Leukemic cell detection model based on deep learning

Hongyi Chen

St. George’s School, Newport, USA
chenhongyi030116@163.com

Abstract. This project examines the feasibility of building a deep learning model that can accurately distinguish leukemic cells from normal cells, as well as the viability of analyzing the black box model so as to explain it in human terms. An accurate and explainable model can be deployed in areas previously inaccessible to the limited existing methods of detecting leukemia in its early stages. It is acknowledged that this is a very difficult process, as cells from individuals vary and cancerous cells are highly morphologically similar to normal cells.

There are two major steps to this project. First, we built the cell classification model by supervised training on a convolutional neural network with provided labeled training data. Many parameters had to be adjusted to yield better results when tested on separate testing data.

Second, we created a program designed to help explain the neural network, and the model is inserted into the program and tested on multiple examples. We examined the results, which highlighted regions the algorithm deems important in making its decision, and attempted to explain the algorithm as much as possible.

Though far from perfect, the results are intriguing. Our algorithm had an accuracy of 68% on the testing data set. This is clearly suboptimal; however, this is also significantly higher than random results, demonstrating that our algorithm learnt useful processes and that this research direction is worth exploring further. The main difficulty is the aforementioned similarity between leukemic and normal cells; thus, the training process must be complex and fine-tuned to yield good results. After analyzing the results from the explanation program, we determined that our algorithm focuses on the contour of cells in its decision-making process, especially unsmooth or protruded sections. To conclude, our algorithm is far from the level required for practical applications; however, we demonstrated that such algorithms are possible and can be explained.

1. Introduction
Leukemia stem cells are both stem-like and leukemic-like. This complicates their detection as rare circulating tumor cells in the peripheral blood of leukemia patients. [1] The cancer here examined is leukemia, cancer of blood and the lymphatic system. The cells examined in this project are B-lymphoblast cells, a type of white blood cells. Leukemia is difficult to identify in its early stages due to the similarity in appearance of cancerous cells to normal cells. Methods exist that can detect leukemia in its early stages, but these require expensive equipment that are not widely available. Relatively easy methods exist to identify leukemia in its later stages; however, late-stage leukemia is difficult to treat and can be deadly. We aim to use machine learning to find a cost-effective method that can reliably detect leukemia in its early stages by examining images of white blood cells.

Deep learning is a very powerful problem-solving tool. It requires a lot of data and allows the computer to learn patterns itself. The specific algorithm used here is supervised convolutional neural network, where the training data set has clear inputs and labels that the algorithm can learn from. Deep
learning is powerful and data-driven because the creator of the algorithm need not know the specifics of the domain knowledge; instead, only a learning model needs to be created and optimized.

However, machine learning has an important issue: it is a black box, a model where the internal workings and processes are unknown to any observer. Hence, we also attempt to interpret the result in this project, which is important for medical applications. Others have made efforts towards the interpretation of machine learning models. For example, Selvaraju et al. examined in their paper the method of highlighting regions of images a neural network finds important, which helps humans explain the neural network. [2]

2. Motivation
This project examines the feasibility of creating a deep learning algorithm that can accurately distinguish leukemic cells from normal cells, as well as the viability of analyzing the algorithm so as to explain it in human terms. The greater goal is to aim for an accurate and explainable algorithm that can be deployed in areas previously inaccessible to the limited existing methods of detecting leukemia.

3. Methods
We first aim to build our model. We built a convolutional neural network [3], whose structure is discussed below, and repeatedly ran it over training data, which is provided by this site [4]. The data consists of stained images of single cells, including normal white blood cells and leukemic cells [5,6]. Training is stopped before accuracy of the algorithm on the training set becomes too high to avoid overfitting.

For the second part of this project, we adapted the theory and code from this paper [7]. We made changes in order to adapt it to our algorithm. This program determines the regions on an image most important in the algorithm’s decision-making by creating masks that hide certain parts of the image and assessing the difference in the algorithm’s output. Masks that significantly alters the algorithm’s result highlight the important regions of the image. The theoretical principles behind constructing the mask-determining program is described below.

4. Structure of Neural Network and Training of Algorithm
The architecture of our network is summarized in the figure. There are 8 layers, 5 of which are convolutional and 3 of which are fully connected. The rectified linear unit (ReLU) activation function is used for all of these layers except for the last one. Max pooling is used for each of the convolutional layers.

The first convolutional layer filters the 450×450×3 input images with 16 kernels of size 5×5. (ReLU, then pooled.)

The second convolutional layer takes the output of the first as input, and it filters it with 32 kernels of size 5×5. (ReLU, then pooled.)

The third convolutional layer takes the output of the second as input, and it filters it with 64 kernels of size 5×5. (ReLU, then pooled.)

The fourth convolutional layer takes the output of the third as input, and it filters it with 128 kernels of size 5×5. (ReLU, then pooled.)

The fifth convolutional layer takes the output of the fourth as input, and it filters it with 256 kernels of size 5×5. (ReLU, then pooled.)

The first fully connected layer takes as input the size 10×10×256 output of the fifth convolution layer, and it processes it to 400 neurons. (ReLU.)

The second fully connected layer takes the size 400 output of the first as input, and it processes it to 84 neurons. (ReLU.)

The third fully connected layer takes the size 84 output of the first as input, and it processes it to 2 neurons.
5. Explanation Program

\[
\min_{m \in [0,1]^A} E \left[ f \left( \Phi(x_0 (\bullet - \tau), M) \right) \right] + \lambda_1 \|1 - m\|_1 + \lambda_2 \sum_{u \in A} \| \nabla m(u) \|^\beta
\]  

(1)

This is the minimization function used to determine the regions of importance on images for our machine learning algorithm.

The first term calculates the score of the targeted region of the image. The mask is applied to the targeted region and the cost is calculated. However, this term alone does not fully constitute the minimization function due to the two reasons below, which are each resolved by the second and third terms.

We need the mask to be sparse. This means that the smallest mask that drops the score should be found. If the size of the mask is not controlled, the function would simply blacken the entire image to fully increase the score. This would yield no useful results. Thus, we find the difference the mask makes in the image through the second term. \( \lambda_1 \) is the constant of adjustment.

We need the mask to be smooth. This means that the change between each pixel position of the mask should be small. If the smoothness is not controlled, the function may yield a highly disjoint mask, which would be impossible to analyze. Thus, we find the sum of the difference between adjacent pixel positions of the mask through the third term. \( \lambda_2 \) is the constant of adjustment.

6. Experiment Results

When our algorithm is applied to the testing data set, the accuracy of the results is around 68%.

We find that the contour of the cell is the important factor in the algorithm’s classification of the cell as normal or cancerous. In the following images, the contours of the cells are highlighted much more significantly than the interior of the cells: many portions of the cell contour are highlighted, while only small spots in the interior are.

Continuing to use lambda 1 and lambda 2 as before, we adjust their values to see their effects on the algorithm. When lambda 1 is increased, the sparseness of the image is emphasized for the algorithm. Fewer regions are highlighted, and less regions in the interior of the cell are highlighted. When lambda 2 is increased, the smoothness of the image is emphasized. When previously there
would be small highlighted spots on the interior of the cell, these spots are smoothed over and only the larger, more important spots in the contour of the cell remain highlighted.

The next page shows a few example cells and their masks. Specifically, example cells 3 and 4 also shows the effects of the increase of either lambda constants of adjustment.

Figure 2. Cell 1

Figure 3. Cell 1 Mark

Figure 4. Cell 2

Figure 5. Cell 2 Mark

Figure 6. Cell 3

Figure 7. Cell 3 Mark

Figure 8. Cell 3 Mask, lambda 1 increased

Figure 9. Cell 3 Mask, lambda 2 increased
7. Conclusion

Though far from perfect, the results are intriguing. Our algorithm had an accuracy of 68% on the testing data set. This is clearly suboptimal; however, this is also significantly higher than random results, demonstrating that our algorithm learnt useful processes and that this research direction is worth exploring further. The main difficulty is the aforementioned similarity between leukemic and normal cells; thus, the training process must be complex and fine-tuned to yield good results. After analyzing the results from the explanation program, we determined that our algorithm focuses on the contour of cells in its decision-making process, especially unsmooth or protruded sections. To conclude, our algorithm is far from the level required for practical applications; however, we demonstrated that such algorithms are possible and can be explained.

References

[1] Cooper, C.L., Leary, J.F., “Advanced flow cytometric analysis of nanoparticle targeting to rare leukemic stem cells in peripheral human blood in a defined model system”.
[2] Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." Proceedings of the IEEE international conference on computer vision. 2017.
[3] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.
[4] Gupta, Anubha, and Ritu Gupta. "ISBI 2019 C-NMC Challenge: Classification in Cancer Cell Imaging.
[5] Anubha Gupta, Rahul Duggal, Ritu Gupta, Lalit Kumar, Nisarg Thakkar, and Devprakash Satpathy, “GCTI-SN: Geometry-Inspired Chemical and Tissue Invariant Stain Normalization of Microscopic Medical Images,” under review.
[6] Ritu Gupta, Pramit Mallick, Rahul Duggal, Anubha Gupta, and Ojaswa Sharma, "Stain Color Normalization and Segmentation of Plasma Cells in Microscopic Images as a Prelude to...
Development of Computer Assisted Automated Disease Diagnostic Tool in Multiple Myeloma,” 16th International Myeloma Workshop (IMW), India, March 2017.

[7] Fong, Ruth C., and Andrea Vedaldi. "Interpretable explanations of black boxes by meaningful perturbation." Proceedings of the IEEE International Conference on Computer Vision. 2017.

[8] Cooper, C.L., Leary, J.F., “High-speed flow cytometric analysis of nanoparticle targeting to rare leukemic stem cells in peripheral human blood: preliminary in-vitro studies”.
