Resource-Limited Automated Ki67 Index Estimation in Breast Cancer

Jessica Gliozzo
jessica.gliozzo@unimi.it
Dipartimento di Informatica,
Università degli Studi di Milano
Milan, Italy
European Commission, Joint Research
Centre (JRC)
Ispra, Italy

Giosuè Marinò
giosue.marino@studenti.unimi.it
Dipartimento di Informatica,
Università degli Studi di Milano
Milan, Italy

Arturo Bonometti
arturo.bonometti@hunimed.eu
Department of Biomedical Sciences,
Humanitas University
Pieve Emanuele - Milan, Italy
Department of Pathology, IRCCS
Humanitas Clinical and Research
Hospital
Rozzano – Milan, Italy

Marco Frasca*
marco.frasca@unimi.it
Dipartimento di Informatica,
Università degli Studi di Milano
Milan, Italy
CINI National Laboratory in Artificial
Intelligence and Intelligent Systems
Rome, Italy

Dario Malchiodi
dario.malchiodi@unimi.it
Dipartimento di Informatica & DSRC,
Università degli Studi di Milano
Milan, Italy
CINI National Laboratory in Artificial
Intelligence and Intelligent Systems
Rome, Italy

ABSTRACT

The prediction of tumor progression and chemotherapy response has been recently tackled exploiting Tumor Infiltrating Lymphocytes (TILs) and the nuclear protein Ki67 as prognostic factors. Recently, deep neural networks (DNNs) have been shown to achieve top results in estimating Ki67 expression and simultaneous determination of intratumoral TILs score in breast cancer cells. However, in the last ten years the extraordinary progress induced by deep models proliferated at least as much as their resource demand. The exorbitant computational costs required to query (and in some cases also to store) a deep model represent a strong limitation in resource-limited contexts, like that of IoT-based applications to support healthcare personnel. To this end, we propose a resource consumption-aware DNN for the effective estimate of the percentage of Ki67-positive cells in breast cancer screenings. Our approach reduced up to 75% and 89% the usage of memory and disk space respectively, up to 1.5× the energy consumption, and preserved or improved the overall accuracy of a benchmark state-of-the-art solution. Encouraged by such positive results, we developed and structured the adopted framework so as to allow its general purpose usage, along with a public software repository to support its usage.

---

*Corresponding author

CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; • **Applied computing** → **Bioinformatics**; • **General and reference** → **Performance**.

KEYWORDS

Tumor infiltrating lymphocytes, Ki67 protein, Resource-limited learning, Resource-limited devices, DNN compression, Deep learning.

1 INTRODUCTION

The nuclear protein Ki67 has been introduced as a proliferative marker to be used along with Tumor Infiltrating Lymphocytes (TILs) as a feature effectively driving the prediction of tumor progression and chemotherapy response [31, 48]. Ki67 estimation in surgical pathology is a specific example of a task in which the result of the evaluation bears a significant clinical consequence in many cancer types. Indeed, the pathology practice considers different positivity cut-offs to discriminate between lesions with different overall prognosis or therapy response [41, 49]. For this reason, an high accuracy in Ki67 estimation is essential to avoid over or under-grading of the scrutinized sample. In breast pathology this task may occupy a major role in the routine of pathologists given the high number of surgical samples produced daily in both referral and peripheral pathology centres, as a consequence of the

effectiveness of the woman health screening policies in different countries. These tasks require the segmentation and visual count of cells by pathologists, which is a relatively time-consuming task subject to high intra- and inter-operator variability, related to the experience of each professional [8, 20]. Furthermore, in the evaluation of all these features it should also be considered the presence of a different degree of intra-tumoral heterogeneity, depending on the cancer histotype and the type of tissue sample [57]. Thus, the development of automatic cell count approaches enables a faster and more reliable diagnosis/prognosis. Indeed, even if the evaluation of a single Ki67 slide may be a relatively fast operation, the in-series estimation of a large number of cancer samples is time consuming and may benefit of automated tools able to provide and integrate such data in the histopathology report, independently of the computational resources available. This latter aspect is also of critical importance, as it should be considered that even though screening programs are commonly employed at least in western countries, not every facility may dispose of enough computational resources to run Deep Learning (DL) software [24, 40]. Along with the problem of Ki67 estimation, also the identification of TIL-score is relevant in breast cancer because it has a prognostic role, as a component of the immune system fighting tumor progression [47]. TIL estimation from histological images present challenges similar to those related to Ki67 (i.e. the task easily becomes time-consuming and operator-dependent), thus also this activity can take advantage from the application of DL.

The successful results of deep convolutional models are not surprising, since these models proved to be one of the most effective and flexible tools to tackle healthcare problems [4, 13]. These methods are currently applied to analyse a wide spectrum of medical data types, such as (I) clinical images, (II) biosignals, (III) high-dimensional omics data, and (IV) Electronic Health Records, achieving state-of-the-art performance on several tasks, including the recognition of mitoses, nodal metastases, or the measurement of prognostic markers [23, 26, 27, 52].

The application of DL models is also pervasive in the analysis of breast cancer histopathological images, where it is effective to tackle different tasks: image classification [2, 10, 36], cell contours and nuclei detection [37] and Ki67-index estimation. Regarding breast cancer image classification, state-of-the-art methods are based on convolutional neural networks. Nawaz et al. [38] proposed the use of the convolutional architecture DenseNet to perform multi-class classification of histopathological images from the BreakHis dataset [46], reaching an accuracy of 95.4%. Xie et al. [51] used the Inception_V3 and Inception_ResNet_V2 networks to classify breast cancer histopathological images through transfer learning, outperforming many previously proposed architectures (AlexNet, CSDCNN, LeNet). Zhu et al. [56] used a fully-CNN exploiting convolution and deconvolution layers to directly regress a density map showing the position of cells. Interestingly, they evaluated the model also on histopathological breast lymph node images. Xue et al. [53] use AlexNet and ResNet with Euclidean loss to predict the number of cells in image patches. The use of DL for cell counting is so well-established in literature that an ImageJ plugin [14] now enables the use of U-Net models to perform cell detection on generic biomedical image data. Finally, the estimation of Ki67-score from immunohistochemical images is another task where DL is largely

adopted. To this end, Saha et al. [44] initially proposed to use a Gamma mixture model with Expectation-Maximization to identify seed points and patch selection, which are fed to a CNN model having a custom decision layer based on decision trees. The method achieved 91% of F-score. Zhang et al. [54] explored the use of CNN for Ki67 image classification, where a probability heatmap can be obtained from repositioning the classified patches to their original location. The ratio of tumor cells in the heatmap can be used as an indicator for Ki67 expression. Moreover, they explored the use of single shot multibox detectors to detect Ki67 positive and negative cells. Fulawka et al. [16] applied a CNN DenseNet model with fuzzy interpretation to obtain a binary mask, which is then used to segment breast cancer cells and compute Ki67 index. Negahbani et al. [39] introduced a pipeline for cell classification and detection of both Ki67 and TILs that exploits a U-Net architecture with novel residual dilated inception modules (see section 2.2 for further details). However, the resources required to run DL applications are not trivial, being these models typically overparameterized [3], and therefore requiring a high amount of computational resources. We refer here to the size of the learned DL models, and not to the ever-growing size of the datasets to be processed [28], or the size of the single images, as these issues have been addressed in literature (see, e.g., [25] and the use of tiles/patches [5]). Just to state some examples, depending on their specific implementation, the space needed to store a CNN trained for image classification purposes, such as the previously mentioned DenseNet, ResNet, or AlexNet models, varies from tens to hundreds of megabytes; this requirement jumps up to several tens gigabytes if we consider modern Large Language Models (e.g., 11 billions of parameters for some variants of the T5 Text-To-Text Transformer Model [42], meaning around 44 terabytes of RAM when using 4 bytes per parameter). To face the issue related to the excessive resource usage, recent works focused on specifically designing deep models for mobile devices [55], which however, despite their effectiveness, is applicable to novel architectures but not to the existing ones. Indeed, most available DNNs have been pre-trained (often using a more than remarkable amount of resources) with the aim of deploying them onto standard computing facilities. Such aspect can easily hamper the adoption of DL-based medical software, especially on devices with limited resources, like smartphones or IoT-devices [1, 18, 30]. Further, this is even more relevant in developing countries, characterized by poor economic conditions—mainly in the rural areas—and often even by shortage of physicians, which would highly benefit from AI systems to support clinical decisions. Low-resource DL models can improve the exploitation and applicability of the basic wearable networked devices available, and foster a speed-up toward revolutionary changes in the healthcare systems of these countries [21]. On the other side, cloud-based solutions might be used for hosting large DL models in order to query them online. However, apart privacy and security issues induced by sensible data, it would also require an active Internet connection, which might be a problem in poor rural areas. Moreover, Internet and cloud-based solutions have a cost which has to be taken into account, in relation with the (scarce) available budget.

In such a context, this study proposes an automated AI method for cell classification and detection of Ki67 and TILs, with a particular attention to the resource usage. We show our solution to

be competitive with the state-of-the-art solution, PathoNet [39], a CNN shown to be top-performing in classifying breast cancer images belonging to three classes *Ki67- immunopositive, Ki67- immunonegative* and *tumor infiltrating lymphocytes*, while reducing its RAM and disk requirements up to 4× and 9×, respectively, along with its energy consumption. Moreover, once verified the effectiveness of our methodology, we have done a further step towards the realization of a more general framework to reduce the resource demand of existing pre-trained DL models. In this sense, PathoNet can be considered as a use case of such a methodology, relatively 'toy' for its memory size of around 13*MB*, but we will argument in the discussion at the end of the paper how recent researches already proposed DNNs for the same problem requiring several hundreds of megabytes of RAM. Summarizing, we provide a Python module to 1) first let the existing pre-trained model to undergo lossy compression of its layers, and then lossless storage of the compressed layers; 2) to run in main memory the model in the resource-saver format and efficiently querying it directly in such a format, without need to reconvert it to the original uncompressed format.

The paper is organized as follow: Sect. 2 illustrates the used data, model and overall methodology. The obtained results are described in Sect. 3 and discussed in Sect. 4. Some concluding remarks end the paper.

2 METHODS

This section depicts the application of the proposed framework to the considered case study. More precisely, Sect. 2.1 describes the used dataset, while Sect. 2.2 illustrates the framework adopted to learn the low-resources model, as well as how to run queries in the compressed storage format.

2.1 Problem Definition and Data

We requested to the authors the breast cancer invasive ductal carcinoma dataset (SHIDC-B-Ki-67-V1.0)[1], which is already divided in a train and a test set (containing 1656 and 700 images, respectively). Each image is associated with a JSON file containing the nuclei coordinates (position on X and Y axes of the cell center) and the ground truth label for each nucleus (where 1=*Ki67-immunopositive*, 2=*Ki67-immunonegative* and 3=*TIL*). From the provided training set, we created a validation set randomly selecting 20% of the labeled images. We checked that the obtained train (composed by the remaining 80% of the images) and validation set had an average number of cells per image (i.e. "avg./IMG") similar to the values published in Table 2 of the reference paper [39]. The obtained values of avg./IMG are presented in Table 1. The train set undergoes data augmentation by flipping images w.r.t. X and Y axes and applying rotations, as done in the reference paper.

2.2 Algorithm

2.2.1 Notation and preliminary definitions. Most memory and disk requirements of a convolutional neural network (CNN) are due to the storage of weight connections for each layer in the model. Thus, the compression of an existing CNN mainly maps to the problem of finding out a succinct approximation W of a given matrix/tensor W^o representing the learned connection weights of

---

[1]https://github.com/SHIDCenter/PathoNet

a network layer. The *compression ratio* is defined as the ratio ψ between the sizes of the uncompressed and compressed matrix, that is $\psi = \text{size}(W^o)/\text{size}(W)$, where $\text{size}(x)$ is the memory size of x. In general, uppercase boldface symbols will denote matrices, and the corresponding lowercase letters will refer to matrix entries (e.g., w^o will be an entry of W^o). Vectors—precisely, row vectors—will be rendered using italic boldface (e.g., x).

2.2.2 Compression Framework. The first phase of the proposed methodology concerns the reuse of an existing pre-trained model, as common nowadays given the large number of publicly available pre-trained models developed for various applications. To show the power and flexibility of the proposed approach we thereby compress the pre-trained PathoNet network, retrieved from the same repository holding the previously mentioned data.

The PathoNet model. PathoNet is a CNN proposed for the accurate detection and count of tumoral cells, appropriately stained for Ki67 and TILs, from biopsy images of malignant breast tumors. The cells present in these images have been labelled as Ki67 positive tumor cells, Ki67 negative tumor cells, and lymphocytes infiltrating the cancer area. The model is designed to detect and classify cells according to these three classes [39].

To deal with the size variability of cells from image to image, PathoNet is mostly composed of *dilated* inception layers. On the one hand, this solves the problem of choosing a fixed kernel size by using different kernel sizes in one module; on the other one, it enlarges the network structure without suffering from the vanishing gradient problem when increasing the number of kernels. In particular, residual dilated inception modules (RDIMs) are used in the encoder and decoder part of the network. Each of these modules consists of two parallel parts, the first composed by two stacked convolutional layers with kernel size 3, and the second built by stacking two 3×3 dilated convolutional layers with dilation 4. To reduce the number of parameters, and consequently the possibility of overfitting, the outputs of these two parts are not concatenated but summed up. Overall, PathoNet utilizes a U-Net-like structure [43], where most convolutional layers are replaced by RDIMs. The structure of the network is the following: the input layer is initially processed using two convolutional layers, in turn stacked over four encoder RDIMs and four decoder RDIMs; the latter are followed by three 1 × 1 convolutional layers with linear activation function, used to produce the three-channel output of the model (see [39], Figure 5, for a visual representation).

Lossy compression of network layers. This step is crucial to achieve good compression ratios while not affecting the model accuracy. Among the vast plethora of compression techniques proposed in literature (see, e.g. [11] for a survey), we designed and exploited the most suitable techniques for the structure of PathoNet. We did not consider weight or structural pruning, since the model network structure is already tuned by the authors, with a well-conceived and precise organization of the individual blocks. Weight pruning, for instance, is rewarded when attaining high sparsity levels (> 0.5), which in turn allow the usage of compressed formats such as CSC. However, an excessive pruning of convolutional layers can highly deteriorate the predictive capabilities of the model [34]. Additionally, our aim is also to not slower the model execution,

Table 1: Average number of cells per image (avg./IMG) and number of annotated cells (# cells) for each set, presented across cell labels, where immunopositive (Ki67 +) and immunonegative (Ki67 -) represent the result of Ki67 staining.

	Training+ Validation		Training		Validation		Test	
	avg./IMG	# cells	avg./IMG	# cells	avg./IMG	# cells	avg./IMG	# cells
Ki67 +	21.21	35106	20.88	27664	22.48	7442	22.51	15755
Ki67 -	45.31	75010	45.04	59683	46.31	15327	46.63	32643
TIL	1.88	3112	1.81	2402	2.15	710	1.97	1380

along with obtaining its space reduction. Accordingly, we operated a weight quantization of convolutional layers, consisting in building the matrix W using a limited number of distinct weights, each represented using less bits than each entry in W^o. The idea is to cast connection weights into categories and substituting all weights in a category with a representative. This approach, named weight sharing (WS), allows to save room when combined with the Index Map storage format, which consists in replacing weights with their category/representative index, at the cost of storing separately the vector of representative weights. The advantage of this format is that it only adds one level memory access and almost preserves the dot time efficiency of the original model. It is worth pointing out that in case the model to compress contains also other types of layers, e.g., fully-connected layers, specific and more suitable lossy compression and lossless storage approaches can be leveraged, exploiting for instance pruning+quantization compression and address map storage [32, 33].

The strategy used to group weights and to select representatives distinguishes the four state-of-the-art quantization algorithms considered in this study and described in the following.

Clustering-based WS (CWS). This strategy groups weights into k clusters via the k-means algorithm [35], and uses the resulting centroids $\{c_1, \ldots, c_k\}$ as representatives to replace the weights in the corresponding cluster [22].

Probabilistic WS (PWS). This technique is based on the *Probabilistic Quantization* method [33], in which a randomized algorithm transforms each weight $w^o \in W^o$ in one of k distinct representatives $c_1, \ldots, c_k$. A nice feature of this method consists in the fact that the obtained W can be seen as the value of an unbiased estimator for W^o (see [33] for details).

Uniform Quantization (UQ). In this scheme, which achieved top compression performance in recent applications to CNNs compression [6], representatives are selected by uniformly partitioning the entire weight domain into k subintervals[2]. Such a selection has been proven to yield an output entropy which is asymptotically smaller than that of any other quantizer, regardless of the source statistics, when the source density function is sufficiently smoothed [19].

Entropy Constrained Scalar Quantization (ECSQ): This is a technique leveraging an iterative optimization algorithm to determine the optimal number of groups. It is driven by the joint optimization of the expected value for the quantization distortion (a measure of the distance between W^o and W)

and the entropy of the resulting discrete distribution for representative weights [7].

A retraining phase is finally applied after quantization, ensuring that the updated weights always assume values in the set of representatives [22].

Lossless storage of the compressed network. The third phase of our framework is the design of a suitable room-saver format for the model, tailored for the compression schemes used in the previous step, and able to perform the model execution without re-expanding the network. As mentioned above, an efficient solution to exploit the quantized tensors is represented by the Index Map (IM) format [22]. Representatives are stored in a vector $c = \{c_1, \ldots, c_k\}$, whose indices are entries of a new matrix/tensor M. Thus, if $w^o \in W^o$ is associated with centroid, say, c_2, then the corresponding entry in M is set to 2. When W^o (and accordingly M) has dimension $n \times m$, denoted by b and $\bar{b}$ the number of bits used to store one entry of W^o and of M, respectively, the *compression ratio* obtained is:

$$\psi = \frac{bnm}{\bar{b}nm + bk}$$

For instance, when $k \leq 256$, $\bar{b} = 8$ is enough to represent $2^8 = 256$ different indices, and assuming a typical FP32 format for W^o ($b = 32$), the compression ratio would be $\psi \approx 4$. We remark that this format does not induce any information loss, while needing only one additional memory access to retrieve a given weight.

Model inference in the resource-saver format. The final step of our framework is the computation of matrix/tensor products directly in the compressed format used at previous step. Without loss of generality, we can assume that the layer weights are represented by a matrix. To compute the output $o = x \cdot W$ of a given compressed layer with weight matrix W on the input x, that is $o_i = \sum_j x_j w_{ji}$ using IM format, we perform $o_i = \sum_j x_j c_{m_{ji}}$. From a memory consumption standpoint, this does not need to expand the compressed matrix, and it still keeps the model memory footprint ψ times smaller than the original one.

Implementation. The PathoNet source code was implemented in Python 3, using Tensorflow and Keras. Our compression techniques and the retraining procedures have been implemented in the same programming environment. Our implementation, available on GitHub[3], allows: (i) to perform the compression and retraining of PathoNet with different quantization techniques; (ii) to compute the compression ratio (relative to the memory space); (iii) to evaluate the compressed model space on disk and the prediction time ratio

---

[2]Note that the actual number of subintervals k can be lower than the input value due to the internal selection of the δ hyperparameter of the method (see [6] for further details).

[3]https://github.com/GliozzoJ/pathonet_compression

compared to the uncompressed version of PathoNet. The repository also contains a Jupyter notebook allowing the replication of our results via direct execution of the compressed models relying on IM representation and another notebook to estimate energy consumption.

3 RESULTS

We tested the four considered quantization approaches (i.e., CWS, PWS, UQ, and ECSQ; see Sect. 2.2) on the breast cancer dataset provided by the authors of PathoNet. As mentioned in Sect. 2.1, we executed an holdout procedure in which the training set is exploited to retrain the model after quantization and a validation set is used to tune a set of hyperparameters by means of grid search. Indeed, the approaches adopted to compress the network present some hyperparameters that influence the obtained performance. One of them is the number k of groups, which has a direct effect on the final size of the compressed model. Moreover, since we have to retrain the network after the quantization step, all the classical hyperparameters related to the training of a neural network (e.g., learning rate, batch size, patience, etc.) play an important role. In particular, we have considered the cumulative learning rate clr and the number of groups k, since they impacted more on the model accuracy. Once the best model is selected (see Sect. 2.1), the corresponding hyperparameters are used to train the network on the train and validation sets, and the generalization performance of the compressed model is assessed on the test set.

3.1 Evaluation Metrics

The successful application of a compression strategy should lead to a model that (I) retains similar generalization performance w.r.t. the uncompressed model, (II) leads to a reduction in terms of space occupancy, and (III) keeps a reasonable execution time when used to make inferences on the test set.

In particular, the generalization performances are evaluated in terms of F1-score, RMSE (Root Mean Squared Error) and aggregated cut-off accuracy for TILs and Ki67 (as defined in the reference paper). Moreover, two additional metrics are used:

- *Compression ratio:* the ratio of the memory size needed by the uncompressed over the compressed model (cfr. Sect. 2.2);
- *Time ratio:* the ratio between evaluation times on the test set of the uncompressed over the compressed model.

3.2 Hyperparameters optimization

The train and validation sets (Section 2.1) are used to perform the tuning of hyperparameters, i.e., the cumulative learning rate clr for the fine tuning of weights after quantization (Section 2.2), and number of clusters k, by means of grid search. Obviously, the validation set is not augmented in this phase. The best combination of hyperparameters is the one that gives the lowest RMSE on the validation set. Then the best model is retrained on the complete augmented training set.

As outlined in [33], the cumulative learning rate needs to be smaller than the learning rate used to train the original model: accordingly, the grid for clr has been set to [0.001, 0.0001, 0.00001, 0.000001]. The number of groups k has been chosen in [256, 1024, 4096]

Table 2: Values of cumulative learning rate (*clr*) and number of groups k selected by the tuning process.

Quantization	CWS	PWS	ECSQ	UQ
clr	0.00001	0.0001	0.00001	0.0001
k	1024	4096	4096	4096

for all compression methods, with the first choice ensuring using 1 byte for each index, and the other 2 choices ensuring lower compression but potentially higher performance. Note that this bidimensional grid yields 12 combinations for each method, for a total of 48 experiments; this prevented us from using more refined grids. After choosing the best couple, a final retraining using all the augmented training set produces the compressed model. The cumulative learning rate and the number of groups selected at the end of tuning are showed in Table 2. The network configuration was kept as in the original PathoNet model, whenever possible, and all the other experimental details are reported in our public repository[4].

3.3 Experimental results.

The generalization performance, in terms of F1-score for the three classes, RMSE and aggregated cut-off accuracy for Ki67-index and TIL-score is presented in Table 3 and compared to the same metrics computed on the original uncompressed PathoNet model [5].

As we can see from the results of this first set of experiments (rows 2–5 in Table 3), the compressed networks achieve comparable performance w.r.t. the original PathoNet model in terms of F1-score for the classes Ki67 immunopositive and immunonegative. Interestingly, the compression methods UQ and ECSQ obtain a significant improvement in F1-score for the TIL class (3% for UQ and slightly lower for ECSQ). Considering RMSE, the performances of compressed networks are slightly worse for the Ki67-index, but they achieve always better results for the TIL-score. Moreover, all the compressed networks match the uncompressed ones for the Ki67-index cut-off accuracy while always consistently improving the corresponding metric for TIL-score. In particular, the improvement in cut-off accuracy for TIL-score ranges from 4.4% to 13.1%, depending on the applied quantization approach. Overall, all the compressed models almost halve the space occupancy in RAM while bringing a slow down during execution of less than 20%.

Best compression. As shown in Table 2, the grid search process led to the selection of an high number of groups k, which was equal to 4096 in most cases. The performance of the compressed models is competitive with the original PathoNet model, and some metrics are often better (especially the ones related to TILs). This behaviour is expected, since an higher number of representatives gives better chances to preserve the network structure. On the other hand, it is interesting to evaluate if a lower number of representatives

---

[4]https://github.com/GliozzoJ/pathonet_compression
[5]The results reported for the uncompressed network in Table 3 are different from the ones showed in the reference paper [39]. We contacted the authors of PathoNet and they agreed with the correctness of the F1-score results using dataset SHIDC-B-Ki-67-V1.0. Moreover, we implemented the function to compute the RMSE and cut-off accuracy, which is now part of the PathoNet package available on GitHub (https://github.com/SHIDCenter/PathoNet/blob/master/evaluation.py).

Table 3: Generalization performance of the uncompressed (first row) and compressed PathoNet models. Ki67+ stands for immunopositives, Ki67- for immunonegatives, while the last two columns respectively represent the RAM compression and the evaluation time ratios (the higher, the better). The best results for each metric are highlighted in bold.

| Experiment | F1-score | | | RMSE | | Cut-off Accuracy | | | |
	Ki67+	Ki67-	TIL	Ki67-index	TIL-score	Ki67-index	TIL-score	Space	Time
Uncompressed	**0.853**	0.776	0.348	0.050	0.054	0.913	0.826	-	-
CWS	0.852	0.774	0.355	0.054	0.044	0.913	0.870	1.942	0.833
PWS	0.848	0.774	0.351	0.053	**0.021**	0.913	**0.957**	1.789	0.838
ECSQ	0.852	0.776	0.375	0.054	0.039	0.913	0.913	1.789	0.837
UQ	0.852	0.775	**0.378**	0.056	0.036	0.913	0.913	1.916	0.835
CWS - k=256	**0.853**	0.778	0.365	0.053	0.041	0.913	0.870	**3.937**	**0.843**
PWS - k=256	0.835	0.746	0.193	0.050	0.028	**0.957**	**0.957**	**3.937**	0.839
ECSQ - k=256	0.848	**0.781**	0.355	**0.049**	0.023	0.913	0.913	**3.937**	0.837
UQ - k=256	0.852	0.777	0.374	0.055	0.038	0.913	0.913	**3.937**	0.840

can lead to competitive results while significantly reducing the space occupancy in main memory. From a practical standpoint, a user could have only limited computational resources available and willing to still use a compressed CNN even at the expenses of a marginal decrease in generalization performance. To test this situation, we executed again the experiments performed in the previous section but avoiding the model selection process. In particular, the number of groups is fixed to 256 for all quantization methods and the cumulative learning rate as the best value selected by means of grid search in the previous set of experiments (see Table 2). The other hyperparameters remained unchanged. The results are showed in Table 3 (rows 6–9). Quite surprisingly, compressed PathoNet models in this setting tend to preserve or even improve their performance when using more representatives (see, e.g., the CWS method), which has the appreciated benefit that the space compression is still increased, namely to $\approx 4\times$ the original uncompressed PathoNet model. An exception is the PWS method, which shows a lower ability to select informative representatives when their number is limited, confirming the results in [34]. On the other side, CWS method exhibited an overall higher stability and effectiveness in choosing the representative weights, as confirmed by the the fact that it performed best during model selection when not using the maximum k allowed (Table 2).

4 DISCUSSION

In this research, we focused on the problem of the automatic computation of Ki67 and TIL indices in breast cancer, with the primary goal of at least preserve the top performance obtained for this task in the literature, along with a special attention to limit the resulting model resource usage, to not hamper its practical applicability. We showed how to obtain a CNN exhibiting performance competitive with PathoNet (reference CNN for the problem), while yielding a model much more resource-cautious, through a novel compression framework. Our solution is around $4\times$ smaller in terms of memory footprint and $9\times$ in terms of disk size (Figure 1), with reference to PathoNet, while still performing the same or better. This is at first of clinical interest, being Ki67 pivotal in the definition of patients' treatment and for the evaluation of their prognosis [12]. Similarly, the number of TILs show a positive correlation with patients survival and therapy response. In fact at least in certain subtypes of

breast cancer, a higher number of TILs indicates a higher activation of tumor-suppressing adaptive immunity and an increased rate of response after adjuvant anthracycline-based chemotherapy [45]. For example, in lung cancer, the evaluation of PDL-1 immunohistochemical positivity became critical after the demonstration of the efficacy of anti-PDL1 drug Pembrolizumab [17]. The inclusion of patients in immunochemotherapy protocols with Pembrolizumab monotherapy or its combinations passes through PDL1 positive cell count, since inter-observer variability can easily shift the therapeutic plan due to the low amount of positive cells needed to reach the TPS cut-off [9]. Indeed, attempts to use DNN for TPS computation in lung cancer already exist in literature [50].

Moreover, given the growing knowledge about disease molecular therapy targets and interest in Precision Medicine, a future increase in the number of routinely analyzed immunohistochemical predictive and prognostic markers becomes a reasonable educated guess [15]. In this context, the application of ML-based models on prognostic/predictive immunohistochemical panels will shorten the time required for their evaluation, and increase the overall accuracy of the tests. The employment of such models will likely reduce the costs and time related to the evaluation of each biomarker, therefore mitigating the pathologist workload; reducing the resource demand of such computational models will thereby still favour their applicability.

Secondly, apart the relevance of cell count operation emphasized so far, our work induces a second benefit related to the model compression, in terms of both RAM and disk occupancy reduction. In general, the disk space required to store a CNN is lower than the amount of RAM needed to load and query the model, due to optimized formats available to serialize models on disk (e.g., the disk serialization provided by the lzma Python module[6]). The disk storage reduction (up to $9\times$, UQ quantization method), represents an advantage in situations in which the serialized representation of models is used to share the latter among several actors in a distributed framework. This happens notably in the *federated learning* setting [29], characterized by a privacy-preserving communication loop in which edge computing devices train "partial" machine learning models on locally acquired data and send them to a centralized server that merges them and shares with all devices the resulting
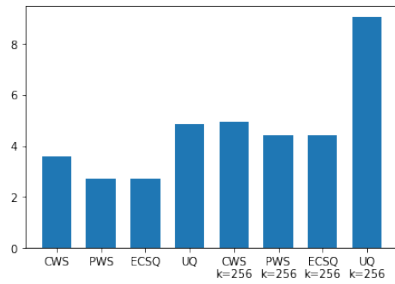
---

[6]https://docs.python.org/3/library/lzma.html

Figure 1: Disk compression ratio of the tested methods with respect to the original size of PathoNet (12.802 MBytes).

"global" model. Having the possibility to send compressed models back and forth would lead in this case to a valuable saving of network bandwidth, which is the bottleneck resource.

Motivated by the promising results obtained, we have made our compression methodology ready and general enough to be applied in other domains with similar needs. In this sense, PathoNet can be seen as a 'toy example' for stating the usefulness of our methodology, being its memory footprint slightly less than 13 MB, taking into account 4 bytes for each model parameter as a floating point number. Nevertheless, for the computation of Ki67-index current researches have also proposed much bigger models: see e.g. [16], where an ensemble of three DenseNet121 models is used to provide a final prediction, which roughly requires 320 MB of RAM. Although in small scale, we give an idea of how much the compression of PathoNet impacts on the energy consumption. We leveraged the Python package *codecarbon* and obtained 0.000573 kWh used for querying the original model on 500 images, and 0.000380 kWh to query the compressed one (i.e. UQ, $k = 256$).[7] Our repository provides a dedicated Jupyter notebook to specifically compare the energy consumption (see Sect. 2), in addition to the code able to reproduce all the experiments proposed in this study, and to the scripts allowing to automatically load serialized saved models, deserialize them, and run their execution.

5 CONCLUSIONS

This study introduces an automated DL approach for the Ki67 - immunonegative, Ki67 - immunopositive, and TILs cell detection on stained images, with a specific attention to the resulting model resource demand. Our framework exhibits performance competitive with state-of-the-art models for the estimation of Ki67 and TIL indices in breast cancer. Further, it successfully tackles the problem of reducing model size and computational resource need, to enhance its applicability in low-resources context, and to limit the energy consumption. With reference to a state-of-the-art top-performing solution, we obtained a model up to 4× and 9× smaller in terms of RAM and disk space respectively, while reducing the energy consumption of around 1.5×, and substantially preserving classification accuracy. Once favorably demonstrated the effectiveness of our approach in the estimation of Ki67 and TIL scores, we have

done a further step to settle a general framework for compressing pre-trained DL models, that coupled with a publicly available repository, allows to extend our approach to other problems and deep models. This study serves thereby as a potential reference for non-expert users who need to downsize existing AI tools based on deep or convolutional neural networks, avoiding building compact DNNs from scratch, mainly for problems where the training has very high costs. Our solution allows to extend existing models to contexts where sufficiently powerful hardware is not available or where devices have inherently few computational resources. Our pipeline can handle both fully-connected and convolutional layer in compression step, and it is not limited to a given storage format in the final compressed layer representation step. The main limitation of the overall framework lies in its applicability to only deep models (which however cover the majority of application domains), and among them, to feed-forward architectures, excluding for instance recurrent neural networks (RNNs), just to state an example. A potential future development of this study would indeed extend the present solution to also support such models.

ACKNOWLEDGMENTS

REFERENCES

[1] Alaa Awad Abdellatif, Lutfi Samara, Amr M. Mohamed, Aiman Erbad, Carla-Fabiana Chiasserini, Mohsen Guizani, Mark Dennis O'Connor, and James Laughton. 2020. I-Health: Leveraging Edge Computing and Blockchain for Epidemic Management. *ArXiv* abs/2012.14294 (2020).
[2] Hanan Aljuaid, Nazik Alturki, Najah Alsubaie, Lucia Cavallaro, and Antonio Liotta. 2022. Computer-aided diagnosis for breast cancer classification using deep neural networks and transfer learning. *Comput. Methods. Programs. Biomed.* 223 (2022), 106951.
[3] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. 2019. Learning and Generalization in Overparameterized Neural Networks, Going Beyond Two Layers. In *Advances in Neural Inf. Process. Syst.*, Vol. 32. Curran Associates, Inc.
[4] Yoshua Bengio, Ian Goodfellow, and Aaron Courville. 2017. *Deep learning*. Vol. 1. MIT press, Massachusetts, USA.
[5] Adam G. Berman, William R. Orchard, Marcel Gehrung, and Florian Markowetz. 2021. PathML: A unified framework for whole-slide image analysis with deep learning. *medRxiv* (2021). https://doi.org/10.1101/2021.07.07.21260138
[6] Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee. 2020. Universal Deep Neural Network Compression. *IEEE J. Sel. Topics Signal Process.* 14, 4 (2020), 715–726.
[7] Philip A Chou, Tom Lookabaugh, and Robert M Gray. 1989. Entropy-constrained vector quantization. *IEEE Trans. Acoust. Speech, Signal Process.* 37, 1 (1989), 31–42.
[8] Yul Ri Chung, Min Hye Jang, So Yeon Park, Gyungyub Gong, Woo-Hee Jung, Korean Breast Pathology Ki-67 Study Group, et al. 2016. Interobserver variability of Ki-67 measurement in breast cancer. *J. Pathol. Transl. Med.* 50, 2 (2016), 129.
[9] Wendy A Cooper, Prudence A Russell, Maya Cherian, Edwina E Duhig, David Godbolt, Peter J Jessup, Christine Khoo, Connull Leslie, Annabelle Mahar, David F Moffat, et al. 2017. Intra-and interobserver reproducibility assessment of PD-L1 biomarker in Non–Small cell lung cancer. *Clin. Cancer Res.* 23, 16 (2017), 4569–4577.
[10] Taye Girma Debelee, Friedhelm Schwenker, Achim Ibenthal, and Dereje Yohannes. 2020. Survey of deep learning in breast cancer image analysis. *Evolving Systems* 11 (2020), 143–163.
[11] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. 2020. Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey. *Proc. IEEE* 108, 4 (2020), 485–532.
[12] Mitch Dowsett, Torsten O Nielsen, Roger A'Hern, John Bartlett, R Charles Coombes, Jack Cuzick, Matthew Ellis, N Lynn Henry, Judith C Hugh, Tracy

---

Lively, et al. 2011. Assessment of Ki67 in breast cancer: recommendations from the International Ki67 in Breast Cancer working group. *J. Natl. Cancer Inst.* 103, 22 (2011), 1656–1664.

[13] Jan Egger, Christina Gsaxner, Antonio Pepe, Kelsey L Pomykala, Frederic Jonske, Manuel Kurz, Jianning Li, and Jens Kleesiek. 2022. Medical deep learning–a systematic meta-review. *Comput. Methods Programs Biomed.* (2022), 106874.

[14] Thorsten Falk, Dominic Mai, Robert Bensch, Özgün Çiçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, Jan Deubner, Zoe Jäckel, Katharina Seiwald, et al. 2019. U-Net: deep learning for cell counting, detection, and morphometry. *Nat. Methods* 16, 1 (2019), 67–70.

[15] Matteo Fassan, Aldo Scarpa, Andrea Remo, Giovanna De Maglio, Giancarlo Troncone, Antonio Marchetti, Claudio Doglioni, Giuseppe Ingravallo, Giuseppe Perrone, Paola Parente, et al. 2020. Current prognostic and predictive biomarkers for gastrointestinal tumors in clinical practice. *Pathologica* 112, 3 (2020), 248.

[16] Lukasz Fulawka, Jakub Blaszczyk, Martin Tabakov, and Agnieszka Halon. 2022. Assessment of Ki-67 proliferation index with deep learning in DCIS (ductal carcinoma in situ). *Sci. Rep.* 12, 1 (2022), 3166.

[17] Leena Gandhi, Delvys Rodríguez-Abreu, Shirish Gadgeel, Emilio Esteban, Enriqueta Felip, Flávia De Angelis, Manuel Domine, Philip Clingan, Maximilian J Hochmair, Steven F Powell, et al. 2018. Pembrolizumab plus chemotherapy in metastatic non–small-cell lung cancer. *N. Engl. J. Med.* 378, 22 (2018), 2078–2092.

[18] Laura García, Jesús Tomás, Lorena Parra, and Jaime Lloret. 2019. An m-health application for cerebral stroke detection and monitoring using cloud services. *Int. J. Inf. Manag.* 45 (2019), 319–327.

[19] H Gish and J Pierce. 1968. Asymptotically efficient quantizing. *IEEE Trans. Inf. Theory* 14, 5 (1968), 676–683.

[20] Douglas S Gomes, Simone S Porto, Débora Balabram, and Helenice Gobbi. 2014. Inter-observer variability between general pathologists and a specialist in breast pathology in the diagnosis of lobular neoplasia, columnar cell lesions, atypical ductal hyperplasia and ductal carcinoma in situ of the breast. *Diagn. Pathol.* 9, 1 (2014), 1–9.

[21] Jonathan Guo and Bin Li. 2018. The Application of Medical Artificial Intelligence Technology in Rural Areas of Developing Countries. *Health Equity* 2, 1 (2018), 174–181. PMID: 30283865.

[22] Song Han, Huizi Mao, and William J. Dally. 2016. Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. San Juan, Puerto Rico. http://arxiv.org/abs/1510.00149

[23] M. G. Hanna, O. Ardon, A. V. Reuter, S. J. Sirintrapun, C. England, D. S. Klimstra, and M. R. Hameed. 2021. Integrating digital pathology into clinical practice. *Mod. Pathol.* 35, 2 (Oct 2021), 152–164.

[24] Nadia Harbeck, Frédérique Penault-Llorca, Javier Cortes, Michael Gnant, Nehmat Houssami, Philip Poortmans, Kathryn Ruddy, Janice Tsang, and Fatima Cardoso. 2019. Breast cancer. *Nat. Rev. Dis. Primers* 5, 1 (2019), 1–31.

[25] Daisuke Hirahara, Eichi Takaya, Mizuki Kadowaki, Yasuyuki Kobayashi, and Takuya Ueda. 2021. Effect of the pixel interpolation method for downsampling medical images on deep learning accuracy. *J. comput. commun.* 9, 11 (2021).

[26] Sebastian Klein and Dan G Duda. 2021. Machine Learning for Future Subtyping of the Tumor Microenvironment of Gastro-Esophageal Adenocarcinomas. *Cancers* 13, 19 (2021), 4919.

[27] Kyubum Lee, John H. Lockhart, Mengyu Xie, Ritu Chaudhary, Robbert J. C. Slebos, Elsa R. Flores, Christine H. Chung, and Aik Choon Tan. 2021. Deep Learning of Histopathology Images at the Single Cell Level. *Front. Artif. Intell.* 4 (2021), 137.

[28] Johann Li, Guangming Zhu, Cong Hua, Mingtao Feng, Ping Li, Xiaoyuan Lu, Juan Song, Peiyi Shen, Xu Xu, Lin Mei, et al. 2021. A systematic collection of medical image datasets for deep learning. *arXiv preprint arXiv:2106.12864* (2021).

[29] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* 37, 3 (2020), 50–60.

[30] Ye Liu, Liqiang Nie, Lei Han, Luming Zhang, and David S. Rosenblum. 2015. Action2Activity: Recognizing Complex Activities from Sensor Data. In *Proceedings of the 24th International Conference on Artificial Intelligence* (Buenos Aires, Argentina) (IJCAI'15). AAAI Press, 1617–1623.

[31] Yan Mao, Qing Qu, Xiaosong Chen, Ou Huang, Jiayi Wu, and Kunwei Shen. 2016. The prognostic value of tumor-infiltrating lymphocytes in breast cancer: a systematic review and meta-analysis. *PloS ONE* 11, 4 (2016), e0152500.

[32] Giosuè Cataldo Marinò et al. 2021. Reproducing the Sparse Huffman Address Map Compression for Deep Neural Networks. In *Reproducible Research in Pattern Recognition*. Springer International Publishing, Cham, 161–166.

[33] Giosuè Cataldo Marinò et al. 2021. Compression strategies and space-conscious representations for deep neural networks. In *2020 25th Int. Conf. on Pattern Recognition (ICPR)*. IEEE, Milan, Italy, 9835–9842.

[34] Giosuè Cataldo Marinò, Alessandro Petrini, Dario Malchiodi, and Marco Frasca. 2023. Deep neural networks compression: A comparative survey and choice recommendations. *Neurocomputing* 520 (2023), 152–170.

[35] J. B. McQueen. 1967. Some methods of classification and analysis in multivariate observations. In *Proc. of fifth Barkley symp. on math. statist. and prob.* 281–297.

[36] André LS Meirelles, Tahsin Kurc, Joel Saltz, and George Teodoro. 2022. Effective active learning in digital pathology: A case study in tumor infiltrating lymphocytes. *Comput. Methods Programs Biomed.* 220 (2022), 106828.

[37] Muhammad Firoz Mridha, Md Hamid, Muhammad Mostafa Monowar, Ashfia Jannat Keya, Abu Quwsar Ohi, Md Islam, Jong-Myon Kim, et al. 2021. A comprehensive survey on deep-learning-based breast cancer diagnosis. *Cancers* 13, 23 (2021), 6116.

[38] Majid Nawaz, Adel A Sewissy, and Taysir Hassan A Soliman. 2018. Multi-class breast cancer classification using deep learning convolutional neural network. *Int. J. Adv. Comput. Sci. Appl* 9, 6 (2018), 316–332.

[39] Farzin Negahbani, Rasool Sabzi, Bita Pakniyat Jahromi, Dena Firouzabadi, Fateme Movahedi, Mahsa Kohandel Shirazi, Shayan Majidi, and Amirreza Dehghanian. 2021. PathoNet introduced as a deep neural network backend for evaluation of Ki-67 and tumor-infiltrating lymphocytes in breast cancer. *Sci. Rep.* 11, 1 (2021), 1–13.

[40] Seung Park, Anil V Parwani, Raymond D Aller, Lech Banach, Michael J Becich, Stephan Borkenfeld, Alexis B Carter, Bruce A Friedman, Marcial Garcia Rojo, Andrew Georgiou, et al. 2013. The history of pathology informatics: A global perspective. *J. Pathol. Inform.* 4 (2013), 7.

[41] Gil Patrus Pena and Joséde Souza Andrade-Filho. 2009. How does a pathologist make a diagnosis? *Arch. Pathol. Lab. Med.* 133, 1 (2009), 124–132.

[42] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21, 1, Article 140 (jan 2020), 67 pages.

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, Cham, 234–241.

[44] Monjoy Saha, Chandan Chakraborty, Indu Arun, Rosina Ahmed, and Sanjoy Chatterjee. 2017. An advanced deep learning approach for Ki-67 stained hotspot detection and proliferation rate scoring for prognostic evaluation of breast cancer. *Sci. Rep.* 7, 1 (2017), 3213.

[45] Roberto Salgado, Carsten Denkert, S Demaria, N Sirtaine, F Klauschen, Giancarlo Pruneri, S Wienert, Gert Van den Eynden, Frederick L Baehner, Frederique Pénault-Llorca, et al. 2015. The evaluation of tumor-infiltrating lymphocytes (TILs) in breast cancer: recommendations by an International TILs Working Group 2014. *Ann. Oncol.* 26, 2 (2015), 259–271.

[46] Fabio A Spanhol, Luiz S Oliveira, Caroline Petitjean, and Laurent Heutte. 2015. A dataset for breast cancer histopathological image classification. *IEEE Trans. Biomed. Eng.* 63, 7 (2015), 1455–1462.

[47] Sasha E Stanton and Mary L Disis. 2016. Clinical significance of tumor-infiltrating lymphocytes in breast cancer. *J. ImmunoTher. Cancer* 4, 1 (2016), 1–7.

[48] Pankaj Taneja, Dejan Maglic, Fumitake Kai, Sinan Zhu, Robert D Kendig, A Fry Elizabeth, and Kazushi Inoue. 2010. Classical and novel prognostic markers for breast cancer and their clinical significance. *Clin. Med. Insights: Oncol.* 4 (2010).

[49] Miaomiao Tao, Shu Chen, Xianquan Zhang, and Qi Zhou. 2017. Ki-67 labeling index is a predictive marker for a pathological complete response to neoadjuvant chemotherapy in breast cancer: a meta-analysis. *Medicine* 96, 51 (2017), e9384.

[50] Xiangyun Wang, Peilin Chen, Guangtai Ding, Yishi Xing, Rongrong Tang, Chaolong Peng, Yizhou Ye, and Qiang Fu. 2021. Dual-scale categorization based deep learning to evaluate programmed cell death ligand 1 expression in non-small cell lung cancer. *Medicine* 100, 20 (2021), p e25994.

[51] Juanying Xie, Ran Liu, Joseph Luttrell IV, and Chaoyang Zhang. 2019. Deep learning based analysis of histopathological images of breast cancer. *Front. Genet.* 10 (2019), 80.

[52] H. Xu, F. Cong, and T. H. Hwang. 2021. Machine Learning and Artificial Intelligence-driven Spatial Analysis of the Tumor Immune Microenvironment in Pathology Slides. *Eur. Urol. Focus* 7, 4 (Jul 2021), 706–709.

[53] Yao Xue, Nilanjan Ray, Judith Hugh, and Gilbert Bigras. 2016. Cell counting by regression using convolutional neural network. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part I 14*. Springer, Springer, Cham, Amsterdam, The Netherlands, 274–290.

[54] Ruihan Zhang, Junhao Yang, and Chunxiao Chen. 2018. Tumor cell identification in ki-67 images on deep learning. *Mol. Cell. Biol.* 15, 3 (2018), 177.

[55] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. 2018. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[56] Runkai Zhu, Dong Sui, Hong Qin, and Aimin Hao. 2017. An extended type cell detection and counting method based on FCN. In *2017 IEEE 17th Int. Conf. on Bioinformatics and Bioengineering (BIBE)*. IEEE, Washington, DC, USA, 51–56.

[57] Dovile Zilenaite, Allan Rasmusson, Renaldas Augulis, Justinas Besusparis, Aida Laurinaviciene, Benoit Plancoulaine, Valerijus Ostapenko, and Arvydas Laurinavicius. 2020. Independent prognostic value of intratumoral heterogeneity and immune response features by automated digital immunohistochemistry analysis in early hormone receptor-positive breast carcinoma. *Front. Oncol.* 10 (2020), 950.