Diagnosing Leukemia Using AI

Capstone Project Three
Springboard - School of Data

David Clark
September, 2021

1. Introduction

Leukemia is a type of cancer of the blood that often affects young people. In the past, pathologists would diagnose patients by eye after examining blood smear images under the microscope. But, this is time consuming and tedious. Advances in image recognition technology have come a long ways since their inception. Therefore, automated solutions using computers would be of great benefit to the medical community to aid in cancer diagnoses.

The goal of this project is to address the following question: How can the doctor's at the Munich University Hospital automate the diagnosis of patients with leukemia using images from blood smears?

This work is based off the A Single-cell Morphological Dataset of Leukocytes from AML Patients and Non-malignant Controls (AML-Cytomorphology_LMU) dataset taken from the Cancer Imaging Archive. The data consists of 18,365 images. Each image is 11 Gb in size and in a TIFF format. The data was acquired from 200 participants in the study. Half the patients were diagnosed with leukemia, while the other half were not.

2. Approach

2.1 Data Acquisition and Wrangling

The dataset was downloaded from the Cancer Imaging Archive. It consisted of three parts, the raw images, an annotations file, and an abbreviations file. The annotations file listed the images location in the downloaded file structure as well as its labeled morphological type. A subset of images were re-annotated, up to two times. For these images, the re-annotated morphologies were also included. However, during EDA and modeling I did not consider how these re-annotations might affect the results.

2.2 Story Telling and Inferential Statistics

2.2.1 Morphology Classes

I began exploration of the dataset by examining the class counts for the 15 different leukocyte morphology classes. In Figure 1, I show a bar plot with class counts. From the plot, it is obvious that this dataset suffers from a large class imbalance. The first four morphology type dominate (NGS, LYT, MYO, and MON), while the other morphological types have much smaller representation. During preprocessing and model, I will explore ways in which I tried to address the class imbalance.

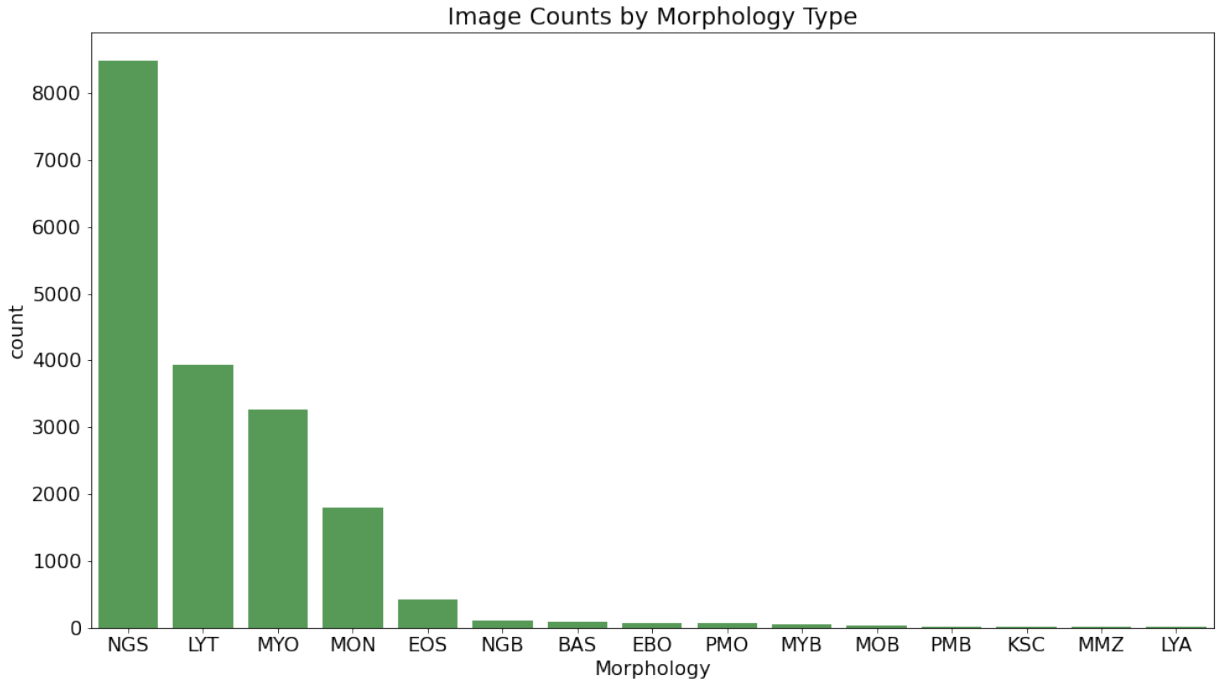

Figure 1. Image counts by leukocyte morphological type.

2.2.2 Mature vs. Immature Leukocytes

This dataset was used in the paper, Human-level recognition of blast cells in acute myeloid leukemia with convolutional neural networks]. In Figure 2 of this paper, the 15 morphological types can be split into two groups, mature leukocytes and immature leukocytes. According to the article, Normal leukocytes, from the Cornel Univesity College of Vetrinary Medicine, EclinPath textbook, mature leukocytes are associated with a healthy body, while immature leukocytes are associated with disease. Often, patients with leukemia release immature leukocytes. Therefore, with this information, I will create an additional column to tag whether a leukocyte morphology is in the mature or immature group.

But, I note that from this information I cannot deduce wether the patient has leukemia or not depending on the maturity of the leukocyte. This additional categorization only serves as an additional exploration of the dataset and relationships between leukocytes.

Using the morphological tree from Figure 2 of the above paper, I have created three lists of morphologies by maturity, `mature`, `immature`, and `smudge`. Figure 2 shows a donut plot with the distributions in maturity type. Almost 70% of the dataset consists of mature leukocyte samples, while the rest are immature leukocytes. Those samples that are labeled smudge are almost negligible.

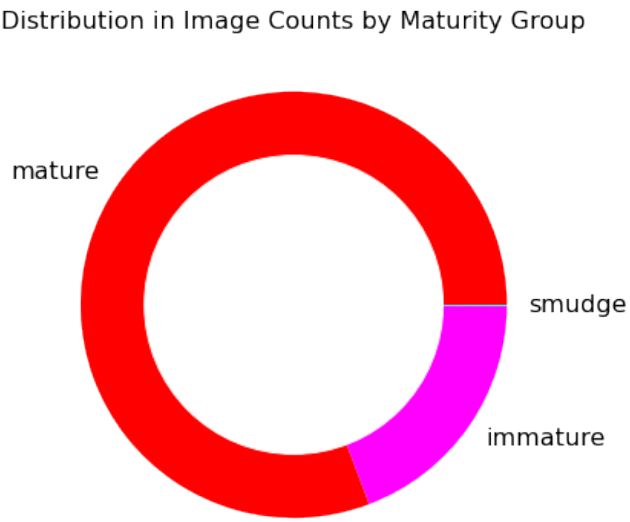

Figure 2: Image counts by leukocyte maturity group.

2.2.3 Image Examination

Next, I examined a representative sample of images from each leukocyte morphology type (Figure 3). Each image is RGB. You will notice that the leukocytes standout as magenta and are roughly centered.

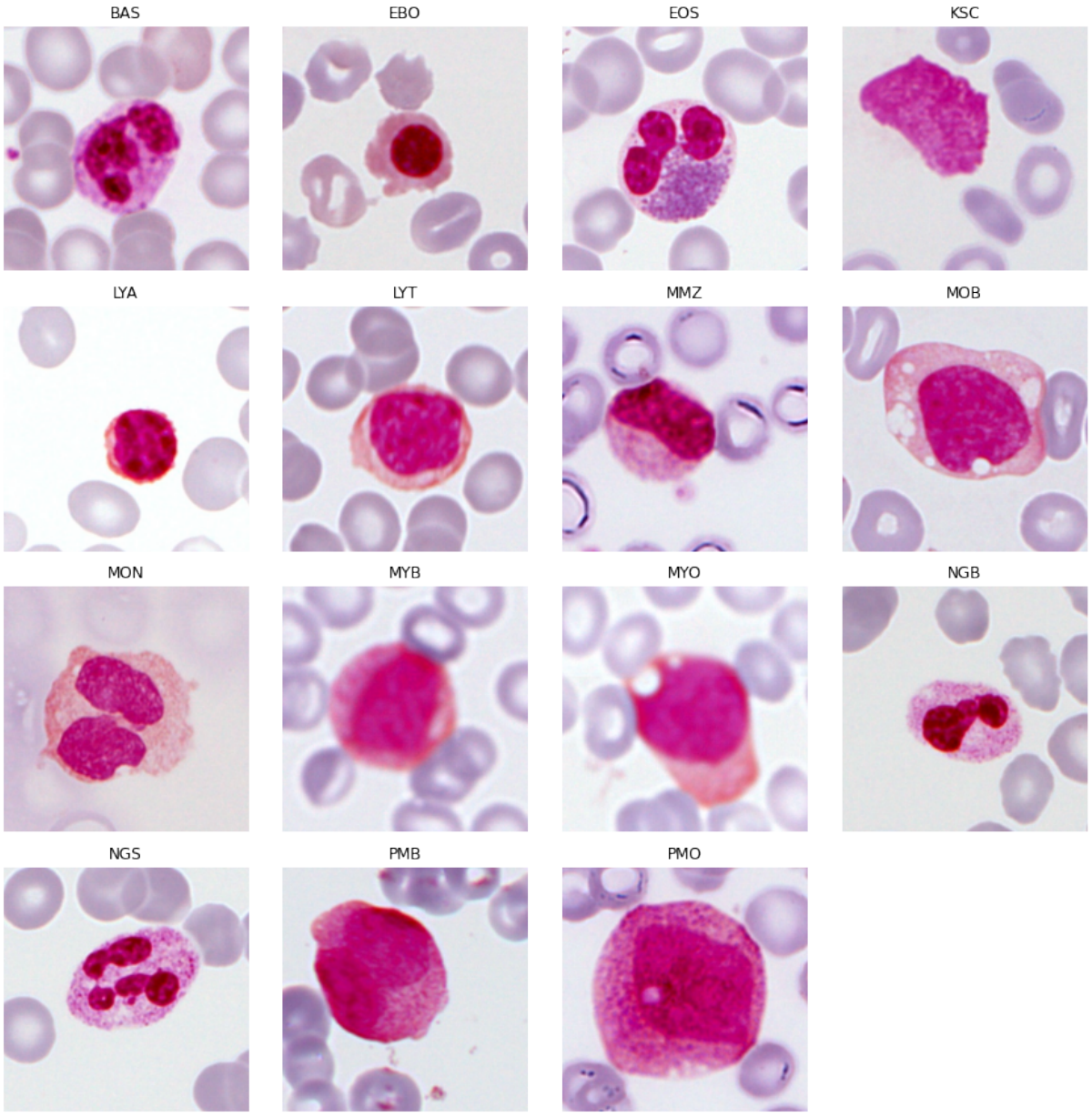

Figure 3: A sample of RGB images for each of the 15 different leukocyte morphology types.

In addition, for each image in the RGB panel, I split it into its separate color channels. This is shown in Figure 4. All leukocytes appear especially bright in blue.

Figure 4: A sample of leukocyte images for each morphological type, broken down by RGB color channel.

Next, I compared image statistics for each color channel and by morphology type, using box plots. The statistics included maximum pixel value, minimum pixel value, mean pixel value, and median pixel value. Figure 5 shows the box plots.

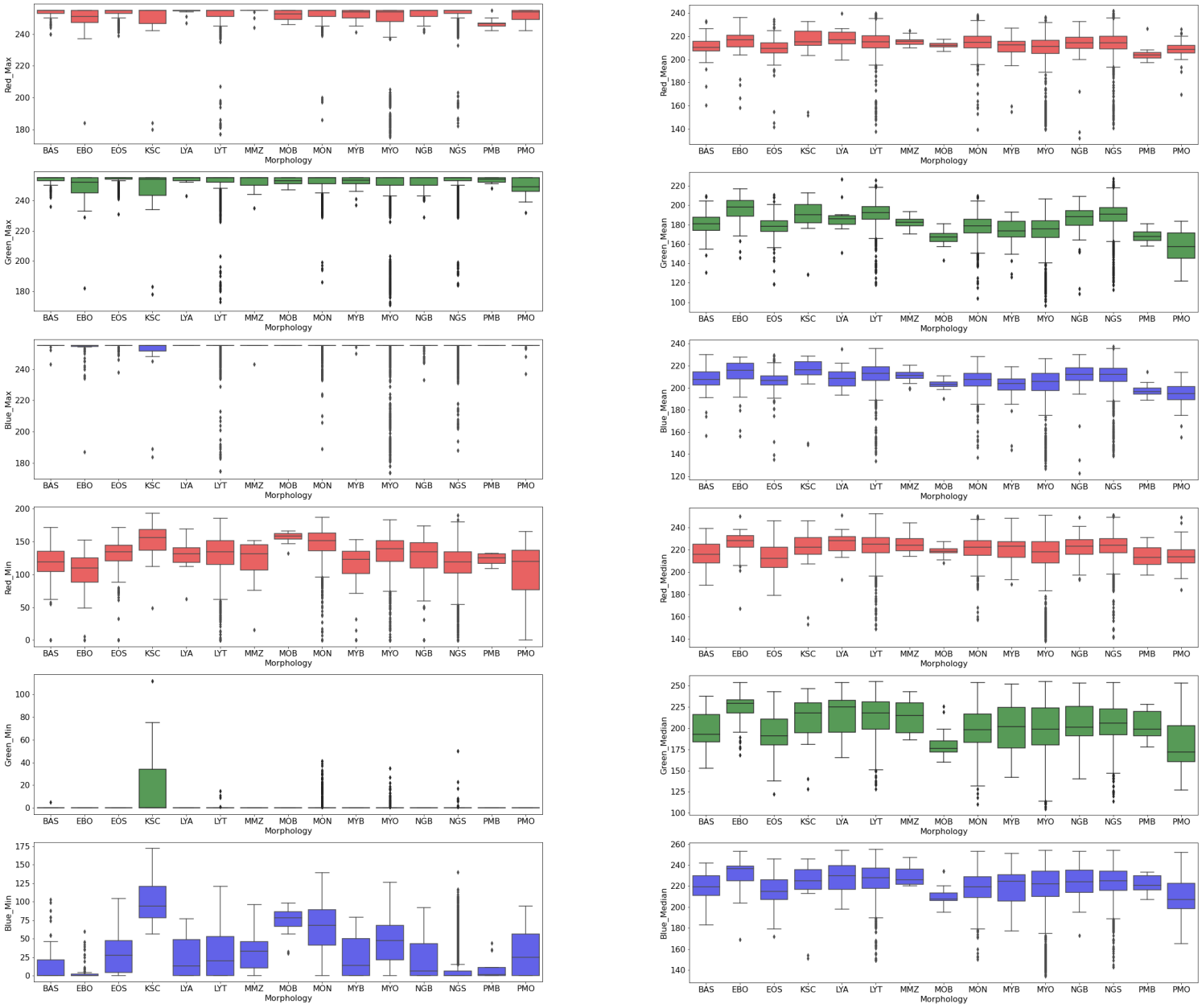

Figure 5: Box plots showing image statistics for each morphology class and by color channel.

2.3 Baseline Modeling

There were several preprocessing steps that needed to be performed before modeling. To begin with, I separated the data set into features and labels, where the features were the images and the labels were the leukocyte morphological types. The modeling algorithms chose required intensive computer resources to process the large

number of 400x400 pixel images. Therefore, I chose to convert the images to grayscale and rescale their sizes by 12%. When rescaling the images to improve training time, information is lost during the transformation. Since I am not a domain expert, I am uncertain if the lost information could be important to identifying leukocyte type. Lastly, I flattened the images into 1-D arrays, so that the feature array would be in the proper format for the initial models.

After these modifications to the images, I split the data set into a training set and a test set, reserving 20% for testing and the rest for training.

As pointed out in section 2.2.1, there is a large class imbalance between the labels. I addressed this imbalance by bootstrapping the training set. First, I bootstrapped the majority class by 10%. Then, I bootstrapped the rest of the classes, to make all classes have a similar representation of samples.

I chose four different models as an initial set of algorithms to try. These includes: logistic regression, random forest classifier, XGBoost, and a support vector classifier. All four are good at classification problems and so are good choices for create a baseline set of models. The following tables show the results on the test set for each of the four baseline models.

Table 1: Logistic Regression Results Summary

morphology	precision	recall	f1-score	support
BAS	0.00	0.00	0.00	16
EBO	0.00	0.00	0.00	16
EOS	0.03	0.02	0.03	85
KSC	0.00	0.00	0.00	3
LYA	0.00	0.00	0.00	2
LYT	0.56	0.58	0.57	787
MMZ	0.00	0.00	0.00	3
MOB	0.00	0.00	0.00	5
MON	0.27	0.27	0.27	358
MYB	0.00	0.00	0.00	8
MYO	0.51	0.55	0.53	653
NGB	0.00	0.00	0.00	22
NGS	0.74	0.73	0.73	1697

Table 1: Logistic Regression Results Summary				
morphology	**precision**	**recall**	**f1-score**	**support**
PMB	0.00	0.00	0.00	4
PMO	0.25	0.14	0.18	14

Table 2: Random Forest Results Summary				
morphology	**precision**	**recall**	**f1-score**	**support**
BAS	0.00	0.00	0.00	16
EBO	1.00	0.06	0.12	16
EOS	0.00	0.00	0.00	85
KSC	0.00	0.00	0.00	3
LYA	0.00	0.00	0.00	2
LYT	0.90	0.75	0.82	787
MMZ	0.00	0.00	0.00	3
MOB	0.00	0.00	0.00	5
MON	0.78	0.27	0.40	358
MYB	0.00	0.00	0.00	8
MYO	0.64	0.89	0.74	653
NGB	0.00	0.00	0.00	22
NGS	0.84	0.98	0.90	1697
PMB	0.00	0.00	0.00	4
PMO	0.00	0.00	0.00	14

Table 3: XGBoost Results Summary				
morphology	**precision**	**recall**	**f1-score**	**support**
BAS	0.00	0.00	0.00	16
EBO	1.00	0.06	0.12	16

Table 3: XGBoost Results Summary				
morphology	**precision**	**recall**	**f1-score**	**support**
EOS	0.83	0.35	0.50	85
KSC	0.00	0.00	0.00	3
LYA	0.00	0.00	0.00	2
LYT	0.89	0.89	0.89	787
MMZ	0.00	0.00	0.00	3
MOB	0.00	0.00	0.00	5
MON	0.67	0.61	0.64	358
MYB	0.00	0.00	0.00	8
MYO	0.74	0.87	0.80	653
NGB	0.00	0.00	0.00	22
NGS	0.94	0.97	0.95	1697
PMB	0.00	0.00	0.00	4
PMO	0.00	0.00	0.00	14

Table 4: Support Vector Classifier Results Summary				
morphology	**precision**	**recall**	**f1-score**	**support**
BAS	0.00	0.00	0.00	16
EBO	1.00	0.06	0.12	16
EOS	0.83	0.35	0.50	85
KSC	0.00	0.00	0.00	3
LYA	0.00	0.00	0.00	2
LYT	0.89	0.89	0.89	787
MMZ	0.00	0.00	0.00	3
MOB	0.00	0.00	0.00	5
MON	0.67	0.61	0.64	358

Table 4: Support Vector Classifier Results Summary				
morphology	**precision**	**recall**	**f1-score**	**support**
MYB	0.00	0.00	0.00	8
MYO	0.74	0.87	0.80	653
NGB	0.00	0.00	0.00	22
NGS	0.94	0.97	0.95	1697
PMB	0.00	0.00	0.00	4
PMO	0.00	0.00	0.00	14

I discovered that all four models strongly over fit and did poorly on the test set. Also, no model was able to give adequate F1 scores for the individual classes. In many cases, the F1 score was zero.

2.4 Extended Modeling

While the above four models discussed in section 2.3 are good at classification problems, convolutional neural networks (CNNs) are ideal for classifying images. I developed a CNN architecture consisting of two convolutional layers, each followed by a max pooling layer, and a softmax output layer. Figure 6, below, shows the architecture in detail.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 48, 48, 8)         80
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 24, 24, 8)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 18, 18, 16)        6288
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 9, 9, 16)          0
_________________________________________________________________
flatten (Flatten)            (None, 1296)              0
_________________________________________________________________
dense (Dense)                (None, 600)               778200
_________________________________________________________________
dense_1 (Dense)              (None, 150)               90150
_________________________________________________________________
dense_2 (Dense)              (None, 38)                5738
_________________________________________________________________
dense_3 (Dense)              (None, 15)                585
=================================================================
Total params: 881,041
Trainable params: 881,041
Non-trainable params: 0
_________________________________________________________________
```

Figure 6: CNN architecture showing each layer and its properties.

All labels were one-hot encoded so that the conformed to the model architecture, specifically the softmax output layer. Also, I did not use the bootstrapped dataset, but did use images that were recalled by 12% and converted to grayscale.

When training a neural network, it is important to use a validation dataset to evaluate the training performance at each epoch of the model run. I created a validation dataset by splitting the training set, reserving 90% for train and 10% for test.

For each model I developed, I used three different methods to evaluate the model performance. Theses were a learning curve, a classification report, a confusion matrix.

- Learning Curve

The the learning curve is a plot of the neural network loss by epoch for both the training set and the validation set. If the model is performing correctly, the training and validation losses should improve quickly and then flatten out, and they should follow closely together. Any deviations from this behavior could indicate a problem with the model or the training data.

- Classification Report

This is a table consisting of each class, or label, used for the model predictions, and various metrics related to that class. In this case, I considered the metrics precision, recall, and F1-Score. The table also includes support, which is the number of samples for each class in the test set.

- Confusion Matrix

The confusion matrix is a heat map comparing predictions with actual values. The predictions are plotted along the x-axis and the actual values along the y-axis. Each row or column value is one of the class labels being predicted. If the model performs perfectly, the predictions and actual values should match up and the diagonal will be filled, with zero values everywhere else.

2.4.1 CNN Model 1a

Using the model architecture described above, I passed the unbalanced class data through model and made predictions.

2.4.1.1 Learning Curve

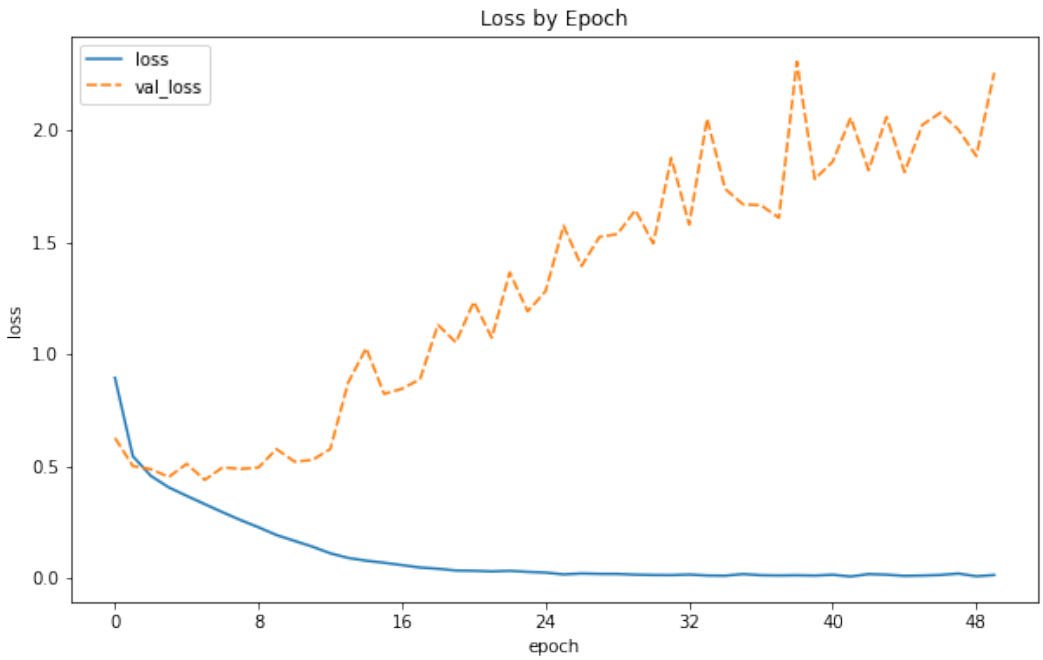

Figure 7: The learning curve for model 1a.

The rapidly diverging validation loss indicates an unrepresentative training set, which means the training data does not provide enough information for the model to learn anything useful.

2.4.1.2 Classification Report

In Table 5, I summarize the results of the model fit. Notice that the top four classes show high F-1 scores, while the F1 score is much worse for the additional classes.

Table 5: Classification Report for Model 1a

morphology	precision	recall	f1-score	support
NGS	0.95	0.95	0.95	1697
LYT	0.88	0.91	0.90	787

Table 5: Classification Report for Model 1a				
morphology	**precision**	**recall**	**f1-score**	**support**
MYO	0.80	0.83	0.81	653
MON	0.59	0.68	0.63	358
EBO	0.45	0.31	0.37	16
EOS	0.40	0.27	0.32	85
PMO	0.12	0.07	0.09	14
BAS	0.00	0.00	0.00	16
KSC	0.00	0.00	0.00	3
LYA	0.00	0.00	0.00	2
MMZ	0.00	0.00	0.00	3
MOB	0.00	0.00	0.00	5
MYB	0.00	0.00	0.00	8
NGB	0.00	0.00	0.00	22
PMB	0.00	0.00	0.00	4

2.4.1.3 Confusion Matrix

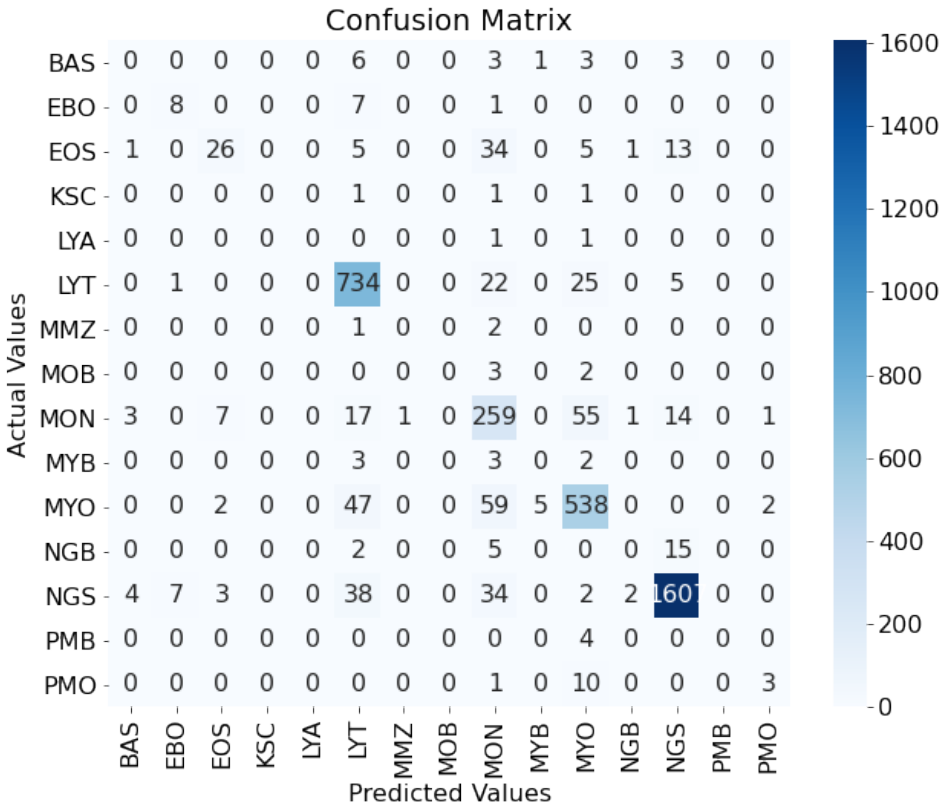

Figure 8: Confusion matrix for model 1a.

The large numbers for the top four classes indicates that the model is doing a good job at fitting the majority classes, but relatively poorly on the smaller classes.

2.4.2 CNN Model 1b

I then tried fitting the same model, but using weighted classes. I gave classes with less representation, more weight.

2.4.2.1 Learning Curve

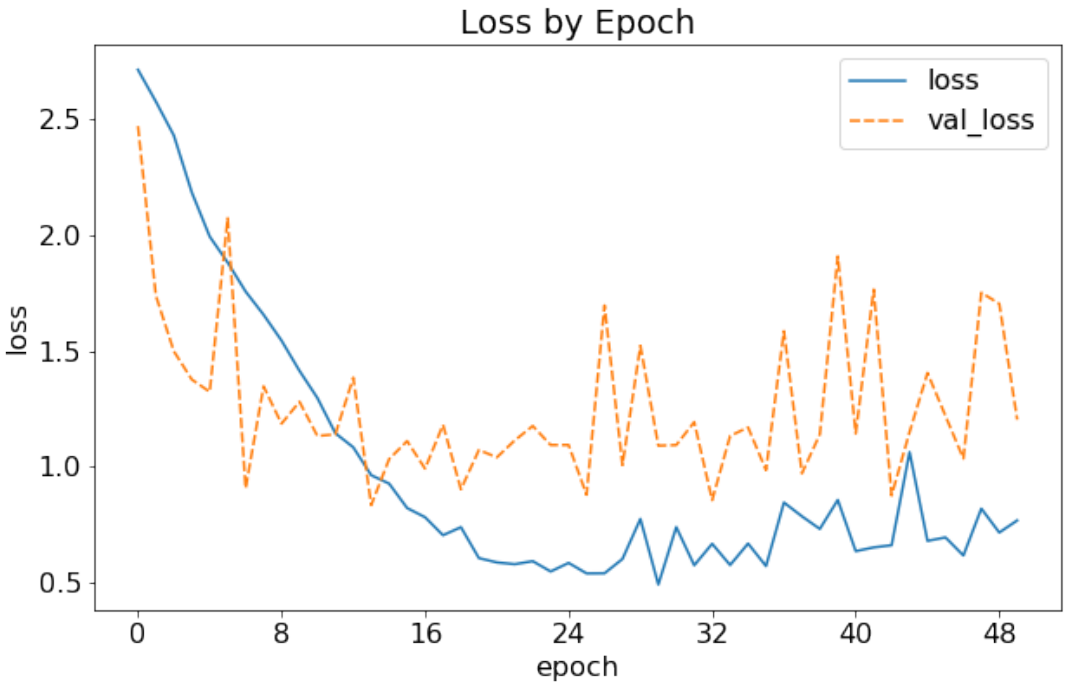

Figure 9: Learning curve for model 1b.

The training performance appears better. But, there is still a displacement between the validation and training losses, and the validation loss is pretty chaotic. Again, insufficient data could be the problem.

2.4.2.2 Classification Report

Table 6 shows the classification report with a summary of this model's results. The model actually fit a lot worse, as shown by the small F1 scores.

Table 6: Classification Report for Model 1b

morphology	precision	recall	f1-score	support
LYT	0.17	0.12	0.14	16
MYB	1.00	0.04	0.08	1697
PMB	0.03	0.29	0.06	14
KSC	0.56	0.03	0.05	358

Table 6: Classification Report for Model 1b				
morphology	**precision**	**recall**	**f1-score**	**support**
MYO	0.02	0.04	0.03	85
EOS	1.00	0.02	0.03	787
NGB	0.02	0.25	0.03	4
NGS	0.00	0.44	0.01	16
LYA	0.01	0.25	0.01	8
MON	0.00	0.00	0.00	3
EBO	0.00	0.50	0.00	2
PMO	0.00	0.00	0.00	3
BAS	0.00	0.00	0.00	5
MMZ	1.00	0.00	0.00	653
MOB	0.00	0.00	0.00	22

2.4.2.3 Confusion Matrix

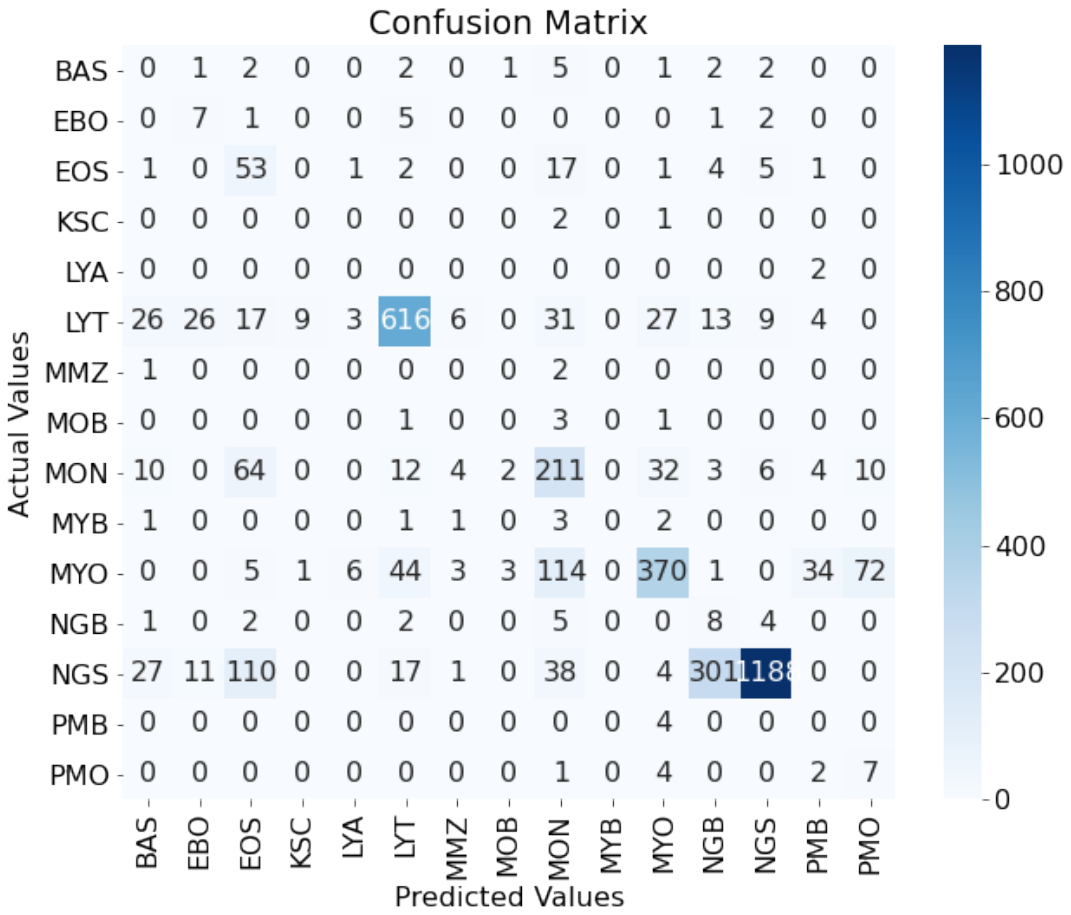

Figure 10: Confusion matrix for model 1b.

The confusion matrix shows the model is doing a messy job of predicting the leukocyte classes. It is getting NGS, MYO, MON, and LYT confused with several other classes. This could be another indication of a lack of information for the model to perform accurate classifications.

2.4.3 CNN Model 2a

In the next couple of models, I decided to explore the model architecture performance by removing the class imbalance issue. Specifically, I converted the problem into a binary classification problem. I selected a dataset only consisting of the top four classes. I then combined the second three classes, creating the binary classes NGC and not NGC.2.4.3.1 Learning Curve

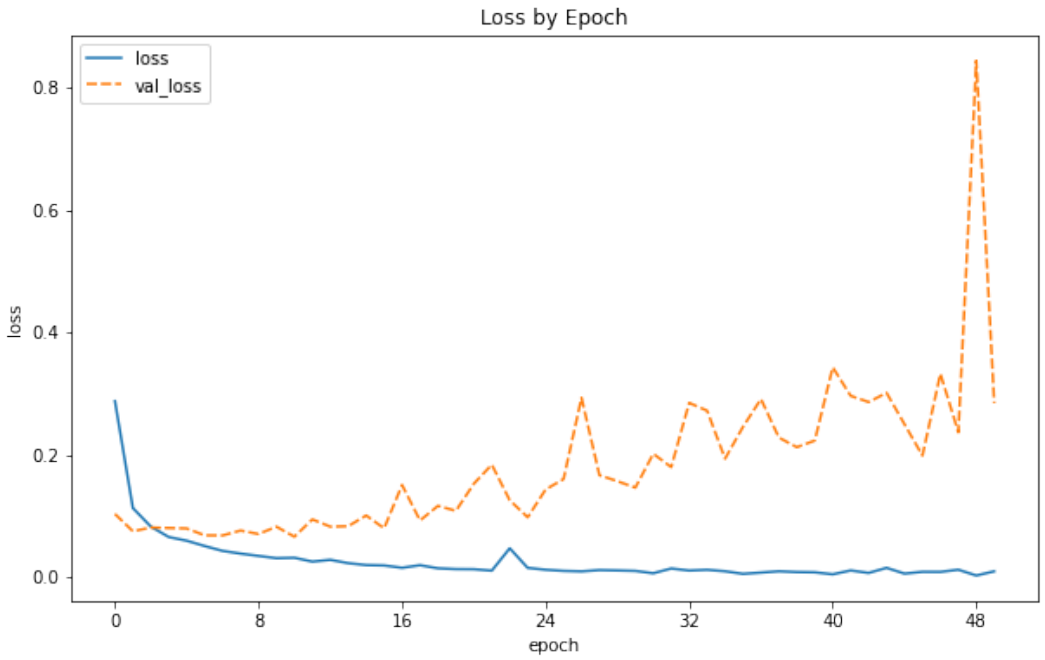

Figure 11: Learning curve for model 2a.

The learning curve is almost identical to model 1a. Again, this indicates the training data contains insufficient information to accurately classify the labels.

2.4.3.2 Classification Report

Table 7 shows the classification report for model 2a. Interesting, the scores across the board are quite good for each class.

Table 7: Classification Report for Model 2a

morphology	precision	recall	f1-score	support
NGC	0.97	0.98	0.97	1697
not NGC	0.98	0.97	0.98	1799

2.4.3.3 Confusion Matrix

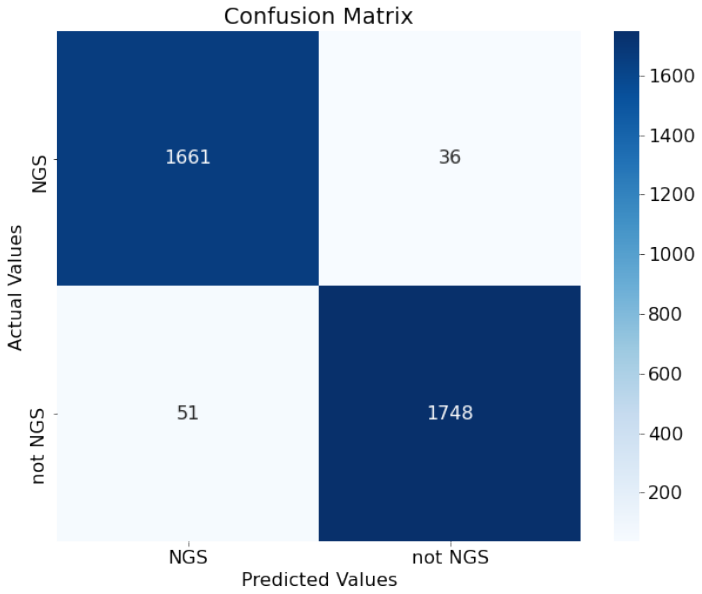

Figure 12: Confusion matrix for model 2a.

The confusion matrix also shows the model is doing quite well at predicting each class.

2.4.4 CNN Model 2b

As I did with model 1a, I now introduced class weights for each class. While the classes are almost equally balanced, I wanted to include this step for completeness, to see if it had any improvements.

2.4.4.1 Learning Curve

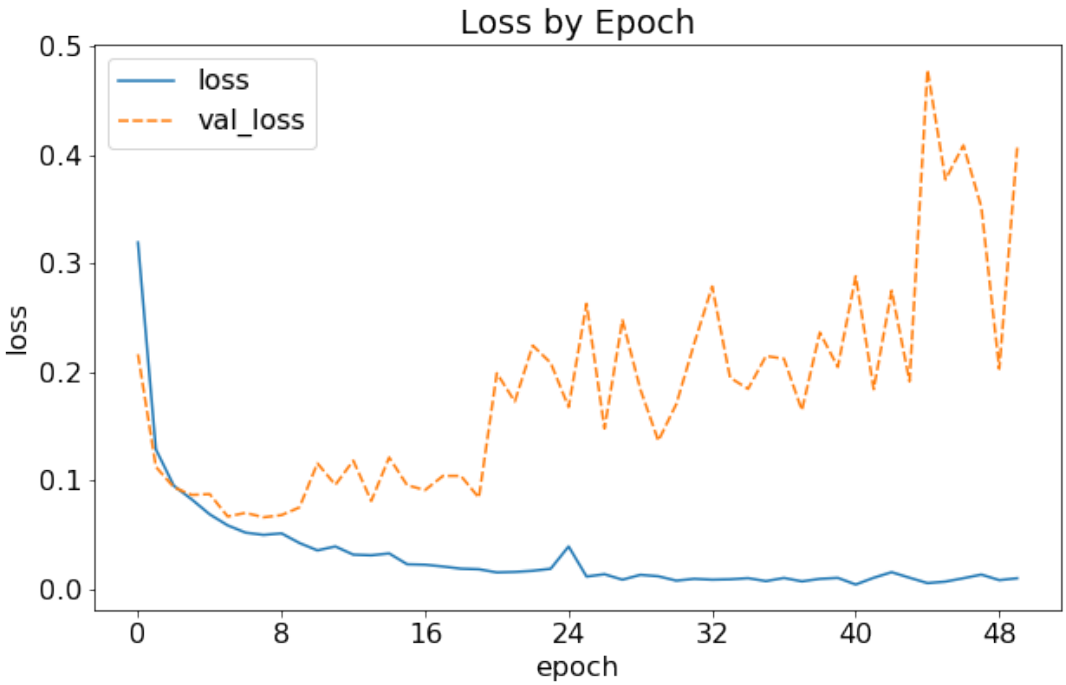

Figure 13: Learning curve for model 2b.

The learning curve appears very similar to model 2a, again showing lack of information in the training set.

2.4.4.2 Classification Report

The model performance is almost identical to model 2a.

Table 8: Classification Report for Model 2b

morphology	precision	recall	f1-score	support
NGC	0.99	0.95	0.97	1697
not NGC	0.95	0.99	0.97	1799

2.4.4.3 Confusion Matrix

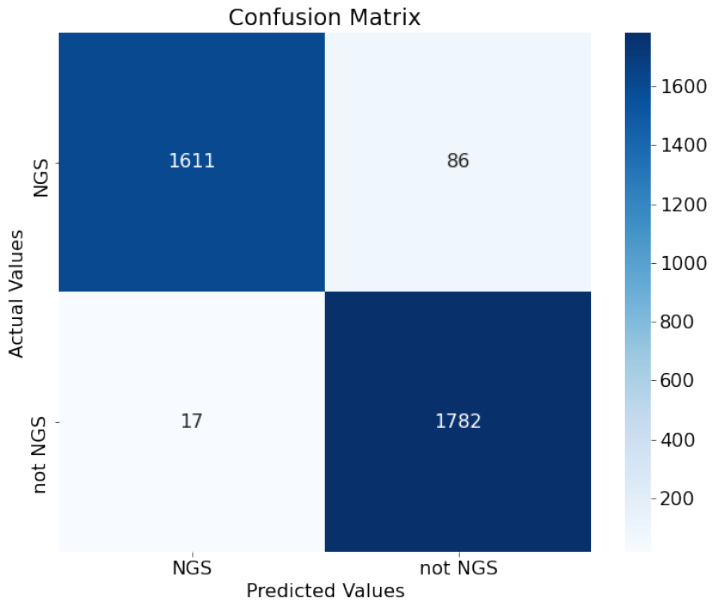

Figure 14: Confusion matrix for model 2b.

Again, almost identical confusion matrices between models 2a and 2b.

3. Findings

Table 9: CNN Model Performance Summary

model	F1 Score	
	macro average	weighted average
model 1a	0.30	0.86
model 1b	0.24	0.73
model 2a	0.98	0.98
model 2b	0.97	0.97

Given this is a multi-classification problem, I chose to use the F1 metric with two different averages to give a concise, numerical evaluation of each model. Table 9 summarizes the F1 metric for each of the four deep learning models that I tried.

The first two models included all 15 leukocyte classes. The unweighted model, model 1a, performed slightly better than the weighted model. You can also see a large difference between the macro averages and the weighted averages, most likely due to the class imbalance issue.

The last two models show very high F1 scores, that are almost identical, and no difference between the macro average and the weighted average. It appears as if the model can do a great job at predicting leukocytes when it is a two class problem. However, as noted in Section 2.4, the learning curves show that the model is under fitting. This discrepancy deserves further investigation, but that is out of scope for the current project.

4. Conclusions and Future Work

4.1 Conclusions

In this project I addressed the following business problem: How can the doctor's at the Munich University Hospital automate the diagnosis of patients with leukemia using images from blood smears? My approach to this problem was to create classification models that could predict leukocyte types from blood smear images. During the exploratory analysis phase, I discovered a large class imbalance between the different leukocyte morphologies.

Initially, I created four baseline models, which included logistic regression, random forest, XGBoost, and a support vector classifier. In all cases, I saw large over fitting and poor performance on the test set. Also, I tried using bootstrapping to counter the class imbalance, but did not see a noticeable improvement.

I extended the modeling to include deep learning and developed a convolutional neural network to address this problem. Specifically, I created four different models: a multi-classification CNN, a weighted multi-classification CNN, a binary classification CNN, and a weighted binary classification CNN. I discovered that the binary classification CNNs performed best, with high average F1 scores (0.98), but that neural network architecture suffered from under fitting as evidenced by the learning curves.

4.2 Future Work

Given more time, here are the next steps I would like to take with this project:

- Explore class imbalance remedies further by using image augmentation.
- Invest in higher performance resources from cloud computing services to train the full dataset without the need for rescaling images.
- Develop a more complex neural network architecture that will not suffer from under fitting.

5. Recommendations for Client

At the current stage of this project, I have developed a model that does a good job at predicting whether a leukocyte morphology is NGC or not. I would recommend that the client use this model as a Phase 1 study to evaluate the time saving performance of classifying leukocyte types using machine learning. Once a more complex model has been developed in the future, the client can incorporate that in phase 2 for the full model rollout.

6. Consulted Resources

Here is a list of the resources I consulted for this current project:

- Data Citation:
  - "Matek, C., Schwarz, S., Marr, C., & Spiekermann, K. (2019). A Single-cell Morphological Dataset of Leukocytes from AML Patients and Non-malignant Controls [Data set]. The Cancer Imaging Archive. https://doi.org/10.7937/tcia.2019.36f5o9ld".
- A Single-cell Morphological Dataset of Leukocytes from AML Patients and Non-malignant Controls (AML-Cytomorphology_LMU).
- Human-level recognition of blast cells in acute myeloid leukemia with convolutional neural networks
- How to use Learning Curves to Diagnose Machine Learning Model Performance