WBC Classifier - User Guide

Welcome to the White Blood Cell (WBC) Classifier App. This tool allows users to upload microscopic images of blood smears to classify white blood cells using a convolutional neural network (CNN) model. It is designed for ease of use and real-time predictions. The application Source Code is available from the github repository located at https://github.com/Tsquareed014/wbc-classifier-app.git .

1. Uploading an Image

To classify a white blood cell:
1. Open the app in your web browser or run locally.
2 **Select** or **drag-and-drop** .jpg/.jpeg/.png files; unsupported formats trigger an alert
3. Wait for the image to display on the screen, images are automatically resized and normalized.

2. Prediction Results & Reports

Once the image is uploaded:

- **Real-Time Classification:** displays predicted WBC type and confidence score.
- **Confidence Threshold Slider:** default 0.85, adjustable to balance sensitivity and specificity.
- **Low Confidence Warning:** suggests manual review for safety.

Batch Prediction Report: Upon batch upload, the system generates a differential report table containing:

- Image filename
- Predicted class and confidence
- Class tallies and summary statistics (counts and percentages)
- Timestamp and processing time per image

These reports support lab workflows and are available for download as CSV.

3. Understanding the Output

You will see:

- The predicted WBC type and its confidence score.
- The Image next to a Saliency Map which illustrates which pixels are informing the model.
- For batch predictions a table will display with the differential, confidence scores, and tallies of each class.

4. Error Handling

If an unsupported file format is uploaded or no image is uploaded, the app will display an appropriate alert message.

5. Sidebar Features

The sidebar provides interactive controls and customization options to tailor the classification workflow:

- **Model Selection Dropdown:** Choose among three available models: MobileNetV2 (head), MobileNetV2 (fine-tuned), and Enhanced CNNv2. Switching models dynamically reloads weights and updates inference speed and accuracy expectations.
- **Confidence Threshold Slider:** Adjust the minimum confidence level (range 0.50–0.99) required to accept a prediction. Lowering the threshold increases sensitivity at the expense of specificity; raising it reduces false positives.
- **Custom Model Upload:** Upload a user-trained model (.h5 or .keras format) via drag-and-drop to evaluate alternative architectures. The app validates and loads custom weights in place of default models.
- **True Label CSV Upload:** Import a CSV file containing ground-truth labels (columns: filename, class to compute live accuracy metrics and generate confusion matrices on batch predictions.
- **URL Image Upload:** Upload an image for classification via exact URL.
- **Normalization Dropdown:** Chose between two normalization methods that fit the model selected.
    - 0–1 Scaling: pixel/255 → [0,1]
    - Mean–Std Z-score: $(x-\mu)/\sigma$ per channel (custom μ/σ)

6. Supported WBC Classes

- Neutrophil
- Eosinophil
- Basophil
- Lymphocyte
- Monocyte

7. Security

Sensitive data (e.g., PHI, credentials, API keys) should be stored locally, never committed to Git, with HTTPS/TLS enforced (Streamlit Cloud and self-hosted). Integration with Laboratory Information Systems would be done within organization networks.

8. About the Models

Three Models are available from the sidebar dropdown menu. Two of the models are variations of the MobileNetV2 architecture while the third is a custom CNN designed for this application and trained on the Raabin and CellaVision DM96 Peripheral Blood Cell Datasets.

MobileNetV2, an evolution of the MobileNet architecture, introduces inverted residual blocks and linear bottlenecks to enhance efficiency and performance, making it ideal for resource-constrained devices. In transfer learning, MobileNetV2's pre-trained weights on ImageNet are leveraged to initialize a model, allowing fine-tuning on a specific task with limited data by adapting the final layers or the entire network.

MobileNetV2 (head) uses a frozen ImageNet backbone with a head, therefore it leverages a pre-trained MobileNetV2 model uses a pre-trained MobileNetV2 with a frozen backbone (trainable=False), where only the added head layers are trained. It is compiled with the Adam optimizer (learning rate 1e-4), categorical cross-entropy loss, and accuracy metric, then trained on our datasets. **MobileNetV2(fine-tuned)** starts with the same pre-trained MobileNetV2, but unfreezes the last convolutional block (excluding BatchNormalization layers) for fine-tuning. It is compiled with the Adam optimizer (learning rate 1e-5), categorical cross-entropy loss, and accuracy metric, then trained on the datasets.

The **Enhanced CNNv2** is a convolutional neural network (CNN) with three convolutional blocks, each consisting of a Conv2D layer with ReLU activation, kernel regularization (weight decay of 1e-4), Batch Normalization, and MaxPooling2D. The network concludes with a GlobalAveragePooling2D layer, a dense layer with 128 units and

ReLU activation, a dropout layer, and a final dense layer with softmax activation for classification.

Model	Parameters	Description	Normalization
Enahnced CNNv2	~1.4M	4-block CNN train from scratch with L2, Dropout 0.6	0-1 Scaling
MobileNetV2 (head)	~2.3M	Frozen ImageNet backbone + head	ImageNet
MobileNetV2 (fine-tuned)	~3.8M	Fine-tuned last conv block	ImageNet

Table 1. Model Summary.

9. Model Training Data

The models were trained on high resolution images of peripheral white blood cells. Below are the model performances on the validation sets. The Enhanced CNNv2 and MobileNetV2 (Fine-tuned) both achieved high performance. The fine-tuned MobileNet achieved greater than 95% accuracy, with better differentiation of eosinophils and basophils, reflecting the benefit of fine-tuning on dataset-specific features, while the Enhanced CNNv2 reached 90% accuracy, performing well across all classes but requiring longer inference time. Below is the evaluation data for review on each of the 3 models.

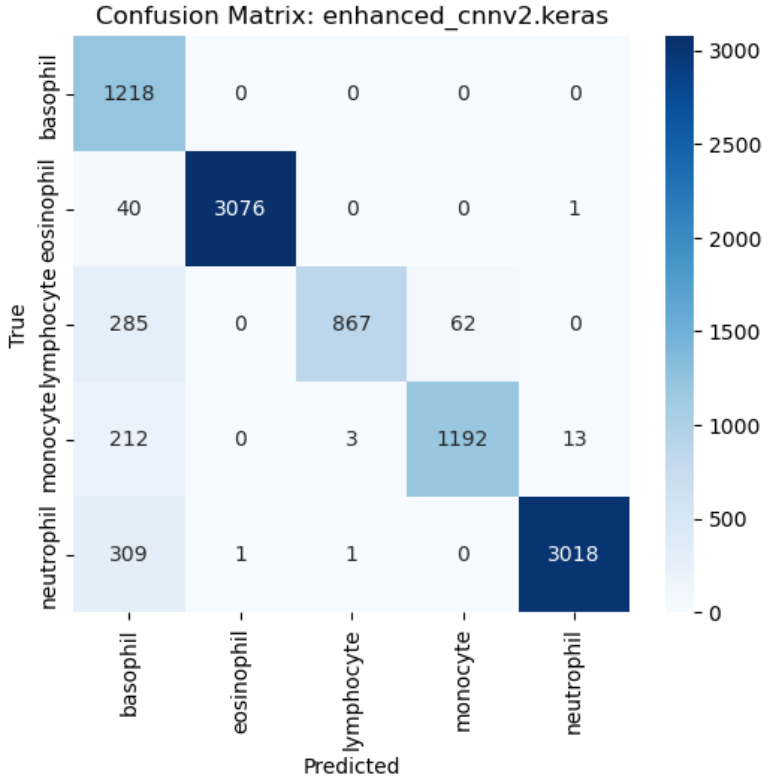
Training Metric from Enhanced CNNv2:

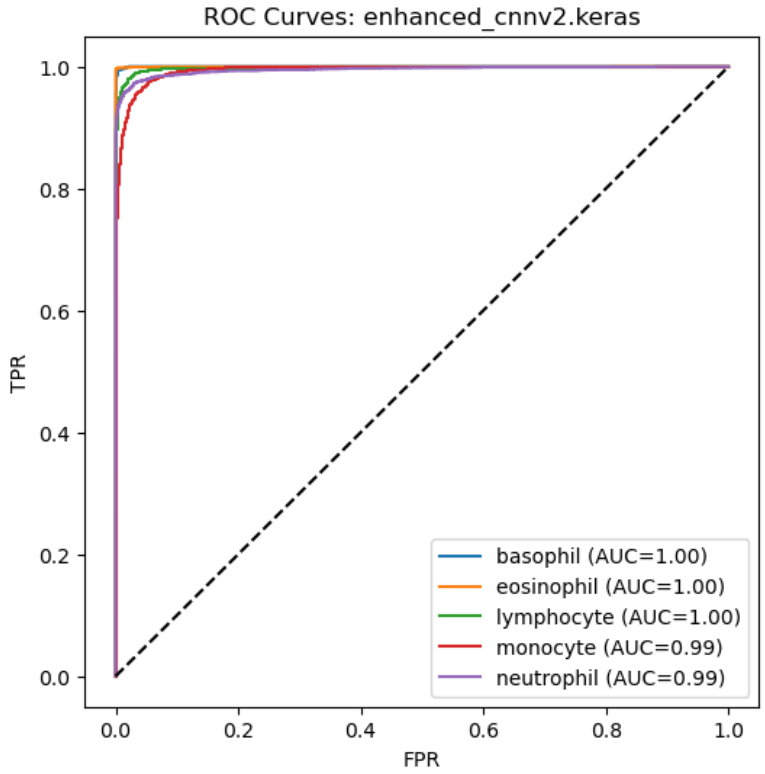
```
Test Loss: 0.3253   Test Accuracy: 0.9100

Classification Report
              precision    recall  f1-score   support

    basophil       0.59      1.00      0.74      1218
  eosinophil       1.00      0.99      0.99      3117
  lymphocyte       1.00      0.71      0.83      1214
    monocyte       0.95      0.84      0.89      1420
  neutrophil       1.00      0.91      0.95      3329

    accuracy                           0.91     10298
   macro avg       0.91      0.89      0.88     10298
weighted avg       0.94      0.91      0.92     10298
```





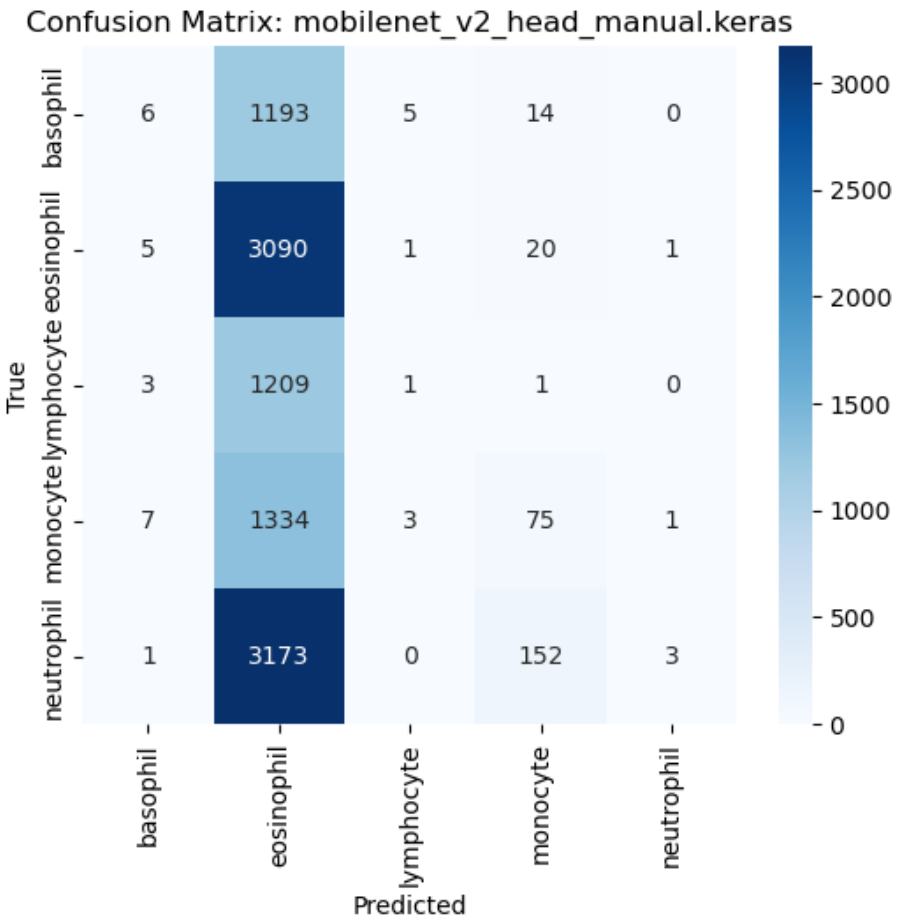
Training Metrics from MobileNetV2(Head)

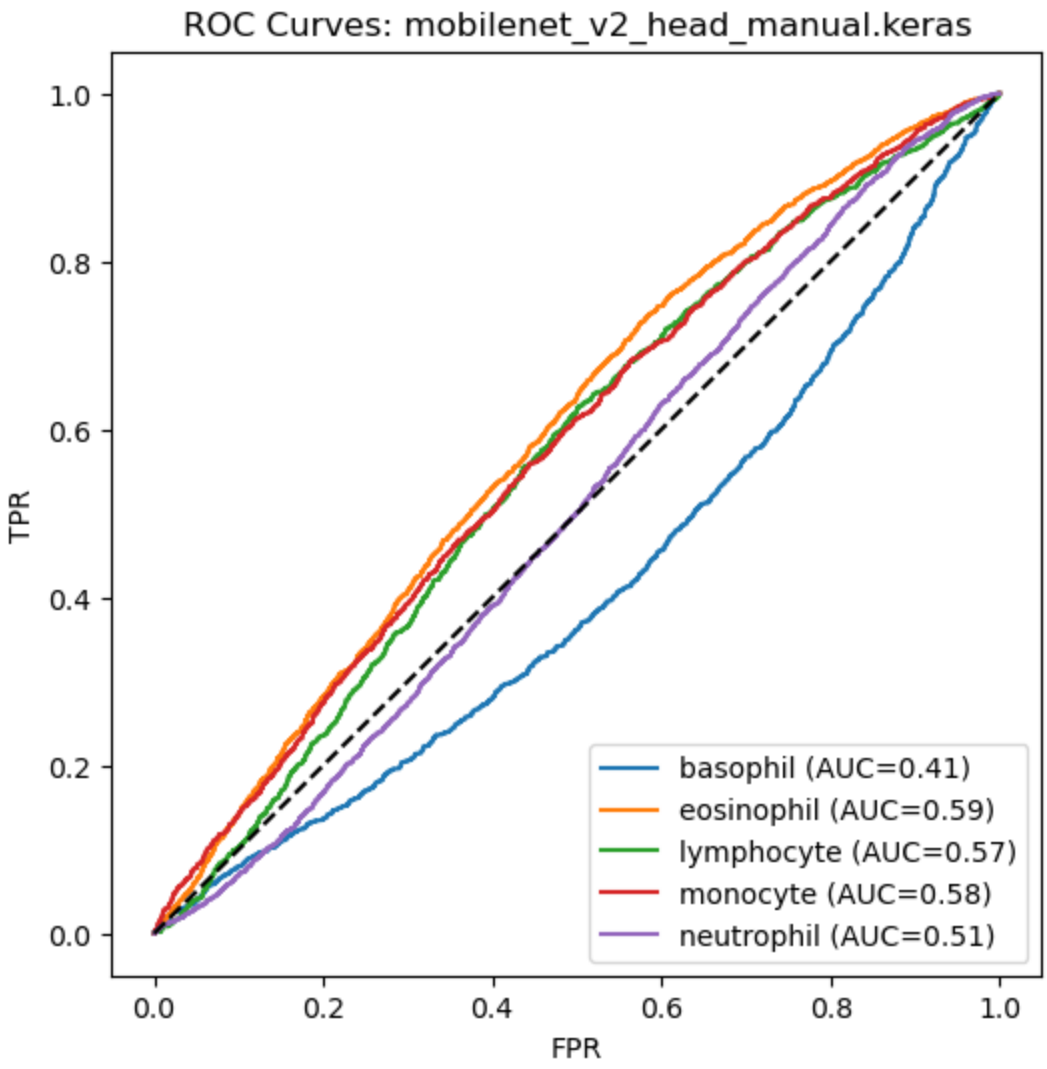
```
Test Loss: 2.1625   Test Accuracy: 0.3083

Classification Report:
              precision    recall  f1-score   support

    basophil       0.27      0.00      0.01      1218
  eosinophil       0.31      0.99      0.47      3117
  lymphocyte       0.10      0.00      0.00      1214
    monocyte       0.29      0.05      0.09      1420
  neutrophil       0.60      0.00      0.00      3329

    accuracy                           0.31     10298
   macro avg       0.31      0.21      0.11     10298
weighted avg       0.37      0.31      0.16     10298
```


Confusion Matrix: mobilenet_v2_head_manual.keras
True
basophil
eosinophil
lymphocyte
monocyte
neutrophil
6
1193
5
14
0
5
3090
1
20
1
3
1209
1
1
0
7
1334
3
75
1
1
3173
0
152
3
3000
2500
2000
1500
1000
500
0
basophil
eosinophil
lymphocyte
monocyte
neutrophil
Predicted


ROC Curves: mobilenet_v2_head_manual.keras
TPR
FPR
0.0
0.2
0.4
0.6
0.8
1.0
basophil (AUC=0.41)
eosinophil (AUC=0.59)
lymphocyte (AUC=0.57)
monocyte (AUC=0.58)
neutrophil (AUC=0.51)

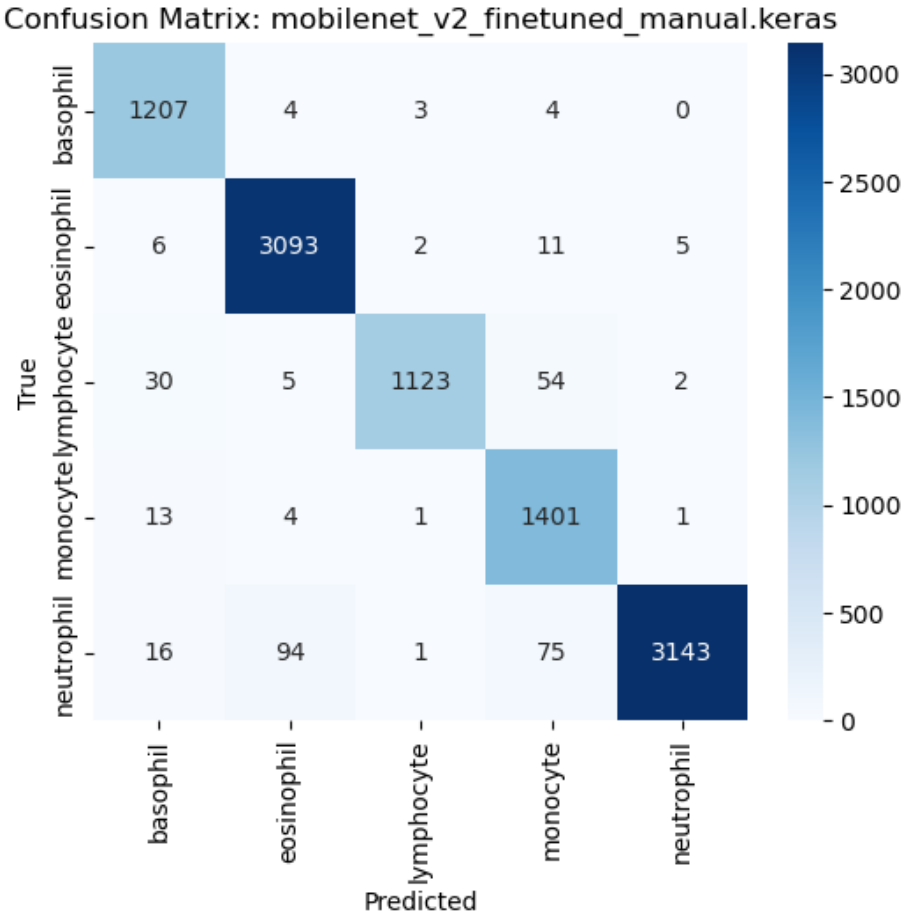
Training Metrics from MobileNetV2 (Finetuned)

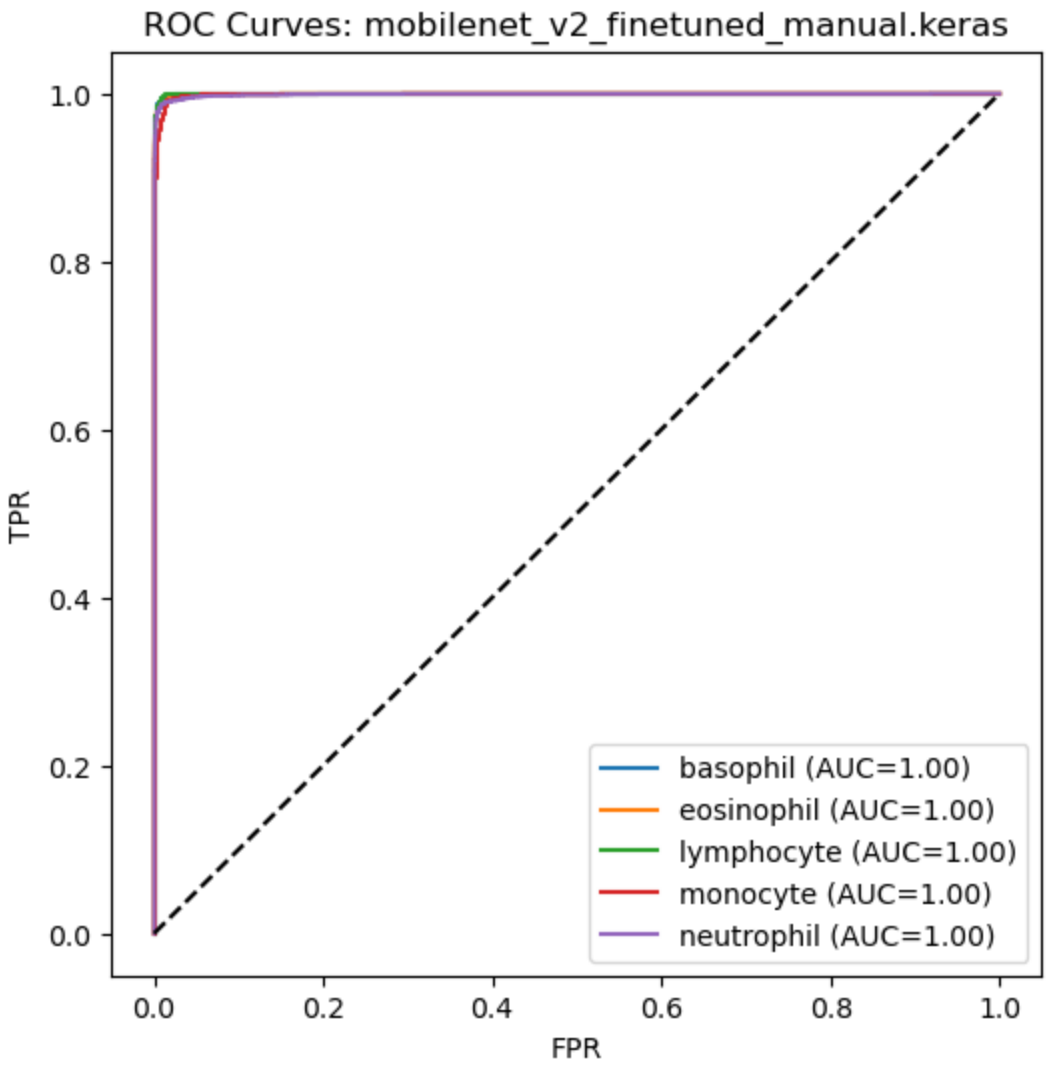
```
Test Loss: 0.0969   Test Accuracy: 0.9679

Classification Report:
              precision    recall  f1-score   support

    basophil       0.95      0.99      0.97      1218
  eosinophil       0.97      0.99      0.98      3117
  lymphocyte       0.99      0.93      0.96      1214
    monocyte       0.91      0.99      0.95      1420
  neutrophil       1.00      0.94      0.97      3329

    accuracy                           0.97     10298
   macro avg       0.96      0.97      0.96     10298
weighted avg       0.97      0.97      0.97     10298
```





References

Acevedo, A., Merino, A., Alférez, S., Molina, Á., Boldú, L., & Rodellar, J. (2020). *A dataset for microscopic peripheral blood cell images for development of*

automatic recognition systems [Data set]. Mendeley Data, V1. https://doi.org/10.17632/snkd93bnjr.1

Blumenreich, M. S. (1990). The white blood cell and differential count. In H. K. Walker, W. D. Hall, & J. W. Hurst (Eds.), *Clinical methods: The history, physical, and laboratory examinations* (3rd ed., Chap. 153). Butterworths. https://www.ncbi.nlm.nih.gov/books/NBK261/

Brownlee, J. (2020). *How to evaluate machine learning models*. Machine Learning Mastery. https://machinelearningmastery.com/how-to-evaluate-machine-learning-algorithms/

Chollet, F. (2018). *Deep learning with Python*. Manning Publications.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42, 60–88. https://doi.org/10.1016/j.media.2017.07.005

Kouzehkanan, Z. M., Saghari, S., Tavakoli, S., Rostami, P., Abaszadeh, M., Mirzadeh, F., Shahabi Satlsar, E., Gheidishahran, M., Gorgi, F., Mohammadi, S., & Hosseini, R. (2022). A large dataset of white blood cells containing cell locations and types, along with segmented nuclei and cytoplasm. *Scientific Reports, 12*, 1123. https://doi.org/10.1038/s41598-021-04426-x

Samek, W., Wiegand, T., & Müller, K. R. (2017). Explainable Artificial Intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 60.

Streamlit. (n.d.). Streamlit: The fastest way to build and share data apps. https://streamlit.io/

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9, 2579–2605.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. *European Conference on Computer Vision*, 818–833. https://doi.org/10.1007/978-3-319-10590-1_53

Zhou, L., Lv, Y., Wang, J., Zeng, Y., Du, M., Chen, Y., & Liu, J. (2022). Research progress of machine learning in the diagnosis of white blood cells. *Frontiers in Medicine,* 9, 9777002. https://doi.org/10.3389/fmed.2022.9777002