Barrett's lesion detection using a minimal integer-based neural network for embedded systems integration

Citation for published version (APA):
Boers, T., Kusters, K., Fockens, K. N., Jukema, J. B., Jong, M. R., de Groof, A. J., Bergman, J. J. G. H. M., van der Sommen, F., & de With, P. H. N. (2023). Barrett's lesion detection using a minimal integer-based neural network for embedded systems integration. In K. M. Iftekharuddin, & W. Chen (Eds.), *Medical Imaging 2023: Computer-Aided Diagnosis* (pp. 1-6). (Proceedings of SPIE; Vol. 12465). SPIE. https://doi.org/10.1117/12.2653890

DOI:
10.1117/12.2653890

Document status and date:
Published: 07/04/2023

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Barrett's lesion detection using a minimal integer-based neural network for embedded systems integration

Tim G. Boers, Carolus H. Kusters, Kiki Fockens, Jelmer Jukema, Martijn Jong, et al.

Tim G. W. Boers, Carolus H. J. Kusters, Kiki N. Fockens, Jelmer B. Jukema, Martijn R. Jong, Jeroen de Groof, Jacques J. Bergman, Fons van der Sommen, Peter H. N. de With, "Barrett's lesion detection using a minimal integer-based neural network for embedded systems integration," Proc. SPIE 12465, Medical Imaging 2023: Computer-Aided Diagnosis, 1246527 (7 April 2023); doi: 10.1117/12.2653890

Event: SPIE Medical Imaging, 2023, San Diego, California, United States

Barrett's Lesion Detection using a minimal Integer-based Neural Network for Embedded Systems Integration

Tim G.W. Boers[a], Carolus H.J. Kusters[a], Kiki N. Fockens[b], Jelmer B. Jukema[b], Martijn R. Jong[b], Jeroen de Groof[b], Jacques J. Bergman[b], Fons van der Sommen[a], and Peter H.N de With[a]

[a]Eindhoven University of Technology, Groene Loper 5, Eindhoven, The Netherlands
[b]Amsterdam UMC, Meibergdreef 9, Amsterdam, The Netherlands

ABSTRACT

Embedded processing architectures are often integrated into devices to develop novel functions in a cost-effective medical system. In order to integrate neural networks in medical equipment, these models require specialized optimizations for preparing their integration in a high-efficiency and power-constrained environment. In this paper, we research the feasibility of quantized networks with limited memory for the detection of Barrett's neoplasia. An Efficientnet-lite1+Deeplabv3 architecture is proposed, which is trained using a quantization-aware training scheme, in order to achieve an 8-bit integer-based model. The performance of the quantized model is comparable with float32 precision models. We show that the quantized model with only 5-MB memory is capable of reaching the same performance scores with 95% Area Under the Curve (AUC), compared to a full-precision U-Net architecture, which is 10× larger. We have also optimized the segmentation head for efficiency and reduced the output to a resolution of 32×32 pixels. The results show that this resolution captures sufficient segmentation detail to reach a DICE score of 66.51%, which is comparable to the full floating-point model. The proposed lightweight approach also makes the model quite energy-efficient, since it can be real-time executed on a 2-Watt Coral Edge TPU. The obtained low power consumption of the lightweight Barrett's esophagus neoplasia detection and segmentation system enables the direct integration into standard endoscopic equipment.

Keywords: Embedded systems, full-integer quantization, Barrett's neoplasia detection

1. INTRODUCTION

To optimize the performance of neural networks (NNs) for real-time embedded systems, efficiency and training strategies should be considered due to constraints on computational power. These embedded architectures are often integrated into devices to develop integration into a cost-effective medical system. A possible advantage with such devices is to favor low-latency processing to enhance user engagement. Yet, simple embedded hardware only has limited computational precision (integer-based) and limited memory capacity, while modern state-of-the-art NNs require high computational resources beyond the capabilities of many embedded processor units. Therefore, to facilitate the implementation of NNs on commercially available medical systems, it is necessary to reduce the computational footprint, memory usage, and simplify and adapt the NN to embedded hardware capabilities, such as integer-based operations.

Optimizations of the NN design can be categorized into two groups, micro and macroarchitecture optimizations. The microarchitecture optimization focuses on improving the operations in the network layers. For example, a widely adopted optimization introduced by Howard *et al.*[1] is the integration of depth-wise separable convolutions.[2–4] In order to improve quantization compatibility Sandler *et al.*[5] introduced ReLU6. Jacob *et al.*[6] presented a method to remove the batch-normalization operations by integrating the normalization into the adjacent convolutional layers. The macroarchitecture search is used to optimize the topological structure of a neural network. These optimizations introduce new modules into the architecture, which can help to improve

Further author information: (Send correspondence to T. Boers)
T. Boers: E-mail: t.boers@tue.nl

Medical Imaging 2023: Computer-Aided Diagnosis, edited by Khan M. Iftekharuddin,
Weijie Chen, Proc. of SPIE Vol. 12465, 1246527 · © 2023 SPIE
1605-7422 · doi: 10.1117/12.2653890

accuracy, such as squeeze-and-excitation[7] and residual modules.[8] Tan *et al.*[9] introduced EfficientNet, which optimizes the scaling of neural networks for depth, width, and input resolution.

To further simply the network, quantization can be used to reduce the memory footprint and simplify the operations of NNs, which require special training and architectural optimizations. This technique involves transforming floating-point operations, typically operated at 32 bits, into low-precision floating-point or integer values. Recent research has shown consistent success in the translation to 8-bit integer-precision calculations,[6, 10] which offers several performance benefits. These benefits include (1) the ability to adapt NNs to processors that can only perform integer-based operations, (2) improved throughput on processors optimized for low-precision data formats, and (3) reduced bandwidth requirements for loading data into memory. There are two main methods for quantizing NNs: post-training quantization (PTQ) and quantization aware-training (QAT). PTQ measures the activation ranges of an already trained NN using (unlabeled) data, and quantizes the weights and activations accordingly. Alternatively, QAT involves introducing quantization noise resulting from rounding errors into the NN during training, in order to optimize the quantized weights and activations to achieve a nearly lossless accuracy.

In this paper, we evaluate the feasibility of quantized neural networks (NNs) for medical applications. In particular, this work concentrates on the use-case of Barrett's neoplasia detection in white-light endoscopy (WLE). The proposed system involves an embedded framework, which combines the EfficientNet-lite[4] encoder plus DeeplabV3 decoder[2] and then transforms the network into a quantized state, based on QAT to achieve a full-integer-based network. Finally, the full-integer model is tested on a Coral edge Tensor Processing Unit (TPU), which is broadly accepted for computing platforms and is optimized for executing NNs.

In summary, our contributions are twofold. (1) We demonstrate that full-integer-based NNs can achieve comparable performances to single-precision floating-point models for Barrett's neoplasia detection. (2) An efficient decoder design is proposed that is specifically optimized for the detection of neoplasia to further decrease the computing footprint of the NN, while maintaining good segmentation details. It is conjectured that our findings generalize to other endoscopic tasks as well.

2. MATERIALS AND METHODS

2.1 Data

Collection: A dedicated data set for Barrett's neoplasia detection in WLE is collected for training, validating and testing. The classification labels for the images are based on a histologically proven ground truth. Clinical research fellows have selected each image and assigned each of them to a set, based on a patient split, while assuring that each set is representative for the various tumor characteristics described by the Paris classification. De-identification is performed in line with the General Data Protection Regulation (EU) 2016/679. The training set consists of 6,237 neoplastic images (1304 patients) and 7,595 Non-Dysplastic Barrett's Esophagus (NDBE) images (1,103 patients), the validation set contains 100 neoplastic images (54 patients) and 100 NDBE images (36 patients). Finally, the test set contains 100 neoplastic images (50 patients) and 300 NDBE images (125 patients).

Annotation: A subset of 2,651 neoplastic images is delineated twice by two experts on Barrett's neoplasia. One delineated area is the Higher-Likelihood (HL), which contains the area that is definitely considered neoplasia by the expert. The second area is the Lower-Likelihood (LL), which is atypical from normal NDBE tissue, which might be neoplasia. In total, 14 international experts have contributed to the delineations. For the HL neoplasia delineation, a minimal consensus of 30% DICE is implemented between experts in order to ensure that both delineate the same area. If the DICE score is less than 30%, then a third expert is invited to annotate the image as well. The two most overlapping delineations are then used to generate the ground truth. Finally, a consensus ground truth to train the model is defined as the union of the two HLs unified with the intersection of the LLs.

2.2 Network Architecture

The proposed network architecture is constructed using an ImageNet-pretrained EfficientNet-Lite1 feature encoder and a MobileNetV2 DeepLabV3+ segmentation decoder, which are both optimized for fast and efficient processing of real-time imagery and compatible for quantization. The network provides two output heads for classification and segmentation. This allows for joint training, and mutual information exchange to the feature extractor, in order to improve feature learning for both tasks. In contrast to similar segmentation models, as in MobileNetV2,[2] the feature maps are downscaled 4 times to 8×8-pixel resolution in the encoder instead of only 2 times, since this is more resource-efficient. These feature maps are then upscaled in the decoder, which outputs a 32×32-pixel resolution segmentation map. The segmentation mask is then subsequently upscaled to the original input resolution. Given that tumors are blob-like shapes, this resolution preserves sufficient detail to clearly segment a neoplastic area.

This network architecture is compared to a standard U-net model with a Resnet-18 backbone, which is often employed in medical settings.

2.3 Training

The training is split into two stages. In the first stage, the model is trained in full-float32 precision, and in the second stage the model is further fine-tuned using QAT, which introduces 8-bit integer rounding errors. This two-stage approach generally leads to better results for QAT, since the pre-quantized weights are already at a good minimum in the loss landscape. The model is trained with a batch size of 32 for 350 epochs in the first stage, and 150 epochs in the second stage. Both stages are trained using Adam and AMS-grad with a weight decay of 10^{-4}, and a learning rate of 10^{-4} and 10^{-5} for the first and second stage, respectively. A cyclic cosine learning-rate scheduler is used to control the learning rate. For the encoder head, we employ a binary cross-entropy (BCE) loss function and for the decoder head of the network, we use a compound DICE+BCE loss function. Images and segmentation masks are randomly rotated with $\theta \in \{0°, 90°, 180°, 270°\}$ and randomly flipped along the $x-$axis and $y-$axis with probability $p = 0.5$. Additionally, random permutations are made to the contrast, brightness and saturation of the images. Since the training set is class-imbalanced, the training images are randomly sampled such that each class is represented 50% on average during each iteration.

The training strategy of the U-net architecture followings the exact training steps of the first stage during training.

2.3.1 Quantization aware-training Scheme

QAT involves introducing quantization noise into the NN weights, resulting from rounding errors during training via a consecutive quantization and dequantization step. These steps are formally expressed in Equations (1) and (3) for 8-bit signed-integer quantization. These quantization functions are applied to the NN weights and activations. The quantization function is specified by:

$$x_q = \text{Quant}(x, S, Z) = \text{Clip}(\text{Round}(\frac{x}{S} + Z)), \tag{1}$$

where parameter x_q is the quantized form of the input x, based on the scale factor S (Real) and the zero-point control value Z (Integer). The Round($\cdot$) operation rounds the input to the nearest integer. The clip function is specified by:

$$\text{Clip}(x) = \begin{cases} -128, & x < -128; \\ x, & -128 \leq x \leq +127; \\ +127, & x > +127. \end{cases} \tag{2}$$

The dequantization function is defined by:

$$\hat{x} = \text{Dequant}(x_q, S, Z) = (x_q - Z) \cdot S, \tag{3}$$

where $\hat{x}$ is the dequantized float32 value of x with quantization noise applied in between. The scaling factor S and the zero-point control value Z are calculated based the moving-average filtering to obtain the maximum value α and minimum value β. The scaling and zero-point control values are computed as follows:

$$S = \frac{\alpha - \beta}{255}, \tag{4}$$

$$Z = -\text{round}(\beta \cdot s) - 128. \tag{5}$$

2.4 Full-integer inference scheme

After QAT, the model can be converted in order to execute in integer-precision mode using the Tensorflow-lite library. In this process, the model weights are converted to int8 precision, batch-normalization folding is applied and all dropout layers are removed in order to save computation power. After the conversion, the model requires new computational graphs, which are provided by Algorithm 1, Algorithm 2 and Equation (6), which represent the Tensorflow-lite reference implementation.

Algorithm 1 Full integer execution of a 2D convolution

```
Input: Four arrays: input, filter, output, bias. Each array carries its own quantization parameters S and Z.
Output: An 8-bit feature map as a product of the quantized convolution of the input and the filter.
  function IntegerConvolution2D(input, filter, output, bias)
    for $x_i, y_i, c_i = 1$ to $X_i, Y_i, C_i$ do        // iterate over the width, height and channels of the input array
      $acc = 0$        // initialize an accumulator with int32 precision
      for $x_f, y_f = 1$ to $X_f, Y_f$ do        // iterate over the filter width and height
        $acc = acc + (input[x_i,y_i,c_i] + zeropoint_i) * filter[x_f,y_f]$
      end for
      $acc = acc + bias[c_i]$
      $acc =$ MultiplyByQuantizedMultiplier($acc$, $multiplier_q$, $shiftvalue$)
      $acc = acc + zeropoint_o$        // acc is shifted by the zeropoint value of the output
      $acc =$ CLIP((acc), Quantize(0), Quantize(6))        // ReLU6 in quantized domain
      output[$x_i$, $y_i$, $c_i$] $\leftarrow$ Cast($acc$)        // cast array to int8 precision
    end for
    return output
```

Algorithm 2 Multiplication step of the quantized feature map

```
Input: An input value, multiplier and shifting value
Output: MultiplyByQuantizedMultiplier($accumulator$, $multiplier_q$, $shiftvalue$)
  function MultiplyByQuantizedMultiplier($accumulator$, $multiplier_q$, $shiftvalue$)
    $totalshift = 31 - shiftvalue$
    $round = 1$ << $(totalshift - 1)$
    $result = accumulator * multiplier_q + round$
    $result = results$ >> $totalshift$
    return $result$
```

The following function returns two output values, i.e. $\text{multiplier}_\text{q}$ and a shiftvalue, which are computed by combining all scaling factors of the input, filter and output stage, using the "frexp" function, resulting in:

$$\text{multiplier}_\text{q}, \text{shiftvalue} = \text{frexp}\left(\frac{S_\text{input} \cdot S_\text{filter}}{S_\text{output}}\right). \tag{6}$$

Here, the first and second output values are together fitting in the expression "$\text{multiplier}_\text{q}$ * 2**shiftvalue", which describes their role as mantissa and exponent value, respectively.

3. EXPERIMENTAL SETUP

Software: For Coral EdgeTPU optimizations, the following software packages are employed: Cuda 11.6, CuDNN 7.6.2, Tensorflow 2.9.1, Tensorflow Model Optimization Toolkit 0.7.2 and PyCoral 2.0.0.

Hardware: All our training experiments are performed on a desktop with an i9-9820X CPU, 32 GB of RAM and an RTX 2080 Ti GPU. The final testing of the quantized model is performed on a Coral Edge TPU, which is integrated into an MSI GS65 laptop. The Edge TPU platform is an ASIC accelerator, which makes it possible to efficiently execute the model on a 2-Watt TPU and achieves real-time performance using quantized 8-bit integer operations.

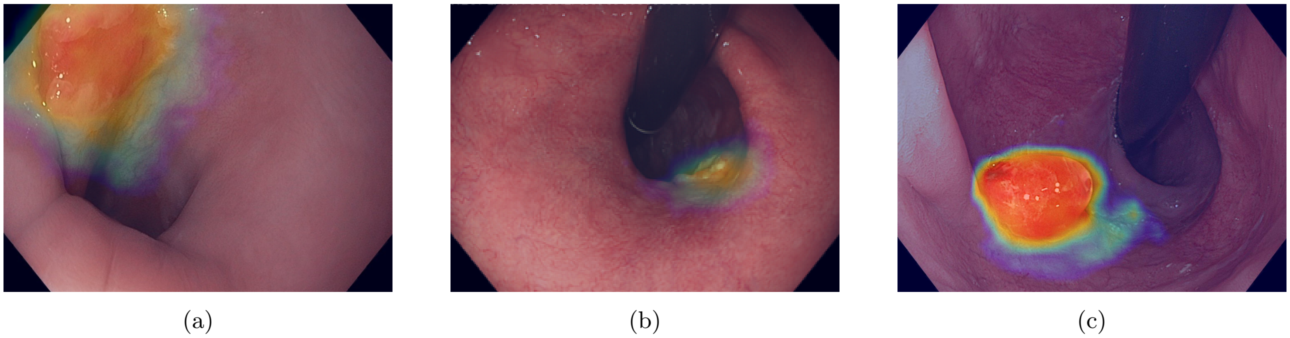

Figure 1: Examples of the obtained heat maps from the segmentation head of the EfficientNet-lite1+DeeplabV3 architecture used for Barrett's neoplasia detection.

4. RESULTS

This section presents the detection results of the NNs. The experiments are repeated 5 times with a different initialization of the network heads and data augmentation for each NN. Table 1 reports the mean results along with the standard deviation between brackets. The results are based on the output of the segmentation head, where the neoplasia score (classification) is defined as the maximum pixel value in the segmentation mask. A detection is therefore regarded positive when this value exceeds a threshold of 0.5. Our quantized network executes at 34.8 frames/second.

Table 1: Performance comparison of the proposed EfficientNet-lite1+DeeplabV3 architecture in 8-bit integer precision, compared with the float32 precision version and a baseline U-Net. The presented results are the average values of 5 full training cycles. The values between the brackets denote the standard deviation.

Design	Exec.	Size	AUC (%)	Accuracy (%)	Sens. (%)	Spec. (%)	DICE (%)
RN18+U-Net	Fp32	56.1 MB	94.10 (0.80)	83.88 (1.76)	91.66 (2.19)	81.40 (1.84)	72.87 (1.65)
Proposed NN	Fp32	19.5 MB	95.86 (0.10)	84.50 (1.16)	94.00 (1.22)	81.33 (1.58)	66.01 (1.14)
Proposed NN	Int8	5.2 MB	95.40 (0.54)	84.90 (1.28)	93.00 (0.71)	82.19 (1.61)	66.51 (1.28)

5. DISCUSSION AND CONCLUSION

We have presented a lightweight quantized 8-bit architecture that is capable of real-time execution on resource-constrained or embedded computing devices. This architecture achieves similar detection performances as a U-Net and our proposed architecture in float32 precision, which is reported in Table 1. However, the quantized version of the proposed model obtains this performance with a 10× and a 4× reduction of the model size compared to the U-Net and the float32 version of the proposed model, respectively. Furthermore, the proposed model achieves a frame rate of 34.8 frames/second, while executing on a 2-Watt Coral Edge TPU in our test setup. This last result is highly relevant for power-constrained solutions to be applied in medical equipment.

Along with a high detection performance, the proposed decoder achieves a DICE score of 66.51%. For the detection of Barrett's neoplasia, the 32×32 segmentation maps contain sufficient resolution to capture the shape of the neoplastic region, in order to alert and direct clinicians to a potential neoplastic area in the esophagus (see Figure 1). Moreover, by implementing this low-resolution segmentation head, the amount of operations can be reduced, while maintaining a detailed segmentation of the tumor for highlighting the neoplastic area. The visualization can be further refined with clinicians, for improvement towards higher user acceptance and optimal matching to the workflow of the clinician.

While the presented model is already rather small, the model size can still be further reduced using model pruning. This is a technique for removing model filters that do not add a meaningful contribution to the prediction result of the network, which thereby maintains the original performance level. Future work could also include the testing of the proposed system on a larger and diverse dataset, and investigating the impact of different quantization strategies on the performance of the model. Moreover, the proposed framework can be extended to other medical imaging modalities and other disease detection tasks.

In conclusion, this work presents a lightweight model for Barrett's neoplasia detection, which solely uses integer-based operations and dramatically limits the amount of memory, which makes it suitable for direct implementation into medical resource-restricted hardware with low power consumption. We have shown that by using these techniques, a similar accuracy as normal floating-point NNs can be maintained. The proposed lightweight approach also makes the model quite energy-efficient, since it can be real-time executed on a 2-Watt Coral Edge TPU. The obtained low power consumption of the lightweight Barrett's esophagus neoplasia detection and segmentation system enables the direct integration into standard endoscopic equipment.

ACKNOWLEDGMENTS

We gratefully acknowledge the research support provided by Olympus Corporation, Tokyo, Japan.

REFERENCES

[1] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv :1704.04861* (2017).

[2] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C., "Mobilenetv2: Inverted residuals and linear bottlenecks," in [*Proceedings of the IEEE conference on CVPR*], 4510–4520 (2018).

[3] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V., "Mnasnet: Platform-aware neural architecture search for mobile," in [*Proceedings of the IEEE/CVF CVPR*], 2820–2828 (2019).

[4] Liu, R., "Higher accuracy on vision models with efficientnet-lite," *TensorFlow Blog* (2020).

[5] Krizhevsky, A. and Hinton, G., "Convolutional deep belief networks on cifar-10," *Unpublished manuscript* (2010).

[6] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 2704–2713 (2018).

[7] Hu, J., Shen, L., and Sun, G., "Squeeze-and-excitation networks," in [*IEEE CVPR Proceedings*], 7132–7141 (2018).

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

[9] Tan, M. and Le, Q., "Efficientnet: Rethinking model scaling for convolutional neural networks," in [*International conference on machine learning*], 6105–6114, PMLR (2019).

[10] Wu, H., Judd, P., Zhang, X., Isaev, M., and Micikevicius, P., "Integer quantization for deep learning inference: Principles and empirical evaluation," *arXiv preprint arXiv:2004.09602* (2020).