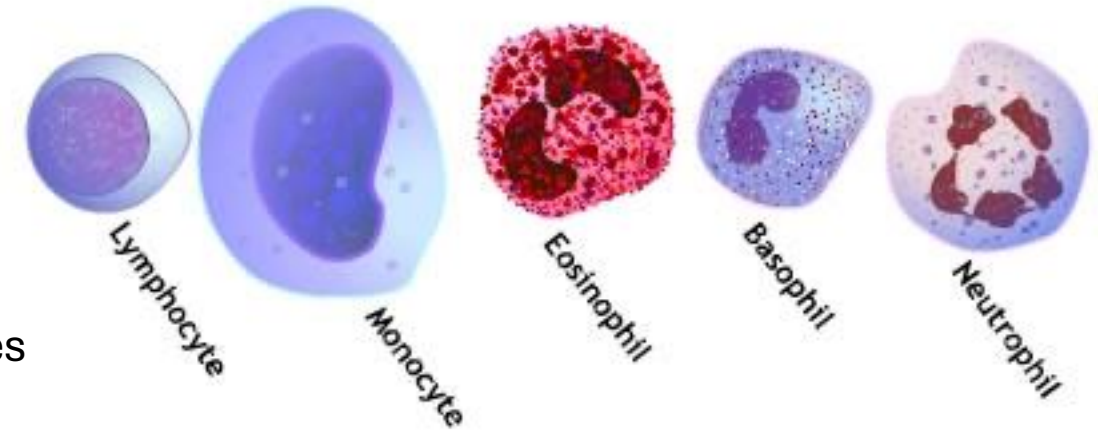
Classification of Leukocytes with Deep Learning

Christian Geils

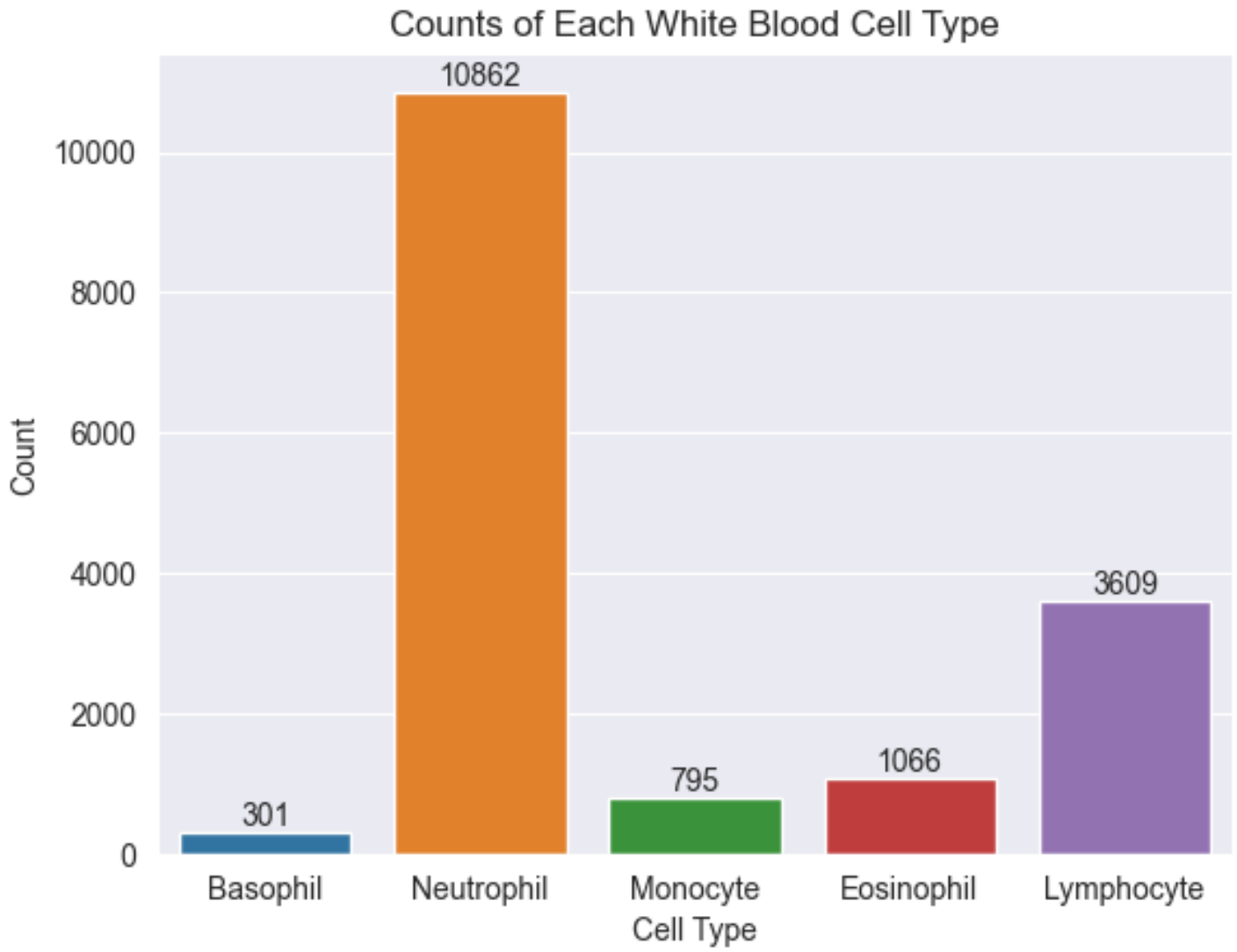
Dataset

Lymphocyte | Monocyte | Eosinophil | Basophil | Neutrophil

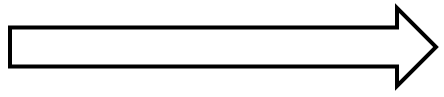


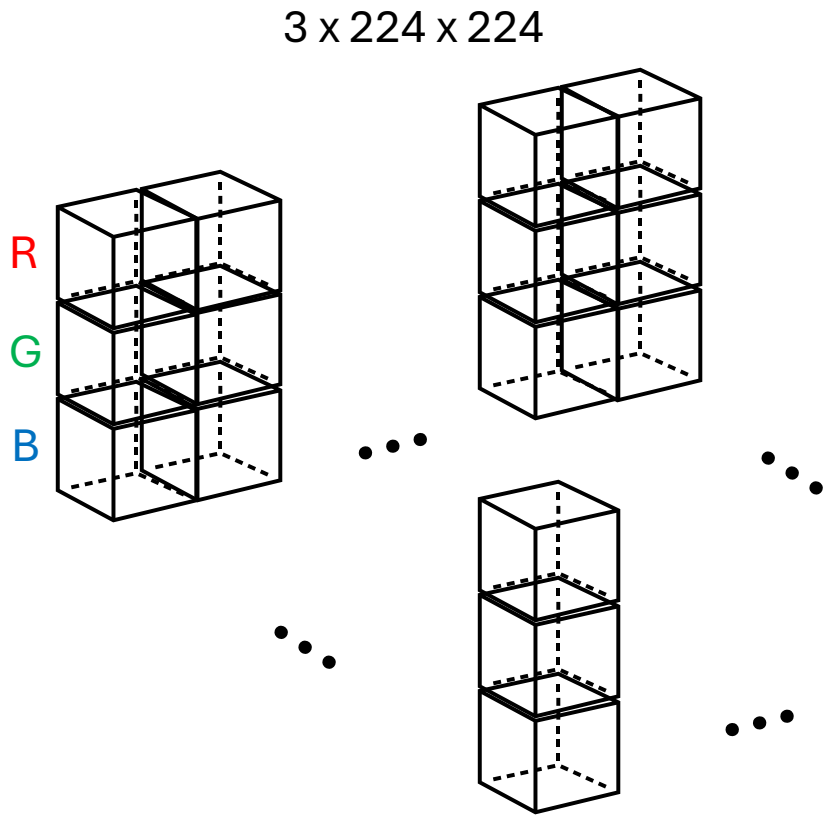
- 16633 expert-labeled images
- Raabin data

Dataset – Unbalanced

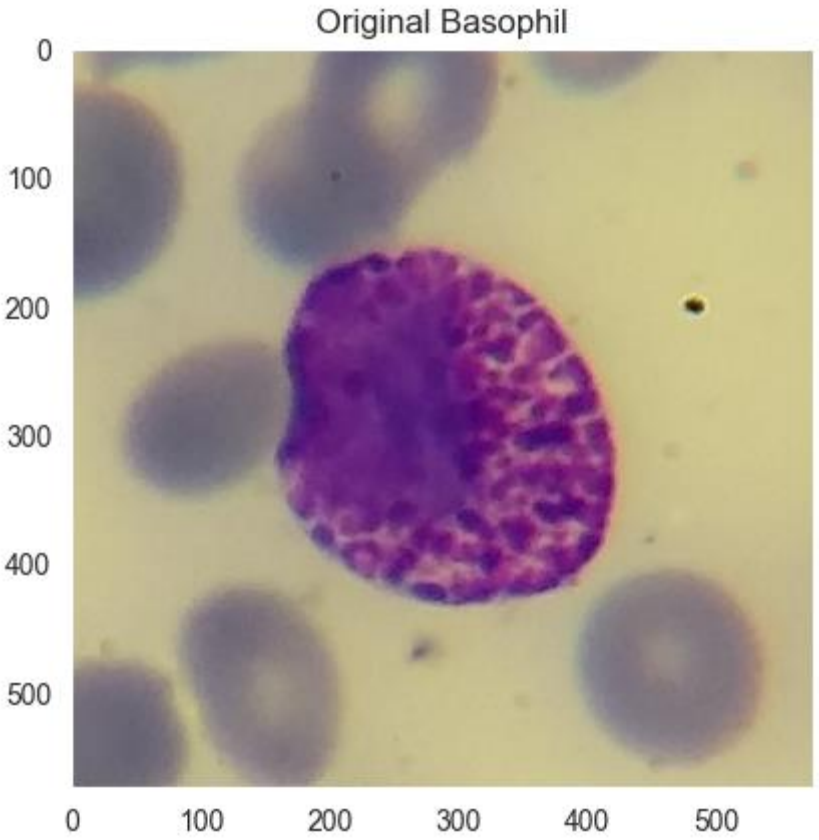


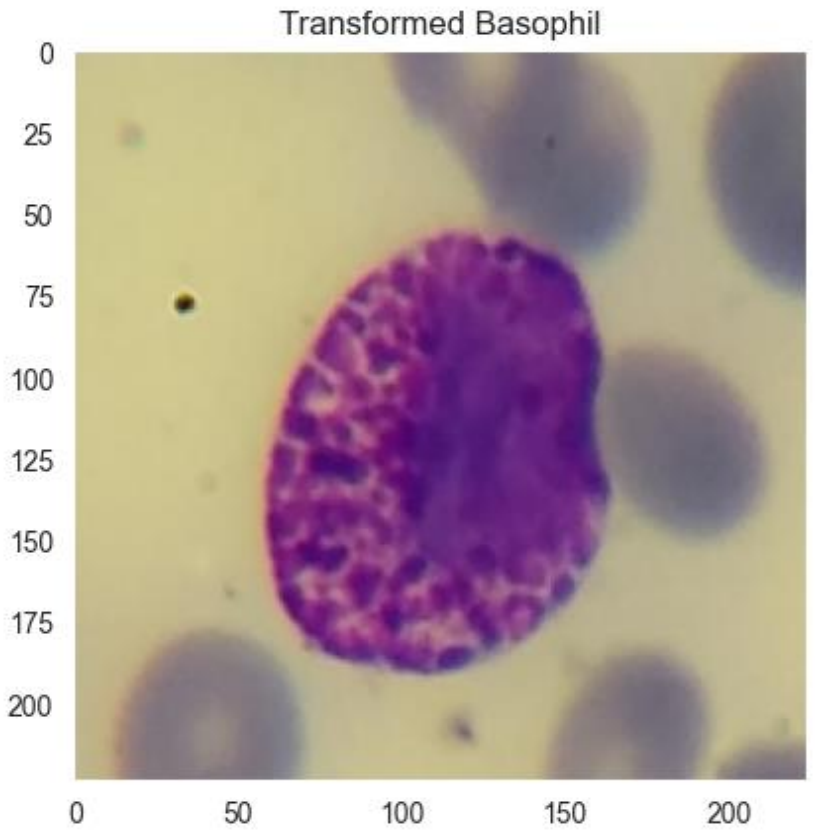
Data Representation





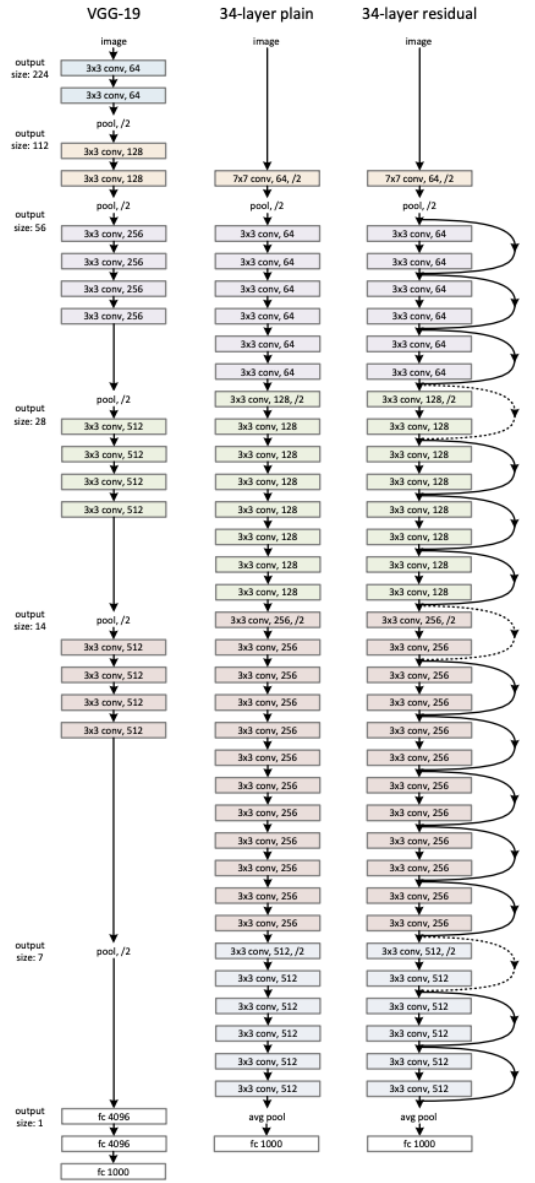
Data Transformation(s)





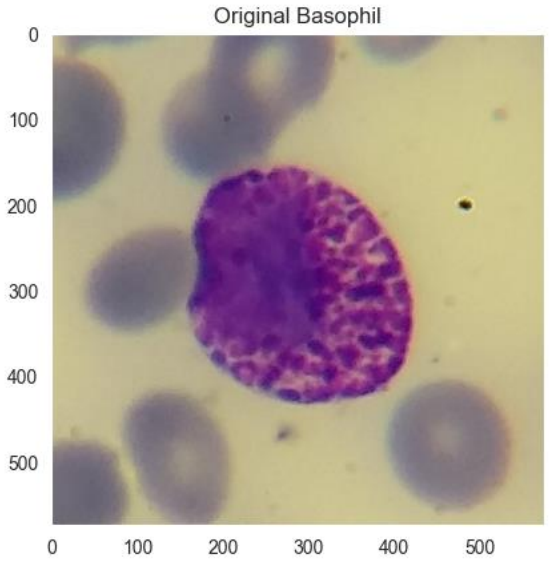
Model Choice: ResNet Family

- Deep CNNs
  - 2D convolutions
  - Skip connections
  - Max 7 avg. pooling
- 3 models: 18, 34, 50
- Pre-training + transfer learning/fine-tuning
- Possible: much smaller CNN w/ grayscale input; efficiency



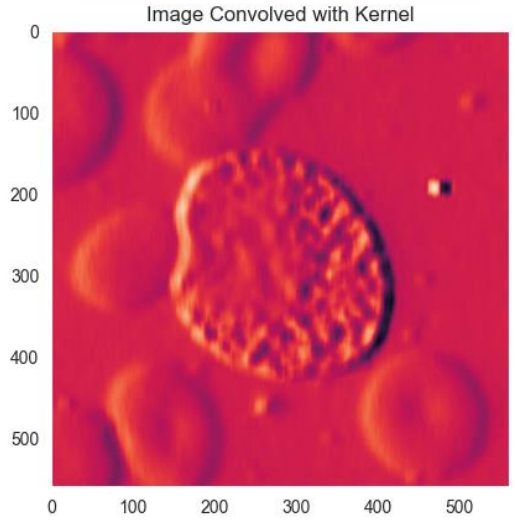
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Model Architecture: Convolution



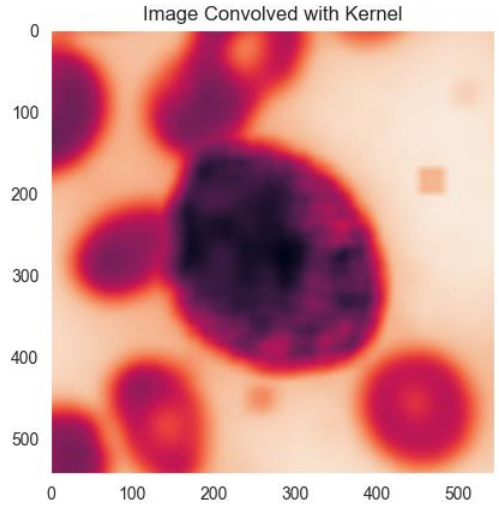
Edge kernel

[[100, ..., -100],
...
[100, ..., -100]]



Box blur kernel

[[1, ..., 1],
...
[1, ..., 1]]



Model Architecture: Residual Block

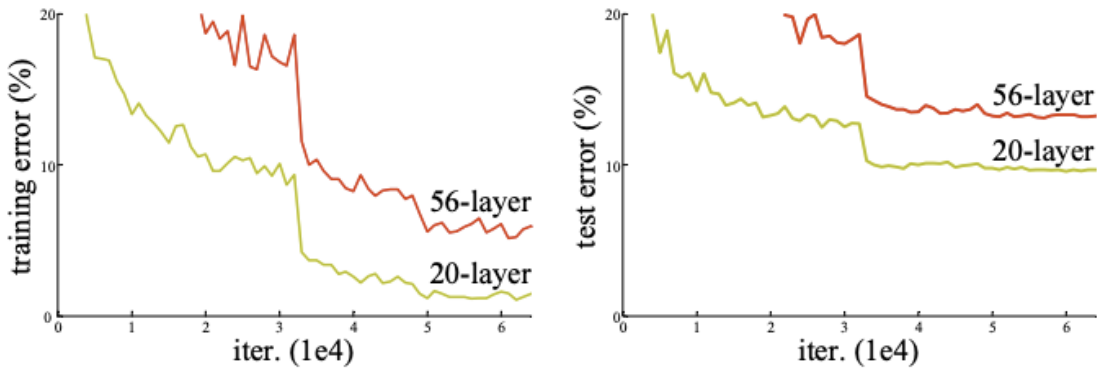


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

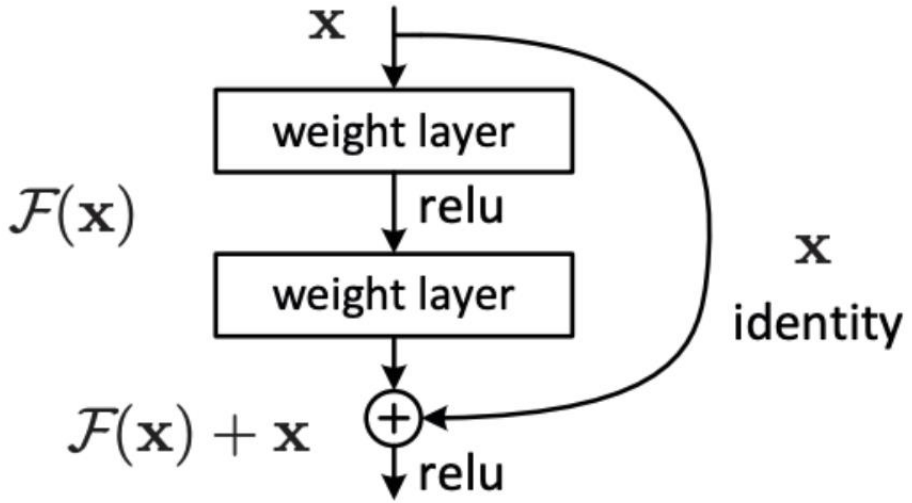


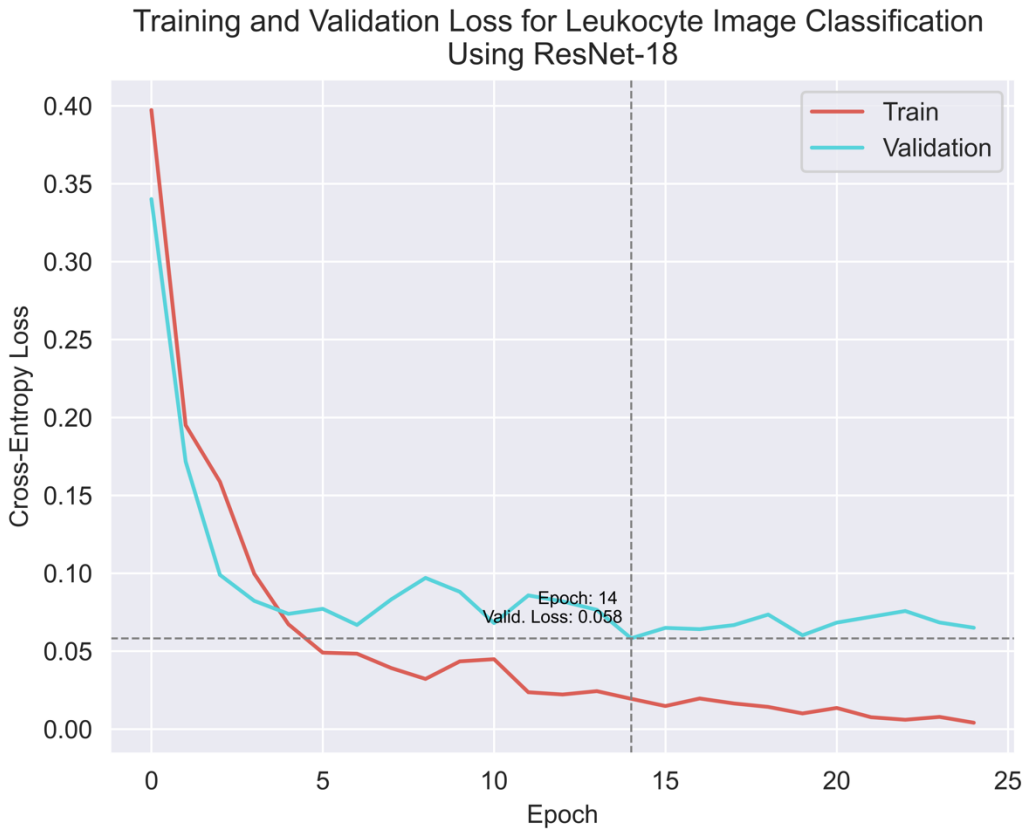
Figure 2. Residual learning: a building block.

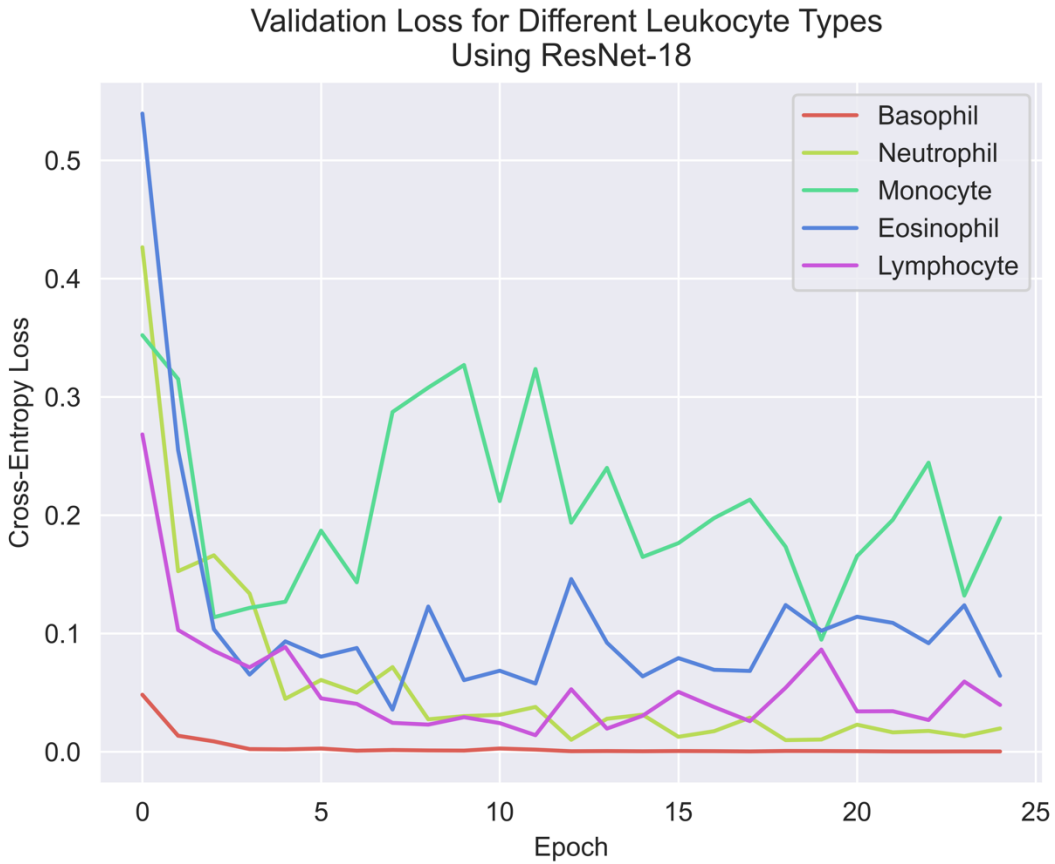
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Model Training Workflow Details

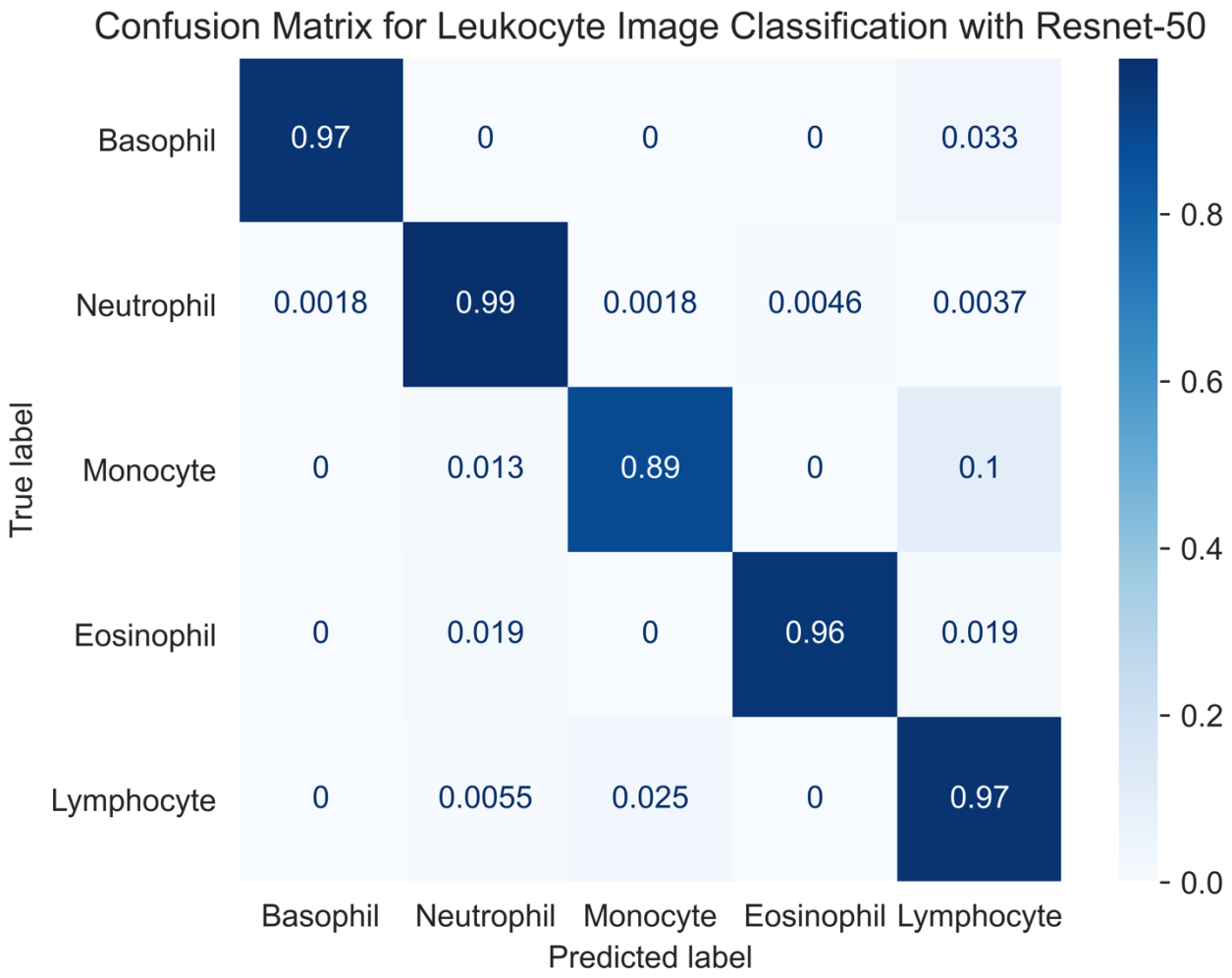
- Split: stratified, 75/15/10 train/validation/test
  - Select model based on minimum validation loss, all final evals on test
- Hardware: Nvidia V100 (16 Gb VRAM), 2 x Xeon Platinum 32-core 2.6GHz processors, 128 Gb RAM (my school's HPC cluster)
- Hyperparameters:
  - Learning rate: 1e-5
  - L2 penalty: 1e-4
  - 25 epochs
  - Batch size of 32
  - Optimizer: ADAM
  - Loss function: Cross-Entropy, weighted by (total samples / class count)
- Time to train: 8.93 min (ResNet-18),10.55 min (ResNet-34), 17.48 min (ResNet-50)

Model Training Results – ResNet-18

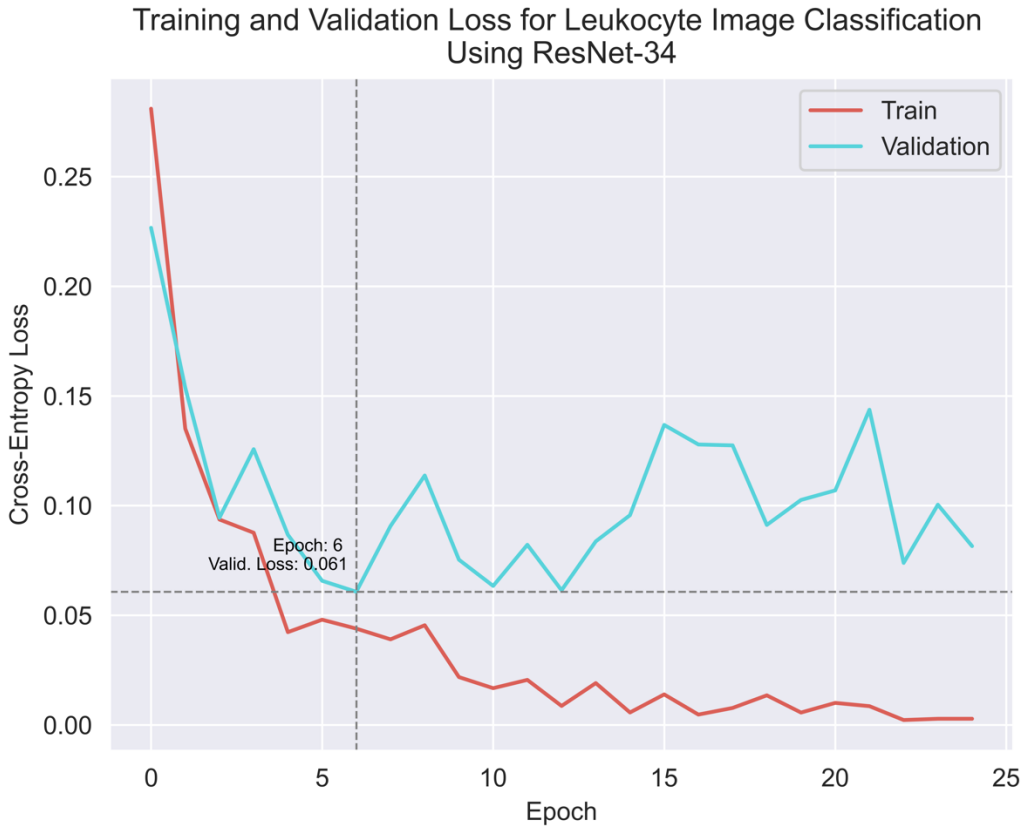


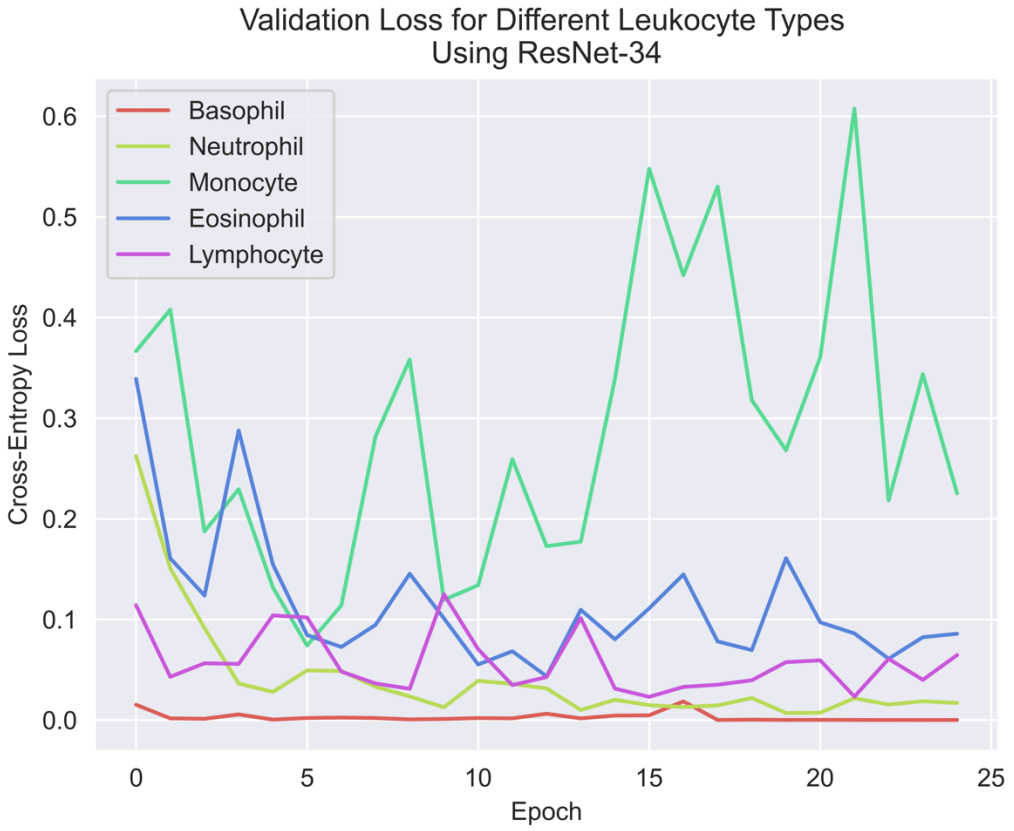


Model Training Results – ResNet-18

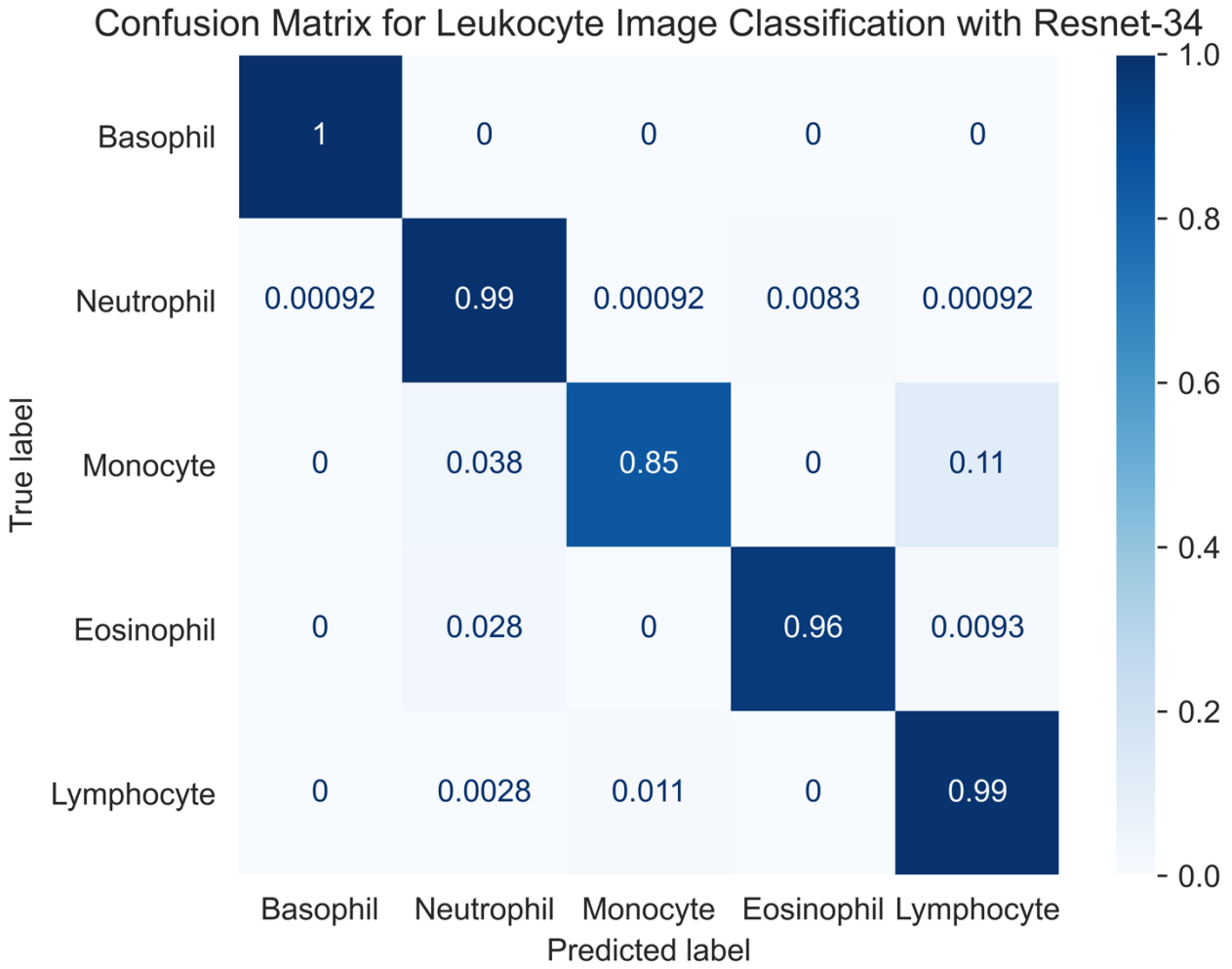


Model Training Results – ResNet-34

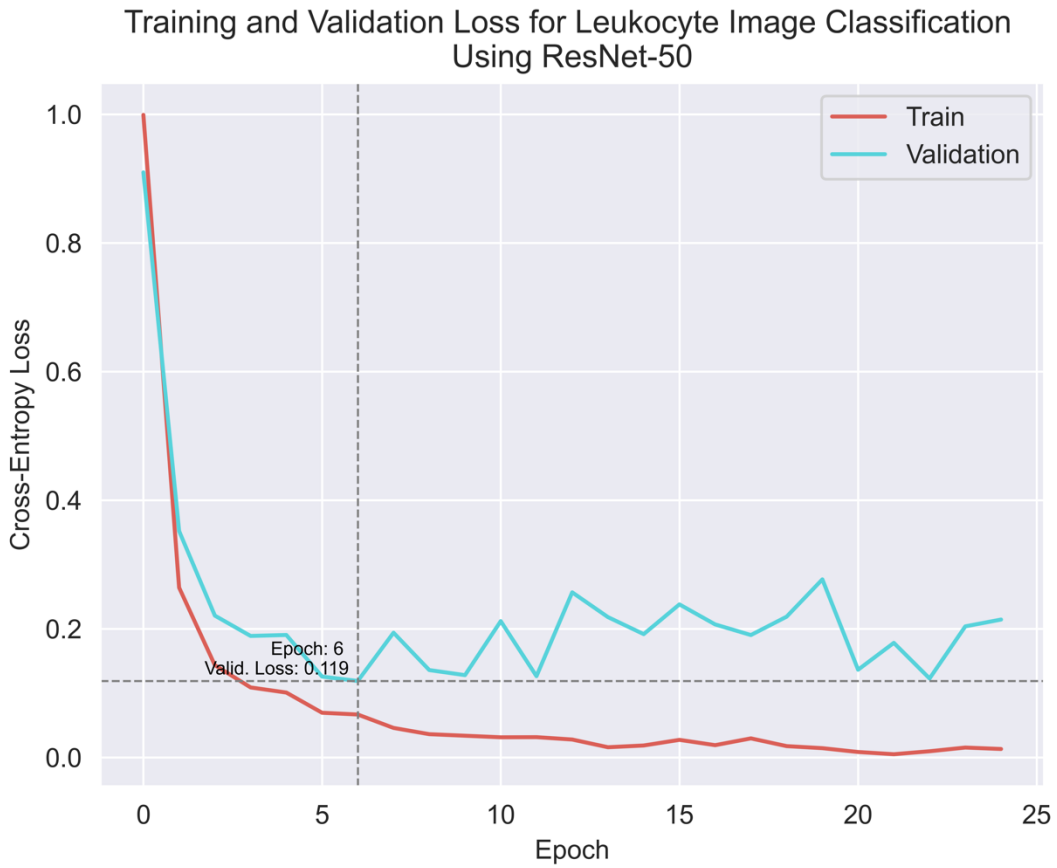


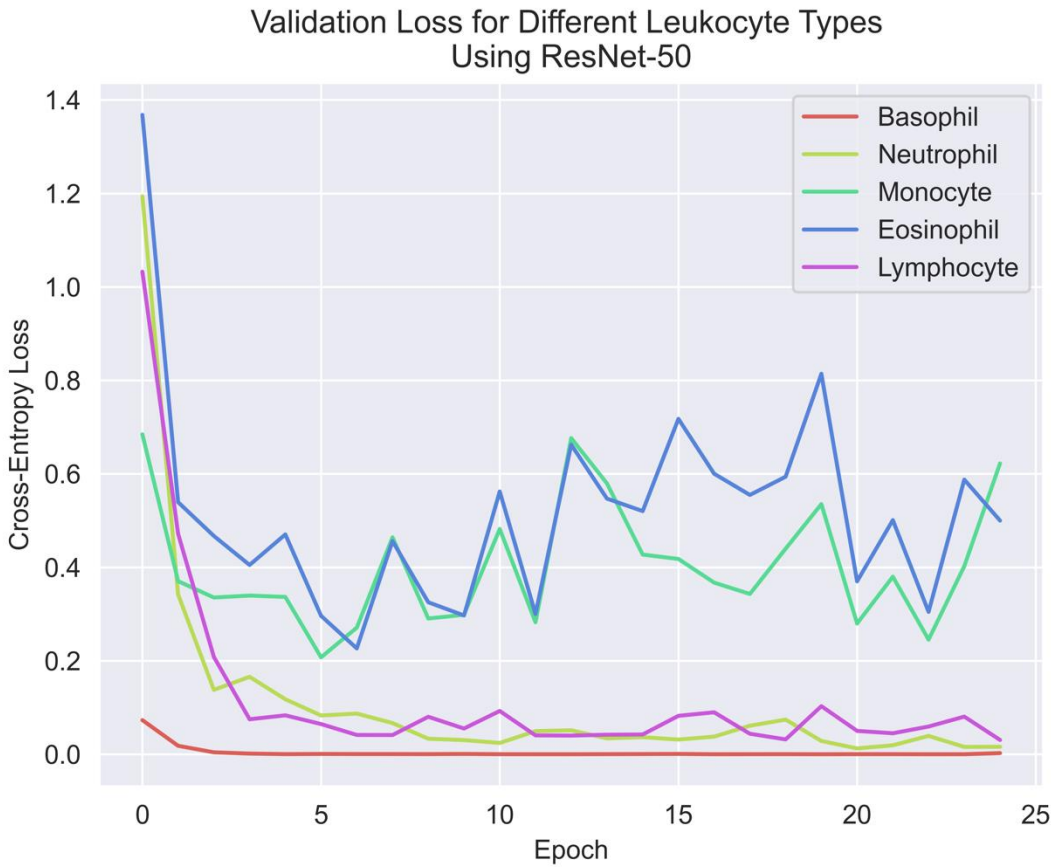


Model Training Results – ResNet-34

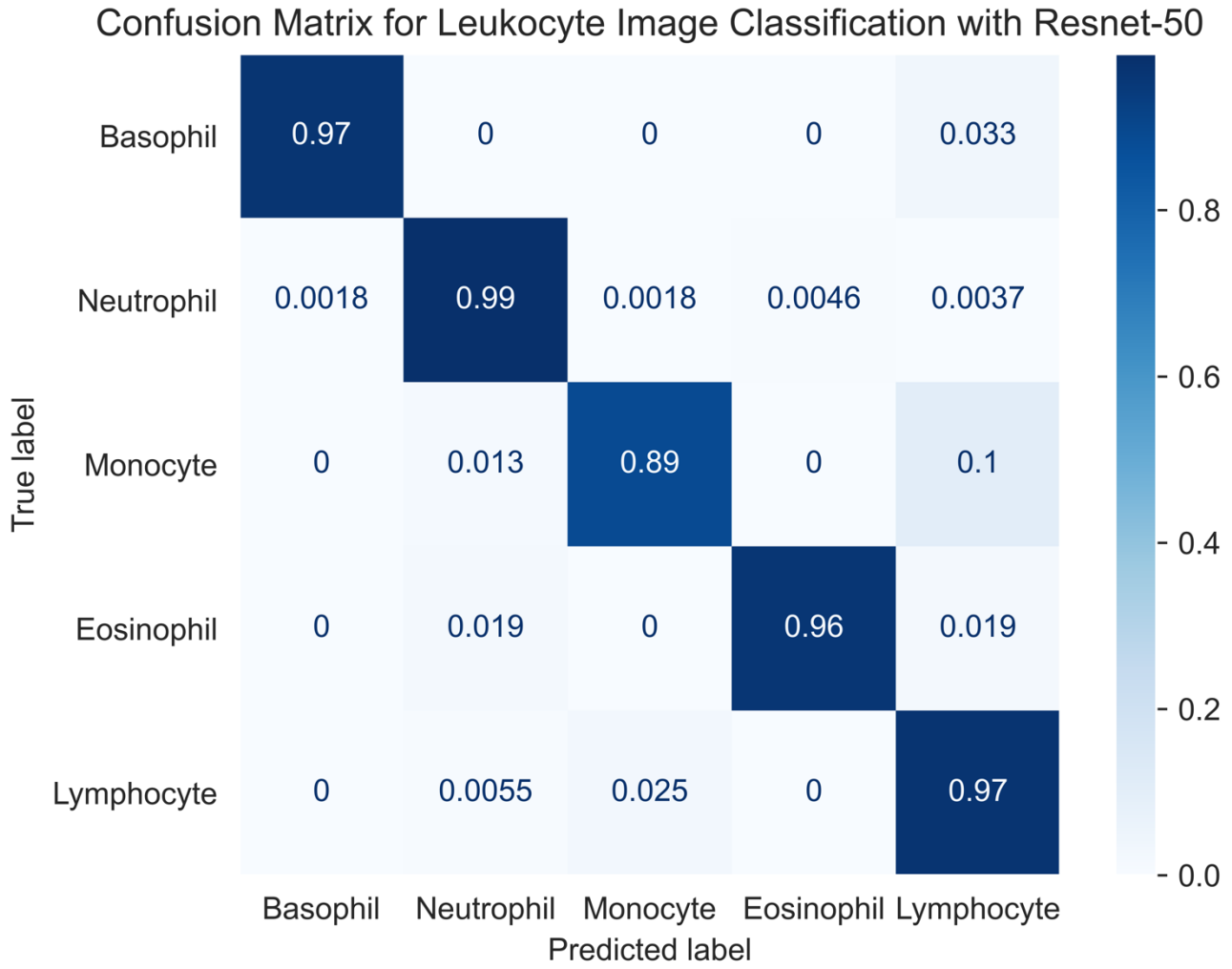


Model Training Results – ResNet-50



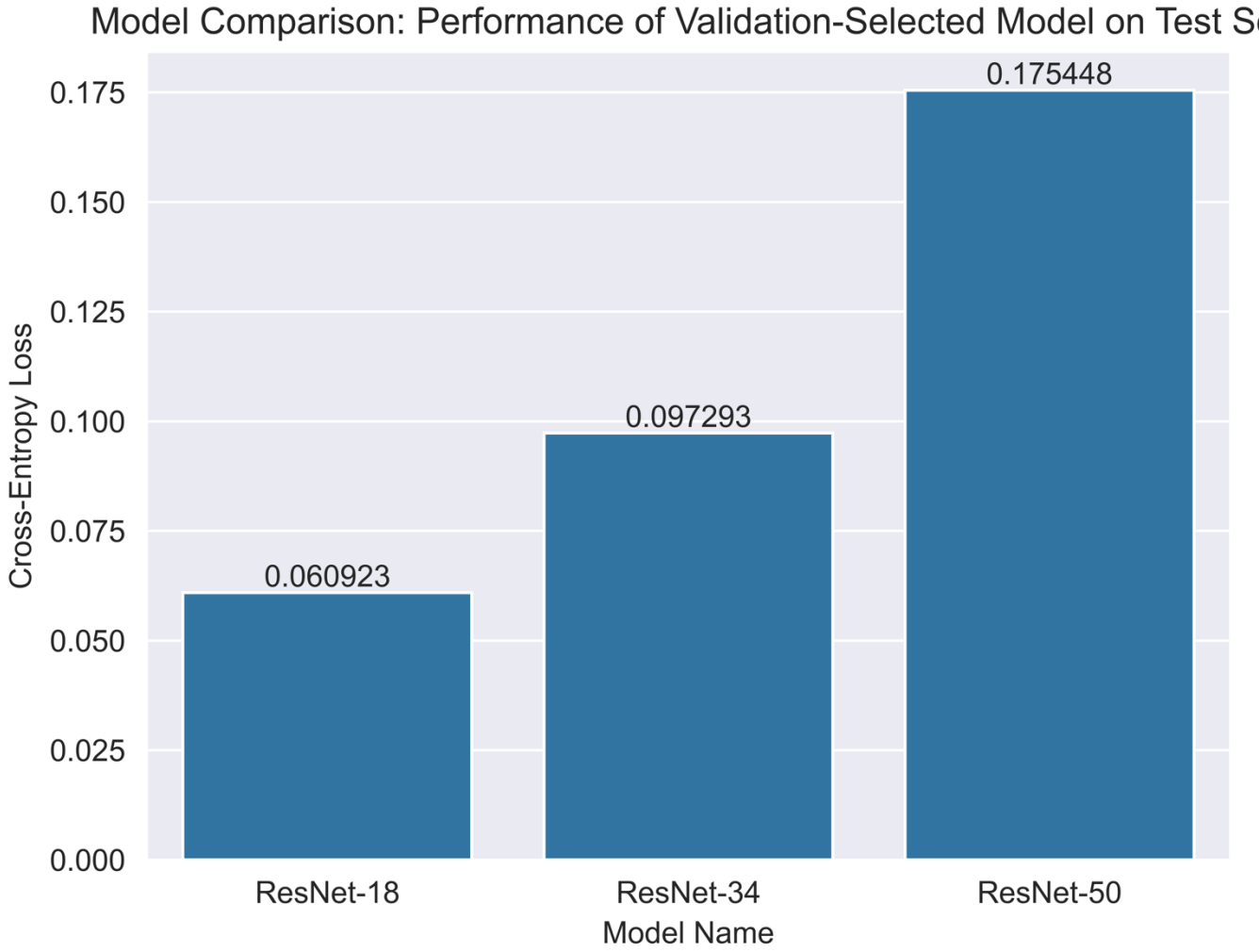


Model Training Results – ResNet-50



Model Training Results: Test Set Comparison

- Same splits for all models
- N=1, not very scientific. Ideally would repeat training over many trials and do a t-test for $H_0 : \mu_{diff} = 0$



Model Training Results – More Classification Metrics for ResNet-18

	Basophil	Neutrophil	Monocyte	Eosinophil	Lymphocyte	accuracy	macro avg	weighted avg
precision	1	0.888	0.967	0.971	0.995	0.981	0.964	0.981
recall	1	0.963	0.989	0.848	0.989	0.981	0.958	0.981
f1-score	1	0.924	0.978	0.905	0.992	0.981	0.960	0.981
support	30	107	361	79	1087	0.981	1664	1664

Model Inference via CLI – Python

...code...

Tech Stack

- Python
  - NumPy
  - PyTorch (+ torchvision)
  - Torchvision
  - Pandas
  - Scikit-Learn
  - Matplotlib + Seaborn
- My school’s HPC system

The End!