Application of Machine Learning to Distinguish Premature Leukemia Cells from Healthy Blood Cells

Noor Elassir

10/28/22

Abstract

Leukemia is a disease that primarily impacts children. The likelihood of beating this deadly disease increases with early detection. Because premature Leukemia blasts are unidentifiable by the human eye, different methods are needed to accurately diagnose cancer in its early stages. To solve this problem Convolutional Neural Networks (CNNS) have been created with classification validation accuracy rates as high as 96.15%. A dataset from the Cancer Imaging Archive [1] was used to train a CNN model. Of the 15,135 images in the dataset, 600 were used to prevent system crashing. With the limited resources available, a 63.6% validation accuracy was achieved.

Introduction

Leukemia is a type of cancer that causes bone marrow to produce an excessive amount of abnormal white blood cells that do not function properly. White Blood cells are vital for helping the body fight infection and disease. Due to the excessive amount of abnormal white blood cells, organ function begins to be compromised. Leukemia is the number 1 most diagnosed form of childhood cancer with children being 80% of all Leukemia diagnoses. The medical methods of diagnosing Leukemia are tedious, complicated, and sometimes invasive. Identification of malignant cells from normal cells using microscopic images is difficult for the human eye because morphologically both cells types appear similar. A method on the less invasive side is a complete blood count test that measures several blood components including red blood cells, white blood cells, hemoglobin, hematocrit, and platelets. Bone Marrow tests are also executed to diagnose Leukemia. Bone Marrow samples are normally taken from the hip bone where a hollow needle is used to remove a small number of marrow cells (aspiration) and a small piece of bone filled with marrow (biopsy). [2] A program to help identify and verify a diagnosis can save physicians precious time and serve as a second opinion. Survival rates decrease as staging advances. The model most suitable for this task is a Convolutional Neural Network that “learns” how to classify these cells by identifying the differences in training images.

Background

There has been recent research done by many scholars to apply AI to diagnose Leukemia. Specifically, transfer learning algorithms were used in their research to identify leukemic B-lymphoblasts. The researchers chose to use a CNN because “CNN is the most extensively used method for image recognition. It has high self-learning, adaptability, and generalization abilities. Traditional image recognition methods need feature extraction and classification, whereas CNN requires only the image data as an input to complete the image classification with the network's self-learning ability.” [3] The researchers developed multiple models and eventually developed an aggregated Deep Learning model— this algorithm scans the data to identify features that correlate and these features are then combined to promote faster learning without being told to do so explicitly. [4] Their model performance is extremely reliable with a

classification accuracy rate of 96.15%. (Wahidul Hasan Abir et.al, 2022) Further confirming that using a CNN is the best approach.

Dataset

The dataset used for the research is from the Cancer Imaging Archive as a part of their 2019 ALL Challenge. The dataset contains three sections of training, testing, and validation data. In total there are 15,135 images from 118 patients labeled either normal cell or leukemia blast. The data used consisted of 65% percent positive labels and 35% negative labels. Due to limited RAM, 600 images were used for the model to prevent crashing.

For usage in the model, the data used is normalized to ensure that each pixel has a similar data distribution. The channel average is subtracted from each input channel and the sum is divided by the standard deviation. The range of the pixel intensity values is changed as a result.

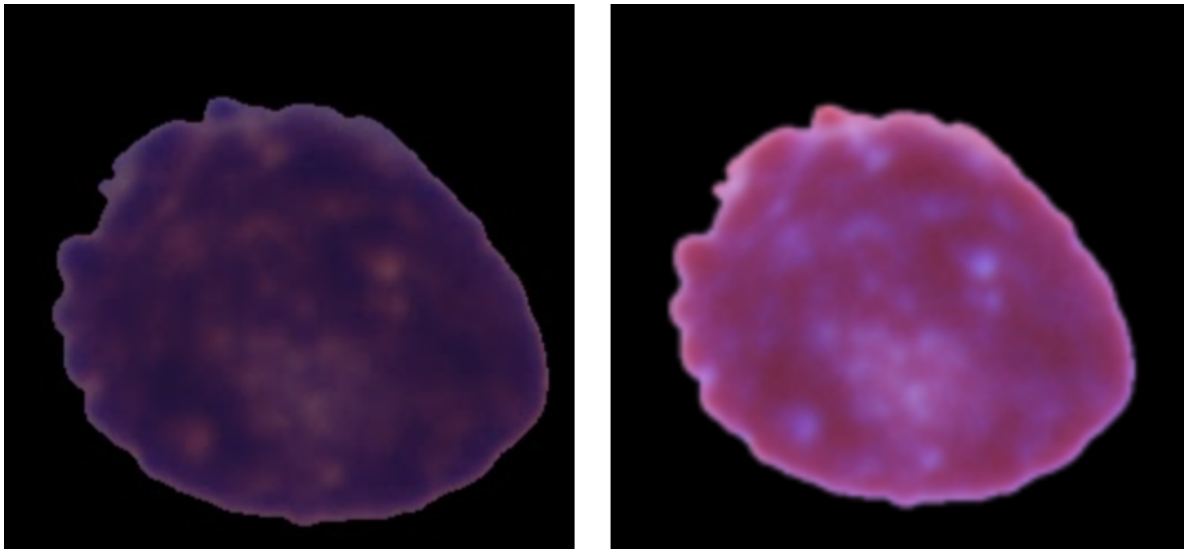

Figure 1: Before + After Normalization

Methodology / Models

We developed a CNN with 3 convolutional layers, each followed by batch normalization and max pooling to avoid overfitting. Batch normalization standardizes the inputs to a layer for each mini-batch. The technique speeds up training and uses higher learning rates, making learning easier. Max pooling is an operation that calculates the maximum value for patches of a feature map, and uses it to create a downsampled (pooled) feature map. Before the final output, there is a dropout layer to further avoid overfitting. (Figure 2)

We started with this simple model, then we added more layers and found no significant improvement in the performance. A translational pre-trained model is available in this case so the model needs to be trained from scratch. The adam optimizer set to a 0.01 learning rate was used.

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 450, 450, 3)]     0

 rescaling (Rescaling)       (None, 450, 450, 3)       0

 conv2d (Conv2D)             (None, 225, 225, 16)      208

 batch_normalization (BatchN  (None, 225, 225, 16)     64
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 113, 113, 16)     0
 )

 conv2d_1 (Conv2D)           (None, 57, 57, 32)        2080

 batch_normalization_1 (Batc  (None, 57, 57, 32)       128
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 29, 29, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 15, 15, 64)        8256

 batch_normalization_2 (Batc  (None, 15, 15, 64)       256
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 8, 8, 64)         0
 2D)

 conv2d_3 (Conv2D)           (None, 4, 4, 128)         32896

 batch_normalization_3 (Batc  (None, 4, 4, 128)        512
 hNormalization)

 dropout (Dropout)           (None, 4, 4, 128)         0

 global_average_pooling2d (G  (None, 128)              0
 lobalAveragePooling2D)

 dropout_1 (Dropout)         (None, 128)               0

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 44,529
Trainable params: 44,049
Non-trainable params: 480
_________________________________________________________________
```

Figure 2: Model Summary

Results and Discussion

With 20 epochs trained on 600 images in the dataset (train 80%, test 20%), the model achieved a validation accuracy of 0.63. (Figure 4) But by looking at the confusion matrix (Figure 3), we noticed that the model actually classifies everything into label 1. But from the history plot, we can see the trend of increasing training accuracy with more epochs, which implies an obvious under-fitted model. And on the test dataset, the model finds a sweet spot. With this trend observed, the model will definitely improve with more training data points and epochs. (not achievable with our resources due to limited RAM on Google Colab Pro). The training accuracy increasing confirms this result.

```
[[ 0 44]
 [ 0 76]]
```

Figure 3: Confusion Matrix

Although there is not enough trainable data, in the confusion matrix the optimizer finds a sweet spot that finds everything to be 1, but because we have a limited number of data and epochs, the accuracy would increase. With more data in the model, the model would perform well for its intended purpose.

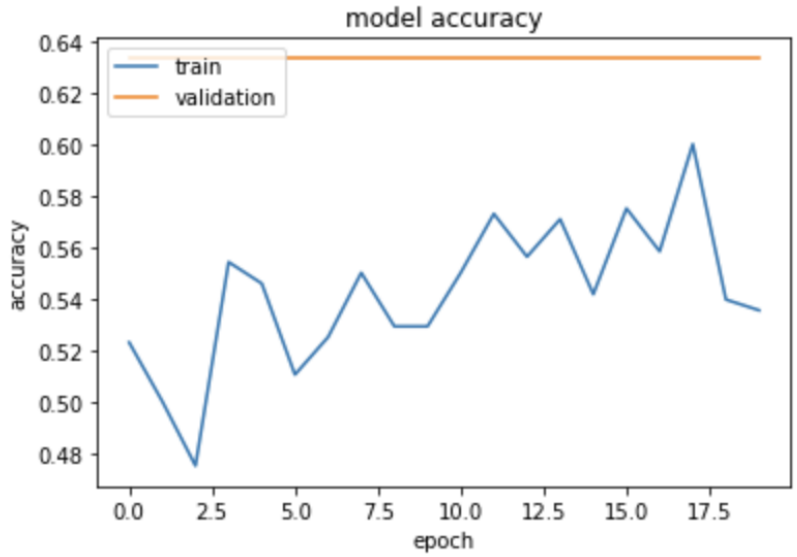

Figure 4: Model History

Conclusions

Although Leukemia is a disease we are still learning how to cure, early detection can be instrumental in saving people's lives. Premature Leukemia blasts are challenging to detect from a microscope—meaning that different methods are needed to accurately diagnose cancer. CNNs with classification validation accuracy rates as high as 96.15% have been created to help with validating a diagnosis. CNNs are the optimal approach to this problem. We have confirmed that the CNN works, so for a better model more data and more epochs are needed. The next steps for research would be to collect and compare data on detection stage vs survival rate and create a model to predict a survival rate based on the stage of cancer when diagnosed.

Acknowledgments

Thank you Yiran Li for mentoring me throughout the project with my research, analysis, and report writing.

References

1. Gupta, A., & Gupta, R. (2019). *ALL Challenge dataset of ISBI 2019* [Data set]. The Cancer Imaging Archive. https://doi.org/10.7937/tcia.2019.dc64i46r
2. LLS (2022) *Bone Marrow Tests*, The Leukemia Lymphoma Society https://www.lls.org/treatment/lab-and-imaging-tests/bone-marrow-tests
3. Abir, W. H., Uddin, M. F., Khanam, F. R., Tazin, T., Khan, M. M., Masud, M., & Aljahdali, S. (2021). Explainable AI in Diagnosing and Anticipating Leukemia Using Transfer Learning Method. Computational Intelligence and Neuroscience, 2022. https://doi.org/10.1155/2022/5140148
4. *Deep Learning and Its 5 Advantages*. (2020, January 28). becominghuman.ai. https://becominghuman.ai/deep-learning-and-its-5-advantages-eaeee1f31c86