Artificial Intelligence Neural Network and Logistic Regression for Blood Cell Image Classification

Anna Carter
6702876
Department of Computer Science, University of Surrey, Guildford, Surrey, GU2 7JP

Background

Early detection of illnesses related to blood cells is crucial in reducing the incidence of major diseases [1]. Current methods of blood cell disease detection analysis are time-consuming, costly, and labour-intensive [1]. However, the emergence of artificial intelligence (AI) has provided a more efficient and automatic disease detection method, leading to improved medical diagnosis and earlier treatment for major diseases [1].

In this project, three different Artificial Intelligence (AI) model architectures for predicting the classes of peripheral blood cell images, such as Logistic Regression, Multi-layer Perceptron (MLP) and Convolutional Neural Networks (CNN), are explored. The classical, widely studied model for image processing classification task is Logistic Regression, but its linearity makes learning complex features a challenge, leading to lower predictions and model accuracy [2]. MLP neural networks can capture complex image features better, however the flattening of the image limits the model's ability to learn spatial features [3]. As a solution to this, Convolutional Neural Networks use convolutional layers to learn spatial features of images better and share parameters, giving an overall improved prediction score and less overfitting [3].

Further challenges in image classification tasks are class imbalance, bias, and overfitting, that are addressed through workarounds such as data augmentation, scaling techniques, early-stopping, dropout, and batch regularisation [3]. Metrics such as cross-validation are used to measure the prediction performance of unseen data on the model to ensure training accuracy, validation accuracy and loss scores are monitored to avoid overfitting. This project also explores reducing the complexity of the model [4].

The objective of this project is to optimise model architecture and hyperparameters to find the optimal accuracy to validation ratio to be able to correctly predict classes for peripheral blood cell images into one of 8 classes. Ultimately, the goal is to leverage the power of AI to improve the early detection of blood cell-related illnesses, accurately predicting blood cell class labels and leading to better healthcare outcomes for patients.

Data Analysis and Pre-processing

The dataset consists of 10,000 training 28x28x3 blood cell images (figure 1), a train.txt file containing 10,000 ground truth labels of the training images for training. A test folder containing 5,000 unlabelled images for testing is made available.

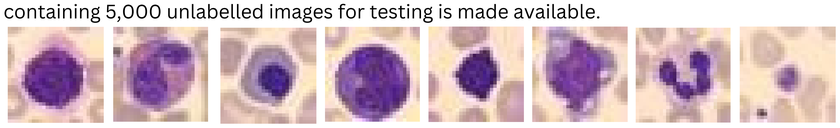


Figure 1: Blood Cell Images in Train and Test Folders (Class Type 0-7)

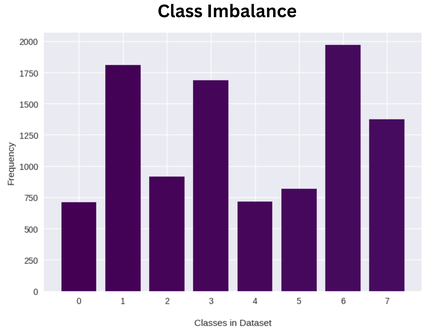


Figure 2: Initial Training Dataset Distribution

Figure 2 visually represents that the training images are not divided into classes equally so there is a degree of class imbalance in the training set. To increase the models ability to learn the features of these under-represented class types so they are predicted correctly on unseen data, data augmentation pre-processing and class weighting techniques are needed to enrich the diversity of the dataset [5]. Under-represented class types are 0, 2, 4 and 5. Furthermore, to reduce bias created by outliers, normalising the pixel values gives equal importance to all features [6].

$$CE = -\sum_{i=1}^{8} y_i * \log(y_i') \quad (1) \qquad Precision\ Score = \frac{TP}{TP + FP} \quad (2) \qquad Recall\ Score = \frac{TP}{TP + FN} \quad (3) \qquad F1\ Score = \frac{Precision * Recall}{Precision * Recall} \quad (4)$$

To train and evaluate the model performance, the 10,000 training set is divided into a 80/20 split (%), 8000 images for model training and 2000 images for cross-validation, validation accuracy, precision, recall and f1-score (equations 2, 3, 4). This split ratio is consistent across all models. Other metrics, such as categorical cross-entropy loss are used to update network parameters in back-propagation algorithm (equation 1). Additionally, the image resolution is low and some classes, 0 and 3 have very similar features so to improve the models ability to differentiate between very similar looking images, thus improve precision scores, rescaling the mean and standard deviation using standardisation techniques is recommended [4].

Model Analysis and Approach

After conducting extensive research on widely used image classification problems and carefully analysing the data, the project began with logistic regression as a baseline model. Recognising the dataset disparity and image similarity challenges, various other models such as MLP's and CNN's were explored. Techniques such as data augmentation, regularisation, learning rate adjustments, and dropout layers were utilised to improve and compare each model's training and validation accuracy [6].

Base Model 1 - Logistic Regression (LR)

Default hyperparameters: C=1, Solver='lbfgs', Penalty=l2.
Default values from library scikit learn were used as a baseline model to compare other models with specifically tuned and redesigned model architecture [7].

Model 2 - LR Hyperparameter Tuning

To address the overfitting problem in model 1 and class imbalance the below hyperparameters were tuned in a second logistic regression model:

- C=0.01
- Class_weights='balanced'

Smaller 'C' adds stronger regularisation and more weighting to under-represented classes was added to address class imbalance [7].

Model 3 - Multi-Layer Perceptron (MLP) + Batch Normalisation

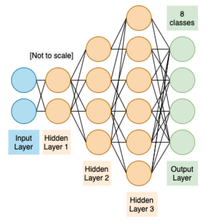


Figure 3: MLP Architecture Model 3

In the third model, to further increase the stability of the model, accelerate learning process and improve the convergence, accuracy and precision scores of Model 1 and 2, MLP architecture and batch normalisation, which forces zero mean and standard deviation 1, is explored [6]. Hidden layers in MLP added to capture and learn complex features more easily than the logistic regression model. Therefore, accuracy should improve during testing the complex image dataset. Neural Networks are prone to overfitting so it is also important to monitor this.

Model 4 - Convolutional Neural Network (CNN)

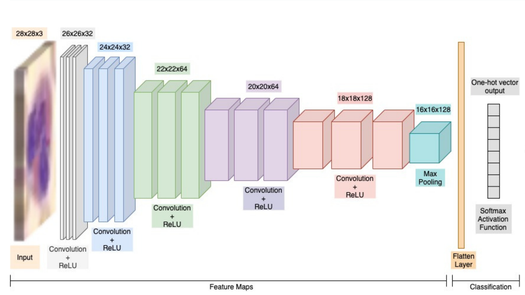


Figure 4: CNN Mode Architecture Final Model 4

Model 4 is the final model (labelled Model 4c in Jupyter notebook). Activation functions ReLU are used and most popular [6]. The parameters of the model are tuned and optimised. Techniques below are also included to reduce interdependency training and improve generalisation. Captures the spatial features better so not to overfit the model.

+ Kernal Regularisation and Dropout Layer

Due to their complexity, neural networks are prone to overfitting so methods such as adding dropout layers that randomly set inputs to 0 before the convolutional layer, to prevent interdependent training and improve its ability to generalise on unseen data. Additionally, kernel regularisation to reduce overfitting can be applied to a fourth model and the performance will be tested [6].

+ Data Augmentation

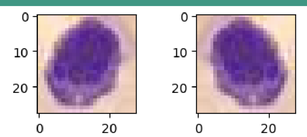


Figure 5: Horizontal Flip Transformation on Blood Cell Image

Since there is disparity in the dataset, one way of improving the models learning ability on under-represented classes is by applying image transformation techniques such as data augmentation to diversify the dataset. Since there is evidence from research that data augmentation improves the accuracy of a models ability to correctly predict image classification tasks, it is also explored in this project to improve training and validation accuracy. [6] Figure. 5 shows an example of a transformed image in Model 4.

Results

Precision

All models were trained with a training set of 8000 images and 2000 validation images were used for model evaluation. The models use the 'Categorical Cross-Entropy' function to monitor loss. Neural networks use the 'Adam' optimiser and Logistic Regression models use the solver 'lbfgs'. The learning rate is explored in Logistic Regression, it initially starts at default C=1 and is reduced to C=0.01, improving validation accuracy by 3.2%. For CNN Model 4, learning rate was set initially to 0.001 and then reduced automatically by 'Reduce LR on Plateau' was used, monitored by 'val_loss', producing the highest validation score 92.4%.

Model*	1 (LR)	2 (LR)	3 (MLP)	4 (CNN)
Training Accuracy (%)	99.7	90.5	96.5	93.7
Validation Accuracy(%)	77.2	80.4	81.2	92.4
Precision Score (%)	77.4	81.1	81.8	92.6
Recall Score (%)	77.2	80.4	81.2	92.5
F1-Score (%)	77.3	80.6	81.1	92.3
Validation Loss	1.15	0.56	1.12	0.25
Training Loss	0.04	0.32	0.21	0.2

Table 1: Accuracy and Precision Results - All Models. *Model used to predict test data labels in green.

Logistic Regression

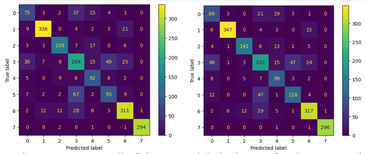
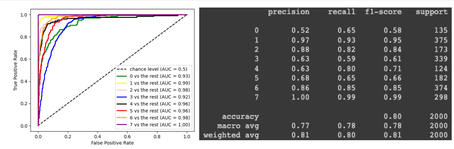


Figure 6: LR 1 (left), LR 2 (right) confusion matrix



	precision	recall	f1-score	support
0	0.52	0.65	0.58	135
1	0.97	0.93	0.95	375
2	0.88	0.82	0.84	173
3	0.63	0.59	0.61	339
4	0.63	0.80	0.71	124
5	0.68	0.65	0.66	182
6	0.86	0.85	0.85	374
7	1.00	0.99	0.99	298
accuracy			0.80	2000
macro avg	0.77	0.78	0.78	2000
weighted avg	0.81	0.80	0.81	2000

Figure 7: LR 2 ROC (left) and classification report (right)

Neural Networks

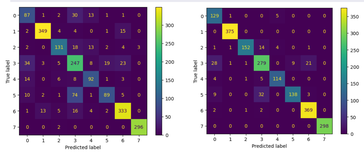
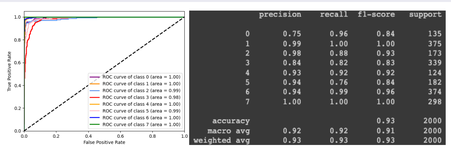


Figure 8: MLP (left), CNN (right) confusion matrix



	precision	recall	f1-score	support
0	0.75	0.96	0.84	135
1	0.99	1.00	1.00	375
2	0.98	0.88	0.93	173
3	0.84	0.82	0.83	339
4	0.93	0.92	0.92	124
5	0.94	0.76	0.84	182
6	0.94	0.99	0.96	374
7	1.00	1.00	1.00	298
accuracy			0.93	2000
macro avg	0.92	0.92	0.91	2000
weighted avg	0.93	0.93	0.93	2000

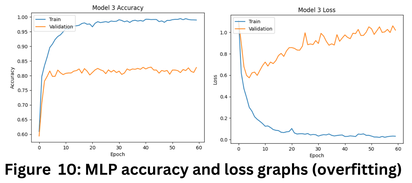
Figure 9: CNN ROC (left) and classification report (right)

Summary

Balancing class weights improved LR generalisation ability, LR 2 - 80% validation accuracy (figure 6). The confusion matrix and ROC curves show accuracy and precision for under-represented classes 0, 3 and 5 further improved with image augmentation and CNN (figure 9). Additionally, neural network models and varying learning rate, 'Reduce LR on Plateau', improved learning capability and validation accuracy together (figure 8). Recall score optimisation was preferred for disease detection so fewer positive diagnosis are missed.

Overfitting, Learning Rates and Loss

Model 3 (MLP)



Model 4 (CNN)

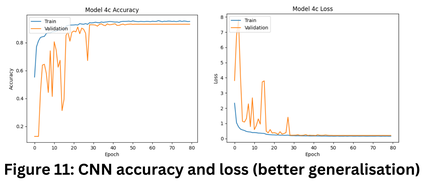


Figure 10: MLP accuracy and loss graphs (overfitting) **Figure 11: CNN accuracy and loss (better generalisation)**

The results show a large gap between mean cross-validation (CV) score (71%) and accuracy score (99%) in LR Model 1 and validation (81%) and accuracy scores (97%) in MLP Model 3 indicating some overfitting (figure 10). CNN model with dropout and regularisation techniques, significantly reduced the gap between validation (92%) and training accuracy (94%) scores, out-performing MLP. In general, the validation and accuracy gap decreases for each consecutive model indicating neural network architecture, batch normalisation, dropout and regularisation improved overfitting and generalisation (figure 11). In conclusion, LR and MLP had higher training accuracy but the CNN model 4 performed the best overall for validation accuracy (table 1) therefore Model 4 CNN is best suited to this data set.

References

[1] Dong, Zhai, Chang, Wu. "A self-adaptive approach for white blood cell classification towards point of care testing", *Applied Soft Computing*, vol. 111, 107709, 2021 https://doi.org/10.1016/j.asoc.2021.107709
[2] Jyothi, Babu, "PIANO: A fast parallel iterative algorithm for multinomial and sparse multinomial logistic regression", *Signal Processing*, vol 194, 108459, 2022. https://doi.org/10.1016/j.sigpro.2022.108459.
[3] Maleki, Ovens, Najafian, Forghani, Reinhold, Forghani, "Overview of Machine Learning Part 1: Fundamentals and Classic Approaches", *Neuroimaging Clinics of North America*, vol. 30:4, SBN 9780323712446 p.g e17-e32, 2020. https://doi.org/10.1016/j.nic.2020.08.007.
[4] Książek, Gandor, Pławiak, "Comparison of various approaches to combine logistic regression with genetic algorithms in survival prediction of hepatocellular carcinoma", *Computers in Biology and Medicine* vol. 134, 104431, 2021, https://doi.org/10.1016/j.compbiomed.2021.104431.
[5] Nam, Lee, "Random image frequency aggregation dropout in image classification for deep convolutional neural networks", *Computer Vision and Image Understanding*, vol. 232, 103684, 2023 https://doi.org/10.1016/j.cviu.2023.103684.
[6] Silva, Luz, Silva, Moreira, Silva, Lucio, Menotti, "COVID-19 detection in CT images with deep learning: A voting-based scheme and cross-datasets analysis", *Informatics in Medicine Unlocked*, vol. 20, 100427 2020, https://doi.org/10.1016/j.imu.2020.100427.
[7] scikit-learn developers. sklearn.linear_model.LogisticRegression. 2023. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html (accessed 2023)