Malaria Classification Report

For this assignment implementation, the first step I did is to import essential libraries and then organize the image dataset folder. Through the reorganization process, the image directory was transformed from a singular folder containing all images to a hierarchical structure. In this structure, images are categorized into subdirectories named according to their labels. After that I did some data exploratory and made decisions on removing outliers. Next, with the cleaned dataset, I load, split, and normalize the training and validation dataset for future use. In the modeling step, I constructed a CNN with some basic layers and augmentations.

Below, I detail several benchmark tests conducted and describe what I did to achieve those results and keep improving my models:

1. **val_accuracy: ~0.9284 - val_loss: ~0.2301**
    a. This is my base case with non-normalizing data
    b. Total of 9 layers are used, with 3 convolutional layers, and each followed by a max pooling layer for feature extraction, a flatten layer to convert the 2D feature maps into a 1D vector, and two dense layers for classification
2. **val_accuracy: ~0.9503 - val_loss: ~0.1703**
    a. Normalized data by its mean and standard deviation before constructing the model (I have tried normalizing before/after and with mean & std vs. 1/255 by pixel and found this performed the best)
    b. Added augmentation on image transformation
3. **val_accuracy: ~0.9566 - val_loss: ~0.1363**
    a. Update the layers based on the paper *Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images*[1]
4. **val_accuracy : ~0.9648 - val_loss: ~0.1003**
    a. Used SGD optimizer (I have tried several different optimizers like Adamax, SGD, Adadelta, Adagrad, and SGD works the best)
    b. Added a L2 regularization to the model (inspired by the paper *Random search for hyper-parameter optimization*[2])
    c. Increased the dense layer size
5. **val_accuracy : ~0.9778 - val_loss: ~0.0803**
    a. Preprocess the training data by extracting features using pre-trained ResNet50 model. Inspired by article '*Detect Outliers with Cleanlab and PyTorch Image Models'* [3]
    b. Remove outliers that deviated significantly front he majority of the data in both groups by Isolation Forest detection
6. **val_accuracy : ~0.9677 - val_loss: ~0.0553**
    a. Added and tune augmentation of image properties: contrast/saturation/brightness. Inspired by the reading: *Examining and Mitigating Kernel Saturation in Convolutional Neural Networks using Negative Images*[4]

After achieving the optimal model configuration, I predicted the label for the normalized test dataset using the same normalization for the train set. Then, I prepared the final submission by exporting the predictions to a CSV file for download.

[1] Rajaraman S, Antani S K, Poostchi M, et al. Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images[J]. PeerJ, 2018, 6: e4568.

[2] Bergstra & Bengio (2012) Bergstra J, Bengio Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research.* 2012;13:281–305.

[3] Cleanlab, https://docs.cleanlab.ai/stable/tutorials/outliers.html

[4] Gowdra, N., Sinha, R., & MacDonell, S.G. (2020) Examining and mitigating kernel saturation in convolutional neural networks using negative images, in Proceedings of the 46th Annual Conference of the IEEE Industrial Electronics Society (IECON2020). IEEE Computer Society Press, pp.465-470. doi: 10.1109/IECON43393.2020.9255147