ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/371094417

A Power Efficient Solution to Determine Red Blood Cell Deformation Type Using Binarized DenseNet

Chapter · May 2023

DOI: 10.1007/978-3-031-33743-7_21

CITATIONS
0

READS
67

5 authors, including:

Md Tanzim Reza
BRAC University
46 PUBLICATIONS **171** CITATIONS
SEE PROFILE

Shakib Mahmud Dipto
University of Liberal Arts Bangladesh (ULAB)
12 PUBLICATIONS **9** CITATIONS
SEE PROFILE

Mohammad Zavid Parvez
64 PUBLICATIONS **1,007** CITATIONS
SEE PROFILE

Prabal Datta Barua
University of Southern Queensland
131 PUBLICATIONS **1,736** CITATIONS
SEE PROFILE

All content following this page was uploaded by Shakib Mahmud Dipto on 02 June 2023.

The user has requested enhancement of the downloaded file.

A Power Efficient Solution to Determine Red Blood Cell Deformation Type Using Binarized DenseNet

Md Tanzim Reza[1], Shakib Mahmud Dipto[1], Mohammad Zavid Parvez[2,3,4,5], Prabal Datta Barua[6,7,8], and Subrata Chakraborty*[7]

[1]Department of Computer Science and Engineering, BRAC University, Bangladesh
[2] Information Technology, APIC, Melbourne, Australia
[3] Information Technology, Torrens University, Melbourne, Australia
[4] Peter Faber Business School, Australian Catholic University, Melbourne, Australia
[5] School of Computing, Mathematics, and Engineering, Charles Sturt University, Bathurst, NSW, Australia
[6] School of Management Enterprise, University of Southern Queensland, Darling Heights, QLD 4350, Australia
[7] School of Science and Technology, Faculty of Science, Agriculture, Business and Law, University of New England, Armidale, NSW 2351, Australia
[8] Cogninet Australia Pty Ltd, Level 5, 29-35 Bellevue St, Surry Hills NSW 2010
{rezatanzim,diptomahmud2,zavidparvez}@gmail.com,prabal.barua@usq.edu.au,subrata.chakraborty@une.edu.au

Abstract. Red Blood Cells (RBCs) play an important role in the welfare of human being as it helps to transport oxygen throughout the body. Different RBC-related diseases, for example, variants of anemias, can disrupt regular functionality and become life-threatening. Classification systems leveraging CNNs can be useful for automated diagnosis of RBC deformation, but the system can be quite resource-intensive in case the CNN architecture is large. The proposed approach provides an empirical analysis of the application of 28 and 45-layer Binarized DenseNet for identifying RBC deformations. According to our investigation, the accuracy of the 45-layer binarized variant can reach 93–94%, which is on par with the results of the conventional variant, which also achieves 93–94% accuracy. The 23-layer binarized variant, while not on par with the regular variant, also gets very close in terms of accuracy. Meanwhile, the 45-layer and 28-layer binarized variant only requires 9% and 11% storage space respectively to that of regular DenseNet, with potentially faster inference time. This optimized model can be useful since it can be easily deployed in resource-constrained devices, such as mobile phones and cheap embedded systems.

Keywords: Binarized Neural Network, Larq, Red Blood Cell, Erythrocyte, Anemia, DenseNet, Convolutional Neural Network

I Introduction

The human body primarily contains three different types of blood cells: erythrocytes, monocytes, and platelets. Erythrocytes, often known as red blood cells (RBC), make up between 40 and 45 percent of the total amount of blood in the human body, making them one of the most significant components. [1] The primary function of RBCs is to transport oxygen to various parts of the body alongside eliminating carbon dioxide by carrying it to the lungs for exhalation. This particular functionality is powered by hemoglobin, a protein predominantly comprised of iron that gives blood its red color.

RBCs are generally round-shaped with a suppressed region at the center. However, a number of RBC disorders, commonly referred to as anemia, can cause RBCs to adopt a malformed shape. The anemia can generally be caused by iron deficiency or hereditary reasons. When RBCs get impacted by diseases, bone marrow cannot produce enough well-formed RBCs to maintain their natural functionalities. Consequently, the optimal flow of oxygen and carbon dioxide gets disrupted, leading to external symptoms. The symptoms include lack of energy, dizziness, fatigue, insomnia, [2] and, in severe cases, heart failure. [3] RBC shortage can occur in children as well, which can cause them to grow more slowly than usual. [4]

It is possible to detect abnormality in the RBC shape by analyzing the formation of the RBC and detecting if there is any deformation. A large presence of deformed cells usually indicates

some form of anemia. However, mass detection of shape-related abnormalities is a time-consuming process, which necessitates the automation of the procedure. CNN-based automation applications are quite popular due to their easy-to-approach training procedures on image data. State of The Art (SOTA) CNNs, however, can be quite heavy and power-hungry in terms of deployment. The proposed paper demonstrates the application of Binary DenseNet (BDN), which is substantially more power efficient than Regular DenseNet (RDN) due to weights and activations being restrained to binary digits only. Empirical analysis showcases that the BDNs can detect RBCs with little to almost no performance impact compared to the RDN variants while taking significantly fewer resources in terms of deployment.

The proposed research contributes by proving that BNNs, which may be used in severely resource-constrained environments, can identify RBC deformation types just as effectively as conventional CNNs. The paper is broken down into five main components. The background study, which includes a literature review and data information, is covered in Chapter two. The proposed model and result analysis are covered in chapters three and four, respectively. The last chapter serves as the paper's conclusion.

II Background Study

A Literature Review

Several machine learning-based studies for detecting RBC abnormalities have been conducted. Hany A. Elsalamony employed basic Neural Networks (NNs) for the classification of four RBC class types: sickle, elliptocytosis, microsite, and cells with undefined shapes. A total of 160 photos - 109 normal images and 51 aberrant images - were used to train and test multiple NNs. For the diseased cells, effective classification scores of 98-100% were achieved. [5] Mengjia et al. used an 11-layer deep Convolutional Neural Network (CNN) model to classify 8 different RBC types based on their shape. The CNN was trained and tested on a total of 7224 individual RBC images with an approximate 87.5% average accuracy on the validation set across five-fold cross-validation. [6] Using their 18-layer deep CNN, Laith et al. developed two different classifiers: one by coupling the CNN with a multilayer perceptron (MLP), and the other by coupling it with a multi-class Support Vector Machine (SVM). Overall accuracy was 88.11% for the variant with MLP attached on top and 92.06% for the variant with SVM. The experimentation was done on 15,000 images of three classes: Normal, Anemia, and miscellaneous. [7] Weiqing and his co-researchers classified 14 different types of red blood cells using the Attention Residual Feature Pyramid Network, and they did it with an average precision score of 86.9 percent. The overall analysis was done on 1,17,839 images, one of the largest amounts we have found. [8]

There are also applications of traditional machine learning approaches for RBC classification. In the work by Shirazi et al., they segmented and detected RBC cells from blood using Fuzzy C-Means and Hough Transform respectively. Afterward, they extracted geometrical and texture features from the segmented RBCs and trained a Support Vector Machine (SVM) alongside Extreme Learning Machine (ELM) classifier with it. The models were tested on 10,000 RBC images, 8000 normal alongside 2000 abnormal. In this binary classification task, ELM achieved better performance with 96% sensitivity and 95% specificity, compared to 88% sensitivity and 85% specificity achieved by SVM. [9] Vadivel et al. classified Sickle Cell Anemia from RBCs using K-Nearest Neighbor (KNN). KNN obtained 93% accuracy in this binary classification challenge of normal and abnormal RBC images. [10]

Although the usage of CNNs for RBC classification is quite common, most of the time the utilized CNNs are small in scale because using pre-trained large CNNs might be overkill for RBC detection. Large-scale SOTA CNNs are power-hungry and resource-intensive for deployment. Therefore, if these CNNs are to be deployed in a resource-constrained environment, some sort of optimization is necessary in order to conserve resources. This criterion gives rise to the concept of RBC-type classification utilizing BDNs since the binarized variant requires less memory.

B Dataset Description

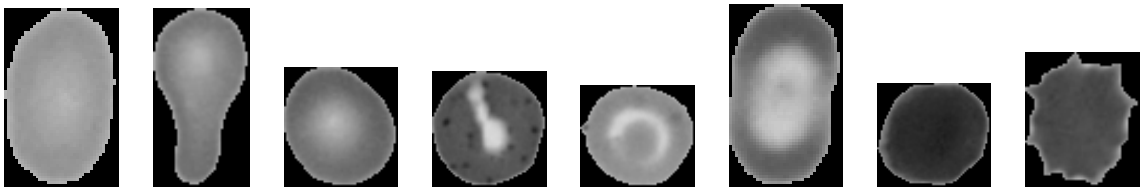
Our dataset was collected from [11], which has 7108 images compiled from nine different classes. Since the sample size of a particular class is extremely small, we omitted that class and utilized images from the rest of the eight classes to train and test the models. The classes include:

Elliptocytes: RBCs with a cigar- or pencil-shaped morphology are known as elliptocytes. Elliptocytes are only produced once an RBC has reached its mature shape, and they are brought on by the stress that RBCs experience during circulation. [12,13] Normal RBCs typically contain very few elliptocytes, although their numbers rise in the presence of certain anemia-related disorders.

Dacrocytes (TearDrop shaped): When infiltrative illnesses like metastatic cancer impact the bone marrow, RBCs are distorted into a teardrop form. Additionally, patients with various anemias also exhibit them. [14]

Normal RBCs: RBCs that are unaffected by abnormalities are classified as normal. They have a typical round-shaped structure.

Stomatocyte: When an RBC's central zone is replaced with a slit-like pattern as a result of severe swelling, stomatocytes are produced. Stomatocytes are created due to genetically flawed RBC transport protein and can result in mild to severe hemolysis and anemia. [15]



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 1: RBC Types: (a) Elliptocyte, (b) Dacrocytes, (c) Normal, (d) Stomatocytes, (e) Codocytes, (f) Hypochromic, (g) Spherocytes, (h) Acanthocytes

Codocytes (Target Cells): Codocytes, also known as target cells, resemble the appearance of a shooting target with a bulls-eye. Codocytes generate in large volumes when there is the presence of liver diseases. [16]

Hypochromic cells: RBCs that are low on hemoglobin are referred to as Hypochromic cells. Usually, a high percentage of Hypochromic cells is an indicator of iron deficiency. [17]

Spherocytes: RBCs lose their typical round, biconcave form and take on a more sphere-like appearance as a result of hereditary spherocytosis and anemia. [16] These sphere-like shaped RBCs are known as spherocytes.

Acanthocytes: Another sign of iron deficiency anemia is a condition known as acanthocytes, in which the RBCs' outer regions take on the appearance of thorny spikes. [18]

Table 1: Dataset Class Distribution

RBC Type	**Distribution** (No. of Images)	**RBC Type**	**Distribution** (No. of Images)
Elliptocyte	1211	Codocytes	851
Dacrocytes	2076	Hypochromic	222
Normal	1426	Spherocytes	563
Stomatocyte	382	Acanthocytes	354

Table 1 refers to the class distribution of the dataset. There is another class in the original dataset that corresponds to RBCs that have a pencil-like form. [11] However, the number of image samples for that class is extremely low, with only 24 images for the particular class. Additionally, empirical findings based on that dataset demonstrate that the lack of samples makes it impossible to detect pencil-shaped RBCs properly, resulting in a zero accuracy score for the class. [19] Hence, we omitted the class from the distribution.

C Binarized Neural Network

Binarized neural networks (BNN) have the same components as the regular NN counterparts, with the only exception of weights and activations being constrained to +1 and -1. As opposed to the standard 32bit/16bit float weights of regular NNs, the weights and activations of these models can be represented by only a single bit because they are limited to two distinct values. As a result, BNNs use a lot less memory than conventional NNs. Typically, a straightforward threshold function is used to binarize the weights and activations:

$$\chi_{\mathbb{B}}(x) = \begin{cases} +1 & \text{if } x > 0 \\ -1 & Otherwise \end{cases} \tag{1}$$

Here, $\chi_{\mathbb{B}}(x)$ is the binarized values after the conversion and x is the original value. Normally, it is not possible to calculate gradients utilizing binary weights since the derivative of the sign function is usually zero. Therefore, original weights are preserved in order to calculate gradients during backpropagation. [20] For our work, we have utilized the larq-zoo [21] library to deploy the binarized and normal variants of DenseNet.

III Proposed Model

We first gathered RBC pictures from the source, resized them into 224x224 resolution, and separated them into train and test sets. 80% of the total photographs were randomly chosen for the models' training, while the remaining images were used for testing. Since we mostly used the default hyperparameters set from the larq library, we did not keep any validation set since no hyperparameter optimization was involved.

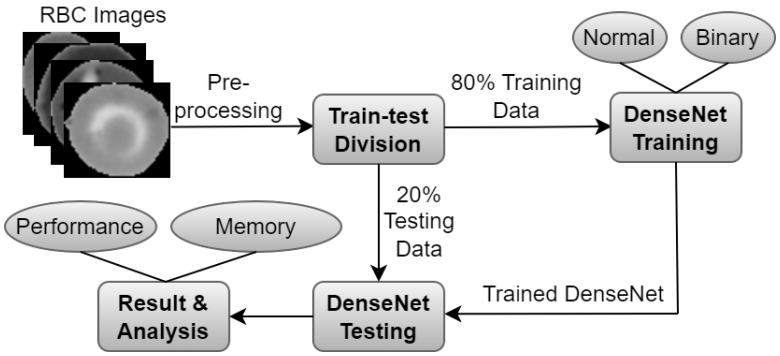


Fig. 2: Proposed Model

For NN selection, we picked the 28 and 45-layer DenseNet variants from the larq zoo library. The 28-layer DenseNet model has 4 densely connected blocks, with a sequence of (6, 6, 6, 5) sub-layers per block, each sub-layer having a series of convolution, concatenation, and batch-normalization layers. There's also the input layer at the top and the average pooling layer at the end. Finally, the feature set is flattened and passed through a densely connected layer of eight output neurons with softmax activation, each output neuron corresponds to a single class. The 45-layer variant has a similar structure, albeit with more sub-layers per block. In this case, the sub-layer sequence for four blocks is (6, 12, 14, 8). The default variants have binarization applied to the forward-propagated weights and activation. We did the initial training with the binarization activated, then modified the source code to remove the binarization and experimented on the float-32 variant for DenseNet28 and DenseNet45 each.

These models are also pre-trained based on imagenet dataset. Instead of initializing the weights randomly, we applied transfer learning and initiated the weights with the pre-trained ones from the library. Afterward, the model was further fine-tuned on the RBC dataset.

The models were trained for 100 epochs with a batch size of 32. Adam optimizer with a learning rate of 0.00001 was used alongside sparse categorical crossentropy loss metrics. After the training phase, the testing accuracy of the model was observed and for each class, the precision, recall, and f1-score were calculated. Finally, an empirical analysis of the performance was done. For analysis, a comparison of the RDNs, BDNs, and other experimental analyses was done.

IV Result and Analysis

The NNs were trained for 100 epochs since each NN began to exhibit overfitting after that point. In general, we retained the model checkpoint recorded at every 10 epochs so that we could load

the prior valid checkpoint if overfitting state appeared. The binarized veriants of the DenseNets hit overfitted state after 100 epochs, where no improvement of validation accuracy was observed. Therefore, after 110 epochs, we loaded the checkpointed version saved at 100 epochs and presented our empirical analysis based on that. Meanwhile, the normal variants hit overfitted state much quickly, between 20-40 epochs. However, since the overfitting did not cause significant decline to validation accuraracy, we kept the 100 epoch checkpointed version for all the models to perform fair comparison.

After training, the training accuracy, testing accuracy, training loss, and testing loss were noted for each of the NNs.

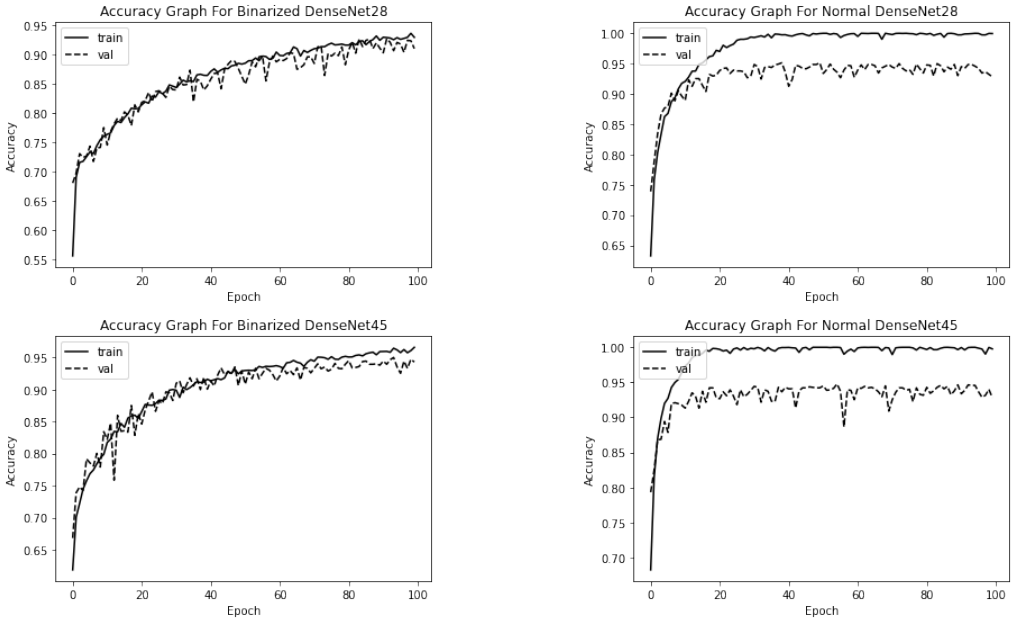


Fig. 3: Accuracy graphs for the BDNs and NDNs

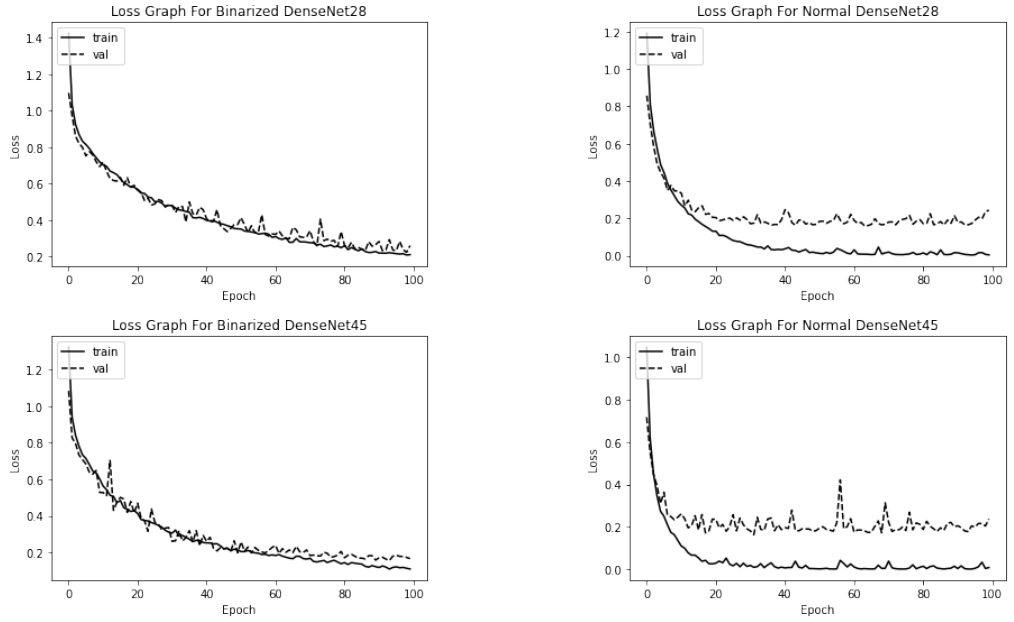


Fig. 4: Loss graphs for the BDNs and NDNs

Generally, we found that the RDNs overfit significantly faster than the BDNs. The training accuracy plateau of 99.5% for the 28-layer RDN is reached after 30 epochs, whereas it is reached even more quickly for the 45-layer variant after just 20 epochs. Due to the overfitting problem, the 45-layer RDN variant also exhibits considerably higher spikes on the validation accuracy graph.

In the next phase, we observed the correct/false prediction for each class through confusion matrices.

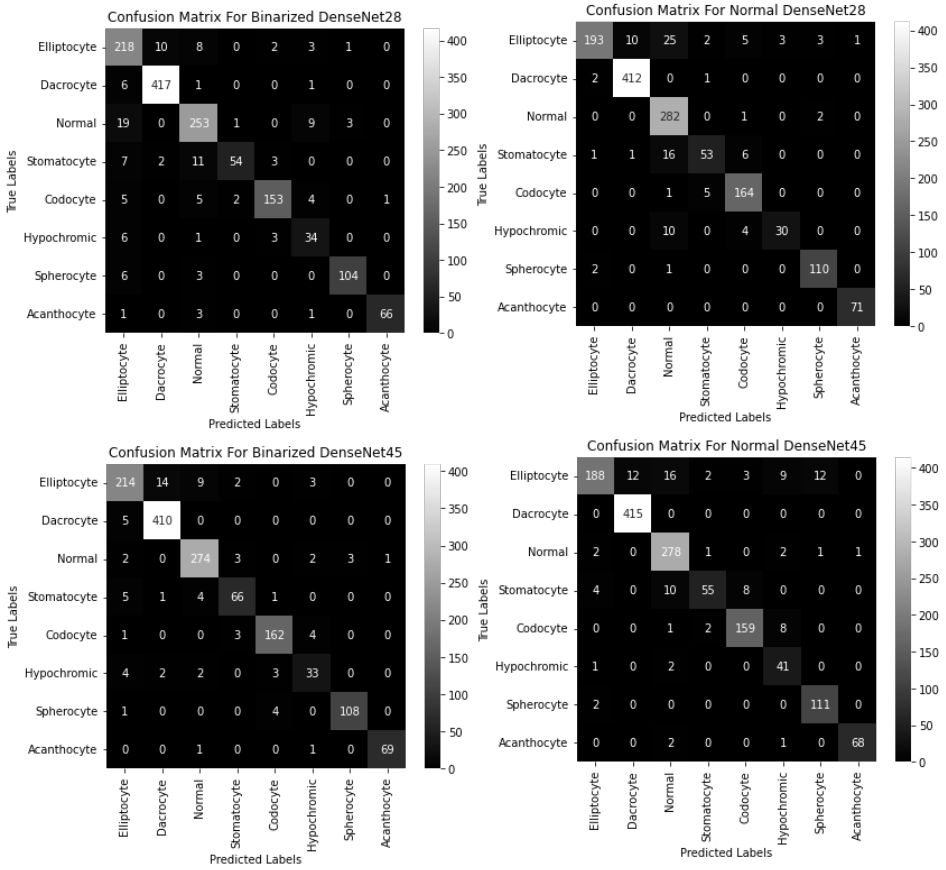


Fig. 5: Confusion matrices for the BDNs and NDNs

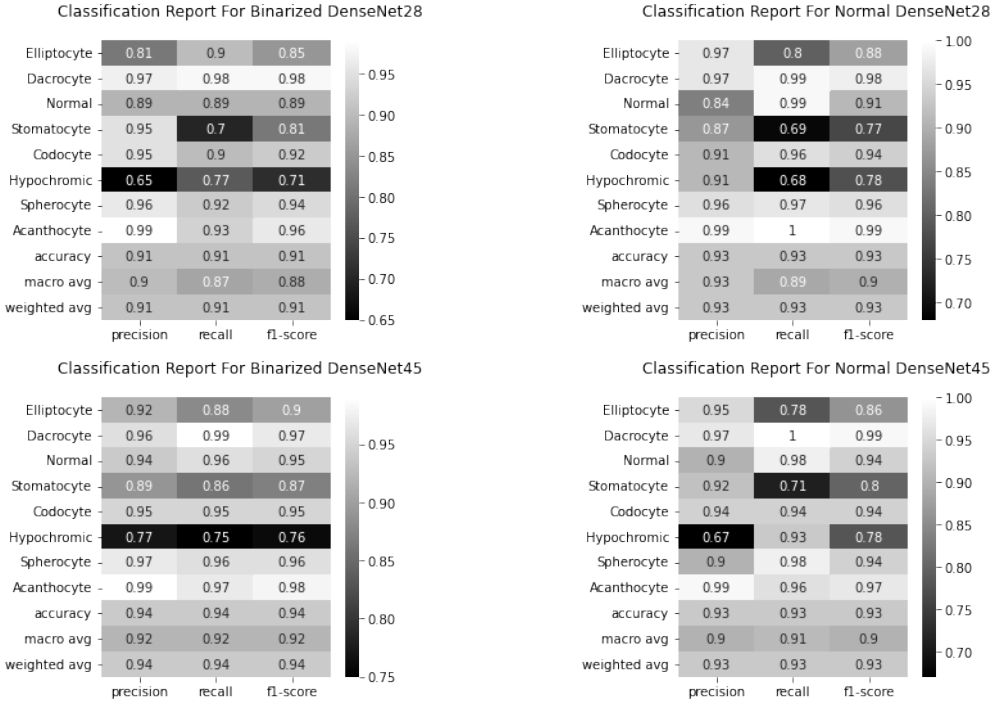


Fig. 6: Classification reports for the BDNs and NDNs

Generally, the NNs perform better for the classes that have a higher number of samples. The precision, recall, and f1 scores for each class alongside the accuracy of each model are given in 6

As visible accross figure 3,4, 5, 6, the BDNs perform almost on par with the RDN variants. Surprisingly, the performance of the 45-layer BDN sometimes exceeds the performance of the 28-layer RDN slightly. This is likely due to the BDN being able to avoid overfitting of RDN by the natural regularization that occurs due to the binarization process.

Table 2: Performance Comparison

Model	**Accuracy Score**
BDN28 (After 100 epochs)	91.2%
RDN28 (After 100 epochs)	92.8%
BDN45 (After 100 epochs)	94.3%
RDN45 (After 100 epochs)	92.8%
Literature [19] (Without Teardrop Class)	94.3%

A comparison between all the models is provided in 2. In literature, [19] an accuracy of 93.8% was attained for the classification of 9 classes. However, since we dropped the teardrop cell class due to the sample distribution being extremely small, we recalculated their scores without the corresponding class and got a score of 94.3%. From table 2, BDNs can evidently perform competitively against RDNs and literature. Meanwhile, BDNs can significantly reduce the memory consumption of RDNs. The details on resource savings are given in table 3.

Table 3: Model Details

Model	**Parameters** (million)	**Binarized Parameters**	**Memory** (Regular)	**Memory** (Binarized)	**Compression Ratio**
DenseNet45	13.1	12.34	49.95 MB	4.32 MB	0.09
DenseNet28	4.56	4.2	17.39 MB	1.86 MB	0.11

As a result, we may draw the conclusion that using BDN to classify RBCs is more beneficial because it uses less memory. Choosing DenseNet for disease classification was a key choice here since the binarization of the weights may cause the neural network to have a higher number of parameters with a deeper architecture to work properly. However, the gradient vanishing issue arises when the number of hidden layers is increased, which disrupts the learning process. DenseNet can tackle this issue by introducing skip connections and this particular property prompted us to use the model.

Finally, this energy-efficient variant of the neural network can be especially useful since it can be deployed on a wide variety of hardware devices. If the models are to be developed for mobile applications, then the low memory consumption can be of huge help. Automated medical image analysis can also get great benefits from distributed learning environments such as a federated setting. In such a federated setting, where models are regularly transmitted over a network, such memory reduction can play a significant role to save bandwidth.

V Conclusion

It is essential to conserve as many resources as possible while implementing DL-based models because DL is becoming more accessible through devices with limited capabilities. BNNs can therefore be quite helpful in making DL more approachable. The primary drawback of BNN is that normal CNN models frequently outperform them. BNNs can, nevertheless, perform on par with regular CNNs when the sample images are simple, such as the RBC images. Obviously, in order to make BNNs more usable, this level of accuracy needs to be attained on more difficult datasets. Future research should be done on this.

References

1. T. W. Secomb, "Blood flow in the microcirculation," Annual Review of Fluid Mechanics, vol. 49, pp. 443–461, 2017.
2. K. F. Lasch, C. J. Evans, and D. Schatell, "A qualitative analysis of patient-reported symptoms of anemia," Nephrology Nursing Journal, vol. 36, no. 6, p. 621, 2009.

3. T. B. Horwich, G. C. Fonarow, M. A. Hamilton, W. R. MacLellan, and J. Borenstein, "Anemia is associated with worse symptoms, greater impairment in functional capacity and a significant increase in mortality in patients with advanced heart failure," Journal of the American College of Cardiology, vol. 39, no. 11, pp. 1780–1786, 2002.
4. A. T. Soliman, M. M. Al Dabbagh, A. H. Habboub, A. Adel, N. A. Humaidy, and A. Abushahin, "Linear growth in children with iron deficiency anemia before and after treatment," Journal of tropical pediatrics, vol. 55, no. 5, pp. 324–327, 2009.
5. H. A. Elsalamony, "Healthy and unhealthy red blood cell detection in human blood smears using neural networks," Micron, vol. 83, pp. 32–41, 2016.
6. M. Xu, D. P. Papageorgiou, S. Z. Abidi, M. Dao, H. Zhao, and G. E. Karniadakis, "A deep convolutional neural network for classification of red blood cells in sickle cell anemia," PLoS computational biology, vol. 13, no. 10, p. e1005746, 2017.
7. L. Alzubaidi, O. Al-Shamma, M. A. Fadhel, L. Farhan, and J. Zhang, "Classification of red blood cells in sickle cell anemia using deep convolutional neural network," in Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018) held in Vellore, India, December 6-8, 2018, Volume 1, 2020, pp. 550–559.
8. W. Song et al., "Red Blood Cell Classification Based on Attention Residual Feature Pyramid Network," Frontiers in Medicine, p. 2638, 2021.
9. S. H. Shirazi, A. I. Umar, N. Haq, S. Naz, M. I. Razzak, and A. Zaib, "Extreme learning machine based microscopic red blood cells classification," Cluster Computing, vol. 21, pp. 691–701, 2018.
10. M. Vadivel, V. Vijaya Baskar, V. Sivakumar, and S. Vimal, "Classification of Sickle Cell Anemia Using Energy-Based KNN Classifier," in Next Generation of Internet of Things: Proceedings of ICNGIoT 2021, 2021, pp. 289–293.
11. D. A. Tyas, T. Ratnaningsih, A. Harjoko, and S. Hartati, "Erythrocyte (red blood cell) dataset in thalassemia case," Data in Brief, vol. 41, p. 107886, 2022.
12. K. Landis-Piwowar, J. Landis, and P. Keila, "The complete blood count and peripheral blood smear evaluation," Clinical laboratory hematology. 3rd ed. New Jersey: Pearson, pp. 154–77, 2015.
13. N. Manchanda, "19 Anemias: Red Blood Cell Morphology and Approach to Diagnosis," vip. persianss. ir, p. 284, 2016.
14. B. Constantino, "Reporting and grading of abnormal red blood cell morphology," International journal of laboratory hematology, vol. 37, no. 1, pp. 1–7, 2015.
15. I. Andolfo, R. Russo, A. Gambale, and A. Iolascon, "Hereditary stomatocytosis: an underdiagnosed condition," American journal of hematology, vol. 93, no. 1, pp. 107–121, 2018.
16. M. A. Parab and N. D. Mehendale, "Red blood cell classification using image processing and CNN," SN Computer Science, vol. 2, no. 2, p. 70, 2021.
17. N. H. Dinh, S. M. Cheanh Beaupha, and L. T. A. Tran, "The validity of reticulocyte hemoglobin content and percentage of hypochromic red blood cells for screening iron-deficiency anemia among patients with end-stage renal disease: a retrospective analysis," BMC nephrology, vol. 21, pp. 1–7, 2020.
18. G. J. Bosman, "Disturbed red blood cell structure and function: an exploration of the role of red blood cells in neurodegeneration," Frontiers in Medicine, vol. 5, p. 198, 2018.
19. D. A. Tyas, S. Hartati, A. Harjoko, and T. Ratnaningsih, "Morphological, texture, and color feature analysis for erythrocyte classification in thalassemia cases," IEEE Access, vol. 8, pp. 69849–69860, 2020.
20. M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," arXiv preprint arXiv:1602.02830, 2016.
21. Larq Zoo, Larq Zoo Pretrained Models, https://docs.larq.dev/zoo/, [Retrieved: 11 December 2022]

View publication stats