Classification Accuracies of Malaria Infected Cells Using Deep Convolutional Neural Networks Based on Decompressed Images

Yuhang Dong, Zhuocheng Jiang, Hongda Shen, W. David Pan
Dept. of Electrical and Computer Engineering
University of Alabama in Huntsville
Huntsville, AL 35899

Abstract—**In many biomedical applications, images are stored and transmitted in the form of compressed images. However, typical pattern classifiers are trained using original images. There has been little prior study on how lossily decompressed images would impact the classification performance. In a case study of automatic classification of malaria infected cells, we used decompressed cell images as the inputs to deep convolutional neural networks. We evaluated how various lossy image compression methods and varying compression ratios would impact the classification accuracies. Specifically, we compared four compression methods: lossy compression via bitplane reduction, JPEG and JPEG 2000, and sparse autoencoders. Decompressed images were fed into LeNet-5 for training and testing. Simulation results showed that for similar compression ratios, the bitplane reduction method had the lowest classification accuracy, while JPEG and JPEG 2000 methods could maintain good accuracies. In particular, JPEG 2000 decompressed images could achieve about 95% accuracy even after 30 to 1 compression. We also provide classification results based on the widely used MNIST dataset, where handwritten digits were found to be much easier to classify using decompressed images, with about 90% accuracy still achievable using only one single bitplane. As a lossy compression method, Autoencoder was also applied to the MNIST dataset, achieving about 85% accuracy with a compression ratio much higher than the other three lossy image compression methods. Autoencoders were also found to provide more scalable compression ratios, while capable of maintaining good classification accuracies.**

I. Introduction

Malaria is a potentially fatal parasitic disease of both human and animals. Half of the world population is at risk of this life threatening infectious disease. Malaria is of great danger to pregnant woman and children, especially those under five [1]. In most cases, malaria infection could be diagnosed by microscopic examination of blood films. In order to provide a reliable diagnosis, necessary training and specialized human resource are required. Patients suffering from malaria disease should be diagnosed at early stage and should be given an effective and affordable treatment within 24 hours [2]. Unfortunately, most infections occur in rural areas, where resources are far from being enough. Also failure to diagnose on time may lead to incorrect treatments. This alarming situation has prompted researchers to develop telemedicine solutions for rapid and accurate identification of malaria infection. To give an idea of the red blood cells involved in this study, we show some samples in Fig. 1.

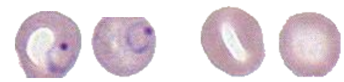

Fig. 1: Red blood cell samples: the two cells on the left are malaria infected, and two cells on the right are non-infected. All these cell images were segmented from a whole slide image scanned by the Department of Pathology, University of Alabama at Birmingham (UAB) [3]. The original image contains more than 900,000 red blood cells, among which around 3,000 are malaria infected. After several morphological operations, isolated cell samples were segmented, which were then curated by pathologists from UAB Medical School [4].

With the emergence of digitized medical images such as the whole slide images [5], more and more images are stored and transmitted through the Internet in the format of compressed images. The purpose of image compression is to reduce the storage space and transmission cost while maintaining good quality. Image compression techniques are categorized into *lossy* compression and *lossless* compression techniques. Lossy compression techniques are capable of offering much higher image compression than their lossless counterparts, albeit at the cost of introducing distortion to the reconstructed images. However, if the decompressed images are used for image classification, the impact of image distortion caused by lossy compression has not been well studied in the literature. Therefore, in this case study, we used decompressed images of red blood cells as the input of an automatic classification system aimed at differentiating between malaria infected red blood cells and those healthy ones.

Studies on malaria cell classification provided many diagnostic methods, most of which were based on machine learning, including unsupervised [6] and supervised learning [7]. However, the performance of these methods are highly sensitive to features extracted from original images. Although many works has been done on feature extraction for malaria cells [6], [7], [8], new feature extraction methods need to be designed for different datasets. In order to achieve fully automated diagnosis without any manual feature extraction, we chose deep convolutional neural network (CNN) as the classifier. CNN can extract hierarchical representations of the input data. In this work, LeNet-5 [9] was used to learn the inherent features of malaria infected and non-infected cells. LeNet-5 [9] is one of the best known CNN architectures. It was first used in handwriting digits recognition and achieved a very low error rate of 0.8%. LeNet-5 consists of three convolutional layers, with two subsampling layers in between, and an output

978-1-5386-1539-3/17/$31.00 2017 IEEE

layer. The convolutional and subsampling layers are organized into planes called feature maps.

It has been shown that bitplanes from different color channels of images contain information that could be used for compression [10] and retrieval [11]. By training CNN using images reconstructed from different bitplanes, we are able to determine how classification accuracy would change accordingly. Similarly, we created reconstructed image datasets using JPEG and JPEG 2000 image compression standards with various compression ratios. Another dataset is achieved by utilizing autoencoder, which can also be viewed as a lossy compressor.

The main novelty of this work is that rather than using original images to train and test the network, we used reconstructed images after compression by four different methods, including bitplane reduction, JPEG, JPEG 2000 image compression methods, and the autoencoder [12]. Four datasets were created using reconstructed images by these compression methods. Autoencoder is an unsupervised learning algorithm. With the help of back propagation, it tries to learn an approximation to the identity function, thereby transforming the input image to an decoded image with the smallest possible amount of distortion. Autoencoders have taken center stage in the deep learning approach. In [12], [13], [14], [15], autoencoders based on Restricted Boltzmann Machines (RBM), are stacked and trained bottom up in a unsupervised mode, followed by a supervised learning stage to fine-tune the entire network. These deep architectures have been shown to provide state-of-the-art performances in many challenging classification and regression problems. In fact, even a single-layer autoencoder is capable of learning a low-dimensional representation of the input image. Thus an autoencoder can be used as a lossy image compressor. We can reconstruct a lossy version of the original image by using the corresponding "autodecoder" based on the learned low-dimensional representations. In our prior work, we designed a four-layer staked autoencoder network to learn the inherent features of red blood cell images [16]. To reduce the training time, we used a simple one-layer autoencoder to encode the input images, and a corresponding one-layer autodecoder to reconstruct the original image with loss. The compression ratios of the autoencoder can be controlled by adjusting the number of neurons.

Similarly, By training the CNN using images reconstructed from different bitplanes, we are able to determine how classification accuracy would change accordingly. It has been shown that bitplanes from different color channels of images contain information that could be used for compression [10] and retrieval [11]. We also created reconstructed image datasets using JPEG and JPEG 2000 image compression standards with various compression ratios.

The rest of this paper is organized as follows. Section II discusses how we compiled new datasets using the four compression methods. The computation platform and simulation results are given in Section III, along with results for the MNIST dataset. Section IV concludes this paper with a discussion of the further work.

II. Reconstructed Image Dataset Compilation

A. Reconstruction of Bitplane Reduction Images

The original cell samples are color images (with three color channels, R, G, and B) of size $60 \times 60 \times 3$, with each pixel being represented by eight bits. In bitplane coding, if only one single bitplane in the R (Red) channel is kept, then all other seven bitplanes will be set to zero, corresponding to an 24 to 1 compression. During image reconstruction, we convert bitplanes back to their corresponding decimal representations. The reconstructed image would be a lossy version of the original image, since only one single bitplane in R channel was retained. In this fashion, each sample image can be decomposed into 24 bitplanes, each representing the nth (from the 1st to the 8th) bitplane in one of the RGB channels. Reconstruction of these bitplanes led to the creation of 24 reconstructed image datasets.

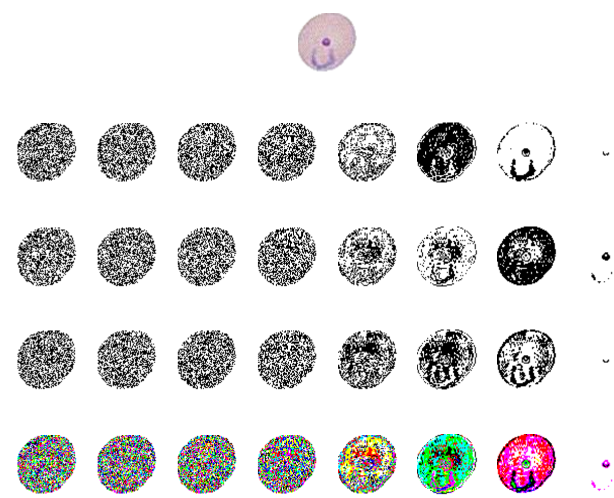

Fig. 2: Bitplane images. The image in the top row is the original malaria infected cell. In each of the lower four rows, there are eight bitplane images, with the leftmost column representing the least significant bitplane (LSB) and the rightmost column the most significant bitplane (MSB). Moreover, the second to forth row are bitplane images from R, G and B channels, respectively. The bottom row are bitplanes for combined RGB channels. Note that in this particular example, the most significant bitplanes retain less features (e.g., the characteristic ring form of the parasite in an infected cell) than the second most significant bitplanes, due to a majority of pixels in the original image have values above 128.

Alternatively, we can choose to retain the co-located bitplane for three color channels simultaneously, leading to 8:1 compression. In this way, we created eight more reconstructed image datasets.Note that while bitplanes are bi-level images, decompressed bitplane images have the same size as the original images. Fig. 2 shows some example bitplanes. All pixels in these bitplane images are normalized to the range of $[0, 255]$ for better visualization. The lowest bitplane is the least significant bitplane (LSB), and the highest bitplane is the most significant bitplane (MSB). It can be seen that, in general, lower bitplanes tend to have more “noise”, and higher bitplanes convey more salient features of the original image.

B. Reconstruction of JPEG Compressed Images

JPEG is a widely used lossy image compression method, where the image source is converted from the spatial domain into the frequency domain using discrete cosine transform (DCT). Then the DCT coefficients are quantized (a lossy step), followed by Huffman coding. In Matlab implementation, the tradeoffs between the compression ratio and distortion of the reconstructed images can be controlled by changing the value of the parameter “Quality”. Some example reconstructed images are shown in Fig. 3.

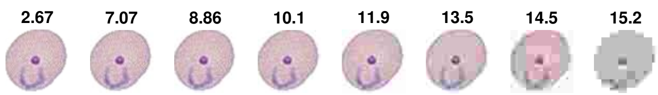

Fig. 3: Reconstructed JPEG images. The number above an image is the corresponding compression ratio. The higher the compression ratio, the lower the quality of the reconstructed image compared to the original image shown in Fig. 2. The ring form of the parasite is barely visible in the rightmost image, which has the largest compression ratio.

C. Reconstruction of JPEG 2000 Compressed Images

JPEG 2000 is a more advanced image compression standard with higher computational cost than JPEG. JPEG 2000 methods provide a wider range of compression ratios with acceptable reconstruction image quality than JPEG, owning mainly to its usage of discrete wavelet transforms and more sophisticated entropy coding schemes such as the Embedded Block Coding with Optimal Truncation [17]. While JPEG 2000 offers both lossy and lossless modes, we used only the lossy compression mode in this work. In Matlab implementation, the reconstruction image quality can be controlled by changing the value of the parameter “Compression Ratio”. Note that this parameter specifies only a target compression ratio. The actual compression ratio achieved may deviate from the target. Some example reconstructed images using JPEG 2000 with distinct compression ratios are shown in Fig. 4.

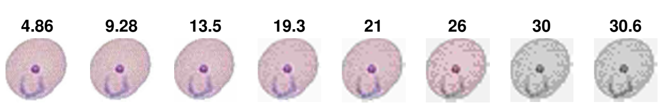

Fig. 4: Reconstructed images from JPEG 2000 compression of the original image shown in Fig. 2. The number above each image is the corresponding compression ratio. The rightmost image has a compression ratio that more than doubles that of co-located JPEG reconstructed image (in Fig. 3), however, the reconstructed image still retains the salient features of the original image such as the nucleus and ring form of the parasite.

D. Reconstruction of Autoencoder Compressed Images

Autoencoder is an artificial neural network that perform unsupervised learning on data. The simplest single-layer autoencoder is made of two components: one encoder and one decoder. The encoder obtains the lower dimension representation (called *codewords*) of the input data. The codewords are vectors of floating-point numbers. Since the encoder performs dimensionality reduction of the input, the autoencoder can be viewed as an input data compressor. Next, the decoder reconstructs the input data from the codewords. The training process seeks to minimize the difference between the input data and reconstructed data. As loss in the reconstructed data is typically inevitable, autoencoder can be used as a lossy compression method. Autoencoder can also be used for lossless compression, if the reconstruction error is maintained [16]. By changing the number of neurons in the single layer of the encoder, we can obtain different effective compression ratios, as calculated by using the following equation.

$$Compression\ Ratio = \frac{\#\ of\ Pixels \times 8}{\#\ of\ Neurons \times 10}, \tag{1}$$

where 8 in the numerator stands for the number bits per pixel. The codewords are real numbers ranging between 0 and 1. We round the codewords to decimal numbers ranging from 0.000 to 0.999. Each of these decimal numbers can be represented by a corresponding binary number with 10 bits (ranging from 0 to 999), hence 10 in the denominator.

For this work, we used Matlab’s built-in sparse autoencoder, which uses regularizers to learn a sparse representation of input. The influence of these regularizers can be controlled by adjusting the following parameters. *L2WeightRegularization* has control over the impact of L2 regularizer on weights of the network. *SparsityRegularization* and *SparsityProportion* control the sparsity of the output from the hidden layer. If the second parameter is set to 0.1, each neuron in hidden layer will have an average output of 0.1 over the training samples. The *UseGPU* option was also set to *true* in order to speed up the training process. Fig. 5 shows the diagram of a single layer autoencoder.

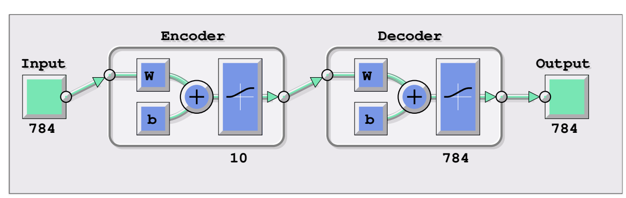

Fig. 5: Autoencoder as a lossy image compressor. Each input image (e.g, with a size of 784 pixels) is encoded into a codeword (a 10-point vector in this example), and then decoded back to a reconstructed image (a 28×28 image in this example). The effective compression ratio depends on the size of the codeword.

Single-layer autoencoders are simple and fast to train. We also found that for the MNIST dataset, the reconstructed images are of such good quality that they work well with classifiers based on deep convolution neural networks. However, for the Malaria datasets, the reconstructed images tend to have poor quality that prevents the classifier from achieving good classification accuracy. In principle, we can improve the reconstruction image quality by stacking several autoencoders together, albeit at the cost of much increased training time and computational complexity. In the following section, we provide detailed simulation results and discussions.

III. Simulation Results and Discussions

We used an NVIDIA DIGITS™ DevBox, which is an efficient platform of training and testing CNN's for classification. Several well known CNN structures have been integrated into and optimized for the DevBox, including LeNet-5, AlexNet and GoogLeNet. LeNet-5 is selected in this paper for its simple structure and relatively less computational time. The DIGITS software can utilize up to four Nvidia Titan X GPUs to accelerate neural network training.

A. Malaria Dataset

The reconstructed image datasets we obtained above were used the train LeNet-5. The datasets were split for training, validation and testing according to the proportion of roughly 5:1:1 [18]. The training epoch number was set to 30 for all datasets. In other words, all the training samples will run the forward propagation and backward propagation through the network for 30 times. The stochastic gradient descent algorithm [19] was used as the solver to optimize the mean square error cost function. After each training epoch, the CNN model will be evaluated on the validation set. If the accuracy of the training set keeps increasing, while validation loss also starts to increase after a certain number of epochs, this is a sign that the model will provide a good fit only for the training set. That is, the trained model cannot be generalized enough to handle unseen data. The loss of validation set will be used to control the potential overfitting issue. The final classification model is achieved after 30 epochs, and then tested on the predefined testing set to demonstrate its generalization capability.

The classification accuracies of 24 individual reconstruction bitplane image datasets, as well as the 8 reconstruction bitplane image datasets for combined color channels are shown in Fig. 6. We can see the general trend of increased classification accuracy as we use higher bitplanes for classification. This owns to the fact that lower bitplanes tend to have less impact on the pixel intensity of the reconstructed images. For example, a flipped bit in the LSB will only change the intensity of the reconstructed image by 1 (2^0), whereas a flipped bit on the MSB would lead to an intensity change by 128 (2^7). However, there are exceptions to this general rule, as suggested by the non-monotonically increasing nature of the curves in Fig. 6. In addition, we can see using bitplanes from combined color channels offers higher classification accuracy than using bitplanes from separate color channels, albeit at the cost of a lower (a $3:1$ reduction) compression ratio.

For both JPEG and JPEG 2000 compression, eight distinct compression ratios are selected, with the classification

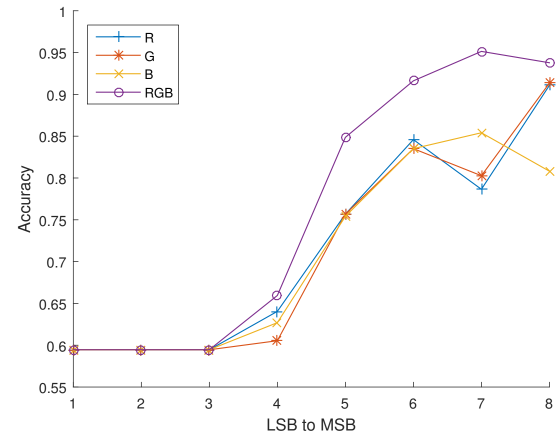

Fig. 6: Classification accuracies of reconstructed bitplane images. "R","G" and "B" stand for datasets for separate color channels, and "RGB" refers to the datasets for combined color channels.

accuracies corresponding to each compression ratio shown in Fig. 7. We can see the relationship between accuracy and compression ratio is as expected, i.e., in general, the higher the compression ratio, the lower the accuracy of classification. The classification results agree well with the visual display of the reconstructed images in Fig. 3 and Fig. 4. Furthermore, we can see that a very high accuracy (above 95%) can still be achieved after JPEG compression of the original images by 10 to 1. More impressively, about 95% accuracy can still be maintained even if the original images are compressed by JPEG 2000 with a large (30 to 1) reduction of the size. Large reduction of image size would be very beneficial to both efficient storage and transmission of medical images in telemedicine applications. These results show that lossy image compression would be a viable solution.

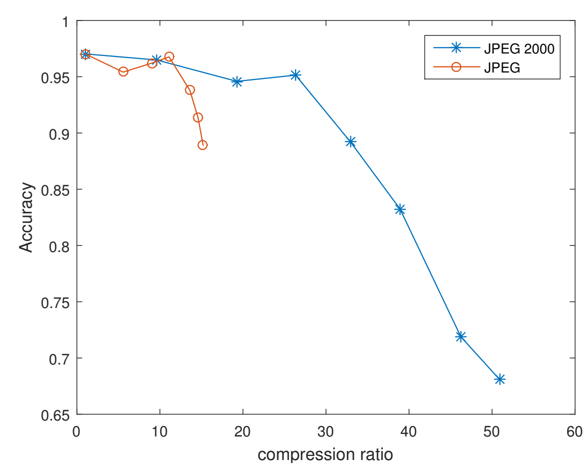

Fig. 7: The classification accuracies of the reconstructed images using the JPEG and JPEG 2000 methods.

One important distinction of JPEG and JPEG 2000 is that the compression are mainly accomplished in frequency domain, which allows the reconstruction quality to more "gracefully" degrade as we increase the compression ratios. In contrast, bitplane reduction is accomplished in the spatial domain. So dropping bitplanes, especially those higher bitplanes, will adversely impact the quality of the reconstructed images, thereby causing lower classification accuracy for similar compression ratios.

B. DIGITS Dataset

To diversify the datasets used, we applied four compression methods on the widely used MNIST dataset [20]. Since MNIST images are all grayscale images, bitplane coding was conducted on just one single channel. By combining the first N bitplanes (going down incrementally from the MSB), eight different reconstructed image datasets were created, with compression ratio ranging from $8:1$ (by keeping only the MSB) to $1:1$ (by keeping all the bitplanes, meaning no compression). Some example decompressed images are shown in Fig. 8.

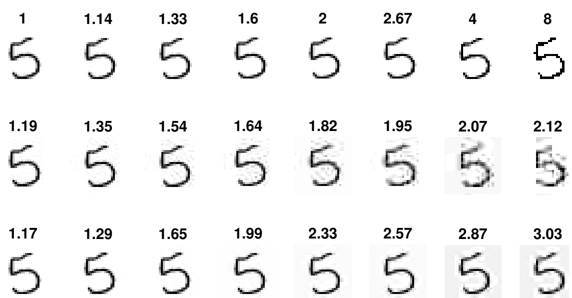

Fig. 8: Reconstructed images after compression with bitplane reduction (the first row), JPEG (the second row), and JPEG 2000 (the third row). The number above each of the images is the practically achievable compression ratio.

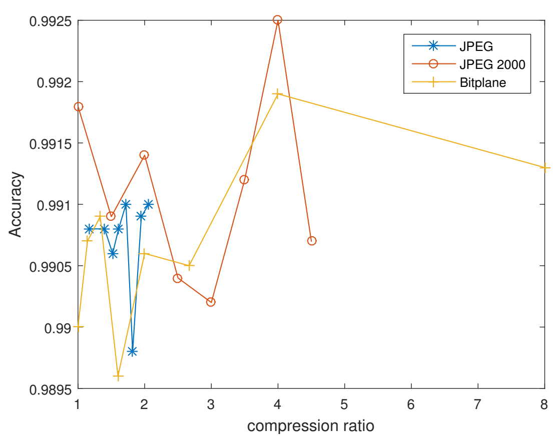

Fig. 9: Classification accuracies of the handwritten digits reconstructed from the compressed images in the MNIST datasets.

After training the LeNet-5 with all these datasets, the testing accuracies obtained are shown in Fig. 9. Unlike the malaria infection dataset, reconstructed images from the compressed MNIST dataset all led to classification accuracies over 90% for the given range of compression ratios, even for the bitplane method. The main reason is that the handwritten digits have high contrast against the background. This means almost all the bitplanes for the background will be "1", and most bitplanes of the foreground will be "0". Thus even if we drop all the higher seven bitplanes, the digit pattern in the reconstructed image is still recognizable. For the same reason, we can achieve very high accuracies by using reconstructed images after JPEG or JPEG 2000 compression. Interestingly, unlike the curves in Fig. 7, for reconstructed images of digits, there is not a clear trend of decreased classification accuracy as the compression ratio increases. Our hypothesis is that a certain degree of compression artifacts might indeed help the distinguishing features stand out better through deep learning. To validate this hypothesis requires more in-depth study.

Autoencoder was also tested as a lossy compression method. As shown in Fig. 5, each 28×28 DIGIT image is reshaped into a vector of 784, and then encoded into a 10-point compressed codeword. The effective compression ratios as calculated by using Eq. (1) are summarized in Table I.

# of Neurons	100	70	50	30	20	15	10
CR	6.27	8.96	12.54	20.90	31.36	41.80	62.70

TABLE I: Effective compression ratios (CR) corresponding to the number of neurons in the single layer of the autoencoder.

The classification results are shown in Fig. 10, where the classification accuracy decreases as compression ratio increases. This is expected, as a high compression ratio typically leads to lower image quality, which will translate into a lower classification accuracy. For example, a blurry hand-written digit "6" may be misclassified as a "0". However, even when compression ratio is higher than 60, the classifier can still achieve an accuracy above 85%. We can see that the autoencoder can offer a much wider range of compression ratios than the other three lossy compression methods, while maintaining a reasonably good classification accuracy.

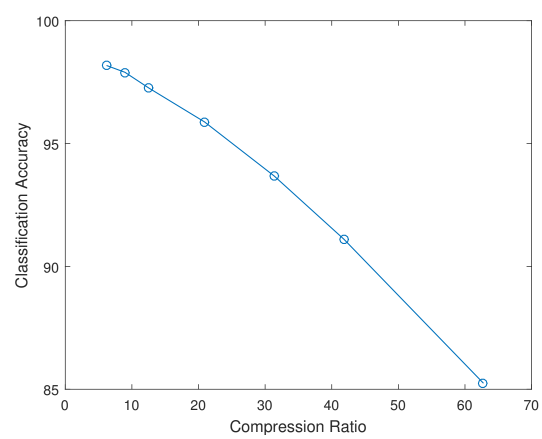

Fig. 10: Classification accuracies of the handwritten digit images in the MNIST datasets reconstructed from the autoencoder.

IV. Conclusion

Large reduction of medical image size would be very beneficial to many telemedicine applications. We evaluated the impact of lossy image compression methods on classification accuracies of convolutional neural networks, which has not been studied before in the literature. Simulation results showed that the bitplane reduction method had lower accuracy than JPEG and JPEG 2000 methods. We found that autoencoders were capable of providing a much more scalable compression ratios than the other three lossy compression methods, while maintaining a reasonably good classification accuracy for the handwritten digit images. We also found that for these type of images, a certain degree of compression artifacts might indeed be beneficial to improving the classification accuracy, an intriguing phenomenon prompting for more in-depth study. As a further work, we seek to improve the reconstruction image quality of more "natural" images such as the red blood cell images in the Malaria dataset by stacking several autoencoders together.

V. Acknowledgement

We acknowledge the support from the UAH College of Engineering Mid-Career Faculty Research Award.

References

[1] M. I. Razzak, "Malarial parasite classification using recurrent neural network," vol. 9, no. 2, pp. 69–79, 2015.

[2] World Health Organization, "Disease burden of malaria," http://www.who.int/mediacentre/factsheets/fs094/en/.

[3] "Whole slide image for malaria infected red blood cells," http://peir-vm.path.uab.edu/debug.php?slide=IPLab11Malaria.

[4] H. Shen, W. D. Pan, Y. Dong, and M. Alim, "Lossless compression of curated erythrocyte images using deep autoencoders for malaria infection diagnosis," in *Proc. of IEEE Picture Coding Symposium (PCS'16)*, Nuremberg, Germany, Dec. 2016.

[5] V. Muralidharan, Y. Dong, and W. D. Pan, "A comparison of feature selection methods for machine learning based automatic malarial cell recognition in wholeslide images," in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, Feb 2016, pp. 216–219.

[6] F. B. Tek, A. G. Dempster and I. Kale, "Parasite detection and identification for automated thin blood film malaria diagnosis," *Journal of Computer Vision and Image Understanding*, vol. 114, no. 1, pp. 21–32, January 2010.

[7] D. K. Das, M. Ghosha, M. Pal, A. K. Maiti, and C. Chakraborty, "Machine learning approach for automated screening of malaria parasite using light microscopic images," *Journal of Micron*, vol. 45, pp. 97–106, Feb 2013.

[8] F. B. Tek, "Computerised diagnosis of malaria," Ph.D thesis, University of Westminster, 2007.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[10] J. Kim, M. Sullivan, E. Choukse, and M. Erez, "Bit-plane compression: Transforming data for better compression in many-core architectures," *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, pp. 329–340, 2016.

[11] N. S. T. Sai and R. C. Patil, "Image retrieval using bit-plane pixel distribution." *International Journal of Computer Science and Information Technology (IJCSIT)*, pp. 159–174, 2011.

[12] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks." *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[13] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets." *Neural Computation*, vol. 18, no. 7, pp. 1527–1544, 2006.

[14] Y. Bengio and Y. LeCun, "Scaling learning algorithm towards ai." *Large-Scale Kernel Machines*, 2007.

[15] D. Erhan, Y. Bengio, A. Courville, P. A. Manzagol, and P. Vincent, "Why does unsupervised pre-training help deep learning?" *Journal of Machine learning reasearch*, vol. 11, pp. 625–660, 2010.

[16] H. Shen, W. D. Pan, Y. Dong, and M. Alim, "Lossless compression of curated erythrocyte images using deep autoencoders for malaria infection diagnosis," in *Picture Coding Symposium*, 2016.

[17] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on image processing*, vol. 9, no. 7, pp. 1158–1170, 2000.

[18] A. Graves, "Neural networks," in *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012, pp. 15–35.

[19] V. P. Plagianakos and G. D. Magoulas, "Stochastic gradient descent," *Advances in Convex Analysis and Global Optimization: Honoring the Memory of C. Caratheodory (1873–1950)*, vol. 54, p. 433, 2013.

[20] "Mnist, database of handwritten digits used for training image processing systems," http://yann.lecun.com/exdb/mnist/, accessed: 2016-05-30.