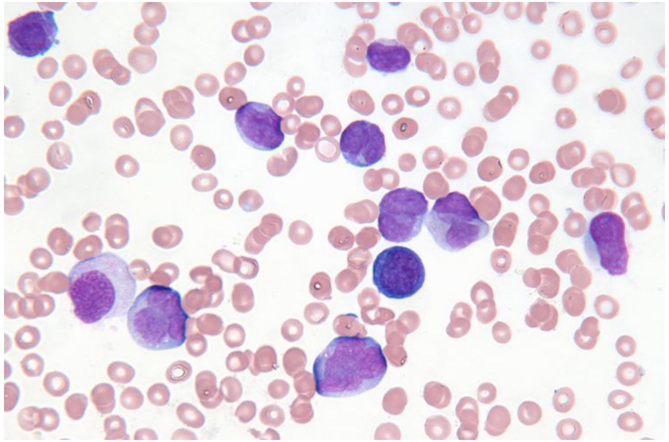
Classifying Leukemia from blood smear images using Machine Learning



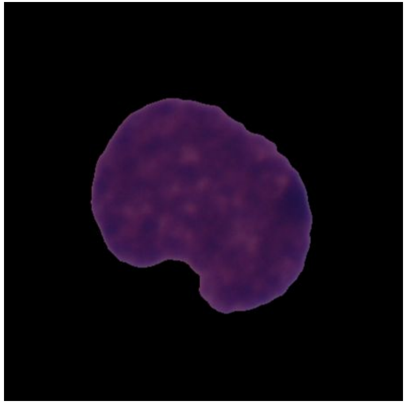
Chris John, Erica Robeen, Simran Gill, Mauricio Perez

MIDS, Fall 2024

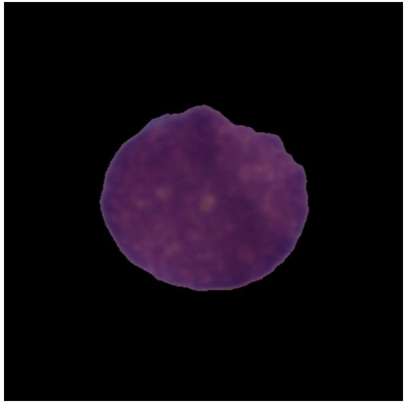
207 – Applied Machine Learning

Image source: https://www.verywellhealth.com/blood-cancer-under-the-microscope-2252200

Motivation

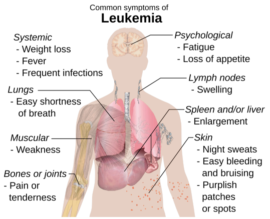


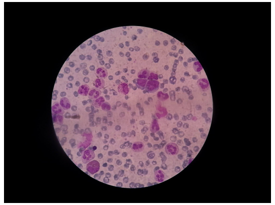
Cancer class

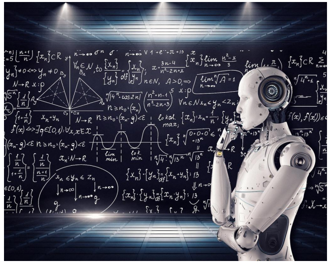


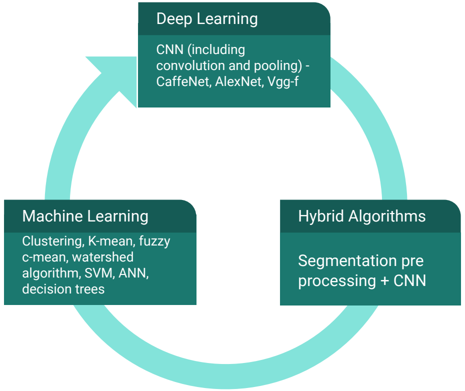
Healthy class

Background



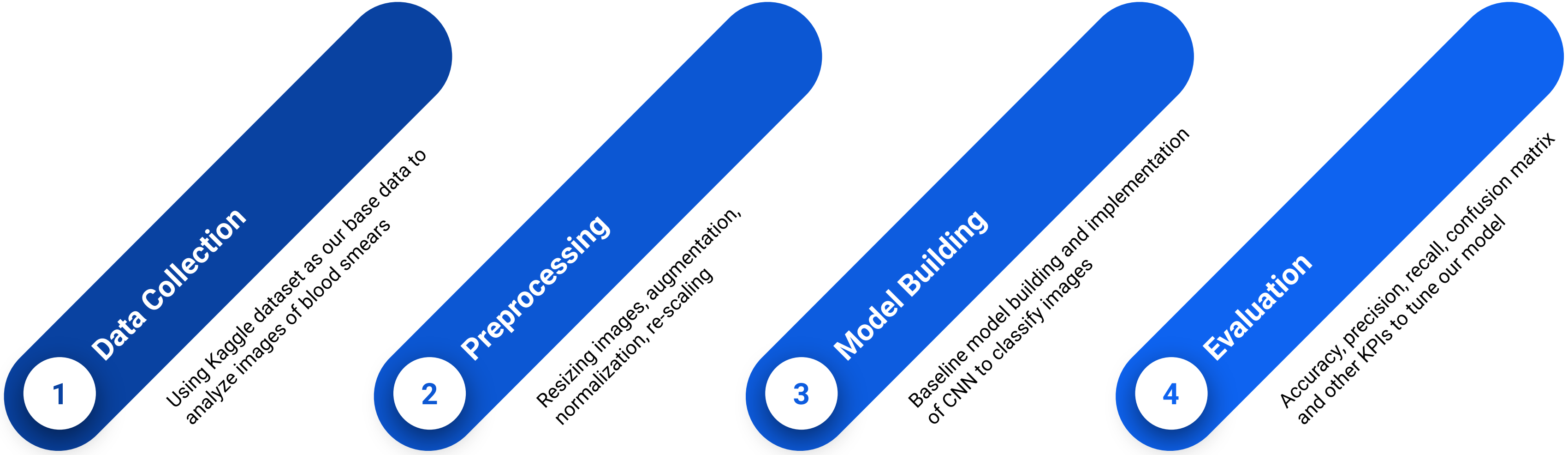





Deep Learning
CNN (including convolution and pooling) - CaffeNet, AlexNet, Vgg-f
Machine Learning
Clustering, K-mean, fuzzy c-mean, watershed algorithm, SVM, ANN, decision trees
Hybrid Algorithms
Segmentation pre processing + CNN

Ghaderzadeh, M., Asadi, F., Hosseini, A., Bashash, D., Abolghasemi, H., & Roshanpour, A. (2021). Machine learning in detection and classification of leukemia using smear blood images: a systematic review. *Scientific Programming*, *2021*(1), 9933481.
Aby, A. E., Salaji, S., Anilkumar, K. K., & Rajan, T. (2024). A review on leukemia detection and classification using Artificial Intelligence-based techniques. *Computers and Electrical Engineering*, *118*, 109446.
Karim, A., Azhari, A., Shahroz, M., Brahim Belhaouri, S., & Mustofa, K. (2022). LDSVM: Leukemia cancer classification using machine learning.
Arivuselvam, B., & Sudha, S. (2022). Leukemia classification using the deep learning method of CNN. *Journal of X-ray science and technology*, *30*(3), 567-585.
Das, P. K., Diya, V. A., Meher, S., Panda, R., & Abraham, A. (2022). A systematic review on recent advancements in deep and machine learning based detection and classification of acute lymphoblastic leukemia. *IEEE access*, *10*, 81741-81763
Talaat, F. M., & Gamel, S. A. (2024). Machine learning in detection and classification of leukemia using C-NMC_Leukemia. *Multimedia Tools and Applications*, *83*(3), 8063-8076.
Daqqa, K. A. A., Maghari, A. Y., & Al Sarraj, W. F. (2017, May). Prediction and diagnosis of leukemia using classification algorithms. In *2017 8th international conference on information technology (ICIT)* (pp. 638-643). IEEE.

Project Objective and Approach Overview



1 **Data Collection**

Using Kaggle dataset as our base data to analyze images of blood smears

2 **Preprocessing**

Resizing images, augmentation, normalization, re-scaling

3 **Model Building**

Baseline model building and implementation of CNN to classify images

4 **Evaluation**

Accuracy, precision, recall, confusion matrix and other KPIs to tune our model

The Data

Dataset Overview

The dataset includes 12,528 images with 67.8% cancerous cells and 32.2% healthy cells.

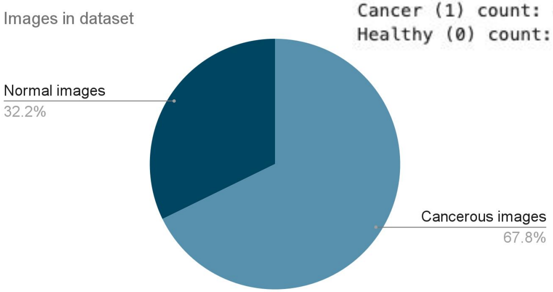
Source and Composition

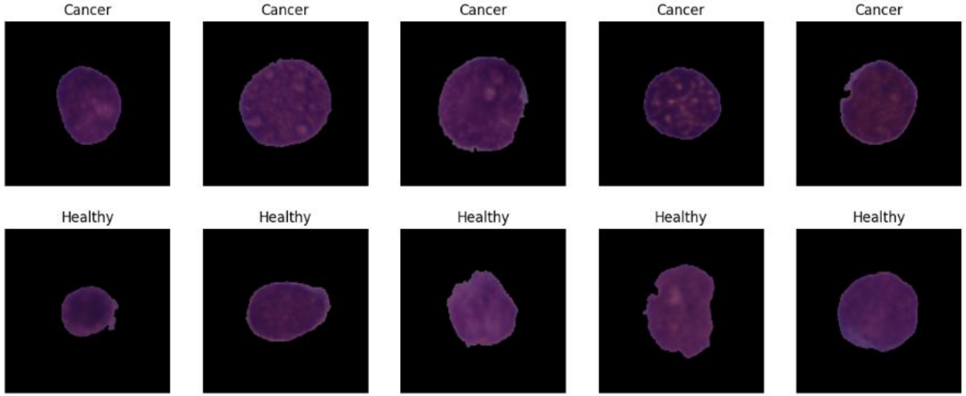
Data sourced from Kaggle with labeled images of normal and leukemia cells for training and validation.

Patient Diversity

Covers 118 unique patients, ensuring a variety of samples for robust modeling.

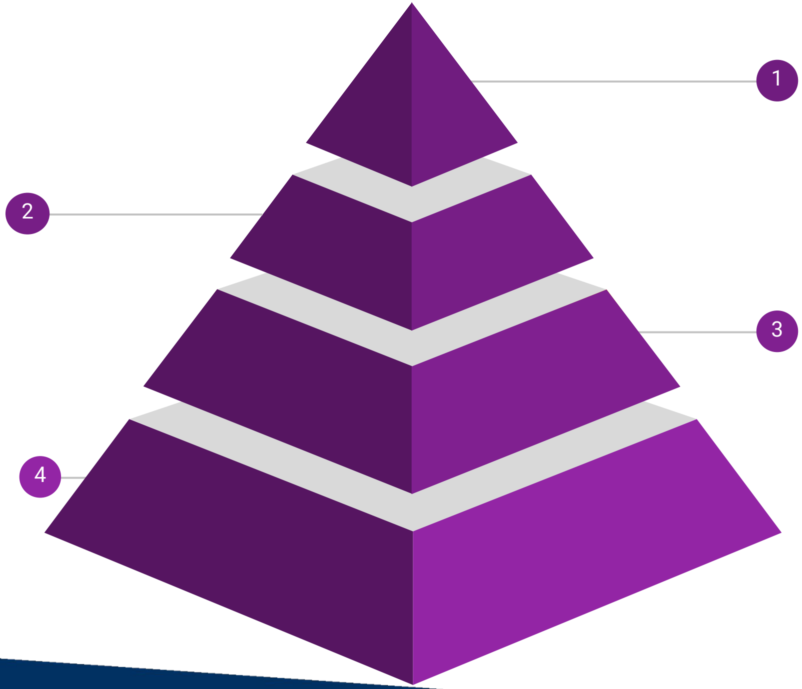
```
Cancer (1) count: 8491
Healthy (0) count: 4037
```





Source: https://www.kaggle.com/datasets/andrewmvd/leukemia-classification?resource=download

Pre-Processing and Augmentation



Re-Sizing

Re-sized images from 450x450 to 128x128 to help reduce computational complexity without losing relevant details in the images.

Normalization/Re-Scaling

Re-scaled pixel values to the [0, 1] range by dividing by 255.0, preparing the data for CNN's.

Data Splitting

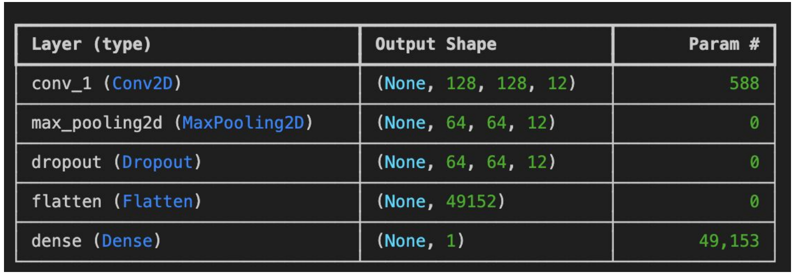
Combined the datasets into one dataset, shuffled the images, and then divided the dataset into 60% training, 20% validation, and 20% testing to ensure balanced evaluation.

Augmentation

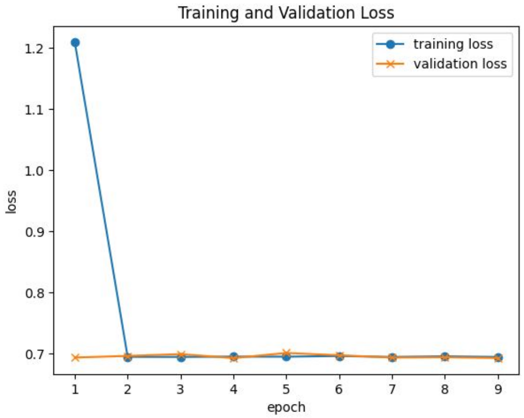
Applied horizontal flips, brightness adjustments, and random rotations to training images to improve model generalization.

Berkeley
UNIVERSITY OF CALIFORNIA

Our First Model: Convolutional Neural Network (CNN) Classifier



Layer (type)	Output Shape	Param #
conv_1 (Conv2D)	(None, 128, 128, 12)	588
max_pooling2d (MaxPooling2D)	(None, 64, 64, 12)	0
dropout (Dropout)	(None, 64, 64, 12)	0
flatten (Flatten)	(None, 49152)	0
dense (Dense)	(None, 1)	49,153



Best epoch:

Baseline Model
`Val Loss: 0.6577`
`Val Accuracy: 0.6780`

Baseline Parameters:

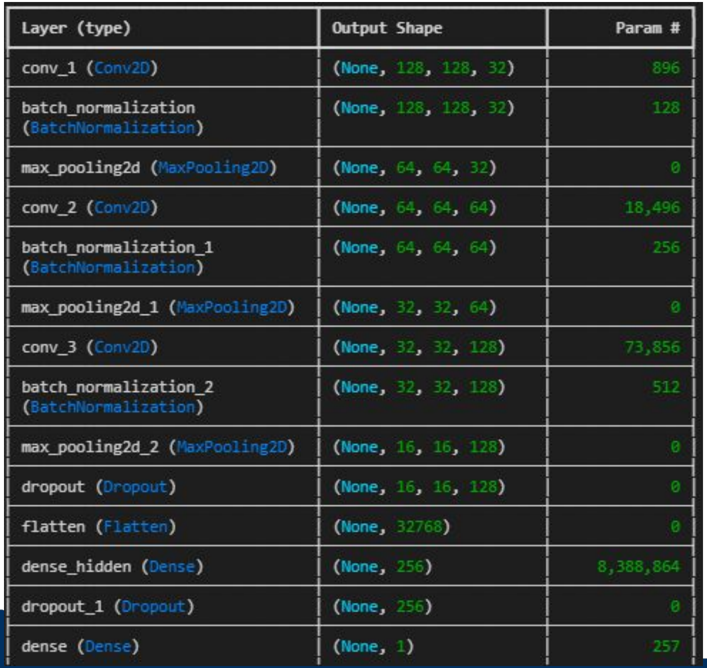
- **1 convolutional block** with 12 filters, 4x4 kernel size, followed by a max-pooling layer with pool size 2x2
- **1 dense layer (represented as classification layer)** with sigmoid activation
- **1 dropout layer** with a rate of .3 to help prevent overfitting
- **Started with a learning rate** of 0.1

Berkeley
UNIVERSITY OF CALIFORNIA

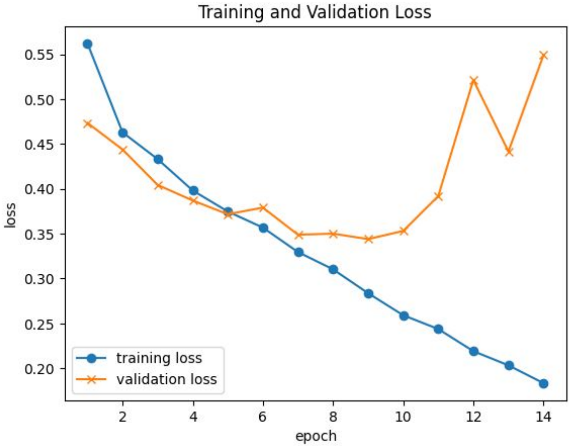
Experimentation

Things we experimented with	**Why**
Number of Convolutional Layers	From 1 to 3 to add complexity to the model and see the effect of deeper networks
Use of batch normalization	Improving training stability and model generalization
Data Augmentation as part of the model	To prevent overfitting and increase model generalization
Dense Hidden layer sizes	Finding optimal sizes for fully connected layers - larger dense layers could potentially find more information, but at the risk of increasing model complexity and overfitting
Dropout layers	To prevent overfitting and try different dropping units during trainings
Max Pooling Layers	To reduce spatial dimensions and control feature extraction - adding and removing pooling layers to reduce dimensionality without losing critical features
Learning rate	Balancing rate of convergence with stability and length of time required to run the training - experimented with high and lower learning rates (and ultimately stuck with lower earning rates given the 3 convolutional layers)

Final Model:



Layer (type)	Output Shape	Param #
conv_1 (Conv2D)	(None, 128, 128, 32)	896
batch_normalization (BatchNormalization)	(None, 128, 128, 32)	128
max_pooling2d (MaxPooling2D)	(None, 64, 64, 32)	0
conv_2 (Conv2D)	(None, 64, 64, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 64, 64, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 32, 32, 64)	0
conv_3 (Conv2D)	(None, 32, 32, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 32, 32, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 128)	0
dropout (Dropout)	(None, 16, 16, 128)	0
flatten (Flatten)	(None, 32768)	0
dense_hidden (Dense)	(None, 256)	8,388,864
dropout_1 (Dropout)	(None, 256)	0
dense (Dense)	(None, 1)	257



Best epoch:

“Model complex”

`Val Loss: 0.3500`
`Val Accuracy: 0.8683`

`vs.` Baseline Model

`post-fine tuning:`
`Val Loss: 0.6577`
`Val Accuracy: 0.6780`

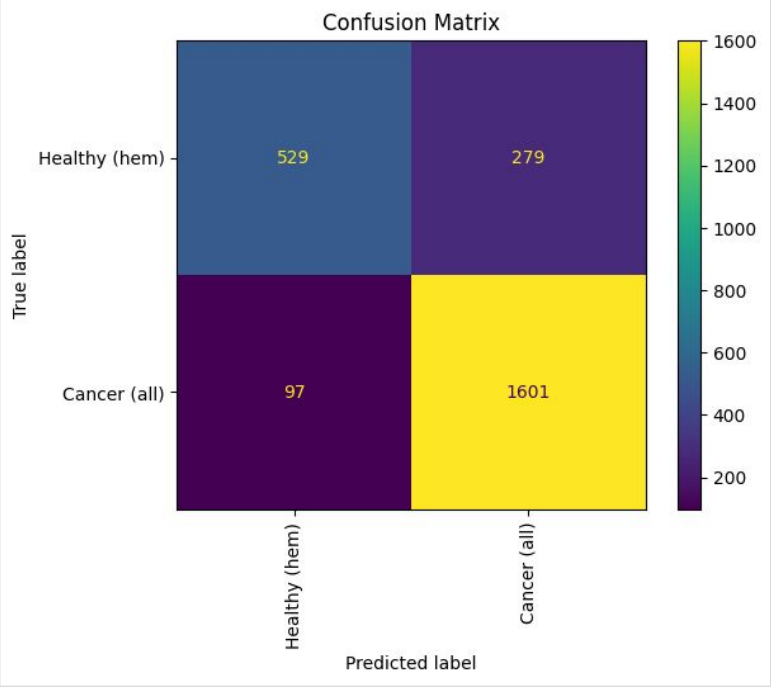
Similar to Homework 10 CCN with key differences:

- **3 convolutional blocks** with progressively increasing filters (32, 64, 128) and smaller kernel size of 3x3 (vs. 4x4 hmwk) with each layer followed by max-pooling for feature reduction
- **Multiple dropout layers** (rate .3 and .4) placed after dense layers to he prevent overfitting
- **Intermediate dense layer** of 256 units (vs. hmwk had one dense layer present in the final classification layer)
- **Lowered the learning rate** to 0.0001 (vs .1 hmwk) for more gradual convergence
- **To better balance classes to reduce bias:** Added batch normalization after each convolutional layer

Evaluating Final Model

Accuracy Scores		
	Baseline Model	**Final Model**
Training	67.77%	90.57%
Validation	67.80%	86.83%
Test	-	84.32%

- Model is generalizing well to unseen data
- **Precision:** 85.16%
- **Recall:** 94.29%



Limitations and Conclusion

- **Increasing validation loss** will accuracy for validation remains high, may be overfitting to training data
- **The images only comes from 118 people and uses a single lab** (69 cancer patient as 49 normal patients) – future work check if model is just able to predict if the patients rather than cancer vs normal cells.
- **The results are limited to 2 classifications**: cancerous and healthy; however, there are at least 9 distinct types of Leukemia– performance may be optimized by increasing number of classifications
- **Images size reduction** from 450x450 to 128x128 – possible loss of extra information

NeurIPS checklist (½)

For all Authors

Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? - NA

1. Have you read the ethics review guidelines and ensured that your paper conforms to them? YES
2. Did you discuss any potential negative societal impacts of your work? YES
3. Did you describe the limitations of your work? YES

Theoretical Results - NA

4. Did you state the full set of assumptions of all theoretical results?
5. Did you include complete proofs of all theoretical results?

If we ran experiments… - YES

6. Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? YES
7. Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? YES
8. Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? YES
9. Did you include the amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? YES

NeurIPS checklist (2/2)

If you are using existing assets (e.g., code, data, models) or curating/releasing new assets… NA

1. If your work uses existing assets, did you cite the creators? NA
2. Did you mention the license of the assets? NA
3. Did you include any new assets either in the supplemental material or as a URL? NA
4. Did you discuss whether and how consent was obtained from people whose data you're using/curating? NA
5. Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? NA

If you used crowdsourcing or conducted research with human subjects… NA

1. Did you include the full text of instructions given to participants and screenshots, if applicable? NA
2. Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? NA
3. Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? NA

Contributions

Team member	Contributions
Erica Robeen	•Built an initial 3 layer CNN model that team iterated on and improved (from loading data, EDA, to the fine-tuned model) •Contributed to slides
Simran Gill	•Developed load_data function - allows of reading through files to load images into arrays •Exploratory Data Analysis – displaying images by Cancer or Normal cells for comparison •Pre-processing – Contributions to pre-processing data, rescale, data-split •Augmentation – testing different augmentation for train set to help with model improvements •Fine tuning model – tried different models to see if accuracy increases without overfitting •Model evaluation – precision, recall, and confusion matrix
Chris John	•Helped identify dataset and topic of evaluation •Contributed to Exploratory Data Analysis – displaying images by Cancer or Normal cells for comparison •Attempted different approach to Pre-processing – pre-processing data, rescale, data-split •Fine tuning model – tried different models to see if accuracy increases without overfitting •Contributed to slides
Mauricio Perez	•Edited some functions (load_data, augment_data and splitting data in training and validation) •Created a few visuals for images and data •Created a few experiments based on the baseline model and fine-tuned model to test different hyperparameters •Made a baseline deck for sharing results and methodology •Did background research on models and other methodologies previous people have done

Resources

- https://onlinelibrary.wiley.com/doi/pdf/10.1155/2021/9933481
- https://www.sciencedirect.com/science/article/pii/S0045790624003732?casa_token=L1Aa89I_8T0AAAAA:e1mdxOyPsSuDhWU5UKZ2rMfeMVdieu8_iS3ZwuSkIgpgxFkAAySTWUkzSXgjUYiesrLvq2Zn5w
- https://manara.qnl.qa/articles/journal_contribution/LDSVM_Leukemia_Cancer_Classification_Using_Machine_Learning/26889151/1
- https://content.iospress.com/articles/journal-of-x-ray-science-and-technology/xst211055
- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9848788
- https://link.springer.com/content/pdf/10.1007/s11042-023-15923-8.pdf
- https://ieeexplore.ieee.org/document/8079919
- https://en.wikipedia.org/wiki/Leukemia