Arman Siddiqui
Katie Rodeghiero
Connor Lydon

Neural Network to Classify White Blood Cell Types As an Early Infection Indicator

1. What problem are you trying to solve?

The problem we were trying to solve is to classify white blood cells from images. Diagnosis of blood based conditions require the classification of white blood cell counts in a patient. Automated methods can help improve accuracy and speed. There are four major blood types that our data includes: Neutrophils, Eosinophils, Lymphocytes, and Monocytes. A description of the major purpose and thumbnail of what they look like is included in the Appendix. Our data didn't include a fifth blood cell type, Basophils, but they only make up ~1% of the number of white blood cells.

2. What is the current state of the art solution for this problem?

Currently, identification of white blood cells primarily relies on people to evaluate the nucleuses and the cytoplasmic granules. This is often done by hand and can be time consuming. While this method certainly isn't broken, there are ways to speed up this process as well as increase the accuracy of those predictions. A model that can classify the type of a given white blood cell can help medical staff in evaluating patient blood. Even if the model can only narrow it down to two out of four types, it can halve the process for staff evaluating the same images. The current state of the solution of this problem is done by hand and can be significantly speed up or automated with classification algorithms done by a deep neural network. In addition, there has been progress made in the field of ML and blood cell identification but in the context of cancer research. This is still a novel and growing field with little backing it as of now.

3. Why is it important?

This application is important because it is a novel solution for deciding if people have infections. Elevated white blood cells over the normal amount likely means infection.

Quickly counting white blood cell counts for each major type allows for users predict or estimate if a person has an infection. “Overall, the most common cause for a high white blood cell count is response to infection” (SightDX). And depending on what type of blood cells present, predict what kind or type infection. Presenting a way to accurately identify distributions and counts of white blood cells can assist medical staff in catching infections early and being able to treat them effectively, rather than waiting or relying solely on physically visible symptoms.

4. **What data are you using?**

The data we are using comes from a **Kaggle** dataset that will be provided in the appendix. The data is fairly balanced already, with the rough proportions of images in each class being about 25% of the images. The white blood cells are roughly bordered and centered in the images used, but there are still significant amounts of extra particles in the images. Originally there were 410 images where the white blood cells are extracted from.

5. **What does an exploratory data analysis tell you about your data?**

Exploratory Data Analysis of the data doesn’t tell a whole lot to be honest. Our data contains images and isn’t interpretable initially. Without the given labels the data looks the same. We had 9957 images in the training and ~2500 images in the testing data. There are 4 classes in the data: Neutrophils, Eosinophils, Lymphocytes, and Monocytes.

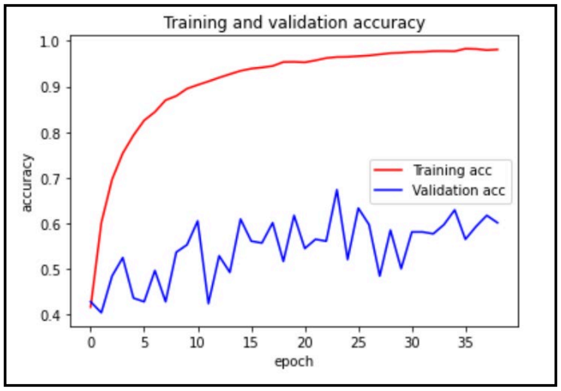
6. **What model did you train? (number of layers, etc) Did you use transfer learning? (if so, from what?) Did you use dropout or regularization?**

6.1. **Transfer Learning:** To speed up the modelling process we used transfer learning. The model we used was VGG16. This is a 16 layer deep convolutional model that can be pre-trained with weights from ImageNet. ImageNet is trained on 14 million images and has ~20,000 categories. We decided to go this route because it can help extract features from the image data. Not only that, but it is already great at it. Though we could’ve spent time working on building a convolutional network, we would rather benefit from using a well defined architecture as well as pre-defined weights from magnitude more training data. The outputs of these models are a .npy bottleneck file. This output is a slice of the

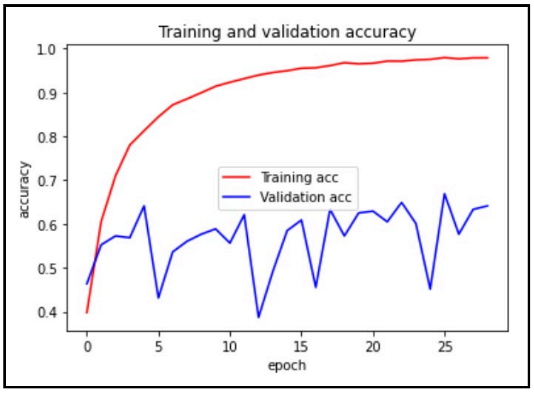
VGG16 model that allows us to use these in our future models. These features are reduced and extracted from the VGG16 model and should allow us to shortcut the convolution phase to use them in our prediction models. This portion of the model is the data preparation.

6.2. Deep Neural Networks

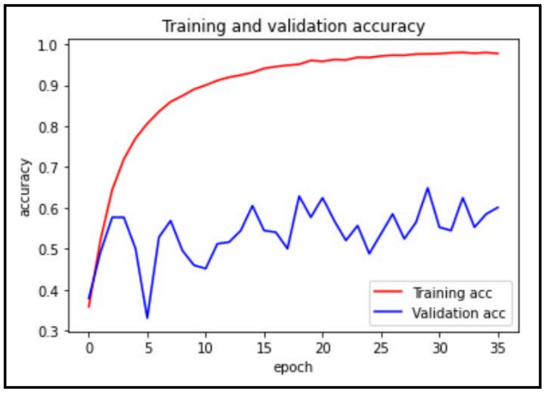
6.2.1. The first base model used contained **two hidden layers**. The model was built in the following order: a layer that flattened the training image data, a layer that had 100 units, each with a leaky-relu alpha of 0.3 (this defines the slope of the negative inputs), a dropout layer of 50% for model flexibility, a dense layer of 50 units, a dropout layer of 30%, and finally a soft-max layer of the number of classes. The leaky-relu is still fast like a relu, but can still use negative values as input. The optimizer was RMSprop. This was used because it uses momentum, so it won't get stuck in local minima and can still learn relatively quickly. 100 epochs were used with a batch size of 32. The model can finish early if the training does not sufficiently drop the loss metric over a few epochs. The learning rate was set to 0.0001 rather than the default of 0.001. This allows the model to learn in a more gradual way rather than jumping in loss and weights/biases. This model attained a testing accuracy of 60%. The plot to the right shows how the validation accuracy increases over time in the model. The jumps are sporadic and there may be some ways we can still optimize input data to attain a more even slope in terms of validation accuracy.



6.2.2. The second model used **four hidden layers**. It used the same optimizer and learning rate as the base model, but used a different unit architecture. It used 256 units of leaky-relu, dropout of 20%, 128 units of leaky-rely, dropout of 20%, 64 units of leaky-relu, dropout of 20%, 32 units of leaky-relu, dropout of 20% and then finally a soft-max layer of each white blood cell type. This model netted an accuracy of 64% and had a seemingly more stable validation accuracy over time.

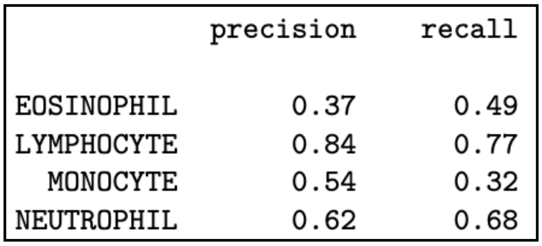


6.2.3. The third model is the same as the second model, but with an additional hidden layer (**five hidden layers**) and reduces the final layer to a dense leaky-relu layer with 16 units. In terms of validation it did slightly worse than the model with four hidden layers, but had a noticeable more stable validation accuracy through the epochs. This is the model that will be used in the model evaluation and usage.



7. What are the results of your model? (accuracy, precision, recall, etc)

7.1. The final model we decided on was the model with five hidden layers that used the transfer learned data as input.

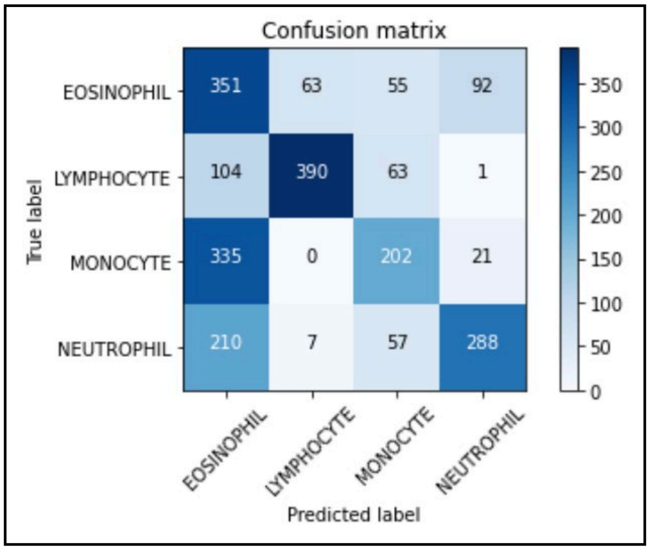


```
             precision    recall

EOSINOPHIL        0.37      0.49
LYMPHOCYTE        0.84      0.77
  MONOCYTE        0.54      0.32
NEUTROPHIL        0.62      0.68
```

7.2. The **precision and recall** values are above. Precision can be interpreted as: what proportion of the positive predictions correct? Recall can be interpreted as: what proportion of that cell type were retrieved by the model. Lymphocytes had the highest precision and recall at 77%. Lymphocytes have the most distinct look, so this makes sense performance wise. Monocytes were especially hard for the model to find, but when they were found, they had a 54% precision score.

7.3. The **confusion matrix** on the right shows how well the model did with respect to each white blood cell type. A perfect model would have a diagonal going from the top left to the bottom right. This model isn't perfect and did fairly well. One major issue was the model would predict Eosinophil the most and would confuse it with the other types. The model did well when only predicting Lymphocyte type cells as well. Looking closely into what defines each blood cell, the size/shape/count of the nucleus can be used to evaluate what kind of blood cell is there. If a submodes was made to pad the data with this information, it would likely do a bit better in terms of predicting what each cell was.



8. Why are your results meaningful?

8.1. The results of the model are meaningful because it can **identify what a blood cell *probably* is**. Though this model isn't classifying blood cells with 100%, it is still helpful. With this number of training and testing data, accuracy and metrics over 25% mean that this model can perform better than just guessing them. Even if the model was classifying blood cells with 100% accuracy, it still isn't likely that the model would be allowed to act in an unsupervised way. The black box nature of neural networks as well as the fact that a wrong decision could cause turmoil means that the final decision would likely be moved to a person that is more qualified and can give understandable reasoning behind their decision.

8.2. Use Case: Even if the model isn't 100% certain, it can be helpful by speeding up the process by which medical staff identify if a cell is of a certain type. A specific use case could be a medial staffer identifying 1000 white blood cells from images. Yes, they could do this by hand, but a model such as ours could act as an aid to staff. There model isn't directly deployable too this case, but it could easily be trained and optimized for it. As detailed in the results, the model can predict what a blood cell is, and it can give 100% prediction or a mixed prediction. A further analysis could look into how the confidence of the model impacts overall accuracy and precision.

9. Future steps for improvement

9.1. Though our models did well on the training and testing data, one drawback to our **data was that it wasn't annotated**. The images used for training were an image of the blood cell. It didn't have a bounding box and it is likely that it lost a considerable amount of performance from the model having to predict blood cell type around a lot of information that didn't even relate to the blood cell

9.2. Another option is to use the data to **create and pad more data**. This dataset in total had 12,500 images. It is likely that even more information can be learned from them, especially if and when they are border boxed and annotated rather than just labeled. If the data were translated into different axis and different sizes (sizes of blood cells), then the model could train on more data and gain a more generalizable

9.3. Translating this model as an aid for medical staff. The translation of this model would make it more explicit in terms of ranking. Neural networks have a tendency to be 100% confident in some scenarios. This happens because the training loss encourages the models to pick rather than be right. This can cause the model to be very wrong in some instances. Instead, the model can try and give a score to do a ranking rather than using a soft-max activation function. Using a linear or pseudo-linear activation function can allow it to assign a score to them. An analysis of how the model currently does in each scenario would also be useful because it can see the differences between real and predictions and how that loss would change as the confidence distribution changes.

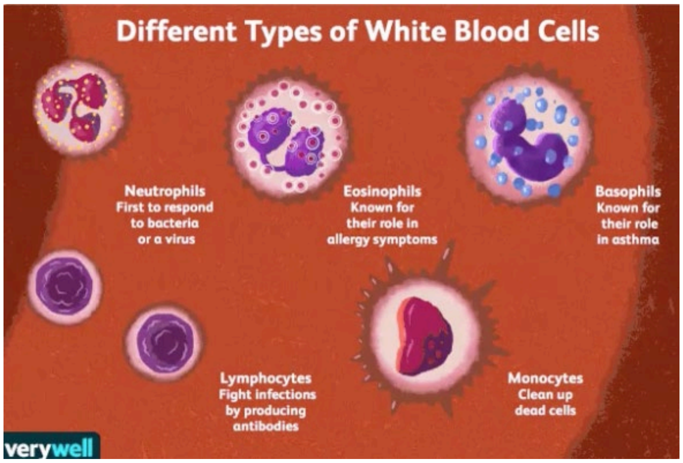
10. Conclusion

This model provides an analytical approach to classifying types of white blood cells. The use cases can include automating or aiding medical staff. This model isn't up to par, but it does have the ability to help staff decide what type of white blood cell exists. Even if it can't definitively say what a cell is, it can help an evaluator in terms of time saved and overall precision and accuracy when it comes to making a decision. It can significantly speed up the process of evaluating a patient. Though the accuracy isn't the smoothest through the epochs, increasing the depth of the model smoothed out these jagged edges and allowed for the model to make more stable predictions. There are many avenues we take to improve the model, but the most influential would be to have a tighter bounding box around the white blood cell and to translate new data into the model. Applications like these have significant ability to improve medical care and aid medical staff in making the best decisions for their patients.

Appendix:

Data: https://www.kaggle.com/datasets/paultimothymooney/blood-cells

Major White Blood Cell types:


Different Types of White Blood Cells
Neutrophils First to respond to bacteria or a virus
Eosinophils Known for their role in allergy symptoms
Basophils Known for their role in asthma
Lymphocytes Fight infections by producing antibodies
Monocytes Clean up dead cells
verywell

References:

"High White Blood Cell Count : Meaning, Causes, Ranges." *Sight Diagnostics - The Future of Blood Diagnostics Is Here*, https://www.sightdx.com/knowledge-center/high-white-blood-cell-count.

"Systems Cell Biology@Yale." *Blood Lab*, http://medcell.med.yale.edu/systems_cell_biology/blood_lab.php.