Searching Architecture and Precision for U-net based Image Restoration Tasks

Krishna Teja Chitty-Venkata and Arun K. Somani
Department of Electrical and Computer Engineering
Iowa State University
Ames, IA, USA
{krishnat, arun}@iastate.edu

Sreenivas Kothandaraman
Advanced Rendering Technologies
Intel Corporation
Santa Clara, CA, USA
sreenivas.kothandaraman@intel.com

2021 IEEE International Conference on Image Processing (ICIP) | 978-1-6654-4115-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIP42928.2021.9506777

***Abstract*—Manually architecting Deep Neural Networks (DNNs) has led to the success of Deep Learning in many domains. However, recent DNNs designed using Neural Architecture Search (NAS) have exceeded manually designed architectures and have significantly reduced the human effort to develop complex networks. Current works use NAS to identify a cell architecture constrained by a fixed order of operations that is then replicated throughout the network. The constraints potentially limit the effectiveness of NAS in converging on a more efficient DNN architecture. In the first part of our paper, we propose "Operation Search," a search on an enlarged topological space for U-net and its variants that retain efficiency. The idea is to allow for custom cells (operations and their sequence) at various levels of the network to maximize image quality while being sensitive to computation cost. In the second part of our paper, we propose custom quantization at various levels resulting in a mixed-precision network. Additionally, we increase the search efficiency by constraining the search space to use the same precision for both weights and activations at any level. This does not result in computational inefficiency because it matches the operand precisions supported by Tensor Core enabled GPUs.**

***Index Terms*—Neural Architecture Search, U-net, Image Reconstruction, Mixed Precision**

I. Introduction

Initially, Convolutional Neural Networks (CNNs) [1], [2] were manually designed by domain experts. The trial and error process involved in designing architectures for a problem is time-consuming and labor-intensive. Neural architecture search (NAS), whose goal is to build a Neural Network without significant human intervention, has been very effective in the last few years. NAS [3], [4] works by constructing a search space that contains all possible network configurations for a given backbone network and then searching for an efficient architecture subject to constraints such as accuracy, size, and latency. DARTS [5], which is one of the most successful NAS techniques, starts by constructing and training a supernet of all possible combinations, followed by sampling a path within the supernet to be the final searched network.

The first and important step in the differentiable search process is to pick a backbone/skeleton on which we search for an architecture. We choose U-Net [6], [7], a low compute cost model, as a backbone, which is suited for a variety of applications. The U-net architecture consists of an Encoder side (Down-Sampling Cells) and a Decoder side (Up-Sampling Cells), as shown in Fig. 1. The downsampling cell (left path in Fig. 1) halves the input feature map, whereas an upsampling cell (right path in Fig. 1) doubles the feature map. The decoder cells are fused with the skip connections from the encoder side to improve the ability to segment details.

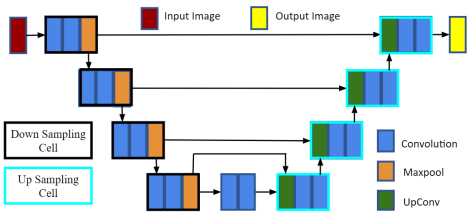


Fig. 1: Unet Model of Depth 4

We design efficient search space for the U-net model, with experiments on Image Restoration tasks such as Super-Resolution and Denoising, and differentiable search as a search strategy [5]. Super-Resolution (SR) [8] is a task to recover High Resolution (HR) images from Low Resolution (LR) ones. Denoising [9] is a task to restore an HR or clean image from a distorted image.

Quantization is a promising solution to solve the difficulties faced by modern CNNs in real-time deployment. Quantization converts the high-precision (floating-point 32) weights to low-precision (Integer 8/4/1), saving memory and compute time. Uniform Quantization [10] is a process of quantizing all the layers in the network to the same precision. Int8/Int4 Quantization of CNN followed by fine-tuning proved to be very successful without significant loss in accuracy [10], [11].

Mixed Precision Quantization [12] strikes a balance between accuracy and low-precision implementation. This enables the optimization of bit widths at the filter level and allows each layer to execute in different precision. The problem then lies in deciding the precision for each layer such that the model is reduced through quantization while maintaining accuracy. For a network consisting of "L" layers and "N" bit widths, there exists N^L different choices. It is very difficult/impossible to try out every possible combination and find the optimal bit-width of each layer. Hence, we need an automatic bit allocation method for solving mixed-precision quantization problem. In the second part of our paper, we work on optimal

bit allocation to different layers in the U-net model using the differentiable search scheme. Few previous works [13], [14] designed differentiable search process for solving the mixed-precision problem for image classification tasks. However, the mixed-precision networks in these search methods are not effective on Tensor Core enabled GPUs, as the weights and activations have different precision. We address this problem by searching for the same precision for weights and activations, as supported by Tensor Cores.

II. Operation Search Process

A. Fundamental Operations

A normal operation, such as a 3*3 convolution of stride 1, does not alter the input feature map size. Depthwise (Depth) convolution [15] reduces the number of parameters and Multiplication and Accumulations (MACs) compared to the traditional convolution. The squeeze and excitation (SE) convolution [16] outperformed the traditional convolution in networks like ResNet50 [17] with little increase in parameter count. We consider Depth and SE convolution in our search space along with the traditional spatial convolution.

Down Sampling and Up Sampling operations halves and doubles the size of the input feature map, respectively. Down/Up Convolution, Down/Up Depth, and Down/Up SE operations can be directly derived from the base Convolution, Depth, and SE operations, respectively, with stride 2. The Up Convolution refers to the transpose convolution, a default operation in the baseline U-net. We consider Max pooling and Average pooling in our Down Sampling search space. We choose parameter-less operations such as Bilinear and Nearest Interpolation in our Up Sampling search space. All the convolution operations have 3*3 kernel size, and a summary of all the operations in the search space is listed in Table I.

TABLE I: Types of Operations

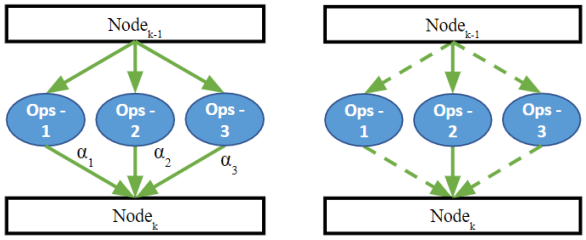
Down Sample OPs	Normal OPs	Up Sample OPs
Down Convolution	Convolution	Up Convolution
Down Depthwise Separable Conv.	Depthwise Separable Convolution	Up Squeeze and Excitation
Down Squeeze and Excitation	Squeeze and Excitation	Bilinear Interpolation
Max Pool	———	Nearest Interpolation
Avg Pool	———	———

B. Over-Parameterized Node

We briefly describe the differentiable method [5], the underlying algorithm in our search process. The key mechanism in this method is to populate every node (operation) in the network with all the operations in the predefined search space, as shown in Fig. 2a. Architecture parameters α (one for each operation) are defined at each node along with the traditional convolution weights. The input feature map at Node_{k-1} (x) is broadcasted to all the edges, and the output Node_k is the softmax weighted (α) sum of all edges. If α_1, α_2, and α_3 correspond to the operations Op-1, Op-2, and Op-3, respectively, the feature map at Node_k is derived as follows:

$$Node_k = \alpha_1 * Op_1(x) + \alpha_2 * Op_2(x) + \alpha_3 * Op_3(x)$$

All the alpha (α) parameters across "N" operations are initialized to the same value (1\N), giving equal weight to all edges. The entire supernet is trained end-to-end using bilevel optimization of θ and α parameters on training and validation datasets, respectively. The final neural architecture from the supernet is derived by sampling one edge at every node which has the highest α value, as shown in Fig. 2b.



(a) Over-parameterized Node (b) Sampled Path

Fig. 2: Differential Architecture Search Process

C. Single Cell-based Operation Search

Traditional NAS algorithms design a complex Down/Up Sampling cell architecture [5], [18] and replicate it throughout the searched network. However, different operations can have a different impact at each level of the network on the overall performance. We resolve this limitation in the cell-search NAS algorithm by our operation search method, which finds distinct operations at each stage of the network. We first describe the construction of an over-parameterized network (Supernet) with all the possible combinations in the operation search method. Fig. 4 shows the sequence of operations in an encoder-decoder (Down Sampling-Up Sampling) pair of U-net backbone. Every operation (normal, down, and up) in this supernet model is populated with its respective operations, predefined in our search space, and respective architecture parameters are initialized, as shown in Fig. 3a. We apply the bi-level optimization of alternatively updating weights and architecture parameters. Towards the end of the search process, one operation with the highest architecture parameter is retained in the final sampled network, as shown in Fig. 3b.

D. Three Cell based Operation Search

The previous Single Cell search consists of a fixed encoder and decoder cells order. The current sequence (Fig. 4) in downsampling and upsampling cells is the most expensive sequence of operation.

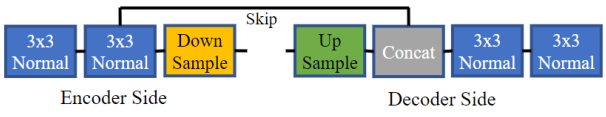
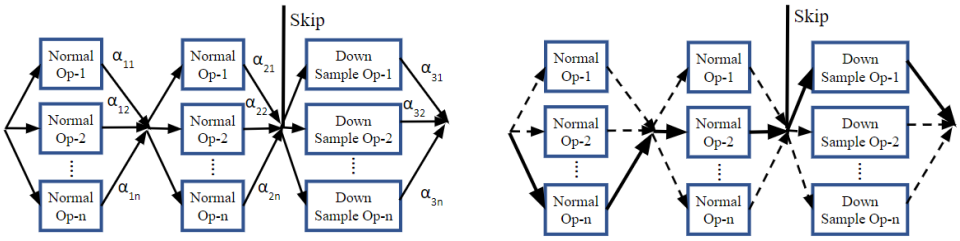


Fig. 4: Encoder-Decoder Pair in Unet Model

However, this sequence can be reordered to increase performance by decreasing the total number of MACs, explained



(a) Over-parameterized Encoder Cell (b) Sampled Path in Encoder Side

Fig. 3: Single Cell Operation Search Process

using the following example. Consider a (3, 3, 64, 128) convolution (Conv1), a (3, 3, 128, 128) convolution (Conv2) and a (2,2) max pool (downsampling) operation. The three operations can be arranged to have three different combinations, which are as follows: 1) Conv1 $\rightarrow$ Conv2 $\rightarrow$ Maxpool 2) Conv1 $\rightarrow$ Maxpool $\rightarrow$ Conv2 3) Maxpool $\rightarrow$ Conv1 $\rightarrow$ Conv2. The total number of weight parameters in the three cell choices are 0.22M. However, the number of MACs in cell choices 1, 2, and 3 are 3.62G, 1.18G, and 0.9G, respectively. This example shows that the order of operation plays an important role in determining the total MACs in the network. The same applies to the decoder cell. The total number of possible combinations for a U-net network of depth 4 with three encoder (2 Normal and 1 Down Sample operations) and three decoder (1 Up Sample and 2 Normal operations) cells are 3^8 = 6561. Hence, we propose a differentiable search process to find the best sequence of operations at every cell level, along with the best operation within each sequence.

This problem can be formulated as a weighted sum of cell outputs, as shown in Fig. 5. Every operation (normal, upsample, and downsample) within each path is populated with its corresponding operation in the search space, as shown in Fig. 2a. Apart from alpha (α) parameters, which find operation at each step, we also define beta (β) parameters (one for each cell choice) at each level to determine the best cell choice out of the three options. The output at each cell level is given as follows:

$$\text{Output} = \beta_1 * \text{Cell}_1 + \beta_2 * \text{Cell}_2 + \beta_3 * \text{Cell}_3$$

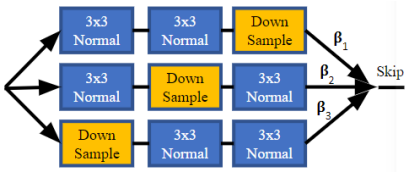


Fig. 5: Three-Cell based Search Process

E. Training and Results

We train the Super-Resolution networks using the DIV2K dataset [19] and infer on Set5 and Set14 test datasets. For Denoising networks, we use BSD400 dataset [20] for training and the BSD68 for inference testing. The loss function used in both the tasks is L_1 loss, and the weights are initialized using the Kaiming uniform initialization method [21]. Adam optimizer [22] is used to update both the weights and architecture parameters. The PSNR, Number of parameters, and MAC results of the two tasks (Super-Resolution (4x) and Denoising (Noise Levels (σ) 5 and 15)) are reported in Tables II and III, respectively. It is evident from the results that Single Cell NAS U-net outperformed Three Cell NAS U-net and baseline U-net in terms of PSNR metric with less number of parameters. However, Three Cell NAS U-net consumes fewer MACs than other networks, with a small drop in PSNR. The total MACs are calculated based on the input image dimension (48, 48, 3) for Super-Resolution and (40, 40, 1) for Denoising.

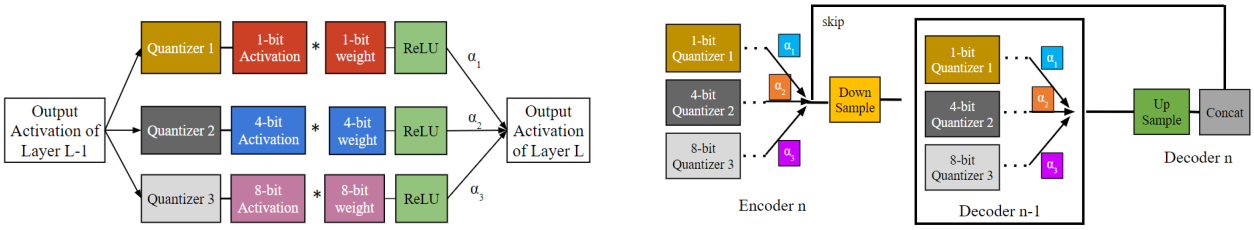
TABLE II: PSNR Metric Comparison

	PSNR		
Test Dataset	Single Cell NAS Unet	Three Cell NAS Unet	Baseline Unet
Set5 (SR 4x)	**31.29**	30.85	29.61
Set14 (SR 4x)	**28.01**	27.66	26.85
DIV2K (SR 4x)	**28.95**	28.19	27.99
BSD68 (Denoising σ 5)	**35.51**	35.43	34.16
BSD68 (Denoising σ 15)	**26.70**	26.64	24.63

TABLE III: Parameters and MACs Comparison

	Single Cell NAS Unet	Three Cell NAS Unet	Baseline Unet
Number of Params. (SR 4x)	**22.12M**	22.19M	35.63M
Number of MACs (SR 4x)	2.86G	**1.98G**	3.4G
Number of Params. (Denoising)	**18.8M**	22.3M	35.5M
Number of MACs (Denoising)	1.38G	**1.19G**	2.4G

Fig. 7 illustrates an example of the three-cell-based searched network on the denoising task. Due to the selection of depth-wise convolution and an efficient sequence of operations, the searched network requires 1.73x fewer MACs than the baseline U-net model. The selection of parameter-less operations (interpolation) contributes to a 1.8x reduction in the number of parameters. Despite the optimization, the inference quality (PSNR) of the searched model improved by 1.5dB compared to the baseline U-net network.



(a) Over-Parameterized Node

(b) Parameter Sharing in Encoder-Decoder Pair

Fig. 6: Mixed Precision Quantization Search on U-net

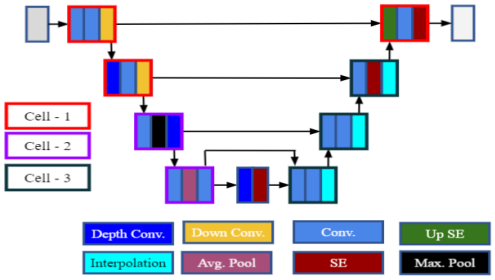


Fig. 7: Searched Model by Three Cell-based Operation Search

III. Mixed Precision Quantization Search

A. Search Space

The design of search space for weight/activation precision is crucial for the mixed-precision networks to be executed on Tensor Core GPUs. The precision allowed for weight and activation matrix in a single layer for efficient execution is Int8, Int4, or Int1 as Tensor Cores in Turing architecture supports only these executions. We use the DoReFa-Net [23] methodology to quantize 1-bit weight and activations, whereas linear quantization [10] for Int8 and Int4 weights. The thresholds are trained in the log space [24] for Int4/Int8 activations.

B. Over-Parameterized Quantized Super-Net

In the supernet, every edge represents the convolution operator with quantized weights and activations in the same precision. The quantized convolution is preceded by an activation quantizer to map previous activations to the respective precision, followed by an activation function (ReLU) and architecture parameter (α), as shown in Fig. 6a. The second operation in a decoder cell of U-net is the depthwise activation concatenation of the upsampled output and the skip connection, which must be in the same quantization range and precision. Hence, we share the activation quantizers and architecture parameters (α) of second convolution of n^{th} encoder cell with the second convolution of $(n-1)^{th}$ decoder cell, as shown in Fig. 6b.

Complexity Loss: Generally, networks with high-precision weights and activations have high accuracy metrics. Also, optimizing only L_1 loss in the mixed-precision supernet tends the differentiable search algorithm to choose high precision weights/activations in the search space. Hence, to ensure that the algorithm chooses low precision weights with a trade-off between accuracy/PSNR metrics and network complexity, we utilize a complexity loss function in the final loss function in the search process. This auxiliary loss is a direct function of quantization architecture parameter (α) with the corresponding bit-width, multiplied directly to the L_1 loss, given as follows:

Total Loss = (L_1 Loss) * $(\sum_{i=1}^{Layers}(\alpha_{i1}*1 + \alpha_{i2}*4 + \alpha_{i3}*8))^{\gamma}$

The gamma (γ) in the complexity loss is used for a trade-off between complexity and accuracy. A higher gamma gives less preference for accuracy. We omit the complexity loss in the final sampled network from the supernet.

C. Training and Results

We first train the U-net model in floating-point precision and quantize the weights in respective edges of the over-parameterized and final sampled networks. The quantized weights and activations in the training process are fine-tuned with fake/simulated quantization training [11]. The PSNR and average bit-width comparison of 8-bit, 4-bit, and our searched Mixed-Precision U-net model for Denoising task (Noise Level 5) is given in Table IV. The 8-bit network outperformed other networks due to the presence of high bit-width and computation cost in all the layers. However, the PSNR of the searched mixed-precision exceeds the PSNR of the 4-bit precision network with less average bit-width. This is because of the presence of few crucial layers executing in 8-bit precision.

TABLE IV: Comparison of Quantized Unets on Denoising Task with Noise Level 5 on BSD68 Test Dataset

Network	PSNR	Average bit-width	Storage Space (MB)
8-bit Unet	**34.02**	8	36
4-bit Unet	33.09	4	18
Mixed-Precision Unet	33.87	**3.47**	**15.6**

IV. Conclusion

For U-net and its topological variants, we devise a Neural Architecture Search (NAS) process to identify the optimal set of operations and the quantization of weights and activations. The results show that our searched U-net models exceed the accuracy (PSNR) metrics of the baseline U-net model with fewer parameters and lesser computation. The mixed precision quantized U-net model exceeds the PSNR of the 4-bit quantized model with a lower average bit-width.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[4] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. Le, and A. Kurakin, "Large-scale evolution of image classifiers," *arXiv preprint arXiv:1703.01041*, 2017.

[5] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[7] M. M. Thomas, K. Vaidyanathan, G. Liktor, and A. G. Forbes, "A reduced-precision network for image reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–12, 2020.

[8] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[9] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[10] R. Krishnamoorthi, "Quantizing deep convolutional networks for efficient inference: A whitepaper," *arXiv preprint arXiv:1806.08342*, 2018.

[11] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2704–2713.

[12] K. Wang, Z. Liu, Y. Lin, J. Lin, and S. Han, "Haq: Hardware-aware automated quantization with mixed precision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 8612–8620.

[13] B. Wu, Y. Wang, P. Zhang, Y. Tian, P. Vajda, and K. Keutzer, "Mixed precision quantization of convnets via differentiable neural architecture search," *arXiv preprint arXiv:1812.00090*, 2018.

[14] Z. Cai and N. Vasconcelos, "Rethinking differentiable search for mixed-precision neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2349–2358.

[15] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] Y. Weng, T. Zhou, Y. Li, and X. Qiu, "Nas-unet: Neural architecture search for medical image segmentation," *IEEE Access*, vol. 7, pp. 44 247–44 257, 2019.

[19] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135.

[20] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2. IEEE, 2001, pp. 416–423.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv preprint arXiv:1606.06160*, 2016.

[24] S. R. Jain, A. Gural, M. Wu, and C. H. Dick, "Trained quantization thresholds for accurate and efficient fixed-point inference of deep neural networks," *arXiv preprint arXiv:1903.08066*, 2019.