

# MedZip: 3D Medical Images Lossless Compressor Using Recurrent Neural Network (LSTM)

O. H. Nagoor, J. Whittle, J. Deng, B. Mora and M. W. Jones  
Department of Computer Science, Swansea University, Swansea, UK

**Abstract**—As scanners produce higher-resolution and more densely sampled images, this raises the challenge of data storage, transmission and communication within healthcare systems. Since the quality of medical images plays a crucial role in diagnosis accuracy, medical imaging compression techniques are desired to reduce scan bitrate while guaranteeing lossless reconstruction. This paper presents a lossless compression method that integrates a Recurrent Neural Network (RNN) as a 3D sequence prediction model. The aim is to learn the long dependencies of the voxel's neighbourhood in 3D using Long Short-Term Memory (LSTM) network then compress the residual error using arithmetic coding. Experimental results reveal that our method obtains a higher compression ratio achieving 15% saving compared to the state-of-the-art lossless compression standards, including JPEG-LS, JPEG2000, JP3D, HEVC, and PPMd. Our evaluation demonstrates that the proposed method generalizes well to unseen modalities CT and MRI for the lossless compression scheme. To the best of our knowledge, this is the first lossless compression method that uses LSTM neural network for 16-bit volumetric medical image compression.

## I. INTRODUCTION

Hospitals acquire a massive amount of data on a daily basis. Among this information, medical images contain a large amount of valuable data, which also consumes a vast amount of storage. According to the UK NHS Diagnostic Imaging Dataset Statistical Release, up to 23.5 million X-rays, 3.74 million Magnetic Resonance Imaging (MRI), and 5.67 million Computed Tomography (CT) scans were produced with a total of 45 million medical images were acquired in a one-year period (Sep 2018 - Sep 2019) [1]. Radiologists use these scans for clinical purposes, including diagnosis or pre-surgery planning. Therefore, keeping these scans' quality and accuracy for accurate diagnosis while reducing storage size form a significant challenge. Although lossy compression methods may achieve a higher compression ratio, they are not recommended in the medical imaging domain due to the loss of critical data or introduction of artefacts, which could influence data fidelity and result in misleading diagnosis [2]. Maintaining quality and accuracy while reducing the file size form the primary intention of lossless compression, which reduces storage overhead and guarantees exact reconstruction.

Compression performance depends on the type of data redundancy which is classified into three main types, spatial, coding, and spectral (psycho-visual) redundancy [3]. Each type represents the correlation between different parts of the data, and by reducing such repetition, a coder gains compression.

Deep learning approaches form a promising and emerging research direction due to its ability in estimating non-linear

transformations and likelihood data estimation. Current state-of-the-art deep learning methods illustrate that neural networks can construct lossy and lossless performance with results comparable to or even better than standard linear codecs. In this work, we propose a lossless compression approach using LSTM specifically for 16 bit volumetric medical images.

Our proposed lossless compression framework integrates a recurrent neural network (LSTM) to learn spatial correlations of neighbourhood sequences within the 3D regions followed by an entropy encoding for the residual errors. A high-level overview of our proposed lossless compression framework is given in Fig. 1. Given a set of neighbouring voxels, the LSTM model predicts the next intensity value in the sequence. The prediction error is calculated as the difference between the Ground Truth (GT) voxel and the predicted one. If many of the error values are negligible (equal zero), this indicates that the model is accurate. All prediction errors are compressed using entropy coding such as Huffman and arithmetic coding to reduce the coding redundancy. Storing and transmitting the compressed error reduces the file size compared to the GT volume. The contributions of this paper are as follows:

- A novel lossless compression method using LSTM recurrent neural network cells that achieves a higher compression ratio compared to state-of-the-art lossless compression methods.
- Our proposed approach MedZip is for a domain-specific application, namely, volumetric medical images (with 16 bit-depth).
- We demonstrate the generalization of our proposed models on many datasets for higher dynamic range.

The rest of this paper is organized as follows. Section II highlights the current research trends in data compression, including non-learned lossy and lossless compression literature and deep learning compression literature. An overview of the proposed methodology and models architectures are given in Section III. A description of the dataset details is presented in Section IV. The same section describes the experimental and implementation details of our LSTM models, including input sequence specifications and sampling scheme. A subsection in IV provides the compression results of our proposed models compared to the state-of-the-art lossless compression methods. Finally, section V concludes the main objectives of this paper.

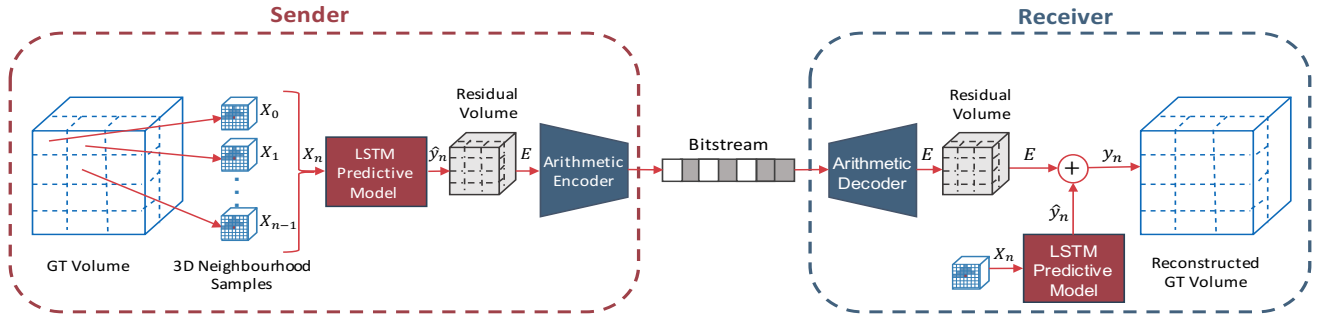


Fig. 1. An overview of our proposed lossless compression framework using LSTM

## II. RELATED WORK

### A. Compression Type

Data compression techniques can be divided into two main types – lossy and lossless compression. Choosing which type to use relies on the application requirements. For medical image compression, the lossless approach is more appropriate since it recovers the original data without any loss in quality.

### B. Entropy

Information theory quantifies the amount of information present in random variables and distributions. Data compression is a field of information theory interested in techniques for reducing the size of a data stream to overcome the bottleneck of data storage, transmission and processing. In the context of data compression, information theory is applied to find the optimal codec that achieves the best compression and the lowest entropy. The more deterministic the data distribution is, the lower entropy the coder gains [4], [5].

Lossless compression is known as reversible compression, which aims to send a stream of data  $m$  with a codelength bounded to the entropy  $L(m) \geq H(m)$ . Given a *message*  $m \in M = \{m_1, m_2, \dots, m_N\}$  with probability distribution  $P_{data}(m)$ , a *sender* will encode the data  $m$  into a sequence of bits with a codelength  $\hat{L}(m)$ . The encoder uses a probabilistic model  $\hat{P}(m)$  to define the codelength  $-\log \hat{P}(m)$  for each symbol  $m_i$  in the stream. If the codelength  $l(m_i)$  is equal to  $-\log P_{data}(m_i)$  for all symbols in the message, the encoder achieves the maximum compression (entropy):

$$H(m) = - \sum_{m_i \in M} \hat{P}(m_i) \log P(m_i) \quad (1)$$

When the *receiver* gets the compressed bitstream, a decoding or reverse operation is applied and the original data  $m$  is fully reconstructed without any loss of information.

### C. Sequence Models

A sequence model is a particular type of neural network that learns a mapping function  $f(x)$  from an input  $x_t$  with sequence length  $l$  to an output  $y_t$ . Such networks maintain an internal state or memory unit, while the sequences are fed through. This internal state is required to aggregate all the information in the sequences through time. Recurrent Neural

Network (RNN)[6], LSTM [7], GRU [8], and Transformers [9], [10], [11] are some of the recent deep learning sequence models.

### D. Classic Approaches

The current state-of-the-art (non-learned) algorithm for medical images lossless compression relies on a hand-crafted scheme, which can be categorised into a prediction-based or transform-based algorithm. If a codec is computed in 2D space (i.e. image), it is classified as image encoder. However, coders computed over three-dimensions are volumetric encoders. Most algorithms involve quantisation or entropy coding for further coding reduction. In adaptive prediction-based approaches, the encoding of values is usually context-sensitive on estimating the local gradients. The encoding mode is selected based on a local causal template. Lossless and Near-Lossless Compression of Continuous-Tone Still Images (JPEG-LS) [12], context-based adaptive lossless image codec (CALIC) [13], and Minimum Rate Predictor (MRP) [14] are some of the standard lossless predictive image codecs. Alternatively, the transform-based algorithm transform an image spatial domain into frequency domain to make it more compressible. A well-known transform-based image coder is Joint Photographic Experts Group 2000 (JPEG2000) [15], [16], which applies a Discrete Cosine Transform (DCT) and Discrete Wavelet Transform.

Most of the volumetric encoders (3D codecs) were introduced by extending the image-based codecs functionality to higher dimensional space. These codecs exploit inter-frames correlation as well as spatial correlation, which further enhances compression performance. JP3D is an extension of JPEG2000 (Part 10), which upgrades wavelet transform to 3D known as 3D-DWT [17]. Another 3D codec that enhances an existing 2D codec (CALIC) is known as 3D-CALIC [18], which gains higher compression performance. 3D-CALIC involves interband and intraband correlation approaches for multispectral images, which achieves better exploits of redundancy in a 3D context. An improved and further optimised version of 3D-CALIC was presented by Magli et al. known as M-CALIC [19]. The essential contribution of that version is achieving near-lossless and lossless performance for hyperspectral data with a multiband algorithm. A more efficient lossless encoder

for medical imaging volumes and an upgrade of the MRP method was presented by Lucas et al. [20], which is known as 3D Minimum Rate Predictor (3D-MRP). The experimental results of this method earned higher compression efficiency compared to the other lossless compression literature for 16-bit medical image resolution. High-Efficiency Video Coding (HEVC) is a standard lossless coder for video compression purposes [21]. Although this coder was developed originally for video compression, it can be applied to 3D medical images compression using Range Extension [22] with 4:0:0 chroma format for one colour component 16-bit data.

The classical hand-crafted codecs may have limited ability in representing non-linear correlations or high-dimensional data distribution. This critical limitation rises the demand for new compression approaches with higher flexibility and generalizability in representing nonlinearity. Recently, state-of-the-art deep neural networks models demonstrate great potential in representing high-dimensional data distribution for both lossy and lossless compression performance. Moreover, a higher compression ratio can be achieved using deep learning methods compared to traditional linear methods.

### E. Learning Based Approaches

Deep learning methods form a promising research direction for data compression. Training ability, learning capacity, model generalization and excellent performance in modelling non-linearity and higher-dimensional data are the most compelling features of deep learning approaches. The research development on compression is classified into developing modelling techniques, coding algorithms or a combination of both (learning an end-to-end compression framework).

1) *Lossy*: Most state-of-the-art deep learning approaches address lossy reduction to assist purposes such as dimensionality reduction (autoencoders), estimating pixel likelihood (autoregressive), and generative compression. A comprehensive survey on learning lossy image compression methods was conducted by [23]. Dimensionality reduction is accomplished by learning an invertible mapping of the data distribution to the quantized latent space, which eventually will losslessly be compressed. Such a mapping is usually estimated using compressive autoencoders [24], variational autoencoders [25] [26], and LSTM-based autoencoders [27] [28]. An end-to-end lossy compression method using encoder-decoder RNN-based architecture combined with an entropy coding was presented by Toderici et al. [28]. Sushmit et al. [29] provide another learnable encoder-decoder based on convolutional Recurrent Neural Networks (RNN-Conv) specifically for chest X-ray images. Estimating image probability distribution based on pixel likelihood is usually known as an autoregressive likelihood model, which is one of the compelling methods in context modelling. Using this approach, predicting any pixel relies conditionally on the probability distributions of all its previous pixels. This dependency process may affect the overall model complexity due to its sequential implementation. Recent approaches reduce the complexity by applying parallelized and dependency reduction approaches or adding regularization

to enhance the training process. Some of the state-of-the-art autoregressive models are PixelCNN [30], PixelCNN++ [31], PixelRNN [32], and Multiscale-PixelCNN [33].

2) *Lossless*: While a considerable amount of effort has been made to address lossy performance using neural networks, less attention was paid to achieve lossless compression. DeepZip [34] is a lossless compression framework that uses RNN (LSTM/GRU) as a probability estimator for compressing genomics and text data. This statistical model combines RNN with arithmetic coder to learn the casual distribution of a symbol based on the previously encountered symbols. Compared to their method, we formulate the lossless compression as a regression problem, not as a classification problem. Our sequence prediction models predict the target voxel intensity given a sequence of samples from its neighbourhood. Another difference is that all our proposed MedZip models are trained once and generalize to unseen medical datasets. Bit-Swap [35] is another lossless method which provides an efficient factorized compression and decompression scheme using a recursive version of Bits-Back coding method [36]. Bits-Back coding integrates an Asymmetric Numerical System (ANS) as an entropy coder. Bit-Swap has a density estimator, which performs a bit-rate closely matches the negative ELBO on average using a hierarchical latent variable model with Markov chain structure.

## III. METHODOLOGY

### A. Overview

As the LSTM model is one of the state-of-the-art sequence models, we formulated our proposed lossless compression approach as a supervised sequence prediction problem and integrated the LSTM model as 3D sequence predictor (see Fig. 1). Our LSTM model solves a many-to-one sequence prediction problem, which takes a sequence of 3D neighbouring voxels as input and predicts the next intensity value. The intention of using LSTM memory cell to solve the sequence prediction is its ability to maintain the gradient flow across the cells in a way considerably better than RNN. The gates mechanisms in LSTM controls updating the long dependencies with less potential to get gradient vanishing or exploding problems. Such flow control (gates) allows the internal memory of LSTM to learn the long-term inter-frame correlation between slices as well as the spatial correlation. As shown in Fig. 1, both sender and receiver have the same LSTM predictive model with the same architectures and sharing the same weights. Given a data distribution defined over  $V \in R^N$ , we extract training sequence samples  $X_n \subset V$ , where each  $X_i = \{x_1, x_2, \dots, x_{l-1}\}$  is a flattened vector containing 3D neighboring intensities of voxel  $y_i$  and  $l$  is the sequence length of the 3D neighborhood. The LSTM model is trained to learn a differentiable mapping function  $\hat{y}_i = f(X_i)$  that maps a sequence input  $X_i$  to a single output  $\hat{y}_i$  to minimize the difference with  $y_i$  (ground truth voxel). After training, the

model applies compression, with the sender computing the residual error (prediction error)  $E$  for each voxel as

$$E = y_n - \hat{y}_n \quad (2)$$

The error is compressed with arithmetic coding and sent or stored with a lower bit-rate. To reconstruct the original data by the receiver, a reversed operation is applied by the arithmetic decoder, which losslessly reconstructs the residual error  $E$ . The LSTM model is used to generate predictions  $\hat{y}_n$ , which are added to the error values  $E$  as illustrated in figure 1.

### B. Network Architecture

The proposed models are Vanilla LSTM models, which are composed of the input layer, LSTM layer with 128 cells, and a linear output layer. The activation functions of the LSTM cells are sigmoid and tanh.

Inspired by [37], a joint loss  $L_{joint}$  (Eq. 3) was chosen as the loss function for all the proposed model. This loss function jointly combines the Mean Absolute Error (MAE) (Eq. 4) with the Pearson Correlation Coefficient (PCC) (Eq. 5) also known as bivariate correlation

$$L_{Joint} = MAE + \lambda(1 - |PCC|) \quad (3)$$

$$MAE = \frac{\sum_{i=1}^n |y - \hat{y}|}{n} \quad (4)$$

$$PCC = \frac{cov(y, \hat{y})}{\sigma_y \sigma_{\hat{y}}} \quad (5)$$

Where,  $cov$  is the covariance,  $\sigma_y$  is the standard deviation of  $y$ , and  $\sigma_{\hat{y}}$  is the standard deviation of  $\hat{y}$ . The intention of using PCC in the loss function is to measure the statistical relationship between the ground truth value  $y$  and  $\hat{y}$ . When  $PCC = 0$  this means no linear correlation between the two continuous variables. In the  $L_{joint}$ , the absolute value of PCC is computed, which limits the PCC value to be less than or equal to 1. If the value is equal to 1, it means the variables are linearly correlated. Incorporating PCC with the MAE for solving the regression problem has a significant impact in enhancing the accuracy and stabilizing the training.

### C. Local Sampling

In a sequence prediction model, the shape and amount of the input sequence form a crucial role in learning the mapping function of the data distribution to the target output. The sequence length is determined by the number of previous observations or features imposed with some explicit order that a model analyses before making a prediction. However, within a recurrent neural network, the challenge that arises is as the sequence length increases, the model will suffer from the vanishing or exploding gradient problem. Practically, by breaking the input into a smaller fixed sequence length, this problem can be avoided. Image compression generally utilises the four previous neighbouring pixels to encode the target pixel. Video and 3D predictor codecs involve neighbouring pixels from previous frames to discover temporal (inter-frame) redundancy.

In our proposed sequence prediction model, we formulate the input sequence to utilise 3D neighbours (i.e. integrating information from previous slices) with a decreased number of voxels. We refer to this sequence pattern as 3D pyramid (see Fig. 2.b). A comparison to block based models is given in section IV. The aim is to learn 3D spatial correlation within the volume space by fetching a fixed-length representative input sequence that can lead to optimal compression. We choose a 3D pyramid sequence with  $(13 \times 13, 9 \times 9, 5 \times 5, 1 \times 1)$  regions on each slice to be the input sequence to our models. To choose the training set we sampled uniformly through multiple volumes with a specific pixel spacing. For each voxel in the set, we extract the 3D pyramid neighbouring sequences (edge voxels are padded with minimum intensity). All volume values are normalized to the range  $[-1, 1]$ . Each sequence prediction model is trained to make one-step ahead prediction until convergence.

## IV. EXPERIMENTAL RESULTS AND DISCUSSION

We used the compression ratio in Bit-per-pixel (BPP) presented in equation 6 as our evaluation criteria to compare the performance of our method to the state-of-the-art lossless compression methods.

$$BPP = \frac{Compressed\ Image\ Size\ (Bits)}{Number\ Of\ Voxels} \quad (6)$$

It is important to understand that we compare the compression size of the **volumetric residual**, the prediction error computed with equation 2 (generated by our proposed LSTM models) to the compressed size of the **GT medical volume** (generated by state-of-the-art lossless compression methods).

### A. Dataset

The two datasets used in this paper consist of a set of DICOM files with two different modalities Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) stored in 16-bit grayscale images. Dataset1 is a private dataset, which contains 43 volumes generated by the same hospital and represents CT scans for a patient's entire trunk. All the scans have the same width and height  $512 \times 512$ ; however, they differ in the depth of the volume  $z \in [750, 1120]$ . The slice thickness is  $0.625mm$  in all patients in Dataset1. The pixel spacing **varies** between patients  $PS \in \{.488, .578, .625, .703\}$ . Intensity values range from -1024 to 3071 (12 bits stored as 16 bit integer). The proposed three models were trained on random subsets extracted from volumes with a specific pixel spacing (i.e. MedZip1 and MedZip2) or a subset obtained from various pixel spacing (i.e. MedZip3 training set). Table I presents an overview of the training set for each model. The evaluation was conducted on TestSet1, which contains the rest of the volumes belonging to Dataset1. We also evaluate our three proposed models on a public dataset (TestSet2), which has a different modality (MRI) and represents a different part of the patient's body, namely, head and neck [38], [39]. TestSet2 contains 12 volumes stored in 16-bit grayscale images. All volumes have  $512 \times 512 \times 120$  resolution, slice thickness  $2mm$ ,

Model ID	Training Set (Pixel Spacing)	Slice Thickness	Hyper Parameters	Preprocessing
MedZip1	Random samples from volumes with pixel spacing .488	.625	Batch size=128, & learning rate=0.00005	-3D pyramid neighboring sequence with $(13 \times 13, 9 \times 9, 5 \times 5, 1 \times 1)$ sequence size. -All samples values are normalized between $[-1, 1]$ .
MedZip2	Random samples from volumes with pixel spacing .625	.625	Batch size=128, & learning rate=0.00005	
MedZip3	Random samples from volumes with pixel spacing .488, .578, .625	.625	Batch size=128, & learning rate=0.0001	

TABLE I

AN OVERVIEW OF THE TRAINING SET AND THE TRAINING HYPER PARAMETERS FOR EACH OF THE PROPOSED LSTM MODELS.

and  $PS = .5$  pixel spacing. The minimum intensity value is 0, and the maximum is 689.

### B. The Proposed Models

This subsection provides the experiential and training details for the three proposed sequence prediction models. For all these models, we used a vanilla LSTM architecture (see section III-B). Since the capacity of the recurrent network is determined by the number of memory cells the network has, we found the LSTM with 128 cells has enough capacity to learn the 3D voxel correlation. This relatively compact network reduces the overhead of model size.

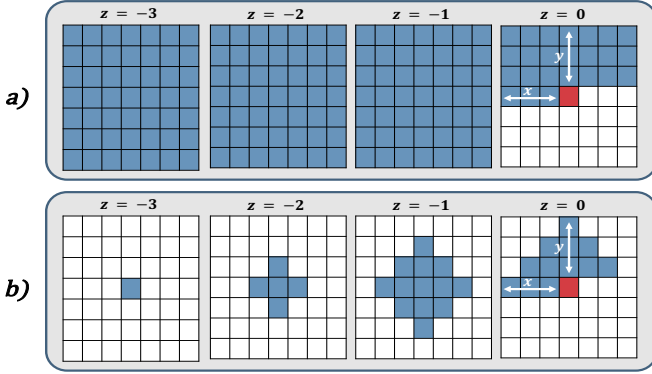


Fig. 2. Two different neighborhood shapes: a) 3D Cube Neighboring Sequence and b) 3D Pyramid Neighboring Sequence.  $z=0$  represents the current slice. The red voxel refers to the target voxel that needs to be predicted, and blue voxels are the input sequences while the white voxels are ignored (masked)

Neighborhood Block Size	3D Pyramid Neighboring Sequence	3D Cube Neighboring Sequence		
	$(13 \times 13, 9 \times 9, 5 \times 5, 1 \times 1)$	$(5 \times 5 \times 5)$	$(7 \times 7 \times 7)$	$(9 \times 9 \times 9)$
Bits-Per-Pixel (BPP)	4.267	4.702	4.478	4.36
Compression Time (hh:mm:ss)	1:23:58	0:44:51	1:17:13	2:27:47

TABLE II

COMPARING THE COMPRESSION PERFORMANCE (COMPRESSION RATIO (BPP) AND COMPRESSION TIME) OF DIFFERENT NEIGHBORING SEQUENCE (3D PYRAMID & 3D CUBE) WITH DIFFERENT BLOCK SIZES.

Two different 3D neighbourhood shapes were applied to find the input sequence that can lead to an optimal compression, namely, the 3D cube and 3D pyramid neighbouring sequence. Each type introduces a diverse coverage of the block around the target voxel. Figure 2 illustrates an example of each 3D

shape. For both types,  $z=0$  represents the current slice. The sequence only includes voxels from the current slice and the previous slices. The red voxel refers to the target voxel that needs to be predicted, and blue voxels are the input sequences while the white voxels are ignored (masked). In the same figure, a) the 3D cube neighbourhood size is  $(7 \times 7 \times 7)$  while b) the 3D pyramid neighbourhood size is  $(7 \times 7, 5 \times 5, 3 \times 3, 1 \times 1)$ .

The intention of choosing the 3D pyramid neighbouring sequence with this specific shape is built upon applying Manhattan distance from the target voxel to a certain distance in each dimension. The 3D local appearance and structures tend to be highly correlated, which can lead to spatial and interframe redundancy. Experimentally, different lengths to the target voxel were applied to select the 3D neighbouring size with the best compression performance as presented in Table II. Both compression ratio in (BPP) and compression time have been computed to select the optimal input sequence. As illustrated in Table II, the pyramid structure was compared to a full block neighbourhood at  $(5 \times 5 \times 5)$ ,  $(7 \times 7 \times 7)$ ,  $(9 \times 9 \times 9)$  block sizes. As expected, with the increase in the 3D cube block size, the compression rate also increases as well as the compression time due to the longer sequence length. The 3D pyramid neighbourhood demonstrates a great balance between the compression time and overall compression achievement. Therefore, the 3D pyramid sequence with  $(13 \times 13, 9 \times 9, 5 \times 5, 1 \times 1)$  sequence size was used as input for all the proposed models. We found that compared to using a full cube block, there was no performance loss in terms of size of compressed file and the training time was substantially reduced because fewer samples were used.

The training hyper-parameters and training set specifications of the three models are provided in Table I. As Dataset1 has variance in the pixel spacing between volumes, we empirically investigate the effect on compression ratio when training on different spacing. MedZip1 was trained on samples selected randomly from volumes with  $[\text{.488}, \text{.488}]$  pixel spacing. MedZip2 was trained on samples from volumes with  $[\text{.625}, \text{.625}]$  spacing. MedZip3 was trained on subsets sampled equally from volumes with all different spacing  $[\text{.488}, \text{.488}]$ ,  $[\text{.578}, \text{.578}]$  and  $[\text{.625}, \text{.625}]$  pixel spacing. Both MedZip1 and MedZip2 were trained with batch size=128 and learning rate=0.00005. However, MedZip3 training hyper parameters are batch size=128 and learning rate=0.0001. All LSTM networks were optimized by Adam optimizer with  $\beta_1 = 0.9$ , and  $\beta_2 = 0.999$ . The parameter  $\lambda$  in the joint loss  $L_{joint}$  (equation 3) was set to  $\lambda = 1$ , which weights the contribution of the two losses to be the same.

### C. Comparisons with the state-of-the-art

In this subsection, we evaluate the compression performance of our three proposed models on two different medical images datasets (TestSet1 and TestSet2). We compare the compression size of our models to the state-of-the-art lossless compression methods. The selected methods include image-based codecs, namely, JPEG-LS [40], and JPEG2000 (OpenJPEG software) [41] and 3D volumetric codecs – JP3D [41], and HEVC (HM-SCC-extensions-4998) using lossless configuration with main-RExt profile available in [42] [22]. The evaluation compression also includes the Prediction by Partial Matching (PPMd) algorithm [43] with ultra compression level. We have benchmarked our approach against a deep learning lossless compression method known as LSTM-Compress [44].

**TestSet1.** The evaluation over TestSet1 is illustrated in Fig.3. The volumes are classified into four sets based on their pixel spacing value – separated with the horizontal black lines. Each row represents a volume except for the last row which is the average of (BPP) for each method through all volumes. The first column presents the pixel spacing of each volume in Dataset1. Cells in the first column are highlighted from minimum pixel spacing .488 (White) to maximum spacing .703 (Black). Each other column represents the bits per pixel (BPP) of a lossless compression method, including PPMd, JPEG-LS, JPEG2000, HEVC, JP3D, MedZip1, MedZip2, and MedZip3, respectively. Cells are highlighted from the maximum compression 3.837 BPP (Blue) to lowest compression 6.236 BPP (Red). Overall, the performance of our RNN models illustrate the best improvement in compression compared to the standard algorithms. Among the standard codecs, JP3D has a better bit-per-pixel than the image-based codecs (i.e. JPEG-LS and JPEG2000) through its exploitation of 3D correlation within frames. HEVC 3D-codec gave worse than average compression results on volumes with pixel spacing .488 and .57 but slightly better performance than image-codecs (JPEG-LS and JPEG2000) on volumes with larger pixel spacing. For the standard compression methods, PPMd produces the least compression ratio overall on volumes in TestSet1. Of our new methods, MedZip1 has the best compression ratio for volumes with .488 pixel spacing – the spacing it was trained-on. Less performance was gained on volumes with larger pixel spacing but still better than classical approaches. MedZip2, which was trained on .625 pixel spacing, gains a considerably good compression ratio overall regardless of volumes pixel spacing. MedZip3 has a good compression performance overall on volumes with various pixel spacing demonstrating generalization. The robustness of this model was gained by training it on samples that belong to volumes with different pixel spacing.

**TestSet2.** To demonstrate the robustness and the generalization ability of our proposed LSTM models, we evaluated our models’ performances on an out of the domain public dataset that has a different modality (MRI) and represents a different part of a patients body (head and neck). We also compare against a deep learning lossless compressor known as

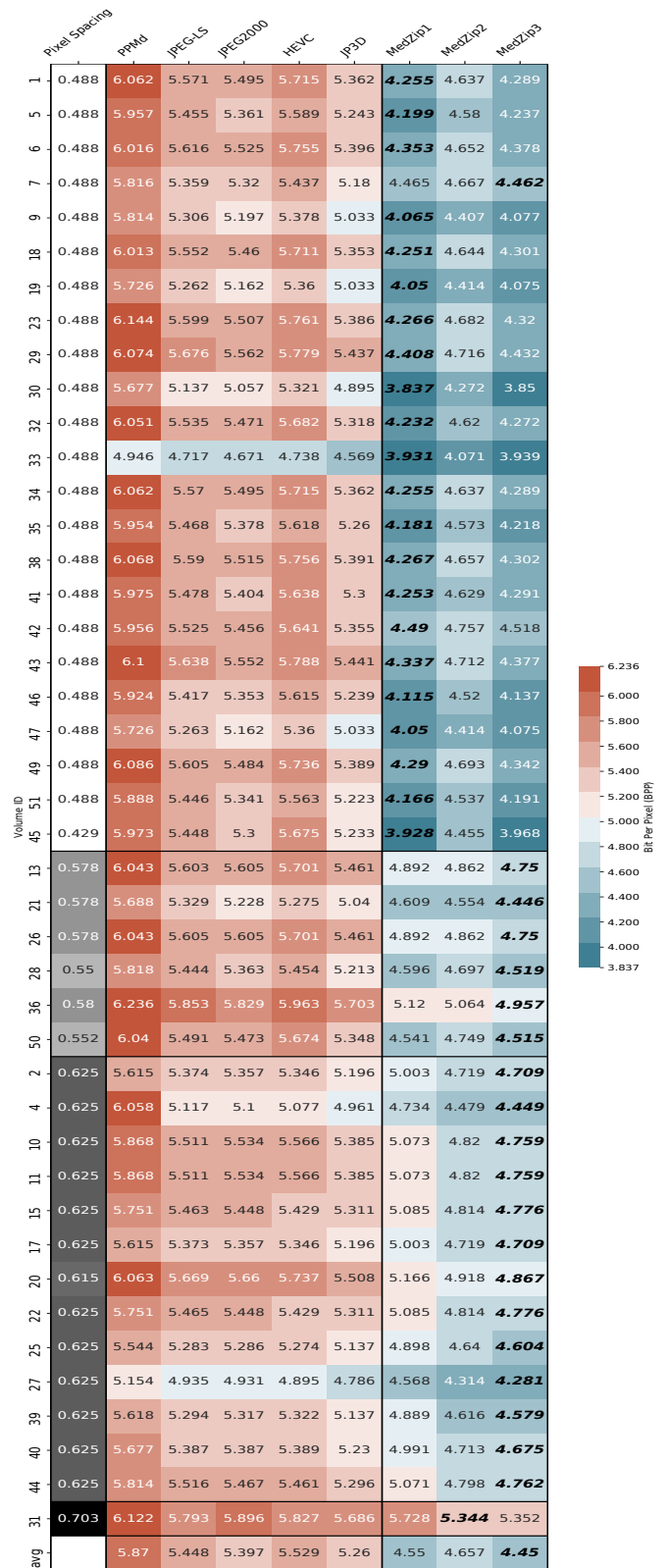


Fig. 3. Bits-per-pixel (BPP) for each lossless compression method on TestSet1. The first column is colour mapped by the pixel spacing value of each volume. The other cells are highlighted from the maximum compression 3.837 BPP (Blue) to minimum compression 6.236 BPP (Red). For the average row, MedZip3 saves 15.4% space compared to the best performer (JP3D).

LSTM-Compress. The evaluation over TestSet2 is illustrated in Fig. 4. The first column illustrates the pixel spacing of TestSet2 which is the same for all volumes (0.5). Each other column represents the bits per pixel (BPP) of a lossless compression method, including PPMd, JPEG-LS, JPEG2000, HEVC, JP3D, LSTM-Compress, MedZip1, MedZip2, and MedZip3, respectively. The last row is the average of (BPP) for each method through all volumes. Cells in other columns are highlighted from the maximum compression 2.949 BPP (Blue) to minimum compression 4.52 BPP (Red). Among the standard codecs, JPEG2000 yields the best bit-per-pixel results followed by JP3D. Unexpectedly, HEVC, which can exploit inter-frame redundancy, gains only better compression than PPMd. Additionally, PPMd algorithm has the worst compression compared to the other methods. Compared to the state-of-the-art techniques, the deep learning approach (LSTM-Compress) achieves better compression. Overall, the proposed RNN approaches gained best compression bit-rate when compared to the standard codecs and the deep learning method (LSTM-Compress). Although our models were not trained on this dataset, they have still obtained the best BPP reduction.

In comparison to the deep learning method (LSTM-Compress), our proposed MedZip models demonstrate the best compression ratio and additionally reduce compression time showing the strategy of using 3D neighbourhood information works well to predict voxel values. MedZip models are notably faster than LSTM-Compress (2.26 hours vs 56.66 hours) to compress all of the volumes in Testset2. In terms of compression ratio over Testset2, MedZip has a size reduction of up to 11% compared to LSTM-Compress. Since each volume takes around 30 hours to be compressed by LSTM-Compress, we compared the compression performance of LSTM-Compress on seven volumes selected randomly from TestSet1. The average compression performance of MedZip3 on those volumes is 4.50 BPP while LSTM-Compress is 5.13 BPP with MedZip3 showing a saving of 12% over LSTM-Compress.

In Fig. 5, a summary of the compression performance over the two test sets is computed for each of the lossless methods. Each row represents the performance of a single compression method over different datasets. Cells are coloured from the maximum compression percentage 100.00% (Blue) to the minimum performance 136.56% (Red) – (Less value indicates better performance). Among the existing codecs, JP3D proposes the best compression performance on TestSet1. However, over TestSet2, JPEG2000 gains the best reduction. An unexpected performance was given by the 3D codec HEVC, which produces only better compression than PPMd on all datasets. Although the compression level of PPMd was set to Ultra setting, this algorithm performs the worst result on average on both datasets. By observing the results of Fig. 5, the overall compressing performance of our MedZip models illustrates the best reduction over the different datasets. MedZip3 outperforms MedZip2 with better reduction up to 5%. The advantage of MedZip1 is more noticeable, reaching

a gain of almost 36% over the PPMd. To conclude, MedZip3 performs the state-of-the-art compression percentage 100.00% on 16-bits volumetric medical images.

Volume ID	Pixel Spacing	PPMd	JPEG-LS	JPEG2000	HEVC	JP3D	LSTM-Compress	MedZip1	MedZip2	MedZip3
1	0.5	4.245	3.941	3.558	4.079	3.707	3.424	<b>3.053</b>	3.267	3.082
2	0.5	4.246	3.966	3.57	4.068	3.702	3.455	<b>3.114</b>	3.294	3.133
3	0.5	4.281	3.969	3.572	4.096	3.709	3.458	<b>3.1</b>	3.299	3.122
4	0.5	4.133	3.875	3.487	3.99	3.621	3.379	<b>3.014</b>	3.212	3.037
5	0.5	<b>4.52</b>	4.158	3.757	4.273	3.885	3.654	<b>3.295</b>	3.478	3.314
6	0.5	4.101	3.827	3.438	3.96	3.579	3.338	<b>2.949</b>	3.172	2.976
7	0.5	4.163	3.899	3.516	4.025	3.662	3.428	<b>3.099</b>	3.279	3.116
8	0.5	4.327	4.04	3.644	4.137	3.776	3.536	<b>3.222</b>	3.382	3.238
9	0.5	4.201	3.922	3.534	4.034	3.679	3.459	<b>3.142</b>	3.304	3.158
10	0.5	4.257	3.969	3.581	4.101	3.719	3.483	<b>3.137</b>	3.32	3.155
11	0.5	4.153	3.879	3.503	4.029	3.655	3.409	<b>3.012</b>	3.236	3.047
12	0.5	4.073	3.803	3.42	3.899	3.553	3.326	<b>2.991</b>	3.17	3.007
avg		4.225	3.937	3.548	4.058	3.687	3.446	<b>3.094</b>	3.284	3.115

Fig. 4. Illustrating the compression ratio in BPP for the proposed models compared to the state-of-the-art lossless compression methods on TestSet2 (16-bits volumes). The first column is colour mapped by the pixel spacing value of each volume. The other cells are highlighted from the maximum compression 2.949 BPP (Blue) to minimum compression 4.52 BPP (Red).

Compression Methods	TestSet1	TestSet2
PPMd	131.92%	136.56%
JPEG-LS	122.43%	127.26%
JPEG2000	121.28%	114.69%
HEVC	124.27%	131.15%
JP3D	118.22%	119.18%
MedZip1	102.26%	<b>100.00%</b>
MedZip2	104.67%	106.15%
MedZip3	<b>100.00%</b>	100.69%

Fig. 5. A summary overview of the compression performance over the two test sets for all the lossless methods. Cells are coloured from the best compression performance 100.00% (Blue) to the worst performance 136.56% (Red). (Less value indicates better performance).

## V. CONCLUSION

This paper proposed a novel lossless compression approach using LSTM, specifically for compressing 3D medical images (16 bit-depths). We solved the lossless compression problem using LSTM-based sequence prediction model and 3D pyramid shaped sequences as models input. MedZip empirically demonstrates a significant reduction in the compressed size (BPP) when compared to the state-of-the-art lossless compression methods. Additionally, our pre-trained LSTM model generalized well to unseen modality (MRI) and achieves a higher compression ratio compared to the other methods. Moreover, we believe that the proposed models would achieve more improvement by integrating it with attention-based mechanisms.

## REFERENCES

- [1] S. Dixon, "Diagnostic imaging dataset statistical release," NHS England, UK, Tech. Rep., Sep. 2018.
- [2] D. Koff and H. Shulman, "An overview of digital compression of medical images: Can we use lossy image compression in radiology?" *Canadian Association of Radiologists journal*, vol. 57, pp. 211–7, 11 2006.
- [3] A. Hussain, A. Al-Fayadh, and N. Radi, "Image compression techniques: A survey in lossless and lossy algorithms," *Neurocomputing*, vol. 300, pp. 44 – 69, 2018.
- [4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [5] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.
- [6] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.
- [7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [8] J. Chung, Çağlar Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *ArXiv*, vol. abs/1412.3555, 2014.
- [9] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *ICML 2018*, 2018, pp. 4052–4061.
- [10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.
- [11] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.
- [12] M. J. Weinberger, G. Seroussi, and G. Sapiro, "The loco-i lossless image compression algorithm: principles and standardization into jpegs," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, Aug 2000.
- [13] X. Wu and N. Memon, "Calic-a context based adaptive lossless image codec," *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 4, pp. 1890–1893 vol. 4, May 1996.
- [14] I. Matsuda, H. Mori, and S. Itoh, "Lossless coding of still images using minimum-rate predictors," *ICIP 2000*, vol. 1, pp. 132–135 vol.1, Sep. 2000.
- [15] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The jpeg2000 still image coding system: an overview," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, Nov 2000.
- [16] D. Taubman and M. Marcellin, *JPEG2000 image compression fundamentals, standards and practice: image compression fundamentals, standards and practice*. Springer Science & Business Media, 2012, vol. 642.
- [17] P. Schelkens, A. Munteanu, A. Tzannes, and C. Brislawn, "Jpeg2000. part 10. volumetric data encoding," *2006 IEEE International Symposium on Circuits and Systems*, pp. 4 pp.–3877, May 2006.
- [18] Xiaolin Wu and N. Memon, "Context-based lossless interband compression-extending calic," *IEEE Transactions on Image Processing*, vol. 9, no. 6, pp. 994–1001, June 2000.
- [19] E. Magli, G. Olmo, and E. Quacchio, "Optimized onboard lossless and near-lossless compression of hyperspectral data using calic," *IEEE Geoscience and Remote Sensing Letters*, vol. 1, no. 1, pp. 21–25, Jan 2004.
- [20] L. F. R. Lucas, N. M. M. Rodrigues, L. A. da Silva Cruz, and S. M. M. de Faria, "Lossless compression of medical images using 3-d predictors," *IEEE Transactions on Medical Imaging*, vol. 36, no. 11, pp. 2250–2260, Nov 2017.
- [21] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.
- [22] "Hevc format range extension (rext)." <https://hevc.hhi.fraunhofer.de/rext>, accessed: 15.07.2019.
- [23] Y. Hu, W. Yang, Z. Ma, and J. Liu, "Learning end-to-end lossy image compression: A benchmark," 2020, arXiv preprint <http://arxiv.org/abs/2002.03711>.
- [24] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," *International Conference on Learning Representations*, vol. abs/1703.00395, 2017.
- [25] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *International Conference on Learning Representations*, vol. abs/1802.01436, 2018.
- [26] L. Zhou, C. Cai, Y. Gao, S. Su, and J. Wu, "Variational autoencoder for low bit-rate image compression," in *CVPR Workshops*, 2018.
- [27] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," *International Conference on Learning Representations*, vol. abs/1511.06085, 2015.
- [28] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *CVPR 2017*, July 2017, pp. 5435–5443.
- [29] A. S. Sushmit, S. U. Zaman, A. I. Humayun, T. Hasan, and M. I. H. Bhuiyan, "X-ray image compression using convolutional recurrent neural networks," in *2019 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, May 2019, pp. 1–4.
- [30] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with pixelcnn decoders," in *NIPS*, 2016, pp. 4790–4798.
- [31] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications," *International Conference on Learning Representations*, 2017.
- [32] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *ICML 2016*, 2016, pp. 1747–1756.
- [33] S. Reed, A. van den Oord, N. Kalchbrenner, S. G. Colmenarejo, Z. Wang, Y. Chen, D. Belov, and N. de Freitas, "Parallel multiscale autoregressive density estimation," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2912–2921.
- [34] M. Goyal, K. Tatwawadi, S. Chandak, and I. Ochoa, "Deepzip: Lossless data compression using recurrent neural networks," in *2019 Data Compression Conference (DCC)*, March 2019, pp. 575–575.
- [35] F. H. Kingma, P. Abbeel, and J. Ho, "Bit-swap: Recursive bits-back coding for lossless compression with hierarchical latent variables," in *ICML 2019*, 2019, pp. 3408–3417.
- [36] J. Townsend, T. Bird, and D. Barber, "Practical lossless compression with latent variables using bits back coding," in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=ryE98iR5tm>
- [37] J. Whittle and M. W. Jones, "A deep learning approach to no-reference image quality assessment for monte carlo rendered images," *CGVC 2018*, pp. 23–31, Sept 2018.
- [38] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior, "The cancer imaging archive (tcia): Maintaining and operating a public information repository," *Journal of Digital Imaging*, vol. 26, no. 6, 2013. [Online]. Available: <https://doi.org/10.1007/s10278-013-9622-7>
- [39] C. Cardenas, A. Mohamed, G. Sharp, M. Gooding, H. Veeraraghavan, and J. Yang, "The cancer imaging archive," *Data from AAPM RT-MAC Grand Challenge*, 2019. [Online]. Available: <https://doi.org/10.7937/tcia.2019.bcfjqfqb>
- [40] "Dcmtk is a collection of libraries, which includes dcmcjpls tool for encoding dicom file to jpeg-ls transfer syntax." <https://support.dcmctk.org/docs/dcmcjpls.html>, accessed: 15.07.2019.
- [41] "Openjpeg an open-source jpeg 2000 codec written in c." <https://www.openjpeg.org/>, accessed: 15.07.2019.
- [42] D. Flynn, D. Marpe, M. Naccari, T. Nguyen, C. Rosewarne, K. Sharman, J. Sole, and J. Xu, "Overview of the range extensions for the hevc standard: Tools, profiles, and performance," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 4–19, Jan 2016.
- [43] "7-zip is a file archiver with a high compression ratio." <https://www.7-zip.org/>, accessed: 15.06.2019.
- [44] "Byron knoll. lstm-compress: data compression using lstm." <https://github.com/byronknoll/lstm-compress>, accessed: 1.07.2020.