wozniak M. Increasing depth of neural networks for life-long learning, *Information Fusion,* 2023, P. 101829. DOI: 10.1016/j.inffus.2023.101829.
16. Yang G., Hu E. Tensor programs IV: Feature learning in infinite-width neural networks, *International Conference on Machine Learning : 38th International Conference, 18–24 July 2021 : proceedings*. San Diego, PMLR, 2021, pp. 11727–11737.
17. Sabottke C. F., Spieler B. M. The Effect of Image Resolution on Deep Learning in Radiography, *Radiology: Artificial Intelligence,* 2020, Vol. 2, No. 1, P. e190015. DOI: 10.1148/ryai.2019190015.
18. Thambawita V., Strümke I., Hicks S. et al. Impact of Image Resolution on Deep Learning Performance in Endoscopy Image Classification: An Experimental Study Using a Large Dataset of Endoscopic Images, *Diagnostics*, 2021, Vol. 11, No. 12, P. 2183. DOI: 10.3390/diagnostics11122183.
19. Smelyakov K., Shupyliuk M., Martovytskyi V. et al. Efficiency of image convolution, *Advanced Optoelectronics and Lasers (CAOL) : 8th International Conference, Bulgaria, 6–8 September 2019 : proceedings*. Sozopol, IEEE, 2019, pp. 578–583. DOI: 10.1109/caol46282.2019.9019450.
20. Jadon S. A survey of loss functions for semantic segmentation, *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Chile, 27–29 October 2020 : proceedings.* Viña del Mar, IEEE, 2020, pp. 1–7. DOI: 10.1109/cibcb48159.2020.9277638.
21. Furusho Y., Ikeda K. Theoretical analysis of skip connections and batch normalization from generalization and optimization perspectives, *APSIPA Transactions on Signal and Information Processing*, 2020, Vol. 9. DOI: 10.1017/atsip.2020.7.
22. Cordts M., Omran M., Ramos S. et al. The Cityscapes Dataset for Semantic Urban Scene Understanding, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), USA, 27–30 June 2016 : proceedings*. Las Vegas, IEEE, 2016, pp. 3213–3223. DOI: 10.1109/cvpr.2016.350.
23. Lingjiao C., Wang H., Zhao J., et al. The Effect of Network Width on the Performance of Large-batch Training, *Advances in Neural Information Processing Systems 31,* 2018.
24. Jerubbaal J., Rajkumar J., Mahesh B. Impact of image size on accuracy and generalization of convolutional neural networks, *IJRAR*, 2019, Vol. 6, No. 1, pp. 70–80.
25. Guocheng Q., Li Y., Peng H. et al. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies, *Advances in Neural Information Processing Systems 35*, 2022, pp. 23192–23204.

Received 19.02.2024.
Accepted 16.04.2024.

УДК 004.93

МЕТОДИ МАСШТАБУВАННЯ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ СЕМАНТИЧНОЇ СЕГМЕНТАЦІЇ

Гмиря І. О. – аспірант кафедри програмної інженерії, Харківський національний університет радіоелектроніки, Харків, Україна.

Кравець Н. С. – канд. техн. наук, доцент, доцент кафедри програмної інженерії, Харківський національний університет радіоелектроніки, Харків, Україна.

АНОТАЦІЯ

Актуальність. Розробка нової архітектури нейронної мережі є складним і трудомістким процесом, який у деяких випадках може бути замінений масштабуванням існуючої моделі. У цій статті ми розглядаємо методи масштабування згорткової нейронної мережі та прагнемо розробити метод, який дозволяє масштабувати оригінальну мережу, яка вирішує завдання сегментації, у більш точну мережу.

Мета роботи. Метою роботи є розробка методу масштабування згорткової нейронної мережі, який досягає або перевершує існуючі методи масштабування, і перевірити його ефективність у вирішенні задачі семантичної сегментації.

Метод. Запропонований асиметричний метод поєднує в собі переваги інших методів і забезпечує таку ж високу точність мережі в результаті, як і комбінований метод, і навіть перевершує інші методи. Метод розроблено для застосування до згорткових нейронних мереж, які слідують архітектурі кодера-декодера, призначеної для вирішення завдання семантичної сегментації. Метод посилює потенціал виділення ознак що відбувається в частині кодера, водночас зберігає початкову архітектуру частини декодера. Через свою асиметричність запропонований метод більш ефективний, оскільки призводить до меншого приросту кількості параметрів.

Результати. Запропонований метод реалізовано на архітектурі U-net, яка застосовувалася для вирішення задачі семантичної сегментації. Оцінка методу, а також інших методів була виконана на семантичному наборі даних. Метод асиметричного масштабування показав, що його ефективність перевершує або досягає результатів інших методів масштабування, при цьому він є більш ефективний за кількістю параметрів.

Висновки. Методи масштабування можуть бути корисними у випадках, коли доступні додаткові обчислювальні ресурси. Запропонований метод був застосований до згорткової нейронної мережі та оцінювався при вирішенні завдання семантичної сегментації, на якому метод показав свою ефективність. Незважаючи на те, що методи масштабування покращують початкову точність мережі, вони значно підвищують вимоги до мережі, для зменшення яких пропонується асиметричний метод. Перспективи подальших досліджень можуть включати процес оптимізації та дослідження оптимального компромісу між підвищенням точності та вимогами до ресурсів, а також проведення експерименту, який включає кілька різних архітектур.

КЛЮЧОВІ СЛОВА: згорточна нейронна мережа, метод масштабування, асиметричне масштабування, семантична сегментація, кодер-декодер, зображення.

DOI 10.15588/1607-3274-2024-2-6

ЛІТЕРАТУРА

1. The Neural Network Models Effectiveness for Face Detection and Face Recognition / [K. Smelyakov, A. Chupryna, O. Bohomolov et al.] // 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Lithuania, 22 April 2021 : proceedings. – Vilnius : IEEE, 2021. – P. 1–7. DOI: 10.1109/estream53087.2021.9431476.
2. Search by Image. New Search Engine Service Model / [K. Smelyakov, D. Sandrkin, I. Ruban et al.] // Problems of Infocommunications. Science and Technology (PIC S&T) : 2018 International Scientific-Practical Conference, Ukraine, 9–12 October 2018 : proceedings. – Kharkiv : IEEE, 2018. – P. 181–186. DOI: 10.1109/infocommst.2018.8632117.
3. Hao S. A Brief Survey on Semantic Segmentation with Deep Learning / Shijie Hao, Yuan Zhou, Yanrong Guo // Neurocomputing. – 2020. – Vol. 406. – P. 302–321. DOI: 10.1016/j.neucom.2019.11.118 2020.
4. Hafiz A. M. A survey on instance segmentation: state of the art / Abdul Mueed Hafiz, Ghulam Mohiuddin Bhat // International Journal of Multimedia Information Retrieval. – 2020. – Vol. 9, No. 3. – P. 171–189. DOI: 10.1007/s13735-020-00195-x.
5. Minaee S. An ADMM Approach to Masked Signal Decomposition Using Subspace Representation / Shervin Minaee, Yao Wang // IEEE Transactions on Image Processing. – 2019. – Vol. 28, No. 7. – P. 3192–3204. DOI: 10.1109/tip.2019.2894966.
6. Dhanachandra N. Image Segmentation Using K-means Clustering Algorithm and Subtractive Clustering Algorithm / Nameirakpam Dhanachandra, Khumanthem Manglem, Yambem Jina Chanu // Procedia Computer Science. – 2015. – Vol. 54. – P. 764–771. DOI: 10.1016/j.procs.2015.06.090.
7. Yu Z. A modified support vector machine and its application to image segmentation / Zhiwen Yu, Hau-San Wong, Guihua Wen // Image and Vision Computing. – 2011. – Vol. 29, No. 1. – P. 29–40. DOI: 10.1016/j.imavis.2010.08.003.
8. A Machine Learning Approach to Brain Tumors Segmentation Using Adaptive Random Forest Algorithm / [T. Hatami, M. Hamghalam, O. Reyhani-Galangashi et al.] // 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), Iran, 28 February – 1 March 2019 : proceedings. – Tehran : IEEE, 2019. – P. 76–82. DOI: 10.1109/kbei.2019.8735072.
9. Image Segmentation Using Deep Learning: A Survey / [S. Minaee, Y. Boykov, F. Porikli et al.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2021. – P. 1. DOI: 10.1109/tpami.2021.3059968.
10. Ulku I. A Survey on Deep Learning-based Architectures for Semantic Segmentation on 2D Images / Irem Ulku, Erdem Akagündüz // Applied Artificial Intelligence. – 2022. – P. 1–45. DOI: 10.1080/08839514.2022.2032924.
11. Ronneberger O. U-Net: Convolutional Networks for Biomedical Image Segmentation / Olaf Ronneberger, Philipp Fischer, Thomas Brox // Lecture Notes in Computer Science. – Cham, 2015. – P. 234–241. DOI: 10.1007/978-3-319-24574-4_28.
12. UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation / [H. Huang, L. Lin, R. Tong et al.] // ICASSP 2020 – 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Spain, 4–8 May 2020 : proceedings. – Barcelona: IEEE, 2020. – P. 1055–1059. DOI: 10.1109/icassp40776.2020.9053405.
13. Swin-Unet: Unet-Like Pure Transformer for Medical Image Segmentation / [H. Cao, Y. Wang, J. Chen et al.] // Lecture Notes in Computer Science. – Cham, 2023. – P. 205–218. DOI: 10.1007/978-3-031-25066-8_9.
14. Zhang S. Modified U-Net for plant diseased leaf image segmentation / Shanwen Zhang, Chuanlei Zhang // Computers and Electronics in Agriculture. – 2023. – Vol. 204. – P. 107511. DOI: 10.1016/j.compag.2022.107511.
15. Kozal J. Increasing depth of neural networks for life-long learning / Jedrzej Kozal, Michal Wozniak // Information Fusion. – 2023. – P. 101829. DOI: 10.1016/j.inffus.2023.101829.
16. Yang G. Tensor programs IV: Feature learning in infinite-width neural networks / G. Yang, E. Hu // International Conference on Machine Learning : 38th International Conference, 18–24 July 2021 : proceedings. – San Diego : PMLR, 2021. – P. 11727–11737.
17. Sabottke C. F. The Effect of Image Resolution on Deep Learning in Radiography / Carl F. Sabottke, Bradley M. Spieler // Radiology: Artificial Intelligence. – 2020. – Vol. 2, No. 1. – P. e190015. DOI: 10.1148/ryai.2019190015.
18. Impact of Image Resolution on Deep Learning Performance in Endoscopy Image Classification: An Experimental Study Using a Large Dataset of Endoscopic Images / [V. Thambawita, I. Strümke, S. Hicks et al.] // Diagnostics. – 2021. – Vol. 11, No. 12. – P. 2183. DOI: 10.3390/diagnostics11122183.
19. Efficiency of image convolution / [K. Smelyakov, M. Shupyliuk, V. Martovytskyi et al.] // Advanced Optoelectronics and Lasers (CAOL) : 8th International Conference, Bulgaria, 6–8 September 2019 : proceedings. – Sozopol : IEEE, 2019. – P. 578–583. DOI: 10.1109/caol46282.2019.9019450.
20. Jadon S. A survey of loss functions for semantic segmentation / Shruti Jadon // 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Chile, 27–29 October 2020 : proceedings. – Viña del Mar : IEEE, 2020. – P. 1–7. DOI: 10.1109/cibcb48159.2020.9277638.
21. Furusho Y. Theoretical analysis of skip connections and batch normalization from generalization and optimization perspectives / Yasutaka Furusho, Kazushi Ikeda // APSIPA Transactions on Signal and Information Processing. – 2020. – Vol. 9. DOI: 10.1017/atsip.2020.7.
22. The Cityscapes Dataset for Semantic Urban Scene Understanding / [M. Cordts, M. Omran, S. Ramos et al.] // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), USA, 27–30 June 2016 : proceedings. – Las Vegas : IEEE, 2016. – P. 3213–3223. DOI: 10.1109/cvpr.2016.350.
23. The Effect of Network Width on the Performance of Large-batch Training / [C. Lingjiao, H. Wang, J. Zhao et al.] // Advances in Neural Information Processing Systems 31. – 2018.
24. Jerubbaal J. Impact of image size on accuracy and generalization of convolutional neural networks / John Jerubbaal, Joseph Rajkumar, Balaji Mahesh // IJRAR. – 2019. – Vol. 6, No. 1. – P. 70–80.
25. PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies / [Q. Guocheng, Y. Li, H. Peng et al.] // Advances in Neural Information Processing Systems 35. – 2022. – P. 23192–23204.

© Hmyria I. O., Kravets N. S., 2024
DOI 10.15588/1607-3274-2024-2-6