# BENCHMARKING WHITE BLOOD CELL CLASSIFICATION UNDER DOMAIN SHIFT

*Satoshi Tsutsui, Zhengyang Su, Bihan Wen*

Nanyang Technological University, Singapore

## ABSTRACT

Recognizing the types of white blood cells (WBCs) in microscopic images of human blood smears is a fundamental task in the fields of pathology and hematology. Although previous studies have made significant contributions to the development of methods and datasets, few papers have investigated benchmarks or baselines that others can easily refer to. For instance, we observed notable variations in the reported accuracies of the same Convolutional Neural Network (CNN) model across different studies, yet no public implementation exists to reproduce these results. In this paper, we establish a benchmark for WBC recognition. Our results indicate that CNN-based models achieve high accuracy when trained and tested under similar imaging conditions. However, their performance drops significantly when tested under different conditions. Moreover, the ResNet classifier, which has been widely employed in previous work, exhibits an unreasonably poor generalization ability under domain shifts due to batch normalization. We investigate this issue and suggest some alternative normalization techniques that can mitigate it. We make fully-reproducible code publicly available[1].

***Index Terms***— White Blood Cells, Leukocytes

## 1. INTRODUCTION

The microscopic examination of white blood cells (WBCs), also known as leukocytes, in human blood smears is an essential task in the fields of pathology and hematology. This task is particularly important in the diagnosis of blood disorders, such as leukemia, anemia, polycythemia, and immune-related diseases like autoimmune anemia, allergies, and others [1, 2]. Typically, the diagnosis process involves a differential count [3], which analyzes the distribution of the five types of WBCs, namely neutrophils, eosinophils, basophils, monocytes, and lymphocytes (see Fig. 1). Precise and automated classification of WBCs improves the efficacy of diagnosis.

The recognition of WBCs entails both signal processing [4, 5, 6, 7, 8] and biomedical expertise [9, 10, 11, 12, 2, 13]. A comprehensive literature review on this topic is provided in the survey by Zolfaghari et al. [14]. Previous studies have made noteworthy contributions to the development of automatic WBC classifiers[15, 16, 17, 18] and publicly available datasets [7, 9, 10, 2]. However, until recently, the size of public WBC datasets was limited to only hundreds of images [7, 9, 10], which was insufficient to apply state-of-the-art image classification models, such as Convolutional Neural Networks (CNNs). Recently, the RaabinWBC dataset [2] was released, comprising a relatively large number of WBC images (16k). Subsequent studies [15, 16, 17, 18] have proposed new approaches to improve the classification performance on RaabinWBC.

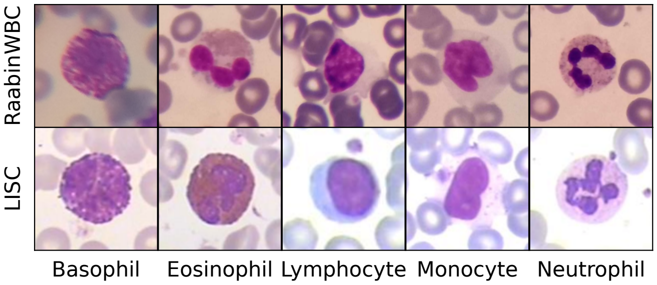


**Fig. 1**: Five types of white blood cells sampled from RaabinWBC [2], which we use for training, and LISC [9], which we use for testing domain generalization. See Table 1 for the dataset statistics.

| | Lymph. | Mono. | Neut. | Eos. | Bas. | Total |
|---|---|---|---|---|---|---|
| RaabinWBC Train | 2,427 | 561 | 6,231 | 744 | 212 | 10,175 |
| RaabinWBC Test-A (Same Microscope) | 1,034 | 234 | 2,660 | 322 | 89 | 4,339 |
| RaabinWBC Test-B (Different Microscope) | 148 | 0 | 1971 | 0 | 0 | 2,119 |
| LISC | 59 | 48 | 56 | 39 | 55 | 257 |

**Table 1**: Summary of datasets used in our Work. While RaabinWBC [2] has a large quantity that allows training modern CNNs, we notice that its test set under domain shifts (Test-B) is not suitable for benchmarking domain generalization due to missing WBC types and the heavy imbalance. Therefore, we suggest using another dataset of LISC [9] for testing domain generalization.

Meanwhile, the community has paid little attention to benchmarking WBC classification. For instance, the highest accuracy we discovered is 99.91% [11] on RaabinWBC Test-A set using VGG CNN [19] , while another study [2] reported 98.09% also using VGG. This difference in performance could be attributed to variations in image preprocessing, data augmentation, or the inherent randomness of stochastic gradient descent. In fact, we found that the range of accuracies can vary by as much as 1% by simply altering the random seeds in our code (see Fig.2), indicating that error bars should be reported. However, we cannot easily compute them for prior work since no paper publishes the implementation for training CNNs on RaabinWBC. Regarding code, some authors have released code for WBC feature extraction [12] or model weights trained on private datasets [13], but we are not aware of any work that publishes reproducible code to train baseline CNNs on publicly available WBC classification datasets and report accuracies with error bars.

**Present Work.** This paper aims to establish solid baselines that can be cited and reproduced by other researchers, which we believe is equally important to developing new methods. Specifically, we demonstrate that standard CNNs can achieve high accuracy (over 98.5%) without the use of more advanced models (see Sec.2.1). However, we have discovered that these high accuracy is only maintained when the model is trained and tested under similar imaging condi-

[1] `https://github.com/apple2373/wbc-benchmark`

tions, such as using the same measuring devices or blood processing methods. When the model is tested under domain shifts, which are more realistic scenarios in the real world, the accuracy of ResNet dramatically drops to around 23% (see Sec.2.2). This finding suggests that the classifier is almost broken, which is surprising given that WBCs only have five categories. Strangely, we find that VGG, which is one generation older than ResNet, can still achieve around 74% accuracy under domain shifts. This is consistent with previous work that reports similar results [11, 12]. Through empirical investigation, we have discovered that batch normalization is the cause of poor generalization and demonstrated that group normalization or pre-trained normalization can mitigate this issue (see Sec.3). Our repaired ResNets perform just as well as VGG and achieve around 74% accuracy under domain shifts, which to our knowledge, is the highest reported accuracy under the same scenario[11, 12]. Although 74% still has room for improvement (see Sec. 4), we believe that our rigorous benchmark on WBC classification will greatly benefit the community. To facilitate progress, we have open-sourced the implementations[1] to reproduce our reported results.

**Contributions.** 1) We establish solid baselines for benchmarking the WBC classification under domain shifts. 2) We empirically demonstrate that group normalization or pre-trained normalization improves the cross-dataset generalization of ResNet, a CNN widely used in previous studies. 3) We open-source the implementation[1], which other researchers can fully-reproduce our results on publicly available datasets.

## 2. BENCHMARKING WBC CLASSIFICATION

### 2.1. Baselines under Similar Imaging Conditions

We aim to establish reproducible WBC classification baselines by training standard image classifiers on publicly available datasets. To accomplish this, we utilize RaabinWBC [2], the largest WBC classification dataset, and train well-known CNN models. Our default choice of CNN is ResNet50 [20], which is widely regarded as the most frequently used image classifier and has reportedly received the most citations in the field of artificial intelligence [21]. We compute accuracy using the Test-A set, which was obtained from the same microscope and blood processing methods as the Train set. ecause we find that the reported accuracies in the previous literature vary more than 1% even with the same CNN, we run our implementation 10 times and report 95% confidence intervals.

*Implementation Details.* We initialize our CNNs with pretrained weights from ImageNet and optimize them using AdamW [22] with a weight decay of 0.005, an initial learning rate of 0.0001, and a cosine learning rate decay [23] for 10 (warm-up) + 90 (decay) epochs, totaling 100 epochs. Images are resized to 224 × 224 with random horizontal and vertical flips. We intentionally avoid using heavy data augmentation, including color changes, as our goal is to establish baselines rather than maximize performance. Future work is encouraged to explore additional data augmentation on top of our approach. For more details, please refer to our implementation[1].

*Results.* We obtained $98.53 \pm 0.18\%$ accuracy on RaabinWBC Test-A (same microscope) and show the corresponding box plot in Fig.2. The highest accuracy is nearly 99% while the lowest is just under 98%, spanning intervals of almost 1%. Given that there are no differences in implementation, we anticipate that the variance of reported results among different papers is likely to be even greater. Since training CNNs inherently involves randomness (due to random weight initialization, order of randomized training samples, etc.), even minor variations in implementation (such as learning rate or preprocessing) can sometimes have a significant impact on relatively small datasets. To address this, we recommend reporting error bars in addition to accuracy. We also experimented with VGG16[19], as it has been previously reported to have the highest accuracy of 99.91% [11], and obtained an accuracy of $98.75 \pm 0.06\%$. While this is slightly higher than ResNet's accuracy of $98.53 \pm 0.18\%$, the confidence intervals overlap.

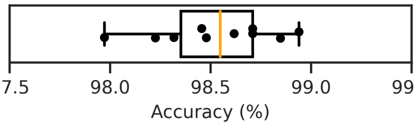


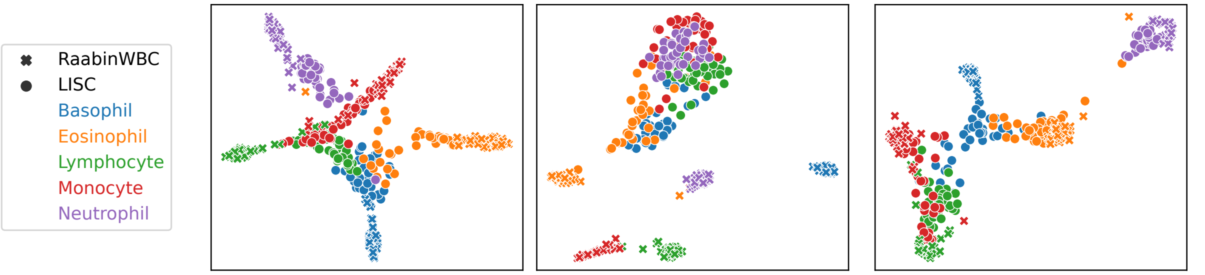
**Fig. 2**: Box plot for ResNet50 classification accuracies (%) on RaabinWBC Test-A set. To ensure a thorough evaluation, we propose reporting error bars. Although no one has published the code for training CNNs on this dataset, we observed that the reported accuracies in the literature vary by more than 1% even for the same CNN model. To investigate the source of this variation, we ran our implementation 10 times with different random seeds, where the black dots represent the accuracies of the 10 runs. The mean and 95% confidence interval is $98.53 \pm 0.18$. We expect even greater variation among different implementations in previous studies.

### 2.2. Evaluate under Domain Shift

Since we achieved very high accuracy (>98.5%) when training and evaluating under similar imaging conditions (Test-A), we believe it is appropriate to move on to a more practical scenario where models are trained and evaluated under different imaging conditions. To do so, we examined the RaabinWBC Test-B set, which was collected using a different microscope. However, we realized that it may not be the best dataset for evaluating classifiers under domain shifts due to its composition: it only includes neutrophils and lymphocytes, with neutrophils comprising 93% (See Table 1). As a result, we believe that the LISC dataset [9] may be more suitable than the Test-B set, as it is more class-balanced, enabling us to use plain accuracy to compare different methods. One drawback of using the LISC dataset is that it contains much fewer images, but reporting error bars with multiple runs can help mitigate this issue.

We propose training models on RaabinWBC and evaluating them with LISC as a practical benchmark for WBC classification. The statistics for the datasets are presented in Table 1, and sample images from LISC are displayed in the second row of Fig.1. In comparison to RaabinWBC images (the first row of Fig.1), LISC images have different coloring due to differences in blood processing conditions, such as staining, in addition to different imaging conditions, such as microscopes or cameras. To confirm the existence of domain differences, we use t-SNE [24] to visualize images from RaabinWBC and LISC using ResNet and VGG, both trained in Sec.2.1. The resulting 2D plots in Fig.3-ab show that the data points for RaabinWBC (✖) and LISC (●) tend to form different clusters, regardless of WBC types (color), indicating a domain discrepancy between the two datasets.

*Results.* We evaluated the two baseline models trained on RaabinWBC with LISC and report their mean accuracies and confidence intervals in Fig.4-a and -b. The accuracies (%) for ResNet50 and VGG16 are $23.11 \pm 3.04$ and $74.44 \pm 2.72$, respectively. As this task only has five classes, where random guessing can achieve 20%, ResNet's performance indicates that the classifier is essentially broken under domain shifts. Interestingly, VGG still has a relatively high accuracy, although it still has significant room for improvement



(a) VGG16 (b) ResNet50 (c) ResNet50-Group-Normalization (GN)

**Fig. 3**: t-SNE [24] plots show image representations from three CNNs. The markers (✖ and ●) indicate the datasets while the colors represent the WBC types. We sampled 39 images (the smallest category of eosinophils in LISC, as described in Table 1) per class and per dataset, and plotted 390 (= 39 × 5 × 2) data points in a 2D space using CNN features trained from RaabinWBC. *(b)*: ResNet50 separates images from RaabinWBC (✖) and LISC (●) more clearly and places them farther apart, even if they belong to the same WBC type (the same color), indicating poor domain generalization ability. *(a, c)*: VGG16 and ResNet50-GN still keep images of RaabinWBC (✖) and LISC (●) relatively close if they belong to the same WBC type (the same color), indicating greater domain generalization ability than ResNet50.

| | | Layer | | Test Accuracy (%) | |
|---|---|---|---|---|---|
| | Model | FC | BN | RaabinWBC [2]-A | LISC [9] |
| (a) | ResNet | - | ✓ | 98.53 ± 0.18 | 23.11 ± 3.04 |
| (a’) | | ✓ | ✓ | 98.45 ± 0.19 | 27.74 ± 3.44 |
| (b) | VGG | ✓ | - | 98.75 ± 0.06 | 74.44 ± 2.72 |
| (b’) | | ✓ | ✓ | 98.68 ± 0.23 | 33.74 ± 8.04 |

**Table 2**: Ablation studies to identify the cause of ResNet’s dramatic performance drop on LISC. The results suggest that Batch Normalization (BN) layers adversely affect the domain generalization ability, while Fully-Connected (FC) layers do not.

compared to the accuracy evaluated under a similar domain. This interesting phenomenon is also supported by the t-SNE plots in Fig.3-a and -b. In Fig.3-a, VGG keeps ✖ and ● (i.e., images from different datasets) of the same color (i.e., the same WBC type) relatively close. However, in Fig.3-b, ResNet separates ✖ and ● more clearly, placing them far apart even if they have the same color, which indicates that the image representations are dramatically changed even if they belong to the same WBC type.

The difference in domain generalization ability between ResNet and VGG is counter-intuitive because ResNet is one generation ahead of VGG and should not be significantly worse than VGG. We infer that an architectural difference between these two models is causing this gap, which we investigate in Sec. 3.

## 3. BATCH NORM ISSUE UNDER DOMAIN SHIFT

ResNet’s performance is drastically reduced under domain shifts when compared to VGG, as seen in Fig. 4-a and -b. In this section, we investigate the cause of this issue and explore possible ways to address it.

**Identifying the cause of the issue.** To identify the building block responsible for the difference in domain generalization between ResNet and VGG, we examine the architectural differences between the models and conduct ablative experiments. We observe that, in addition to the residual connections, **(a)** *ResNet removed Fully-Connected (FC) layers from VGG*, and that **(b)** *ResNet added Batch Normalization (BN) layers, which were not present in VGG*. Therefore, we perform two ablation studies on ResNet: **(a’)** *adding FC layers to ResNet*, and **(b’)** *adding BN layers to VGG*. We summarize our observations and ablations in Table 2. We note that (b’) is a compromised solution as we cannot stably train very deep CNNs without a normalization technique like BN [25, 26]. We also note that adding BN to VGG requires ImageNet pretrained weights of VGG with BN, which are available in PyTorch. We do not conduct ablative experiments on the residual connections as they are the essence of ResNet: Res(idual)-Net(work). The results are shown in Table 2. The addition of VGG-style fully-connected layers to ResNet, (a) → (a’), improves the accuracy (%) on LISC set from 23.11 ± 3.04 to 27.74 ± 3.44, but is still far behind VGG’s 74.44 ± 2.72. On the other hand, the addition of ResNet-style BN layers to VGG, (b) → (b’), causes a sudden drop in accuracy on LISC from 74.44 ± 2.72 to 33.74 ± 8.04. This indicates that *batch normalization is the cause of the poor generalization*.

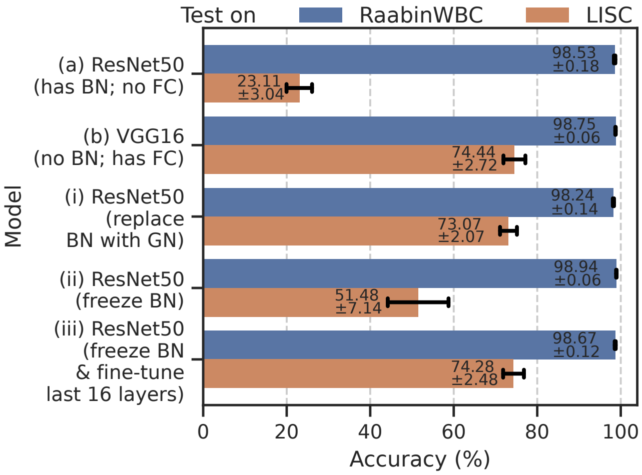


**Fig. 4**: Accuracies of different models trained on RaabinWBC, while tested on RaabinWBC (no domain shift) and LISC (domain shift) datasets. *(a)*: ResNet50 has unreasonable accuracy drop when tested under domain shifts, but *(b)*: VGG does not have it. *(i)*: Replacing Batch Normalization (BN) with Group Normalization (GN) prevents the unreasonable accuracy drop. *(ii)* and *(iii)*: Freezing BN layers and some more layers can also alleviate it. See Sec. 3 for details.

**Literature about BN's problems.** Now we know that batch normalization (BN) [25] is the problem, so we briefly review the related literature to find ways to address it. The BN normalizes the intermediate feature maps of CNNs to have a mean of zero and a standard deviation (std) of one over the training set. During training, it estimates the mean and std using moving averages from mini-batches of SGD, and during testing, it normalizes the feature maps using the mean and std estimated in training. Intuitively, this makes it harder to generalize to different domains during testing, which has also been reported in a preprint[27] on natural image classification. However, its proposed solution [27] is unfortunately not applicable to our work, as it trains domain-invariant batch normalization by having multiple domains during training, which we do not have. Additionally, we found a preprint [28] highlighting BN instability issues due to reliance on the batch for mean/std estimation. A potential solution is Group Normalization (GN), which normalizes feature maps within images using channel/height/width dimensions instead of batch dimension.

**Fixing BN's issue.** Inspired by the literature mentioned above regarding BN, we discuss two potential solutions for our generalization issue. Simply removing batch normalization from ResNet is not a viable option, as training a deep network without a technique like BN can be very unstable [25, 26]. Additionally, we require ImageNet pretrained weights, which are unavailable for ResNet without BN.**(i)** The first option is to *Replace BN with GN.* The idea behind this approach is that if BN normalizes approximately over the training set, making the model dependent on the training domain, then normalizing within each image using GN can make the model independent of the training domain. A disadvantage is the need to prepare ImageNet pretrained weights for ResNet with GN, which fortunately can be avoided since the authors of GN have already published the ImageNet pretrained weights.

**(ii)** The second option is to *freeze the BN layers with ImageNet pretrained parameters.* The concept behind this approach is that we can prevent overfitting to the training domain by enforcing the use of mean and std from a completely different domain, namely, ImageNet.

**Repaired ResNet results:** We show the performance of our proposed fixes in Fig. 4-i and -ii. (a) → (i): Replacing BN with GN significantly improved ResNet accuracy (%) on the LISC dataset from $23.11 \pm 3.04$ to $73.07 \pm 2.07$, which is comparable to VGG's $74.44 \pm 2.72$. (a) → (ii): Freezing the BN layers with ImageNet pretrained parameters improved ResNet's accuracy on the LISC set to $51.48 \pm 7.14$, although still trailing behind VGG's. To address this, we observed that ResNet is deeper than VGG and may be more susceptible to overfitting to a specific domain. Consequently, we experimented **(iii)** *freezing the first layers and finetuning only the last 16 layers*, the same number as VGG16. This improved ResNet's accuracy on the LISC set to $74.28 \pm 2.48$, matching VGG's. However, we note that this solution is not based on a solid foundation and is more of a hack, since we cannot explain why ResNet with GN did not require freezing some layers if the deeper layers were the issue.

## 4. PERFORMANCE ANALYSIS

This section examines the performance of the baseline CNN, ResNet50 with group normalization. Out of the 10 trained models, we selected the one with an accuracy of 74.31% (the closest to the mean) for further analysis and present the precision, recall, and F-measure (harmonic mean of them) in Table 3 and the confusion matrix in Fig. 5.

The classifier achieves a high F-measure for neutrophils, and relatively high F-measures for eosinophils and lymphocytes, both of which exhibit lower precision than recall. Monocytes have the second lowest F-measure, with both precision and recall equally low. Monocytes and lymphocytes are frequently confused with each other. Basophils have the worst F-measure, where recall is particularly low and precision is high. Basophils are the most difficult to classify and are often misclassified as eosinophils or monocytes. This poor performance may be due to the heavy class imbalance in the training data (see Table 1), reflecting the fact that basophils are the rarest white blood cells, composing only 1% or less. We observe that the majority of classes tend to have higher F-measures, while the minority classes have lower F-measures, indicating that addressing the heavy class imbalance in the training phase could be future work.

| | Bas. | Eos. | Lymp. | Mono. | Neut. |
|---|---|---|---|---|---|
| Precision | 94.44 | 69.23 | 68.35 | 54.90 | 98.25 |
| Recall | 30.91 | 92.31 | 91.53 | 58.33 | 100.00 |
| F-measure | 46.58 | 79.12 | 78.26 | 56.57 | 99.12 |

**Table 3**: Precision and Recall on LISC

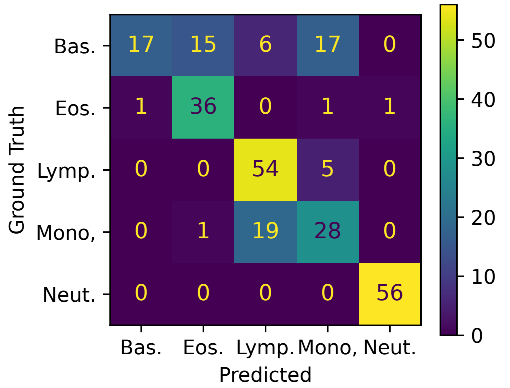


**Fig. 5**: Confusion matrix tested on LISC while trained on Raabin-WBC. Basophils (Bas.) is the most challenging WBC to recognize, which may be due to the extremely low quantity (see the first row of Table 1) in the training data than other WBC types.

## 5. CONCLUSION

In this work, we established a benchmark for white blood cell (WBC) classification. Building on excellent prior work and publicly available datasets, we demonstrated that standard CNNs perform well when evaluated under imaging conditions similar to the training data. Consequently, we suggest shifting focus towards evaluation under unseen imaging conditions, which is more realistic. We trained baseline models, evaluated them under domain shifts, and identified opportunities for further improvement. We also discovered an issue with batch normalization, a commonly used technique in many baseline CNNs, and proposed ways to address it. We have made our code[1] publicly available for full reproducibility of our results.

**Acknowledgements.** This work was supported in part by Sysmex Corporation, Japan. It was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. We thank Winnie Pang for helpful discussions.

**Appendix.** We encourage readers to refer to our arXiv preprint version, which includes supplementary material.

## 6. REFERENCES

[1] Lorenzo Putzu, Giovanni Caocci, and Cecilia Di Ruberto, "Leucocyte classification for leukaemia detection using image processing techniques," *Artif. Intell. Med.*, vol. 62, no. 3, pp. 179–191, 2014.

[2] Zahra Mousavi Kouzehkanan, Sepehr Saghari, Sajad Tavakoli, Peyman Rostami, Mohammadjavad Abaszadeh, Farzaneh Mirzadeh, Esmaeil Shahabi Satlsar, Maryam Gheidishahran, Fatemeh Gorgi, Saeed Mohammadi, et al., "A large dataset of white blood cells containing cell locations and types, along with segmented nuclei and cytoplasm," *Sci. Rep.*, vol. 12, no. 1, pp. 1–14, 2022.

[3] Martin S. Blumenreich, "The white blood cell and differential count," in *Clinical Methods: The History, Physical, and Laboratory Examinations*, H Kenneth Walker, W Dallas Hall, and J Willis Hurst, Eds., chapter 153. Butterworths, Boston, 1990.

[4] Sawsan F Bikhet, Ahmed M Darwish, Hany A Tolba, and Samir I Shaheen, "Segmentation and classification of white blood cells," in *ICASSP*, 2000.

[5] Angelo Genovese, Mahdi S Hosseini, Vincenzo Piuri, Konstantinos N Plataniotis, and Fabio Scotti, "Acute lymphoblastic leukemia detection based on adaptive unsharpening and deep learning," in *ICASSP*, 2021.

[6] Shubhangi Khobragade, Dheeraj D Mor, and CY Patil, "Detection of leukemia in microscopic white blood cell images," in *ICIP*, 2015.

[7] Ruggero Donida Labati, Vincenzo Piuri, and Fabio Scotti, "All-IDB: The acute lymphoblastic leukemia image database for image processing," in *ICIP*, 2011.

[8] Puneet Mathur, Mehak Piplani, Ramit Sawhney, Amit Jindal, and Rajiv Ratn Shah, "Mixup multi-attention multi-tasking model for early-stage leukemia identification," in *ICASSP*, 2020.

[9] Seyed Hamid Rezatofighi and Hamid Soltanian-Zadeh, "Automatic recognition of five types of white blood cells in peripheral blood," *Comput. Med. Imaging. Graph.*, vol. 35, no. 4, pp. 333–343, 2011.

[10] Mostafa M. A. Mohamed, Behrouz Homayoun Far, and Amr Guaily, "An efficient technique for white blood cells nuclei automatic segmentation," in *IEEE SMC*, 2012.

[11] Sajad Tavakoli, Ali Ghaffari, and Zahra Mousavi Kouzehkanan, "Generalizability in white blood cells' classification problem," *bioRxiv 2021.05.12.443717v3*, 2021.

[12] Sajad Tavakoli, Ali Ghaffari, Zahra Mousavi Kouzehkanan, and Reshad Hosseini, "New segmentation and feature extraction algorithm for classification of white blood cells in peripheral smear images," *Sci. Rep.*, vol. 11, no. 1, pp. 1–13, 2021.

[13] Changhun Jung, Mohammed Abuhamad, David Mohaisen, Kyungja Han, and DaeHun Nyang, "WBC image classification and generative models based on convolutional neural network," *BMC Med. Imaging*, vol. 22, no. 1, pp. 1–16, 2022.

[14] Mohammad Zolfaghari and Hedieh Sajedi, "A survey on automated detection and classification of acute leukemia and WBCs in microscopic blood cells," *Multimed. Tools. Appl.*, vol. 81, no. 5, pp. 6723–6753, 2022.

[15] Khaled Almezhghwi and Sertan Serte, "Improved classification of white blood cells with the generative adversarial network and deep convolutional neural network," *Comput. Intell. Neurosci.*, 2020.

[16] Hua Chen, Juan Liu, Chunbing Hua, Zhiqun Zuo, Jing Feng, Baochuan Pang, and Di Xiao, "Transmixnet: An attention based double-branch model for white blood cell classification and its training with the fuzzified training data," in *BIBM*, 2021.

[17] Saba Saleem, Javeria Amin, Muhammad Sharif, Muhammad Almas Anjum, Muhammad Iqbal, and Shui-Hua Wang, "A deep network designed for segmentation and classification of leukemia using fusion of the transfer learning models," *Complex Intell. Syst.*, vol. 8, no. 4, pp. 3105–3120, 2022.

[18] Hua Chen, Juan Liu, Chunbing Hua, Jing Feng, Baochuan Pang, Dehua Cao, and Cheng Li, "Accurate classification of white blood cells by coupling pre-trained ResNet and DenseNet with SCAM mechanism," *BMC Bioinform.*, vol. 23, no. 1, pp. 1–20, 2022.

[19] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2014.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[21] Bec Crew, "Google Scholar reveals its most influential papers for 2019," *Nature Index*, 2019.

[22] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," in *ICLR*, 2018.

[23] Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher, "A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation," in *ICLR*, 2018.

[24] Laurens Van der Maaten and Geoffrey Hinton, "Visualizing data using t-SNE.," *J. Mach. Learn. Res.*, vol. 9, no. 11, 2008.

[25] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[26] Yuxin Wu and Kaiming He, "Group normalization," in *ECCV*, 2018, pp. 3–19.

[27] Lian Qing, Lin Yong, and Tong Zhang, "Invariant batch normalization for multi-source domain generalization," *Openreview.net*, 2020.

[28] Yuxin Wu and Justin Johnson, "Rethinking "batch" in BatchNorm," *arXiv:2105.07576*, 2021.

| Index | Model | Test Accuracy (%)<br>RaabinWBC-A | <br>LISC |
|---|---|---|---|
| (a) | Default ResNet50 (has batch norm; no fully-connected layers) | $98.53 \pm 0.18$ | $23.11 \pm 3.04$ |
| (a') | ResNet50 (has batch norm; add fully-connected layers) | $98.46 \pm 0.19$ | $27.74 \pm 3.44$ |
| (i) | ResNet50 (replace batch norm with group norm; no fully-connected layers) | $98.24 \pm 0.14$ | $73.07 \pm 2.07$ |
| (ii) | ResNet50 (freeze batch norm; no fully-connected layers) | $98.94 \pm 0.06$ | $51.48 \pm 7.14$ |
| (iii) | ResNet50 (freeze batch norm; no fully-connected layers; fine-tune last 16 layers only) | $98.67 \pm 0.12$ | $74.24 \pm 2.46$ |
| (b) | Default VGG16 (no batch norm; has fully-connected layers) | $98.75 \pm 0.06$ | $74.44 \pm 2.72$ |
| (b') | VGG16 (add batch norm; has fully-connected layers) | $98.64 \pm 0.18$ | $32.33 \pm 6.17$ |
| (c) | ViT-Base-16 (use layer norm instead of batch norm) | $98.33 \pm 0.14$ | $69.77 \pm 3.09$ |
| (d) | ConvNeXt-Tiny (similar # params with ResNet50 but many incremental updates; use layer norm instead of batch norm) | $98.83 \pm 0.09$ | $67.35 \pm 2.51$ |

**Table 4**: Results of ViT and ConvNeXt models along with those presented in the main paper. The Index is the same as in Table 2 and Fig. 4.

## 7. SUPPLEMENTARY MATERIEL

### 7.1. PBC Dataset

We regret that we did not include the PBC dataset [29] in our study. This dataset comprises a total of 17,092 images that can be used for training and testing, while the specific data split employed by the authors is not available. In addition, it includes more classes than the five classes that we used in our experiments. It is worth noting that this dataset is as large as the RaabinWBC dataset [2], and was actually published before the RaabinWBC dataset. However, neither dataset cited the other. We became aware of this after the acceptance of our paper, and have included this information here for the sake of completeness.

### 7.2. Recent Models without Batch Norm

In our main paper, we demonstrated that batch normalization makes it difficult for models to generalize beyond the training dataset. To address this issue, we proposed the use of group normalization as an alternative normalization technique that is not dependent on batch. We note that some recent image classification models, including ConvNeXt[30] and Vision Transformers [31], have already abandoned batch normalization in favor of layer normalization, which is a special case of group normalization. We evaluated the performance of these models pretrained on ImageNet and report the results in lines (c) and (d) of Table 4. As expected, these models did not exhibit a significant drop in accuracy on the LISC dataset, in contrast to the default ResNet model with batch normalization.

## 8. REFERENCES

[29] Andrea Acevedo, Anna Merino, Santiago Alférez, Ángel Molina, Laura Boldú, and José Rodellar, "A dataset of microscopic peripheral blood cell images for development of automatic recognition systems," *Data in Brief*, vol. 30, 2020.

[30] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie, "A convnet for the 2020s," in *CVPR*, 2022.

[31] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.