BloodShot: Few-shot Learning On Blood Cells

Tomer Laor

August 12, 2022

1 Introduction

Annotated data is expensive, especially in a case that the annotators have to be domain specialists. Neural network based classifiers preforms very well on images, but requires a lot of data to train from scratch. It requires researchers to create vast image datasets in order to train neural network based classifiers to solve the task at hand. Recent studies in the field of few-shot learning such as [7], [4] shows that it's possible to train a neural network classifier on a generic dataset and use it to classify images from a novel dataset without further training. Other studies in the field of semi-supervised learning, such as [10] shows that it's possible to train a neural network from scratch on a mostly unlabeled dataset, and use as little as 15 labels per class to provide a reasonable performance. Presenting a few-shot learning pipeline in the domain of cells from microscopy images is important since these datasets are expensive to create and expensive to label (requires domain specialists). Given a dataset, even a small one, a few-shot learning model requires to label only few images from each class before helping researchers and doctors making decisions. An important aspect of our work is generality - the presented few-shot learning pipeline is generic and can be used for lots of datasets in the microscopy domain since no actions were taken to make only the specific datasets that we worked on closer visually or conceptually.

In this work, we will use "A Single-cell Morphological Dataset of Leukocytes from AML Patients and Non-malignant Controls" (AML for short) dataset as our generic dataset and "ALL Challenge dataset of ISBI 2019" (ALL-challenge) as the novel dataset. We will compare our results with [5] on the AML dataset, and with [8] on the ALL-challenge dataset. Our goal is to achieve an accuracy that is bigger than base rate (random guess) on the ALL dataset using only few labeled images.

2 Datasets

Both datasets contains humans' white blood cells, both aim to give predictions regarding a single cell resolution. The datasets contain white blood cells of healthy patients and white blood cells of patients with leukemia. Each dataset contains blood cells of a single type of leukemia. While both datasets contain blood cells of patients with leukemia, each exhibits a different type of leukemia. In each dataset, the leukemia can be detected by looking on different cells [1] [2].

As seen in figure 1, the datasets look different; two main differences that are observable immediately are background color, black vs white and background noise, clean background vs noisy background.

AML Dataset. The AML dataset is used to classify the type of the white blood cell. This dataset consists of 18,365 images from 15 classes. This dataset was created from white blood cells of healthy patients and from white blood cells of patients with Acute Myeloid Leukemia. In Acute Myeloid Leukemia, the myeloid stem cells usually become a type of immature white blood cell called myeloblasts (or myeloid blasts). The myeloblasts in AML are abnormal and do not become healthy white blood cells [1].

ALL-challenge Dataset. The ALL-challenge dataset is used to classify whether or not a cell is cancerous. This dataset was already split into train and test since it was used for a competition. There are 10,661 images in its train dataset and 1,867 images in its test dataset. This dataset was created from white blood cells of healthy patients and from white blods cells of patients with Acute Lymphocytic Leukemia. In Acute Lymphocytic Leukemia, the bone marrow makes too many lymphocytes [2]. In this dataset we have only B lymphoblasts, which should mature into adult lymphocytes [3], and

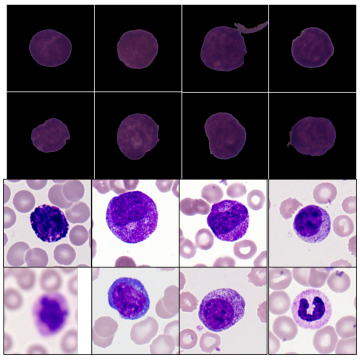

Figure 1: Images from the datasets used in BloodShot. The images with the black background originate from the ALL-challenge dataset, while the images with the white background originate from the AML dataset.

B lymphoids. In the ALL-challenge dataset, B lymphoblasts are labeled as cancerous cells while B lymphoids are labeled as normal cells. The ALL-challenge dataset contains only these classes.

3 Method

Our few-shot learning pipeline has 3 phases: 1. Training of the embedder, in which a neural network is trained on a generic dataset. 2. Memorizing of few images from the novel dataset, by extracting embeddings of these images using the trained embedder and creating representative embedding for each class. 3. Inferencing on the novel dataset, using cosine similarity between the embedding of an incoming image and the memorized represntative embeddings. The inferenced label is the label of the representative embedding.

Creating an embedding for an image that can be compared with another embedding is not an easy task. The task of comparing an embedding to another embedding and decide whether they came from the same class or not is called verification. Previous studies, for example [6] has showed that softmax loss, which is normally used for classification tasks, preforms badly in a verification context. In order to create a neural network that extracts embeddings that can be used for verification, dedicated losses should be used for example AAM-Softmax [6] or triplet loss [9]. [5] was trained using regular softmax loss, which is not suitable for verification tasks. While they provide the weights of the model and the inference code, they don't provide their training code, which mean that we have to implement our own training code.

All of the experiments were conducted on a server with Intel Xeon Silver 4110 CPU, Nvidia RTX 2080 Ti GPU and 128 GB of RAM.

Data Processing. We removed images with the classes KSC (Smudge cell), LYA (Atypical Lymphocyte), MMZ (Metamyelocyte), and MOB (Monoblast) from the AML dataset since they have less than 30 samples per class. We prefer our embedder to learn class centroids from a lot of samples per class, thus classes with a small number of samples are not helpful to our embedder training. For the AML dataset, we used 70% of the data for training, 10% for validation and 20% for testing. For the AML dataset, we used the training dataset to train the neural networks, the validation dataset for validation metrics while training the neural networks and the test dataset to report the final classification metrics. The ALL-challenge dataset came split into train and test, and the train dataset was split into 3 folds. We used the first fold to pick images during the memorizing phase and we used the preliminary test dataset to report the final metrics. Our preprocess is resize into a width and a height of 224 pixels, take only the first 3 channels of the image and rescale the pixels to be between 0 and 1.

Loss. Our loss consists of 2 losses. The first loss is AAM-Softmax [6], also known as ArcFace loss. The purpose of this loss is to make the embeddings of the neural network behave in a suitable way for a verification task. This loss punishes the neural network for creating an embedding in which the angle between it and the center of the sample's class is large. This property allows us to use cosine similarity between 2 embeddings that the neural network extracts. The second loss is consistency loss, also known as consistency regularization. The purpose of this loss in BloodShot is to train the neural

Description	Accuracy	F1-Score of Cancer Cells	F1-Score of Non-Cancer Cells
Softmax loss	0.442±0.088	0.501±0.097	0.353±0.116
AAM-Softmax loss	0.590±0.029	0.709±0.037	0.294±0.031
AAM-Softmax and consistency loss	0.639±0.047	0.703±0.060	0.531±0.037

Table 1: Results on the ALL-challenge dataset using 10 images to memorize

network to make similar embeddings for similarly geometric cells. This loss is computed by rotating every image in a random angle, and measure the L2 distance between the embedding of the rotated image and the embedding of the original image. This loss is taken from semi-supervised learning approaches, for example [10]. Both losses are combined using addition. Our loss can be denoted by: $Loss(embeddings, labels) = AAMSoftmax(embeddings, labels) + \lambda * ConsistencyLoss(embeddings)$. After some experimentation, we found that $\lambda = 1$ works well.

Training Process. We used EfficientNetB0 [11] as our backbone. We trained for 6 epochs with a batch size of 32 using the Adam optimizer with a learning rate of 0.001. We picked the weights of the last epoch. We noticed that when we train the model for more than 6 epochs, while the train and validation loss decreases, the accuracy in few-shot conditions on the ALL-challenge dataset decreases. We hypothesize that training the model for more than 6 epochs causes the model to overfit the AML dataset domain. We tried to mitigate this issue by adding dropout between the last layer of the backbone and the classification head, but it resulted in an accuracy in few-shot conditions on the ALL-challenge dataset that is similar to a random guess.

Evaluation Process. The evaluation for the AML dataset is a simple evaluation. We took the AML test dataset, feed it through the neural network including the classification layer and evaluate the classes that the neural network outputs in a simple feed forward. The evaluation for the ALL-challenge dataset is more complicated since it's a few-shot evaluation. Our evaluation in few-shot conditions is inspired by [4]. In the memorizing phase, we randomly choose k images per class, extract embeddings from them by feed them into the neural network without the classification layer and take the mean embedding of each class to be the representative embedding of this class. In the inference phase, we extract the embedding of every image (similarly to the memorizing phase) and apply cosine similarity between the embedding of this image with the representative embeddings that we found in the memorizing phase. We say that the label of a given image is the label of the closest representative embedding. We repeat this process for 100 times and report the mean and std of the metrics. This process' random generator was seeded to be able to evaluate different models on exactly the same scenario.

4 Results

In the 10-shot conditions, we are comparing between different BloodShot models. In all the other evaluations we are presenting results that were obtained using the BloodShot model that was trained using AAM-Softmax and consistency loss since this model yields the best results in few-shot conditions.

10-shot Results on the ALL-challenge Dataset. The results in 10-shot scenario can be found in table 1. The base-rate accuracy in these experiments is 0.5, which is a uniform distribution over 2 classes. The base-rate is uniform since the classifier don't know anything about the distribution of the test dataset - it knows only k images from each class. Using softmax loss results in an accuracy which is worse than a random guess. Using AAM-Softmax by itself results in accuracy of 0.590 which is significantly higher than the base-rate. Using AAM-Softmax and consistency loss results in an accuracy of 0.639, the highest accuracy that BloodShot achieved. Although the accuracy is significantly higher than base-rate accuracy, it's still low compared to non k-shot methods, such as [8] that achieved an impressive accuracy of 0.879. [8] used over 10,000 images from the ALL-challenge dataset while we used only 20 images (10 per class) in this scenario. Since we used only a 0.2% images from the ALL-challenge dataset compared to [8], our results are very impressive.

2D Visualization. Figure 2 contains 400 random embeddings of images from the ALL-challenge dataset after PCA dimensionality reduction to 2 dimensions. This figure indicates that our embeddings create some separability between cancer cells and normal cells. It's important to note that this

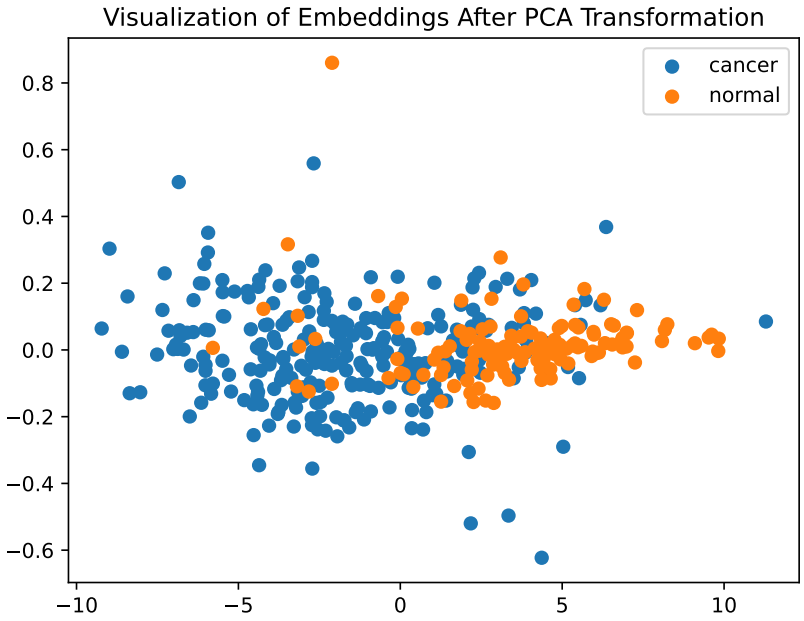

Figure 2: Random chosen embeddings after PCA transformation to 2D.

Class	Class Short Name	Precision	Sensetivity (Recall)	Number of images
Neutrophil (segmented)	NGS	0.99	0.97	1702
Neutrophil (band)	NGB	0.0	0.0	16
Lymphocyte (typical)	LYT	0.87	0.94	813
Monocyte	MON	0.86	0.81	334
Eosinophil	EOS	0.79	0.72	99
Basophil	BAS	0.0	0.0	20
Myeloblast	MYO	0.84	0.95	631
Promyelocyte	PMO	0.0	0.0	12
Promyelocyte (bilobled)	PMB	0.0	0.0	4
Myelocyte	MYB	0.0	0.0	10
Erythroblast	EBO	0.0	0.0	20

Table 2: Classification Results on the AML dataset

seperability was created although the neural network didn't see images from the ALL-challenge dataset during its training.

Classification Results on the AML Dataset. The results, as shown in table 2 indicates that our classifier can classify between different classes from the AML dataset. The classifier wasn't able to detect images of classes that are not very common in the dataset. These results are worst than the results from [5], especially on rare labels. [5] used all the classes in the dataset, while we filtered out the rarest classes. [5] used augmentations while we didn't. [5] used a different backbone that has 4 times more parameters compared to the backbone that we used. [5] used softmax loss while we used AAM-Softmax and consistency loss. This comparison is important, but these results are not the goal of BloodShot. The goal of BloodShot is to deliver an accuracy higher than base-rate in few-shot conditions.

Number of Memorizing Images. We evaluate how the number of memorizing images affects accuracy. The accuracy per number of memorizing images is averaged over 20 random memorizing images choices instead of 100 random choices to reduce run time. As observed in figure 3, more images to memorize usually mean higher accuracy. The accuracy using only 1 image per class for memorization the is almost 0.60 while using 50 images per class for memorization the accuracy climbs

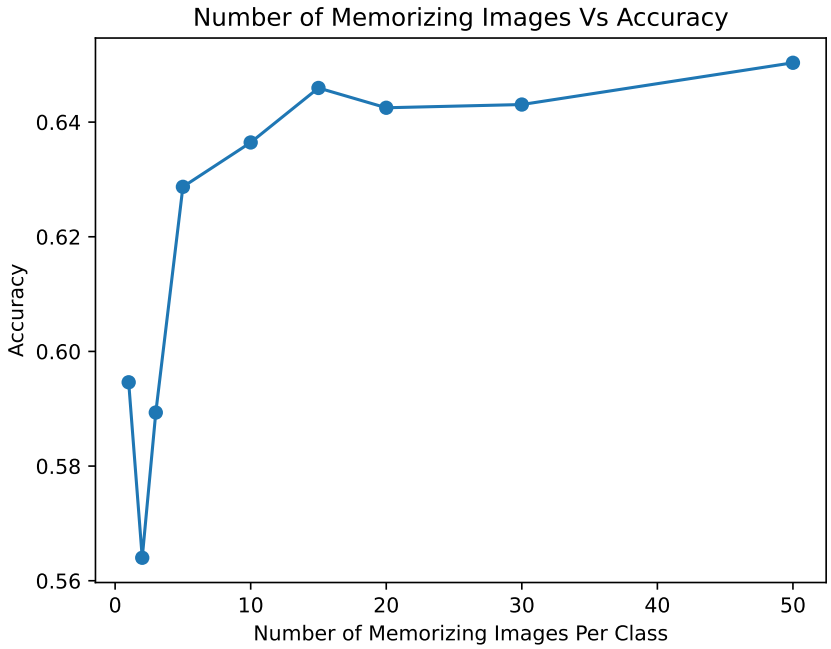

Figure 3: Accuracy given k images for memorization. k is [1, 2, 3, 5, 10, 15, 20, 30, 50].

up to 0.65. We can observe that in a case that less than 5 images per class are annotated, it's better to use only 1 image. We hypothesize that this phenomenon is caused by the use of mean to calculate the representative embedding and that less than 5 embeddings probably creates a bad mean.

5 Discussion

We introduced a technique to preform few-shot learning tasks on images of cells which includes training a neural network with a loss that is constructed from AAM-Softmax loss and consistency loss. BloodShot's scheme is generic and can be applied to other cell images datasets as well. In this work we tested our technique by training a neural network on the AML dataset and using the resulted neural network in few-shot conditions on the ALL-challenge dataset. We showed that our technique provides an accuracy that is significantly higher than the base-rate accuracy, even when using only 1 image per class for memorization. The training and evaluation code as well as the model's weights can be found on https://github.com/TomerMe2/BloodShotLearning.

While making this work I learned that every dataset looks different, even in the case that it should show similar cells in the same context. I learned that the domain gap is a big issue in the microscopy field, since every processing step makes the datasets look very different from one another. I insisted on making the translation between the datasets as generic as possible and leaving out preprocessing steps that could have help, such as segmenting the AML dataset and make its background black, similarly to the ALL dataset, since it would overfit to the adaptation between these 2 datasets. This work has a lot of future work options. For example: try to avoid domain overfitting better, find augmentations that will improve the pipeline, find better losses, evaluate the few-shot conditions on more datasets and find a more suitable neural network architecture.

References

[1] Acute myeloid leukemia treatment (pdq®)–patient version. *National Cancer Institute*.

[2] Adult acute lymphoblastic leukemia treatment (pdq®)–patient version. *National Cancer Institute*.

[3] Lymphoblast definition. *National Cancer Institute*.

[4] Bendou, Y., Hu, Y., Lafargue, R., Lioi, G., Pasdeloup, B., Pateux, S., and Gripon, V. EASY: Ensemble augmented-shot y-shaped learning: State-of-the-art few-shot classification with simple ingredients.

[5] Christian Matek, Simone Schwarz, K. S. C. M. Human-level recognition of blast cells in acute myeloid leukemia with convolutional neural networks.

[6] Deng, J., Guo, J., Xue, N., and Zafeiriou, S. ArcFace: Additive angular margin loss for deep face recognition.

[7] Hu, S. X., Li, D., Stühmer, J., Kim, M., and Hospedales, T. M. Pushing the limits of simple pipelines for few-shot learning: External data and fine-tuning make a difference.

[8] Prellberg, J., and Kramer, O. Acute lymphoblastic leukemia classification from microscopic images using convolutional neural networks.

[9] Schroff, F., Kalenichenko, D., and Philbin, J. FaceNet: A unified embedding for face recognition and clustering.

[10] Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. FixMatch: Simplifying semi-supervised learning with consistency and confidence.

[11] Tan, M., and Le, Q. V. EfficientNet: Rethinking model scaling for convolutional neural networks.