Deep Learning for Cancer Detection

Ethan Li

University of
St Andrews

Supervised by Dr David Harris-Birtill

BSc Computer Science Senior Honours Project, University of St Andrews

Date of Submission: 22nd March 2024

Abstract

Blood cancer affects approximately 1 in 16 men and 1 in 22 women during their lifetime (Blood Cancer UK). This study explores deep learning to assist hematologists in classifying bone marrow cells for diagnosing various blood cancers. Using images from the Bone Marrow Cytology in Hematological Malignancy dataset containing over 170,000 samples and 21 classes, a convolutional neural network (CNN) was trained. Through iterative refinement via hyperparameter tuning and data augmentation, results revealed the best model employed data augmentation to obtain a validation accuracy of 78.2%, outperforming the baseline model's accuracy of 74.0% and the hyperparameter-tuned model's accuracy of 76.0%. Generalisation to unseen data was demonstrated with a test accuracy of 78.3% for the data-augmentation model whilst the baseline and hyperparameter-tuned models obtained a test accuracy of 74.5% and 75.9% respectively. However, concerns regarding the black-box nature of CNNs in medical diagnosis were raised due to potential false predictions, necessitating interpretability for gaining trust. Most research regarding medical imaging fails to address this issue while some attempts to segment an image into the most important features, known as superpixels, have been made by using local interpretable model-agnostic explanations (LIME). However, depending on the number of superpixels chosen to visualise, an imprecise explanation as to why these models produce particular output may be produced. This study addresses this issue by determining the number of superpixels required for an explanation of bone marrow smears to justify the optimised model's prediction. From this, it was discovered that on average, 43 superpixels are present in any given image and that 40 superpixels are required for explanations to be sufficient enough to justify the best model's prediction.

Declaration

I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated.

The main text of this project report is 15,975 words long, including project specification and plan.

In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

Acknowledgements

I would like to first thank my supervisor Dr. David Harris-Birtill for his invaluable guidance and confidence in me throughout this academic year. Coming into this project without any prior machine-learning experience was both intimidating and challenging but completing this body of work has been the most academically enriching experience during my time at St Andrews. I will be forever grateful for the invaluable knowledge I have gained along the way that will undoubtedly serve me well in the future. I would also like to thank my flatmates and family for their unwavering support throughout this dissertation and my time at the University. These past years at St Andrews have been a truly transformative journey, and I owe much of my growth and accomplishments to them.

Table of Contents

1. Introduction

1.1 Problem Statement

In the US, someone dies every 9 minutes from blood cancer (American Cancer Society). The primary diagnosis method involves a hematopathologist examining bone marrow samples from a patient to evaluate its response to chemotherapy (Meem and Hasan 1). When samples are evaluated, they are also further examined to identify various elements, such as the subject's cellularity - the composition and density of cells, blast excess - the presence of an increased number of immature or undifferentiated blood cells, and dysplastic cellular morphology - abnormality or atypical appearances of cells (Meem and Hasan 1). Hematopathologists gather this combined data to arrive at a final diagnostic interpretation (Meem and Hasan 2). In Western countries such as the Netherlands, patients can expect to pay around $1,995 USD for a diagnosis due to the complex procedures and infrastructure required (Uyl-de Groot et al. 1; Ananthakrishnan et al. 3). Despite these costs, manual examinations can still present an error rate between 30% and 40% depending on the type of cancer and the hematologist's level of experience (Reta et al. 2). As a reflection of this, a study by Zhu et al. found that in European countries and China, leukemia is among the top five leading causes of death among adolescents aged 10–14 years (Zhu et al. 2). As required complex treatment regimens are not readily available worldwide (Fallah 61), it is of utmost importance to help classify cells in the bone marrow correctly so that we can diagnose patients early to determine the best course of action, assisting the prevention of adolescent death. This begs the question of whether there are other methods available to us to improve diagnosis statistics whilst reducing the burden on hematopathologists. With developments in technology, various studies have attempted to utilise new machine learning techniques to examine cell images to determine either the presence of blood cancer or categorise the type of bone marrow presented in the slide. In general, the most popular approaches investigated are the use of Convolutional Neural Networks (CNN) with the help of transfer learning (TL). However, trust issues can arise between medical practitioners and CNNs as these models do not justify why they have come to a certain conclusion. To address these challenges and opportunities, this study aims to create a CNN that can predict the type of bone marrow cell in a given image from the Bone Marrow Cytology in Hematologic Malignancies dataset (See *Figure 1.1.1*) whilst evaluating the quality of predictions through an explainable AI methodology known as local interpretable model-agnostic explanations (LIME).

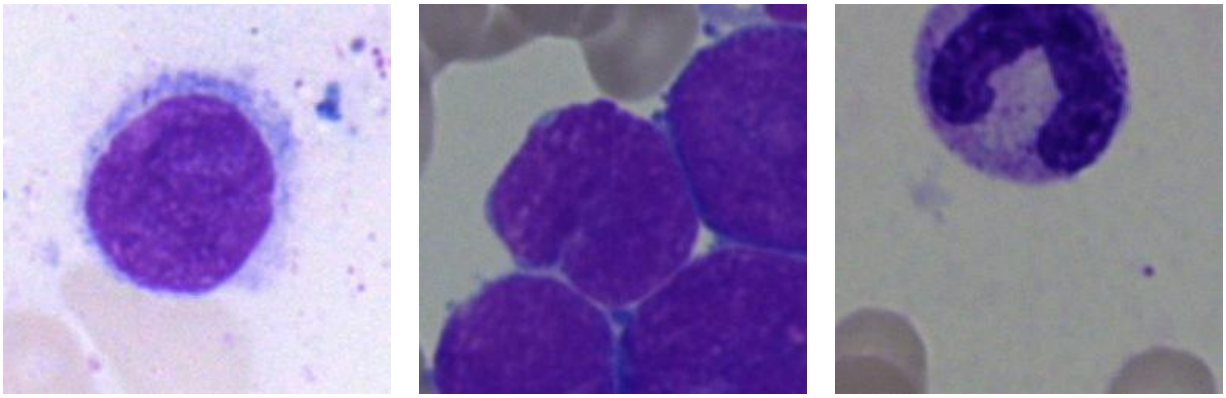


Figure 1.1.1: Example images from the Bone Marrow Cytology in Hematologic Malignancies dataset

1.2 Objectives

1.2.1 Primary

1. Perform a literature review on existing machine learning classification models using the Bone Marrow Cytology in Hematologic Malignancies dataset (and other datasets) to identify their state-of-the-art, advantages and disadvantages, as well as potential research gaps
2. Create an unoptimised minimum viable analysis model that can classify the type of bone marrow cell in a given image
3. Create extension(s) of the minimum viable analysis model using different methods and parameters to optimise the model for the accuracy of predictions
4. Perform an in-depth investigation around a research gap identified by the literature review.

1.2.2 Secondary

1. Explore the impact data augmentation has on the accuracy of the model
2. Compare and contrast the models implemented throughout the investigation

1.3 Report Structure

The report structure is as follows:

- Chapter 2 is a literature review to explore existing solutions that utilise the dataset sourced from The Bone Marrow Cytology in Hematological Malignancy dataset as well as existing solutions to other medical imaging classification problems.
- Chapter 3 is dedicated to addressing the ethical considerations in this project.
- Chapter 4 describes the methodical process of creating an unoptimised minimum viable analysis model capable of classifying bone marrow cells, followed by iterative enhancements to refine its performance.
- Chapter 5 presents the model's performance on previously unseen data.
- Chapter 6 delves into an experiment on explainable AI based on the research gaps found in the literature review. This explores the interpretability of the model's predictions.
- Chapters 7 and 8 offer a critical discussion, evaluation, and conclusion, explaining the significance and implications of this study's findings within the context of medical imaging and deep learning.

2. Context Survey

2.1 Literature Review

With developments in technology, various studies have attempted to utilise new machine learning techniques to examine cell images to determine either the presence of blood cancer or categorise the type of bone marrow presented in the slide. In general, the most popular approaches investigated are the use of Convolutional Neural Networks (CNN) with the help of transfer learning.

2.1.1 Convolutional Neural Networks

2.1.1.1 Background

CNNs have emerged as a powerful deep-learning technique for blood cancer diagnosis through image analysis. They have gained prominence in various computer vision tasks, including medical image classification due to their ability to automatically learn hierarchical features from raw images. They consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. These layers are designed to automatically detect and extract features at different levels of abstraction from input images (See *Figure 2.1.1.1.1*).

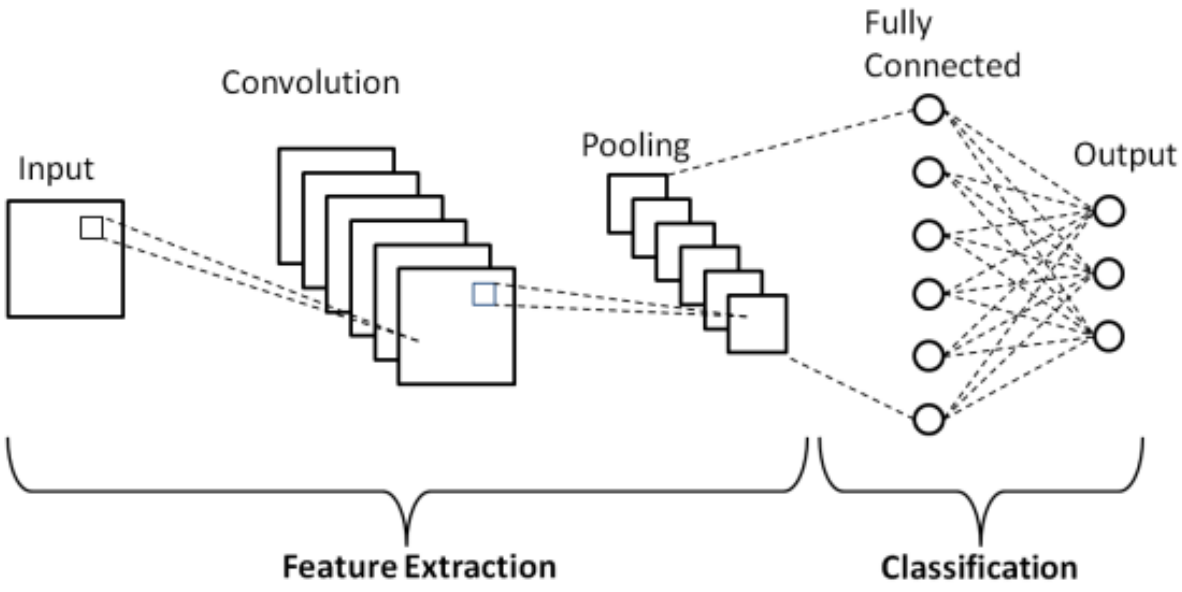


Figure 2.1.1.1.1: Image of an example convolutional neural network structure (Phung and Rhee 3)

From an image classification viewpoint, an image can simply be thought of as matrix of pixel values. This matrix gets passed on to the convolutional layer where a matrix that is orders of magnitude smaller, known as the kernel, slides across the input image to retrieve the dot product. This creates a new image known as the feature map.

After a convolutional layer, pooling is performed to downsize the feature map. This keeps the most important detected features whilst discarding the rest. Similar to the convolutional layer, a kernel is used to determine what features get mapped. An example of this is max pooling, where a kernel slides across a feature map to take the largest pixel value as its method to downsize. This feature map is then passed down to the next convolutional layer where the process restarts. An example of this can be seen in *Figure 2.1.1.1.2* where a 2x2 kernel slides across a 4x4 feature map with a stride length of 2 to obtain a new 2x2 feature map.

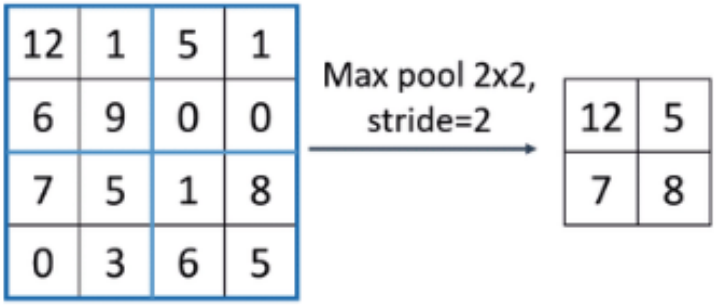


Figure 2.1.1.1.2: Max Pooling (Nirthika et al. 5330)

As more layers are added to the neural network, the kernel used in the convolutions becomes more abstract. In practice, this means that early layers might detect simple features such as geometric shapes whereas in the future layers, the kernels become too complex to classify from a human point of view. Once at the last pooling layer, the feature map is then passed on to the fully connected layers which is then responsible for classifying the input to an output.

2.1.1.2 Comparison Between CNNs and Other Classifiers

In the context of blood cancer diagnosis, CNNs have demonstrated remarkable capabilities in both blood cancer detection and classification. A key advantage of using CNNs for classification as opposed to other methods is that it exclusively relies on images as their input, enabling the model to be trained on all the visual features present in the cell (Engström and Kutakis 17). Other types of learning methods used for this type of classification such as support vector machines (SVMs) require rigorous handmade feature extraction and image pre-preprocessing to work optimally (Engström and Koutakis 17).

Examples of heavy preprocessing can be seen in Amin's investigation using K-Means Clustering and SVM classifier for the recognition of Acute Lymphoblastic Leukaemia (ALL) cells. A large portion of their time was spent in curating a dataset as they needed to obtain samples taken under similar conditions to train their model accurately (Amin et al. 52). Despite this, a degree of image quality variation was still present due to images taken at different times throughout the day whilst also using different slides, leading to a different level of illumination (Amin et al. 52). To combat this fluctuation, images were converted from an RGB colour space to an HSV colour space where a histogram equalisation to the value band to equalise the intensity of the colour was performed (Amin et al. 52).

Engström and Koutakis also claim CNNs usually yield the highest accuracy among all methods for medical imaging (17). An example of this can be seen in Rajpurohit et al. study where they performed ALL identification using CNNs, feedforward neural networks (FNNs), K-nearest neighbours algorithm (KNN), and SVMs. These classifiers required a CSV file containing the extracted attributes of cells. Furthermore, the researchers had to experiment with which sets of attributes were the most significant and achieved the highest accuracy. With their CNN however, no extra pre-processing was performed. Despite this, their CNN achieved an accuracy of 98.33% whilst FNN, SVM, and KNN achieved 95.40%, 91.40%, and 93.30% respectively (2361). One aspect to note was that the CNN was trained on a different set of images compared to the other classifiers so comparisons between accuracies may be biased.

On the other hand, Rehman et. al.'s study had all classifiers use the same image set such that comparisons between classifiers could be properly compared and contrasted. The study showed that using a CNN for classification of ALL accurately classified 97.78% of images correctly whilst Naive Bayesian, KNN, and SVM classifiers only achieved 78.34%, 80.42%, and 90.91% respectively (Rehman et al.). Ahmed et al.'s study on the identification of leukemia subtypes from microscopic images showed that their CNN classifier outperformed four other classifiers by over 29% in terms of accuracy (9). This also extends outside of blood cancer classification as Shin and Balasingham achieve over 90% of classification accuracy, sensitivity, specificity, and precision for screening polyps in the colon whilst SVMs achieved around 70-80% in all areas (3279).

2.1.1.3 Transfer Learning

One disadvantage CNNs have in comparison to other classifiers is that they require a large amount of training data to be efficient. This is because the classifier can fall victim to the overfitting problem, a

problem where a model begins to memorise the training data, including its noise and outliers, instead of generalising the underlying patterns (Ying 1). This can be a challenge in the medical field due to the size of medical image datasets being small as they typically require a skilled radiologist to manually inspect and annotate the images, a process that is both time-consuming and expensive (Swati et al. 35). Moreover, expertise is essential for iteratively fine-tuning the model as modifying the parameters that specify details of the learning process (the hyperparameters) to enhance its performance can be difficult (Swati et al. 35). Thus, training a deep CNN from scratch for use in medical diagnosis is often a challenging task and this is why transfer learning has become the established approach for utilising deep learning in medical imaging applications (Raghu et al. 2).

Transfer learning is a technique where one can leverage CNN models pre-trained on large datasets to apply them to a related task. The models that are frequently employed are typically trained on the ImageNet dataset, an extensive compilation of images specifically designed for the advancement of visual object recognition software (Meem and Hasan 3). This technique works because as mentioned in Chapter 2.1.1.1, the earlier convolutional layers often concentrate on detecting more concrete features of an image such as the edges, curves, and corners. The latter layers tend to observe more abstract features. With this in mind, two directions can be taken to apply transfer learning for a new classification task. The first approach is to leave the pre-trained model untouched to obtain the values from the last fully connected layer of the CNN before passing it to another classifier such as an SVM or KNN for classification (Loey et al.). The second approach involves changing the structure of the network by removing the high-level layers (Loey et al.). Typically this refers to the freezing of the lower layers while replacing the fully connected layers to carry out a new classification task. By utilising the learned representations and features captured by the pre-trained models, this technique enables the transfer of knowledge from one domain to another, even when the target domain has limited annotated data.

The first approach was used in Abunadi and Senan's model to classify the presence of ALL in a cell. Their approach involved using pre-trained models such as AlexNet, GoogLeNet, and ResNet-18 to extract feature maps for the SVM algorithm to classify. All these hybrid systems achieved promising results with AlexNet + SVM achieving 100% accuracy, GoogLeNet + SVM achieving 98.1% accuracy and ResNet-18 + SVM achieving 100% accuracy.

This second approach can be observed in Basymeleh et al.'s study where they compare a model trained from scratch to different pre-trained CNN architectures for an ALL image classification task. A small dataset containing 3,625 images from 89 patients was used. Their results show that the pre-trained

VGG16 model proved the most stable with a 97.50% accuracy, 99.96% sensitivity and 100% specificity in comparison to the model trained from scratch with a sensitivity and specificity of 99.92% and 93.54% respectively (Basymeleh et al.).

Matek et al.'s study also uses the second approach to show the use of a CNN architecture known as ResNeXt to classify bone marrow morphology into 21 classes. A large dataset of 171,374 expert-annotated single-cell images from 945 patients diagnosed with an array of haematological diseases was used. Their findings showed that ResNeXt outperformed in accuracy in all bar two classes against a simple feature-based CNN.

Despite the wide use of transfer learning using ImageNet, their use in medical imaging is not without its scepticism as the precise effect of learning through the use of natural images on medical images is not clear (Raghu et al. 2). With large datasets like ImageNet, they contain images with differing shapes, colours, resolution, and dimensionality whilst in general, medical images are more homogenous and lack this kind of variation (See *Figure 2.1.1.3.1*) (Alzubaidi et al. 2). Moreover, standard classification tasks usually have thousands of classes whilst, in medical imaging, this number is often much lower (Raghu et al. 2). A study by Raghu et al. found that transfer learning using ImageNet architecture offers little benefit to performance as simple models trained from scratch perform at a similar capacity.

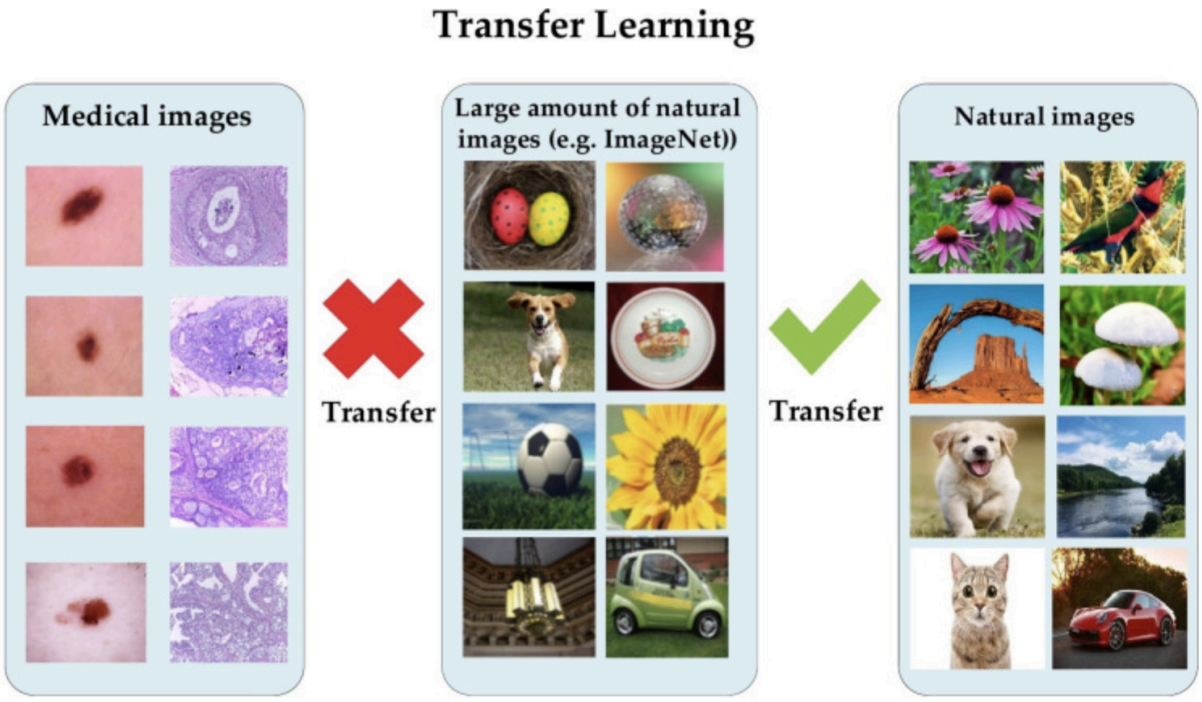


Figure 2.1.1.3.1: Image showing the difference between medical images and natural images (Alzubaidi et al. 2)

In Matek's study, despite the pre-trained model outperforming the sequential model in all but the segmented neutrophils and lymphocytes classes, these classes contained significantly more images than the rest of the dataset as they were the largest and third largest classes. This potentially suggests that transfer learning may negatively impact majority classes. Despite this, Raghu et al. also show that using architectures trained on ImageNet does have some benefits if used through a hybrid approach. An example of this approach is using the pretrained weights of ResNet up to the first two blocks whilst completely redesigning the top layers of the network to make it more lightweight (Raghu et al. 9).

Alzubaidi et al. propose a novel transfer learning approach to train the model to classify skin cancer and breast cancer on over 200,000 unlabeled medical images which were fine-tuned using a smaller dataset of 33,000 labelled skin cancer images and 400 images respectively (3). They found that their breast cancer classification ranks higher than the state-of-the-art classification models in terms of accuracy whilst also showing that the skin cancer classification model performs better than a model that does not utilise transfer learning. They do not however compare their skin cancer imaging to other classification models. Furthermore, they show that this method of transfer learning within other domains in the medical field is effective as they retrained their skin cancer classification model to classify diabetic foot ulcers into either normal or abnormal. This model obtained an F1-score of 99.03% in comparison to the state-of-the-art results having a maximum of 94.5% (17).

2.1.2 Challenges

2.1.2.1 Class Imbalance

Class imbalance is a prevalent challenge in many medical classification tasks as most existing medical datasets are often skewed in their class labels (Rahman and Davis 224). This can pose an issue because classification techniques typically prioritise maximising overall accuracy as they tend to be biased towards the majority class, often overlooking the proportionate representation of individual classes (Rahman and Davis 224). Oftentimes this leads to suboptimal performance in accurately detecting the minority class.

To counteract class imbalance, data augmentation techniques have emerged to alleviate the challenges posed by artificially increasing the size and diversity of the minority class. Matek et al. upsampled the training data to roughly 25,000 images per class by performing image rotations, image flipping, height and weight shifts, and image shears (Deep Neural Networks, 1918). However, for classes with a very

limited number of images, for example 8 images in the abnormal eosinophils class, the extent to which data augmentation can compensate for the lack of data is unclear as there is no comparison using the dataset without augmentation.

Ananthakrishnan et al. use the same dataset as Matek et al. and attempt to address the problem through the use of a Siamese Neural Network. As the name suggests, a Siamese Neural Network uses two identical networks to obtain two feature maps. A third feature map is obtained by calculating the difference between the two feature vectors before being processed by multiple dense layers to obtain a scalar. A sigmoid function is then applied to obtain a value between 0 and 1 which indicates whether the two images are dissimilar or similar respectively.

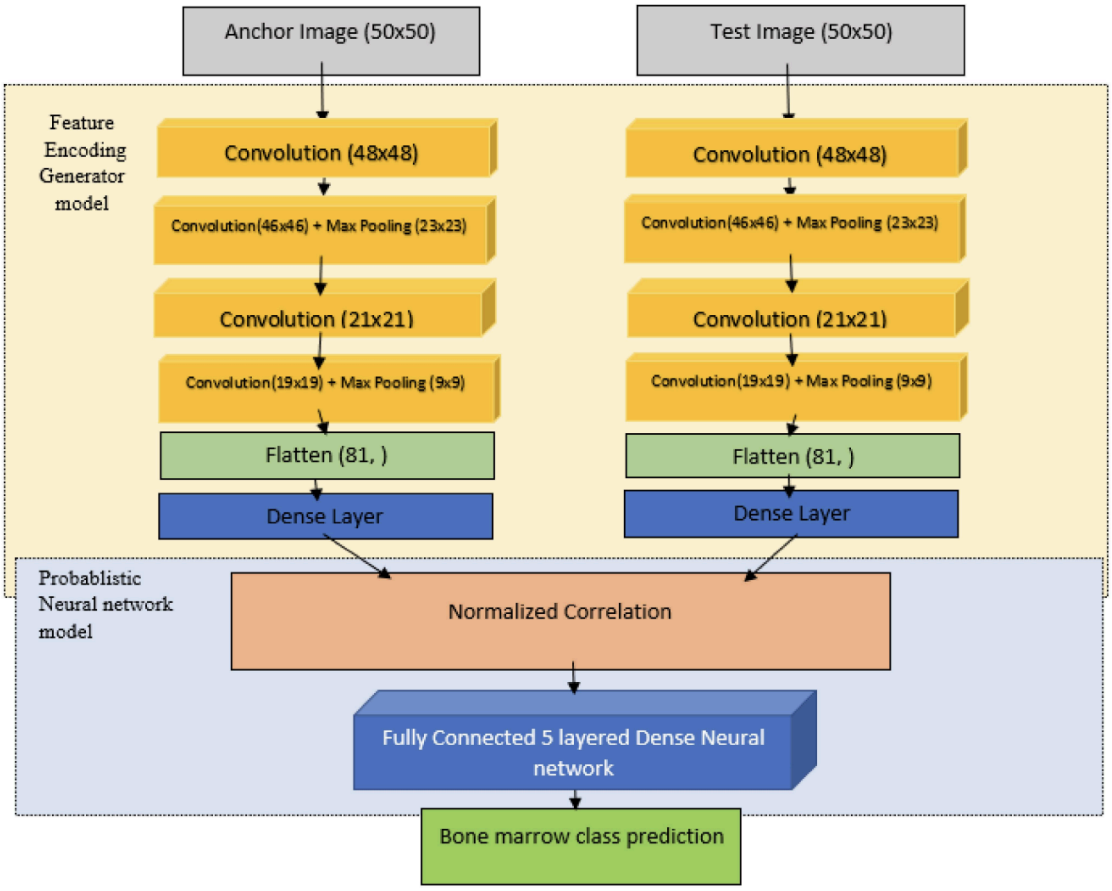


Figure 2.1.2.1.1: Ananthakrishnan et al.'s Siamese Neural Network Architecture (Ananthakrishnan et al. 6)

To train this model, The Triplet Loss method first selects an anchor image from the entire image set. Then, a positive image from the same class and a negative image from a different class are then selected. The anchor image along with either the positive or negative image is passed to the network where the distance is calculated. The distance between positive and negative instances should then be small and large respectively. This inherently reduces class imbalance as one can balance training pairs by matching images from classes with a smaller sample size with more common samples. This study showed a significantly higher precision and recall on unseen data for classes with smaller data samples using their model in comparison to Matek et al.'s model.

Cell Class	Number of Images	Matek et. al		Ananthakrishnan et.al	
		$Precision_{Tolerant}$	$Recall_{Tolerant}$	Precision	Recall
Immature lymphocytes	65	0.35	0.57	0.91	0.74
Abnormal Eosinophil	8	0.02	0.20	1.00	0.54
Faggot cells	47	0.17	0.63	0.53	0.67
Smudge cells	42	0.28	0.9	1.00	0.63

Table 2.1.2.1.2: Table showing a comparison between Matek's and Ananthakrishnan's networks in classifying four minority classes

2.1.2.2 Interpretability

The recent influence of deep learning in medical research has shown to be highly advantageous in the realm of diagnostic methods but to implement this in practice, there must be a high level of transparency and trust in such a system due to the consequences of an AI decision on human life (Kolarik et al. 1). With CNNs, they are often considered "black-box" models that do not give reasoning behind their decisions. In the context of helping hematologists diagnose cancers, explainability is an essential aspect. This is because blindly trusting a system's output can significantly alter the course of action taken by medical practitioners and patients.

To identify regions of interest within the input images that play a significant role in the ResNeXt's classification choices, Matek et al. analysed the model using SmoothGrad and Grad-CAM.

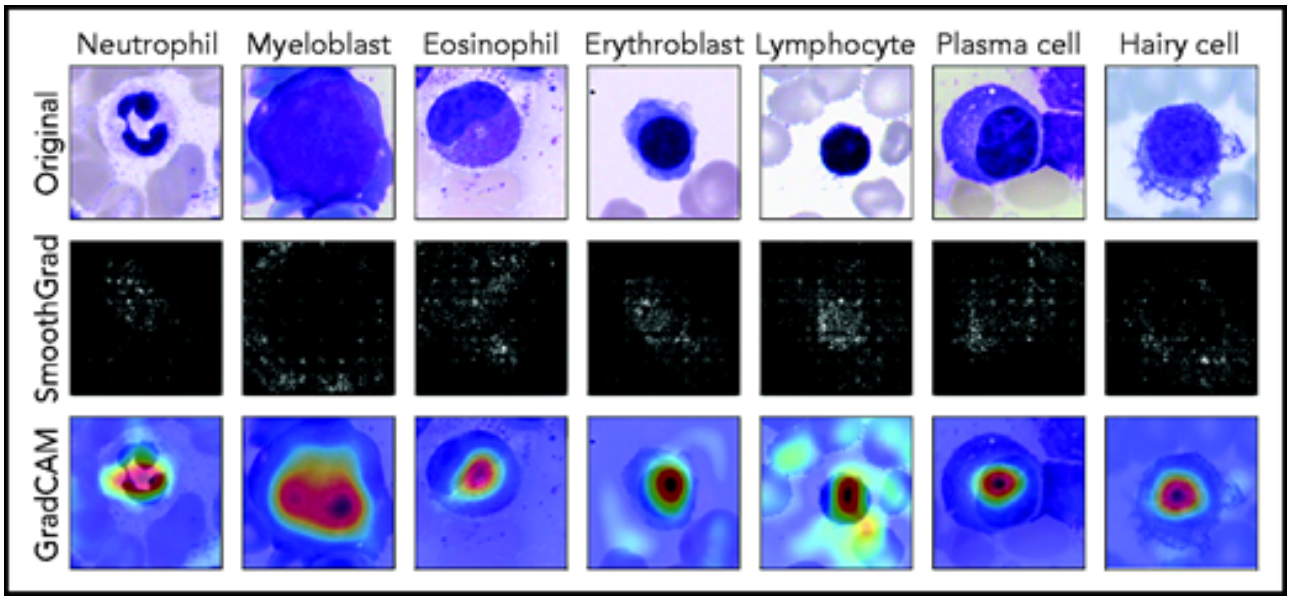


Figure 2.1.2.2.1: Grad-CAM and SmoothGrad's detection of significant features present in the ResNeXt architecture used by Matek et al. (Matek et al., Deep Neural Networks, 1924)

Abir et al. used a transfer-based learning strategy that included the ResNet101V2, VGG19, and InceptionResNetV2 architectures to classify ALL that was later validated using local interpretable model-agnostic explanations (LIME). In image classification, LIME is a method where image explanations are created by segmenting large sections of an image into superpixels to predict which superpixels are deemed most significant (See *Figure 2.1.2.2.4* and *Figure 2.1.2.2.5*). This works by feeding multiple perturbed versions of an image into the CNN to generate a set of predictions. These predictions are then used to train a simpler linear regression model to identify predicted probabilities for each superpixel's significance. An explanation can then be created by taking *n* superpixels with the largest probabilities found through the linear model to create a segmented image that denotes the most significant sections used in determining the image's class (See *Figure 2.1.2.2.4*).

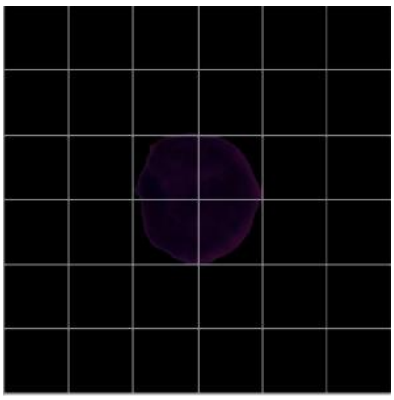


Figure 2.1.2.2.2: Image of a sampled cell (Abir et al.)

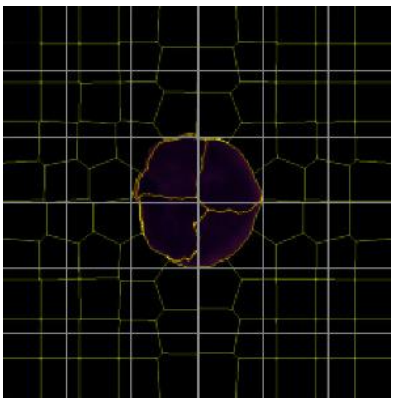


Figure 2.1.2.2.3: Superpixel boundaries used for image perturbations (Abir et al.)

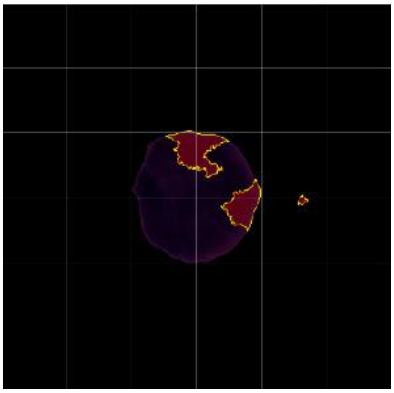


Figure 2.1.2.2.4: Explanation described by LIME (Abir et al.)

With Grad-CAM and LIME, it could reduce the efforts of a hematologist as they can verify whether the area in which the classifier produced its output is significant whilst ignoring the majority of the image. The model-agnostic aspect of the LIME acronym implies that this method works with any machine learning classifier. Thus, the knowledge of the model's architecture is not needed. Grad-CAM on the other hand leverages the gradient data that flows into the final convolutional layer of the CNN to assess the significance of individual neurons concerning a specific decision, meaning one can visualise the inner workings of the CNN to some extent (Selvaraju et al. 1).

A concern regarding interpretability, however, is the tendency for these explanations to provide overly abstract information to humans. For instance, a model may require more features than the top superpixels in a LIME explanation to arrive at the correct prediction. Consequently, these explanations will only offer an overly simple and imprecise reasoning as to why these models produce particular predictions. Evaluation of explanations generated by these tools and methodologies appear to not have been widely performed in the domain of medical image classification. It was also further discovered that throughout this investigation, Abir et al. performed systematic manipulation of the publication and peer-review process leading to the paper's retraction. This further compromises the reliability or integrity of the effectiveness of this procedure. Other domains have, however, attempted to address this issue. Shah and Sheppard experimented to determine the minimum number of superpixels required for an explanation to sufficiently justify a model's prediction. Two classifiers were used in this investigation: the first classified whether an image had a dog or cat, and the second classified different types of flowers. For the cat and dog classifier, they found that this method showed the model was able to isolate animal features well to correctly predict the class with few superpixels present in the image. This is especially true for cat images

where only 1 superpixel enabled achieved a precision of over 80%. For dogs, however, 15 superpixels were required to achieve a precision of just under 80%. For 2 of the classes in the flower classifier, the explanations generated also achieved a precision of around 80% with as few as 5 superpixels. 2 classes in the flower classifier, however, failed to achieve a precision above 40% unless the full image was passed through. This highlights that for different classes and problems, explanations may only accurately reflect the model's prediction if a certain number of superpixels are activated. This also highlights that in some cases, LIME may provide overly simple explanations which do not accurately reflect the reasoning as to why models produce a prediction.

2.1.2.3 Generalisability

Another challenge of using CNNs to classify bone marrow cells is that variations are present in the datasets used. Matek et al., Engström and Koutakis, and Ananthakrishnan et al. used a dataset of single-celled images which contained 21 different forms of malignant white blood cells. Each cell was stained using the May-Grünwald-Giemsa/Pappenheim method and photographed using a camera mounted on a brightfield microscope, using 40x magnification and oil immersion (Engström and Koutakis 13).

Amin et al. acquired a dataset containing multi-celled images split into 6 classes using a Nikon1 V1 camera coupled to Nikon Eclipse 50i light microscope under 100X power objective oil immersed setting and with an effective magnification of 1000 (Amin et al. 52).

Abunadi and Senan use the ALL_IDB1 and ALL_IDB2 datasets which contain multi-celled images obtained with an optical microscope with a Canon PowerShot G5 in JPG format. The ALL_IDB1 dataset comprises 108 images, with 49 images depicting lymphomas and 59 images representing individuals without the condition. Each image was meticulously analysed by experts in lymphoma, categorising around 39,000 blood elements within them whereas ALL-IDB2 is a subset of ALL_IDB1 images where specific regions are cropped from blast cells and normal cells (Abunadi and Senan 4).

Given that medical practitioners in the real world may have differing equipment and methods for acquiring cell images, a large degree of variability is present. All models reviewed were trained on one specific dataset and thus may struggle to generalise to a range of different images.

In an attempt to address the issue of generalisability, Matek et al. included stain-colour augmentation transformations. They test this through an annotated dataset containing 627 single-celled images with variations in illumination and resolution from the erythroid and myeloid lineages (Matek et al., Deep

Neural Networks, 1923). These images were then scaled to the 250 x 250 pixels ratio accepted by their CNN. 380 images were classified into their respective lineages but 247 of them were assigned to "artefact" and "not identifiable" categories (Matek et al., Deep Neural Networks, 1923). Being unable to make predictions on 40% of an external dataset shows the difficulty that these classifiers face regarding generalisability. Moreover, they explain that "very few publicly available data sets that include single [bone marrow] cells in sufficient number, imaging, and annotation quality exist" (Matek et al., Deep Neural Networks, 1923). This further highlights the issue a lack of standard procedure in collecting cell images has on the performance of these machine learning classifiers.

2.2 Dataset

The Bone Marrow Cytology in Hematological Malignancy dataset from the Cancer Image Archive is the dataset employed in this investigation (Matek et al., The Cancer Image Archive). Images were acquired using a brightfield microscope with 40x magnification and oil immersion. All samples underwent processing at the Munich Leukemia Laboratory (MLL) where they were scanned using Fraunhofer IIS-developed equipment and post-processed using Helmholtz Munich software to produce images with a ratio of 250x250 pixels. This dataset contains 21 classes of over 170,000 de-identified, expert-annotated single-celled images from 945 patients with a variety of hematological diseases (with Myeloma being the most prominent). Each sample was stained using the May-Grünwald-Giemsa/Pappenheim method. Due to the rarity of occurrence of certain cell types and abnormalities, the distribution of data between classes is not even. Furthermore, to prevent influencing the labelling of cell images that are easily classifiable during the training of deep learning models, distinct categories are introduced for artefacts[1], unidentified cells, and other cells that fall into morphological classes not covered by the classification system (Matek et al., Deep Neural Networks, 1918). The distribution of images can be seen in *Table 2.2.1, Figure 2.2.2* and *Figure 2.2.3*. Example images for each class can be viewed in *Figure 2.2.4*.

[1] In the context of histopathology, an artefact is a slide that has been inadequately fixed or mishandled during tissue processing, leading to changes in the appearance or structure of a cell or tissue (Taqi et al.)

Cell Name	Class Name	Number of Images	Image Distribution (%)
Abnormal Eosinophils	ABE	8	0.005
Artefact	ART	19,630	11.454
Basophils	BAS	441	0.257
Blasts	BLA	11,973	6.986
Erythroblasts	EBO	27,395	15.986
Eosinophils	EOS	5883	3.433
Faggot Cells	FGC	47	0.027
Hairy Cells	HAC	409	0.239
Smudge Cells	KSC	42	0.025
Immature Lymphocytes	LYI	65	0.038
Lymphocytes	LYT	26,242	15.313
Metamyelocytes	MMZ	3,055	1.783
Monocytes	MON	4,040	2.357
Myelocytes	MYB	6,557	3.826
Band Neutrophil	NGB	9,968	5.817
Segmented Neutrophil	NGS	29,424	17.169
Not Identifiable	NIF	3,538	2.064
Other	OTH	294	0.172
Proerythroblasts	PEB	2,740	1.599
Plasma Cells	PLM	7,629	4.452
Promyelocytes	PMO	11,994	6.999

Table 2.2.1: Table showing the distribution of images in The Bone Marrow Cytology in Hematological Malignancy dataset

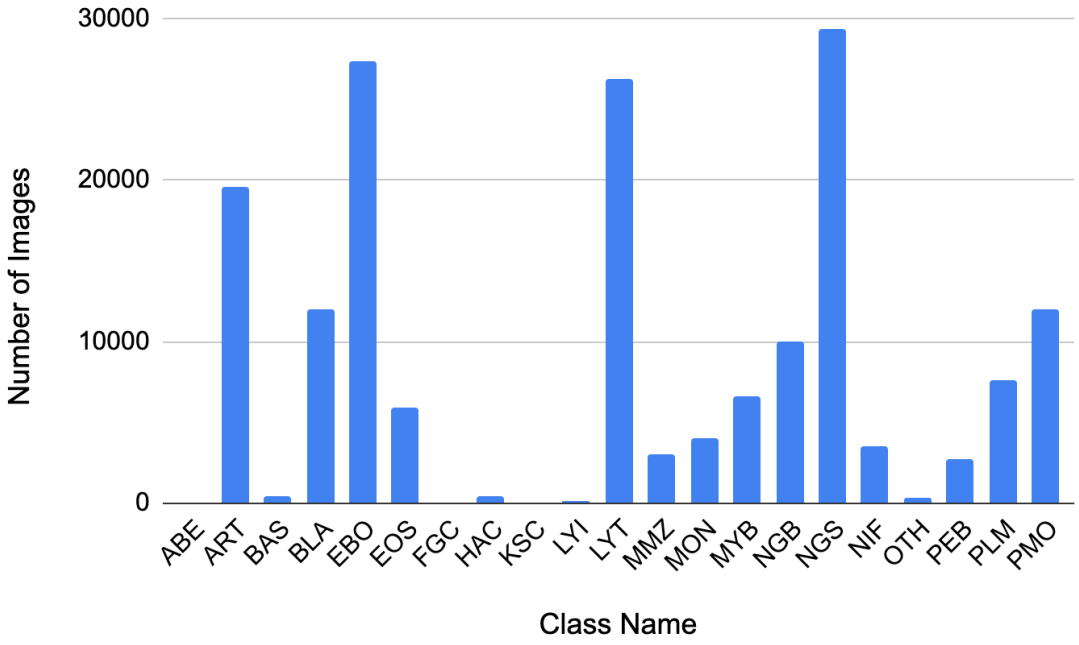


Figure 2.2.2: Bar graph showing the distribution of images in The Bone Marrow Cytology in Hematological Malignancy dataset

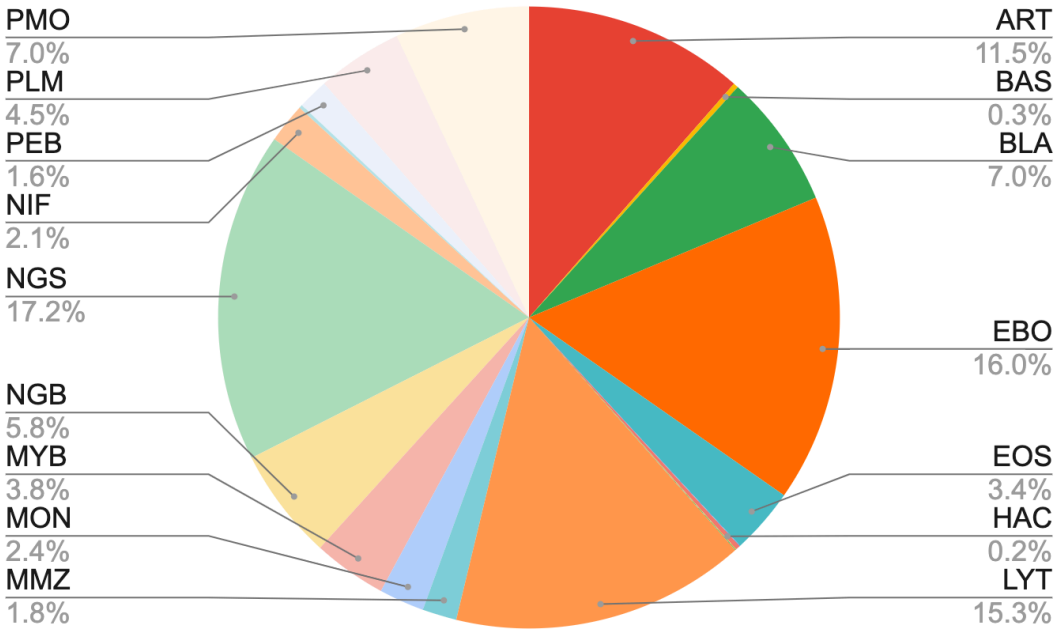
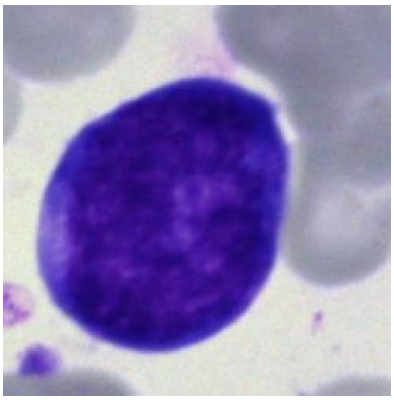
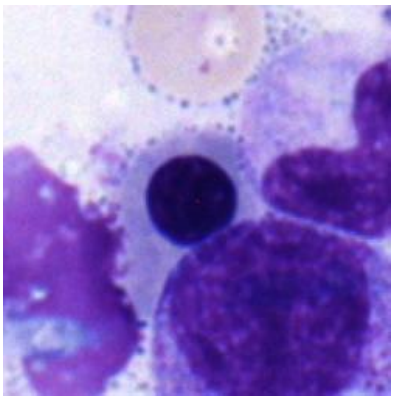


Figure 2.2.3: Pie chart showing the percentage distribution of images in The Bone Marrow Cytology in Hematological Malignancy dataset. Classes not represented in the pie chart hold less than 0.1% of images in the dataset

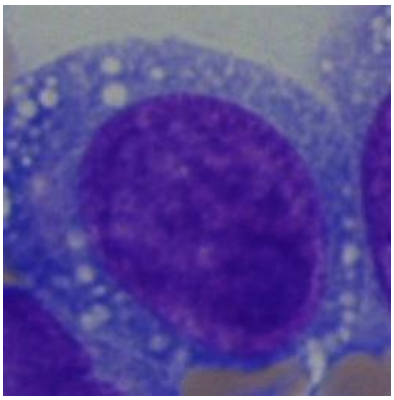
Myelopoiesis:



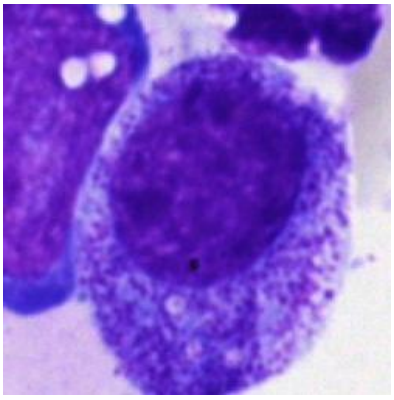
Proerythroblast (PEB)



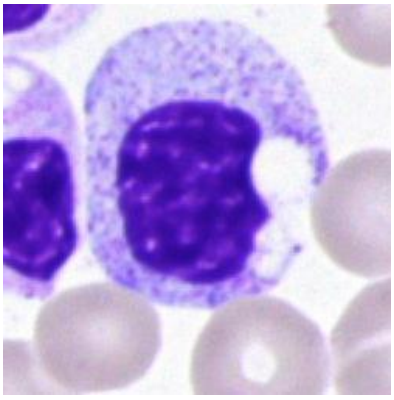
Erythroblast (EBO)



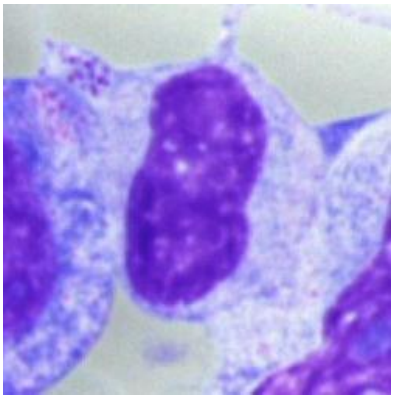
Blast (BLA)



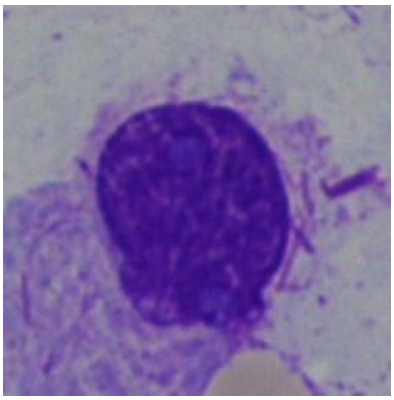
Promyelocyte (PMO)



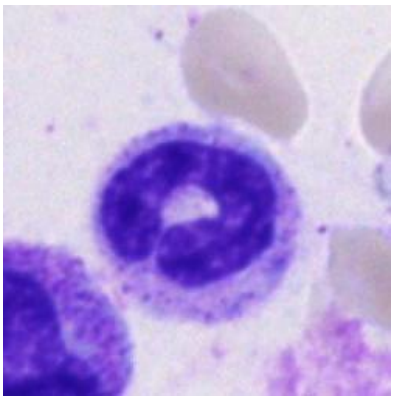
Myelocyte (MYB)



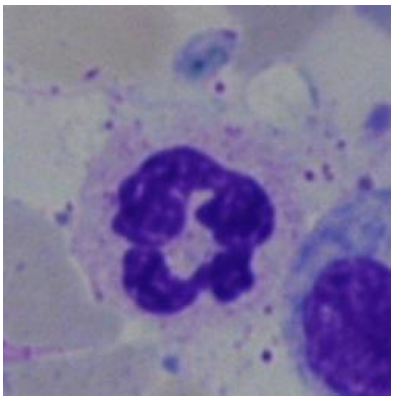
Metamyelocyte (MMZ)



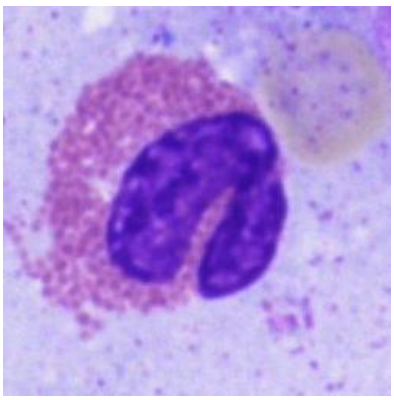
Faggot Cell (FGC)



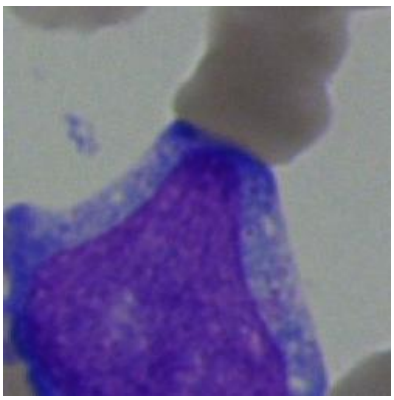
Band Neutrophil (NGB)



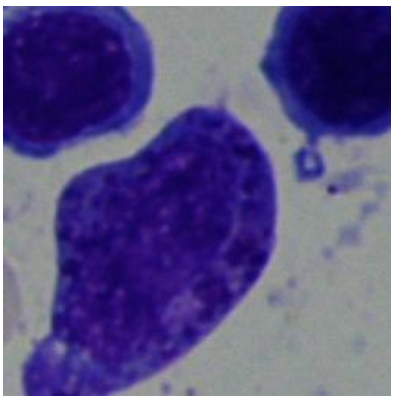
Segmented Neutrophil (NGS)



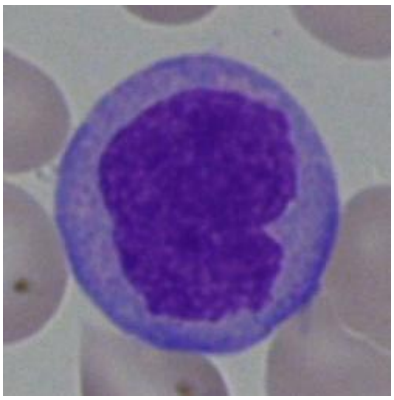
Eosinophil (EOS)



Basophil (BAS)

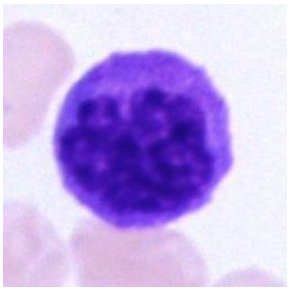


Abnormal Eosinophil (ABE)

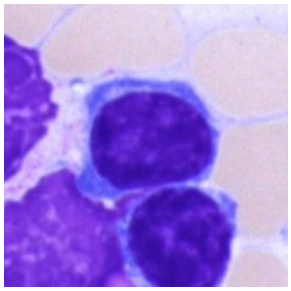


Monocyte (MON)

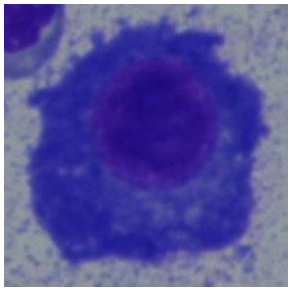
Lymphopoiesis:



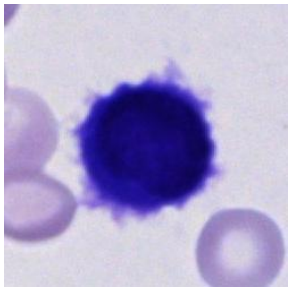
Immature Lymphocyte (LYI)



Lymphocyte (LYT)

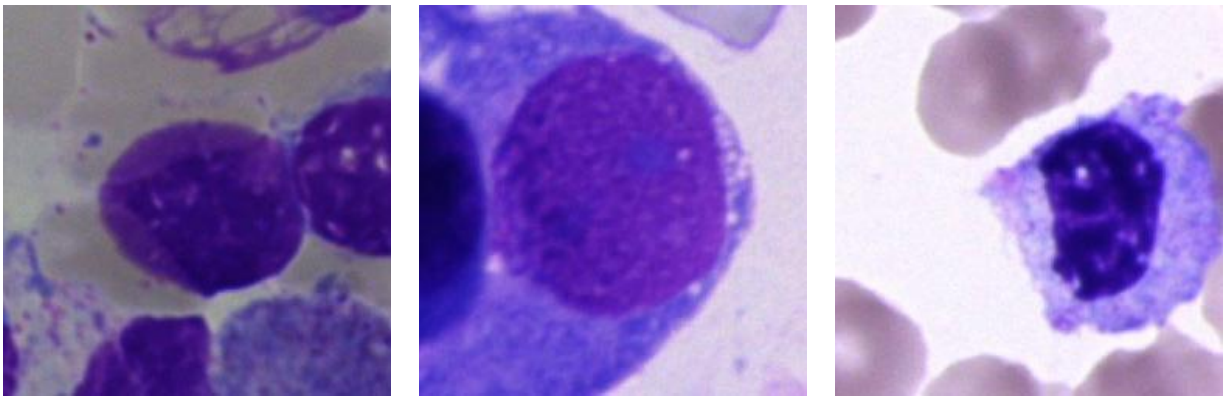


Plasma Cell (PLM)



Hairy Cell (HAC)

Other:



Artefact (ART) Other (OTH) Not Identifiable (NIF)

Figure 2.2.4: Figure showing an example image for each class of bone marrow cell. The cells are split into 3 main physiological classes, myelopoiesis, lymphopoiesis, and other (Tripathi et al.)

Different cell classes carry differing levels of importance for identifying different cancers. For example, blast cells are absent in a healthy individual's bloodstream and should constitute less than 5% of all cells in the bone marrow ("Updated on Blood Cancer"). A blast count of 20% or more is categorised as acute myeloid leukemia (AML) ("Updated on Blood Cancer"). Smudge cells are an important prognostic factor for chronic lymphocytic leukemia (CLL) (Higuchi et al.), while an excess in plasma cells can indicate multiple myeloma (National Cancer Institute, Plasma Cell Neoplasm). This highlights that although most cell types are normally present in samples, it is the excess of any cell type that can indicate cancer. This makes it difficult to assign an order of importance to the classes themselves. However, some extremely rare cell types do indicate cancer. Hairy cells are an example of a rare cell type as they represent 0.2% of the dataset but guarantee the presence of a leukemia known as hairy cell leukemia (National Cancer Institute, Hairy Cell Leukemia). Given that the highest proportion of patients in this dataset had multiple myeloma (Matek et al., Deep Neural, Networks), the performance of plasma cell (PLM) classification will be often referred to throughout the study.

3. Ethics

As this project uses real biological data from human subjects, there are ethical concerns regarding sensitive and private data. However, the dataset used in this project only contains images of single cells with labels referring to information about the cell type and nothing about the patient's private information (See *Chapter* 2.2). This means the data is completely anonymised and de-identified so that no information about the patient can be gained from these images, including being able to use the images to re-identify patients through an image or a combination of images. This research also has the potential benefit of expanding how AI can be used to help reduce the burden on hematologists through automatic cell identification. This can be a cause for ethical concern as these models will not give a 100% accurate prediction, leading to false negatives and false positives which can impact diagnosis outcomes and can therefore be considered as a cause for distress. However, the scope of this project is only to investigate machine learning techniques on the existing dataset from Chapter 2.2 which only contain cell images from patients who have already previously been confirmed to have some form of hematological disease. Moreover, no primary data will be considered and used within the model.

This work has received full ethical approval from the University of St Andrews.

4. Implementation

4.1 Technology

Python's Tensorflow and Keras libraries were employed to manage the dataset and construct the CNN models. The code was completed and executed within a GPU-accelerated Docker container that supported Tensorflow version 2.15.0. To train the models the University provided a GPU machine containing a GeForce RTX3060 to ensure training occurred within a reasonable time frame.

4.2 Pipeline

All images are initially checked whether they can be opened using OpenCV to catch and remove any potentially corrupt images. Afterwards, the images are split into two subsets: a train/validate set and a test set using a 70/30 split. This test set is used as “held-out” images to evaluate the models implemented on completely unseen data at the end of the investigation. The train/validation set is further split into two separate train and validation sets using an 80/20 split. The images on the training set will contain the images used by the models to learn the patterns and relationships whilst the validation set is used to measure the model’s training performance on unseen data. As there are 21 classes, each containing a differing number of images, it is important to ensure that the same proportion of images are taken from each of the classes. This makes the distribution of all images within each set similar to the whole dataset. For training, images are then imported using TensorFlow’s using tf.keras.utils.image_dataset_from directory to ensure efficient files are not stored in memory when training. An initial batch size[2] of 32 with shuffling was chosen such that the batches contained images from a variety of classes. Overall this creates 3,000 batches for the subset of images. As mentioned in earlier sections, images are a 3D matrix of values that range from 0 to 256 to describe the RGB values of a given pixel. This is normalised to values between 0 and 1 as this can help the model converge in a faster and more stable manner.

[2] Batch size refers to the number of training samples utilised in one iteration of training a neural network.

4.3 Performance Metrics

To evaluate the effectiveness and reliability of all methodologies used in this project, it is essential to employ appropriate performance estimation metrics. This section presents an explanation of key metrics used to assess the performance of this minimum viable model and all future models used. This includes accuracy, precision, recall, F1-score, and balanced accuracy.

4.3.1 Accuracy

Accuracy is a metric used to measure the correctness of the model's predictions. It represents the proportion of correctly classified instances about the total number of instances. Mathematically, accuracy can be expressed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$ (Dalianis 48)

where TP (True Positives) refers to the number of correctly predicted positive instances, TN (True Negatives) refers to the number of correctly predicted negative instances, FP (False Positives) refers to the number of incorrectly predicted positive instances and FN (False Negatives) refers to the number of incorrectly predicted negative instances.

4.3.2 Precision

Precision provides insights into the model's ability to correctly identify positive instances among the instances it predicted as positive. It is defined as the ratio of true positive predictions to the total number of positive predictions, and it can be calculated as:

$$Precision = \frac{TP}{TP + FP}$$ (Dalianis 47)

A high precision score indicates a low rate of false positives, which implies that the model has a low tendency to label negative instances as positive.

4.3.3 Recall

Recall measures the model's ability to identify all positive instances correctly. It represents the ratio of true positive predictions to the total number of actual positive instances, and it can be computed as:

$$Recall = \frac{TP}{TP + FN}$$ (Dalianis 47)

A high recall score indicates that the model has a low rate of false negatives, meaning it can effectively identify positive instances.

4.3.4 F1-Score

The F1-score is the harmonic mean of precision and recall and provides a balanced measure of a model's performance. Because it combines both precision and recall into a single metric, it can be useful if a certain class dominates the dataset. The F1-score is calculated as:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$ (Dalianis 47)

The F1-score ranges from 0 to 1 or as a percentage from 0 to 100%. A value of 1 or 100% indicates perfect precision and recall.

4.3.5 Balanced Accuracy

While accuracy calculates the overall correctness of predictions, balanced accuracy is a metric commonly used in machine learning to assess the performance of a classifier, especially when dealing with imbalanced datasets where one class may dominate the others. An example of this is in a binary classification problem with classes A and B. If A has 99 samples and B has 1 sample, if the model were to predict A 100% of the time, the model would have an accuracy of 99%. This is misleading as this high accuracy score does not take into account that the model has learnt nothing about the minority class B. This is where balanced accuracy is useful.

Balanced accuracy considers the recall and specificity of all classes separately and averages them. This is equivalent to averaging the recall scores for each class (Grandini et al., 4). This metric provides a fairer assessment of classifier performance across all classes, making it particularly useful in scenarios where class distribution is uneven. A balanced accuracy score close to 1 (or 100%) indicates excellent performance, while a score near 0.5 (or 50%) suggests random guessing in a binary classification problem. As there are 21 classes in this dataset, a balanced accuracy of around 0.05 (or 5%) would suggest random guessing.

$$Balanced\ Accuracy = \frac{Recall + Specificity}{2}$$ (Akosa 4)

$$Specificity = \frac{TN}{TN+FP}$$ (Akosa 4)

4.5 Minimum Viable Analysis Model

The main minimum viable analysis model is a sequential CNN proposed by Matek et al. (See *Figure 4.5.1*). In Matek et al.'s paper, this model was used as a comparison to the ResNeXt-50 architecture. Although not state-of-the-art, they showed that it performed similarly to ResNeXt-50 using the same dataset. However, this model did not run into out-of-memory issues on the University-provided GPU machines like some of the state-of-the-art architectures like ResNeXt, ResNet explored in the literature review. This model also showed better results when compared to Krappe et al.'s hierarchical tree classification approach using feature extraction, making it a good choice for a minimum viable analysis model.

Other than the number of epochs (13), no other hyperparameters were provided. Therefore, the initial learning rate was set to 0.001 as this is a good default learning policy for most problems (Wu et al.). The ADAM optimiser was also used due to its popularity amongst all other optimisers and its efficiency (Raitoharju 59). Overall, the model was trained on 50 epochs to further understand the behaviour. In total this model took 44 minutes to train.

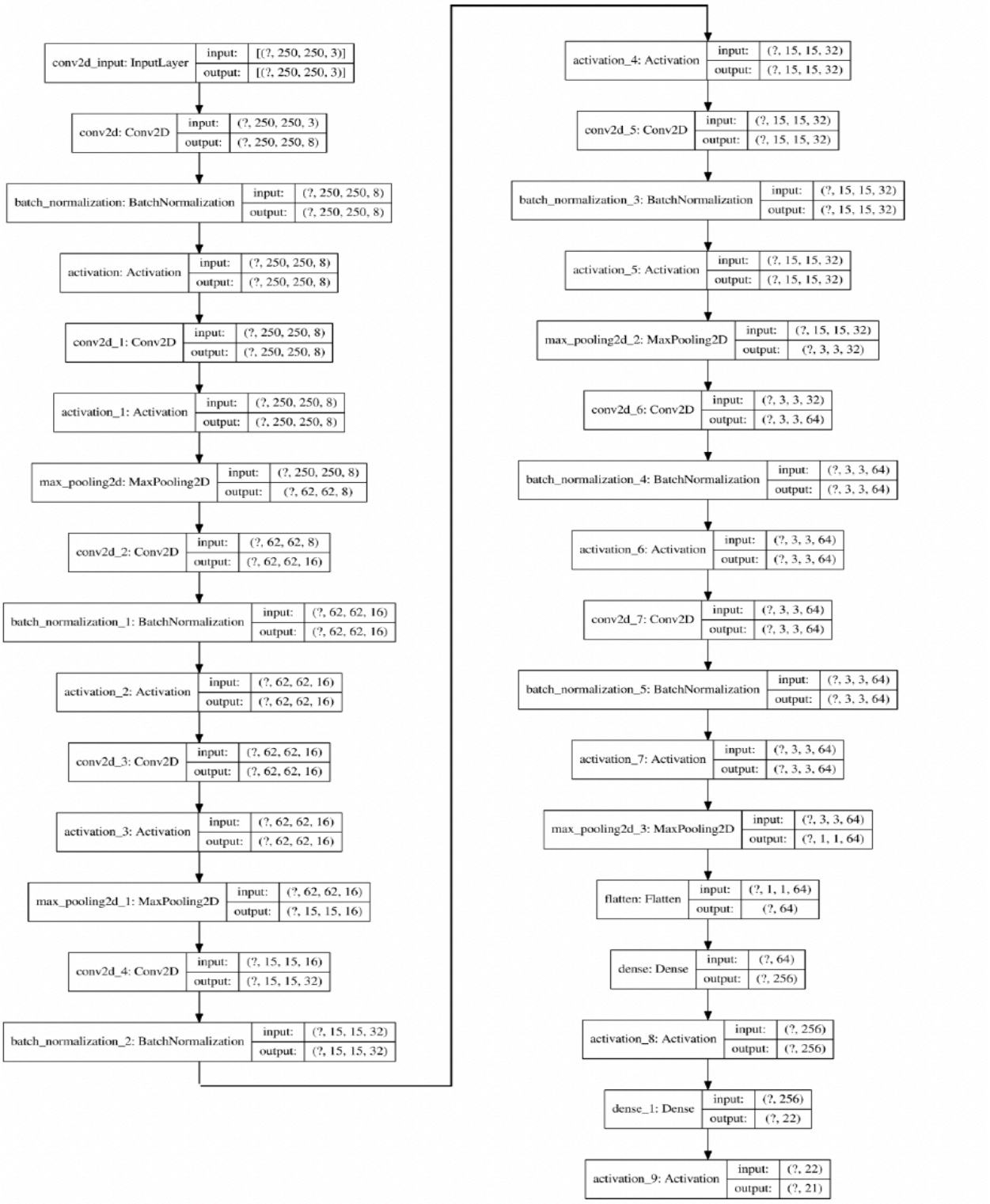


Figure 4.5.1: Structure of the sequential model used in this work for comparison with the results obtained using the ResNeXt architecture (Matek et al., Deep Neural Networks)

From *Figure 4.5.2,* the model achieves a high accuracy in the training data over 50 epochs. However, after just 5 epochs, the validation accuracy plateaus at around 73% to 76% with 74.0% being the final validation accuracy at the end of 50 epochs. *Figure 4.5.3* shows the training and validation loss of the minimum viable analysis model over each epoch. The loss represents the error between the actual target values and the predictions made by the model on the training data and validation data. With a steady validation loss increasing after 5 epochs, this might indicate that the model could be overfitting to the training data or that the initial learning rate is too high. This learning rate hyperparameter determines the size of the steps an algorithm known as the optimiser takes while updating the parameters of the model during training to converge to a minimum point on the loss function. This point indicates the optimal performance of a model. From both *Figure 5.5.2* and *5.5.3*. The value of 0.001 used in this model led to fast convergence but could indicate the model is unable to converge around the minimum point.

In terms of precision, recall, and F1-score, the model achieved a 74% for all metrics. Balanced accuracy for this model was 44%. From *Figure 4.5.4* and *Figure 4.5.5*, classes that hold a significant proportion of images within the dataset can be correctly classified a large majority of the time using this model, with the most successful classification being EBO which is correctly identified 90.7% of the time. However, for minority classes such as FGC, HAC, OTH, and LYI with less than 0.3% of images within the dataset, the model rarely predicts these classes and instead chooses to classify these cells in one of the majority classes. ART, BLA, and PMO are among the most popular. This contributes to the low balanced accuracy score. For other classes with a significant proportion of images such as MMZ, MON, MYB, NGS, and PLM with around 1.5-6% of images in the dataset, the model can predict these images a large proportion of the time but overall, this shows that this model is far from optimal and improvements could be made in terms of generalisability. A summary of the results can be seen in *Table 4.5.6.*

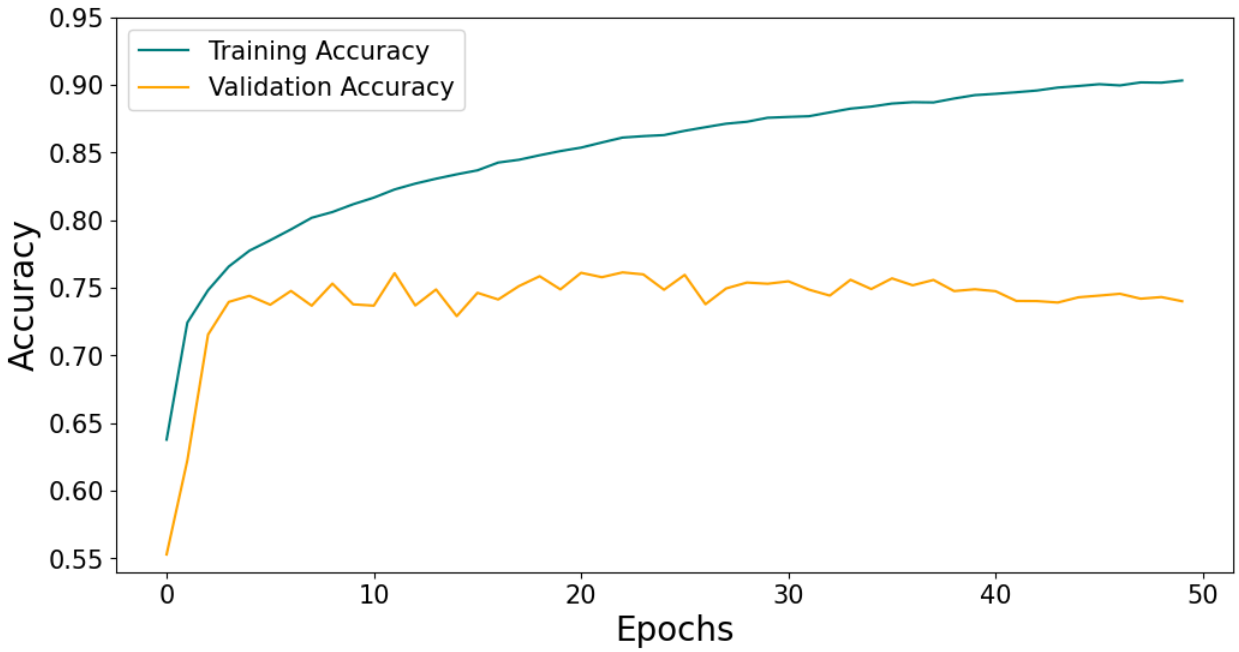


Figure 4.5.2: Training and validation accuracy for the main minimum viable analysis model

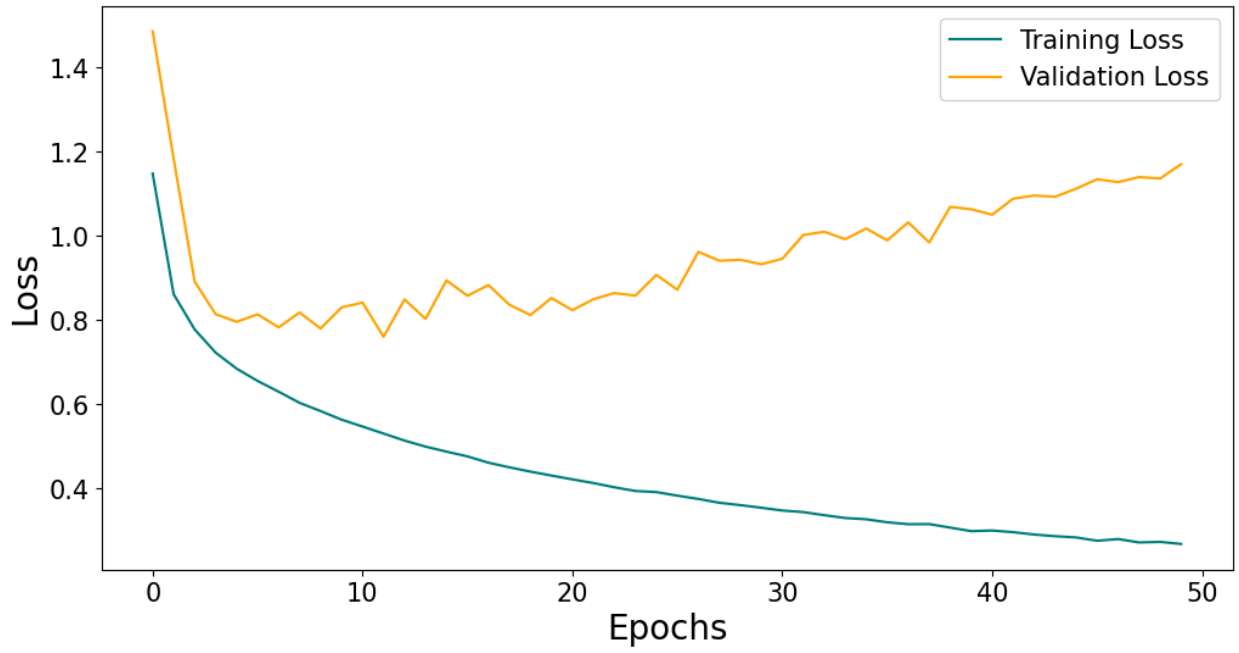


Figure 4.5.3: Training and validation loss for main minimum viable analysis model

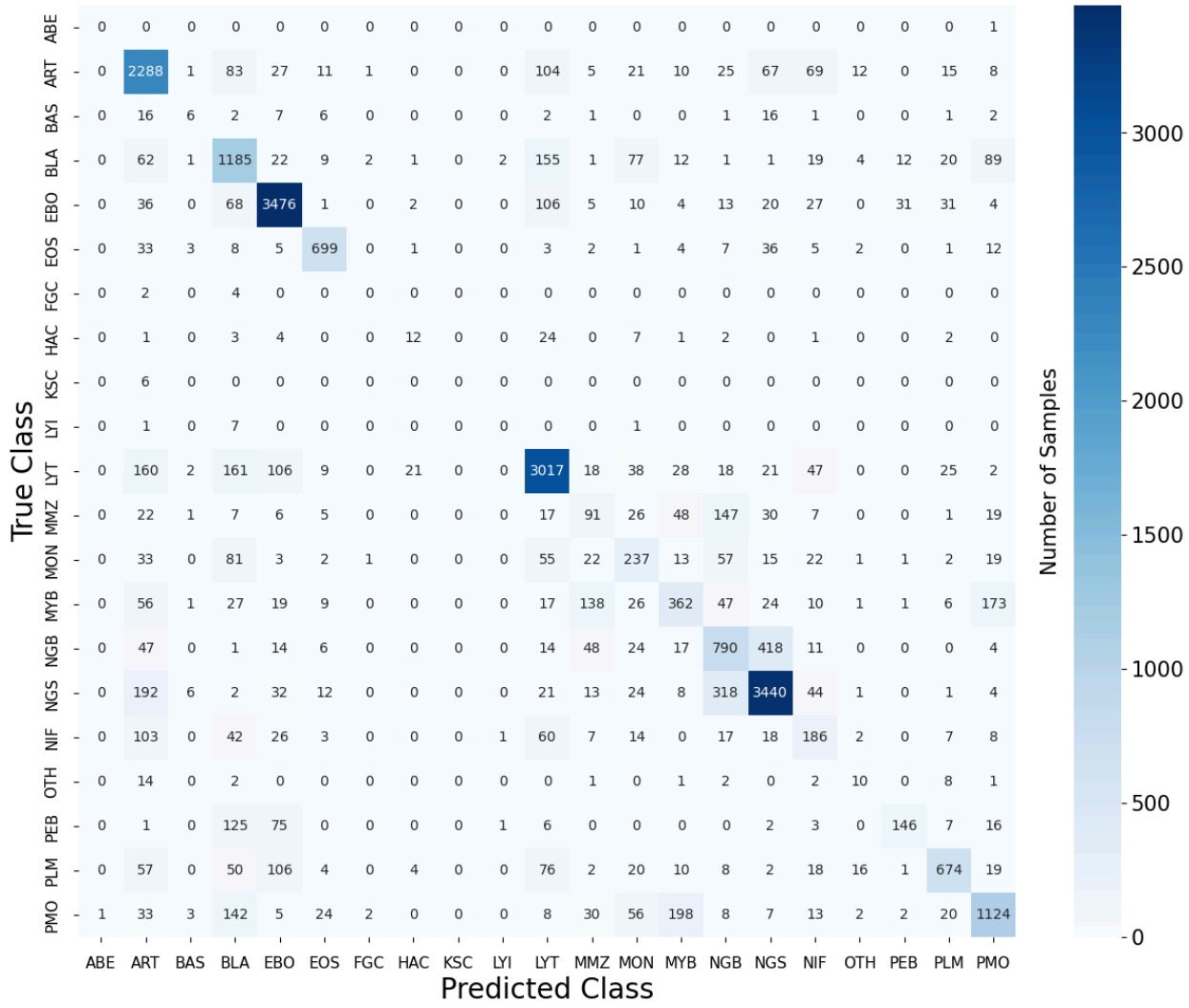


Figure 4.5.4: Confusion matrix for the main minimum viable analysis model on the validation set

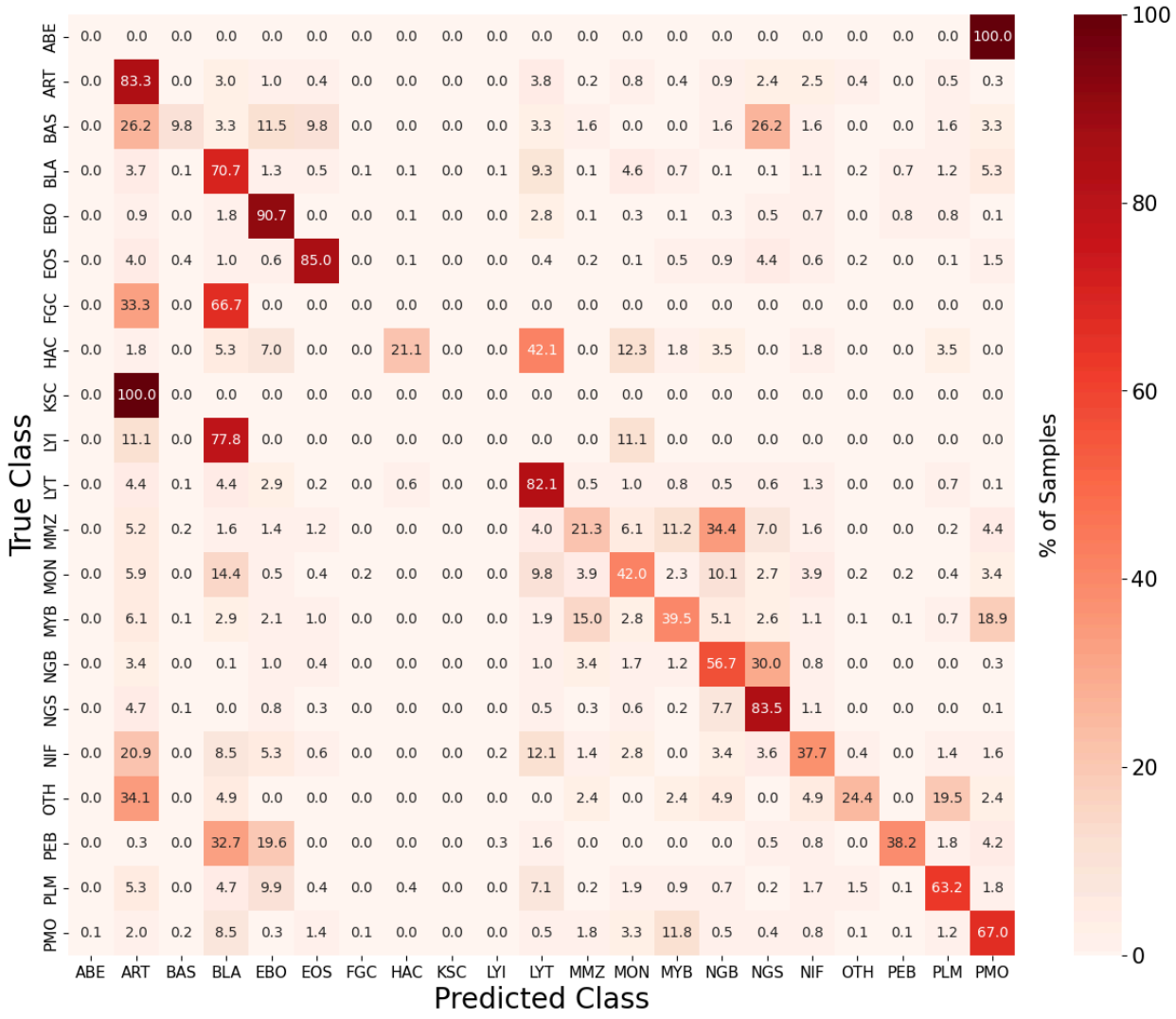


Figure 4.5.5: Normalised confusion matrix showing the distribution of predictions for each class for the main minimum viable analysis model on the validation set

Class	Precision (%)	Recall (%)	F1-Score (%)	Support
ABE	0	0	0	1
ART	72	83	77	2747
BAS	25	10	14	61
BLA	59	71	64	1675
EBO	88	91	90	3834
EOS	87	85	86	822
FGC	0	0	0	6
HAC	29	21	24	57
KSC	0	0	0	6
LYI	0	0	0	9
LYT	82	82	82	3673
MMZ	24	21	22	427
MON	41	42	41	564
MYB	51	39	44	917
NGB	54	57	55	1394
NGS	84	84	84	4118
NIF	38	38	38	494
OTH	20	24	22	41
PEB	75	38	51	382
PLM	82	63	71	1067
PMO	75	67	71	1678

Table 4.5.6: Classification report showing precision, recall, and F1-score for all classes using the validation set images

4.6 Hyperparameter Tuning

4.6.1 Hyperparameter Tuning Setup

As mentioned in Chapter 2.1, expertise is essential for choosing the right set of hyperparameters manually, thus various hyperparameter tuning methods help facilitate the process of choosing a more optimal set of hyperparameters for a given problem. One popular technique is grid search. When providing grid search a mapping of hyperparameters to a set of values, every combination of hyperparameters and their values are used to train a model to find the best-performing model. The advantage of this type of search method is that it is exhaustive, meaning all hyperparameter value combinations are trialled to find the optimal combination (Bergestra and Bengio 282). However, a disadvantage to this type of search is that it is computationally expensive. Adding more hyperparameter values causes the total number of training instances to multiply by the number of values for any given hyperparameter. Moreover, when performing a grid search, it might become apparent that certain hyperparameters do not contribute as much to improving a model, further wasting computational time. For a problem that uses a large dataset, this is particularly inefficient (Bergestra and Bengio 291). An alternative method is random search. As the name suggests, it is an algorithm that randomly selects a set of hyperparameters and values within the same mapping as grid search. The advantage of this method is that not every configuration has to be tested, and thus, will be computationally faster than grid search. Moreover, despite not testing every combination, for any finite number larger than M, random search can find the top 5% of hyperparameters 95% of the time in 60 trials (Zheng). A study by Bergestra and Bengio reveals that in a 32-dimensional configuration space, random search performed statistically equal on four of seven datasets whilst even performing better in one dataset when compared to both grid search and manual search (Bergestra and Bengio 281). As the dataset is large for a medical image classification task, random search was selected to perform hyperparameter tuning for this project.

As observed in Chapter 4.5, the model mainly struggled with overfitting and generalising. Below is a table of hyperparameters and values chosen, along with a justification on why they were chosen in an attempt to address these issues.

Hyperparameter	Values	Justification
Learning Rate	0.001, 0.0001, 0.00001	The minimum viable analysis model with a 0.001 learning rate was unable to improve validation accuracy after 5 epochs, resulting in an increasing validation loss. This may indicate that this learning rate is too large because whilst the model was able to escape spurious local minima, the convergence in 5 epochs may indicate an oscillation between a local minimum (You et al. 1). Reducing the learning rate could lead to improved accuracy since it allows the optimiser to navigate towards a minimum point more smoothly, potentially resulting in better convergence (You et al. 1).
Dropout Rate	0.2, 0.3, 0.4, 0.5	In the context of CNNs, a dropout layer is a layer that randomly drops a percentage of neurons and their connections (Baldi and Sadowski 1). As a consequence this means that the entire network cannot rely on any one neuron as other neurons might be dropped out, forcing it to learn more generalisable and informative features as opposed to noise (Baldi and Sadowski 1). It is widely known that dropout is known to work well in fully-connected layers with typical values ranging from 0.2-0.5 (Wu and Gu; Srivastava et al 1953).
Batch Size	32, 64, 128	A common practice is to decay the learning rate to achieve better convergence but Smith et al. show that an alternative to this is increasing the batch size instead. This is because providing more samples per iteration can lead to a better gradient estimation, which in turn can compensate for the lack of learning rate decay (Smith et al. 1). Moreover, increasing the batch size can act as a form of implicit regularisation (Kandell and Castelli 313), leading to models that generalise better to unseen data. In testing, I have tried experimenting with larger batch sizes but in some instances ran into out-of-memory issues. Therefore as such, I have decided to limit batch sizes to 128 whilst also introducing the different learning rates.
Optimiser	Adam, SGD	In the baseline model, the ADAM optimiser was used as it is both computationally efficient (due to its ability to adjust the learning rate during training) and the most commonly used optimiser for CNNs (Raitoharju 59). However, SGD can generalise better and thus may be more suitable for this type of problem (Zhou et al. 1). It was found that the learning rate with 0.00001 rarely converged after 200 epochs for the SGD optimiser. Thus to save computation time, a learning rate of 0.00001 was replaced with 0.0001.

Table 4.6.1.1: Table showing the hyperparameters and values used in random search along with a justification

As a scoring metric for the random search, accuracy was chosen as the metric to maximise despite not reflecting well on imbalanced datasets like the one used in this study. However, as explained in Chapter 2.2, the presence of most cell types does not indicate that a type of malignancy is present. Instead, it is the overaccumulation of these cells present in the bloodstream that leads medical practitioners to diagnose cancer. Moreover, the dataset itself is more reflective of the frequency of cells that occur in the real world. Therefore, these factors lead me to believe that having better accuracy is more important than having better-balanced accuracy.

To save computation efforts, early stopping was also used. This is a type of regularisation commonly used to avoid overfitting when dealing with iterative optimisation procedures (Lauriola). In the context of this hyperparameter investigation, it checks whether a model's validation loss has decreased after 5 epochs. If it has not, it is likely that the model has stopped learning the general patterns and began to memorise the training data.

4.6.2 Hyperparameter Tuning Results

After 60 iterations of random search spanning over 6 days, 5 hours, and 9 minutes, the following optimal hyperparameters were determined for the model:

- Dropout rate: 0.4
- Learning rate: 0.001
- Optimizer: Adam
- Batch size: 128

Using these hyperparameters, the model achieved a validation accuracy score of 77.6%. When refitting the model, it achieved a validation score of 76.0% with a loss score of 0.8184 after 17 epochs utilising a 5 epoch patience (See *Figure 4.6.2.1* and *Figure 4.6.2.2*). This represents a 2.0% accuracy improvement over the minimum viable analysis model's validation accuracy of 74.0%. Precision, recall, and F1-score also saw an increase from 74% each to 76%, 76%, and 75% respectively. Balanced accuracy however decreased from 44% to 43%. Overall, the hyperparameter tuning process did prove successful in optimising the model's performance in most metrics.

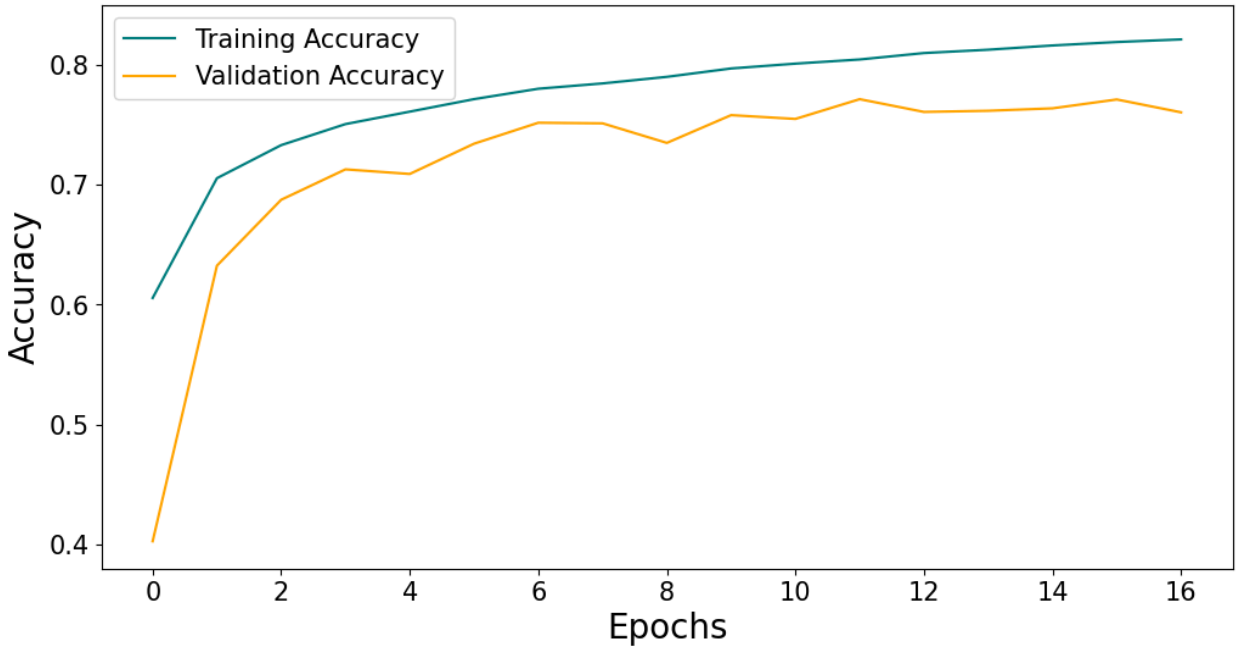


Figure 4.6.2.1: Training and validation accuracy for the optimised model

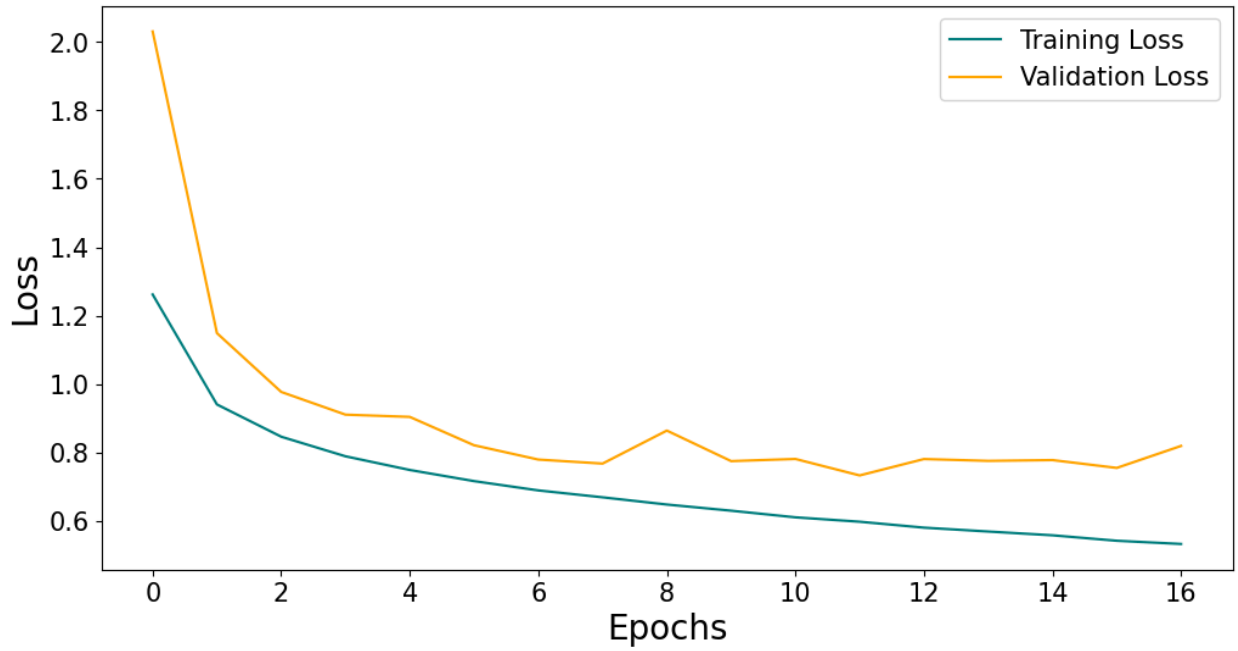


Figure 4.6.2.2: Training and validation loss for the optimised model

By looking at the confusion matrices in *Figure 4.6.2.3* and *Figure 4.6.2.4*, we can see a major improvement in accuracy for the PLM class. 889 samples are accurately predicted as PLM compared to the baseline 674 samples. This represents an increase in recall from 63.2% to 83.3% for that class. Other classes such as NGB and PMO also presented significant improvements in accuracy from 56.7% and 67.0% to 69.8% and 77.8% respectively. However, minority classes that make up less than 0.1% of the dataset such as ABE, BAS, FGC, KSC, and LYI are correctly assigned 0% of the time. A breakdown of class-level precision, recall, and F1-score can be seen in *Table 4.6.2.5*.

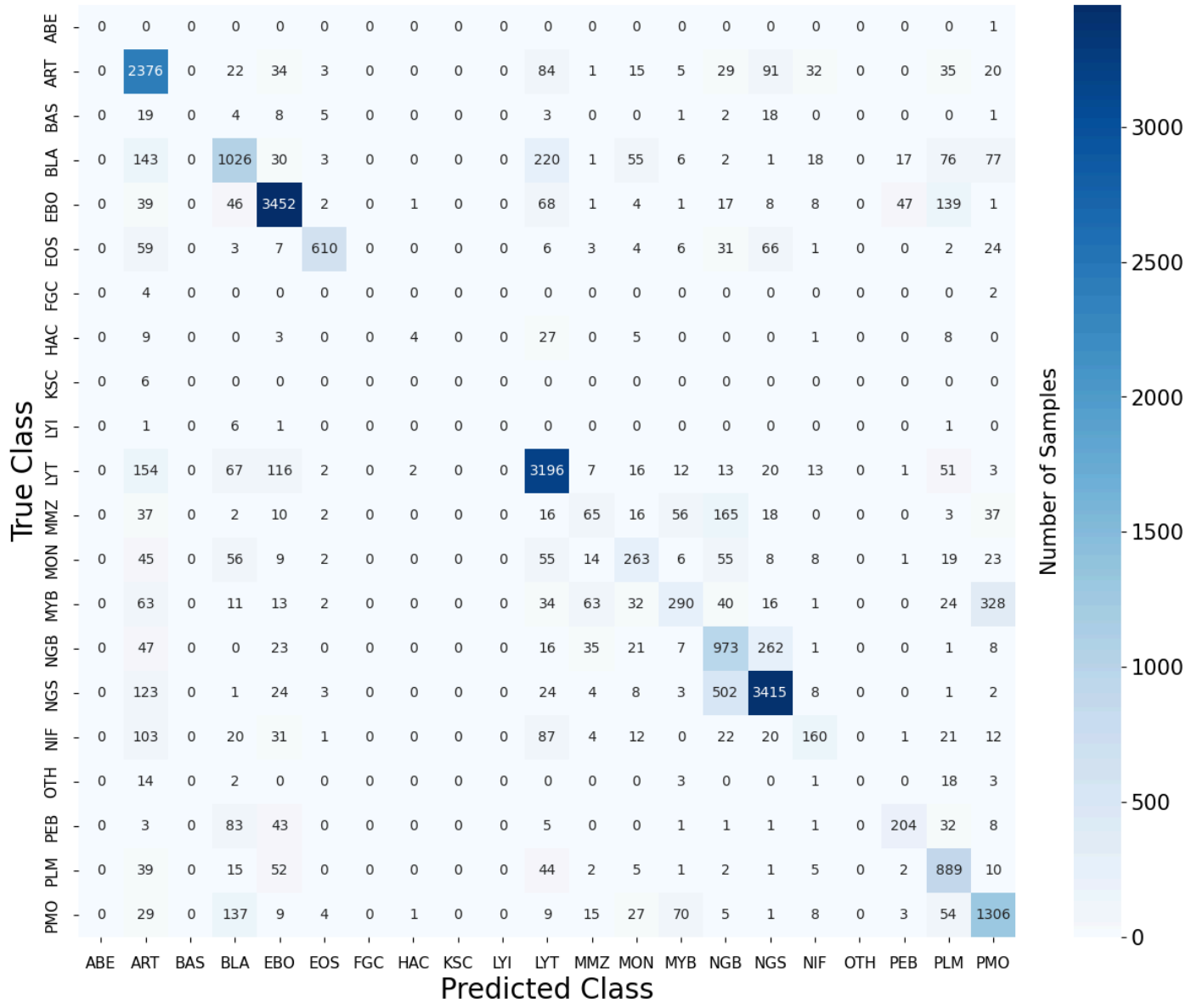


Figure 4.6.2.3: Confusion matrix for the optimised model on the validation set

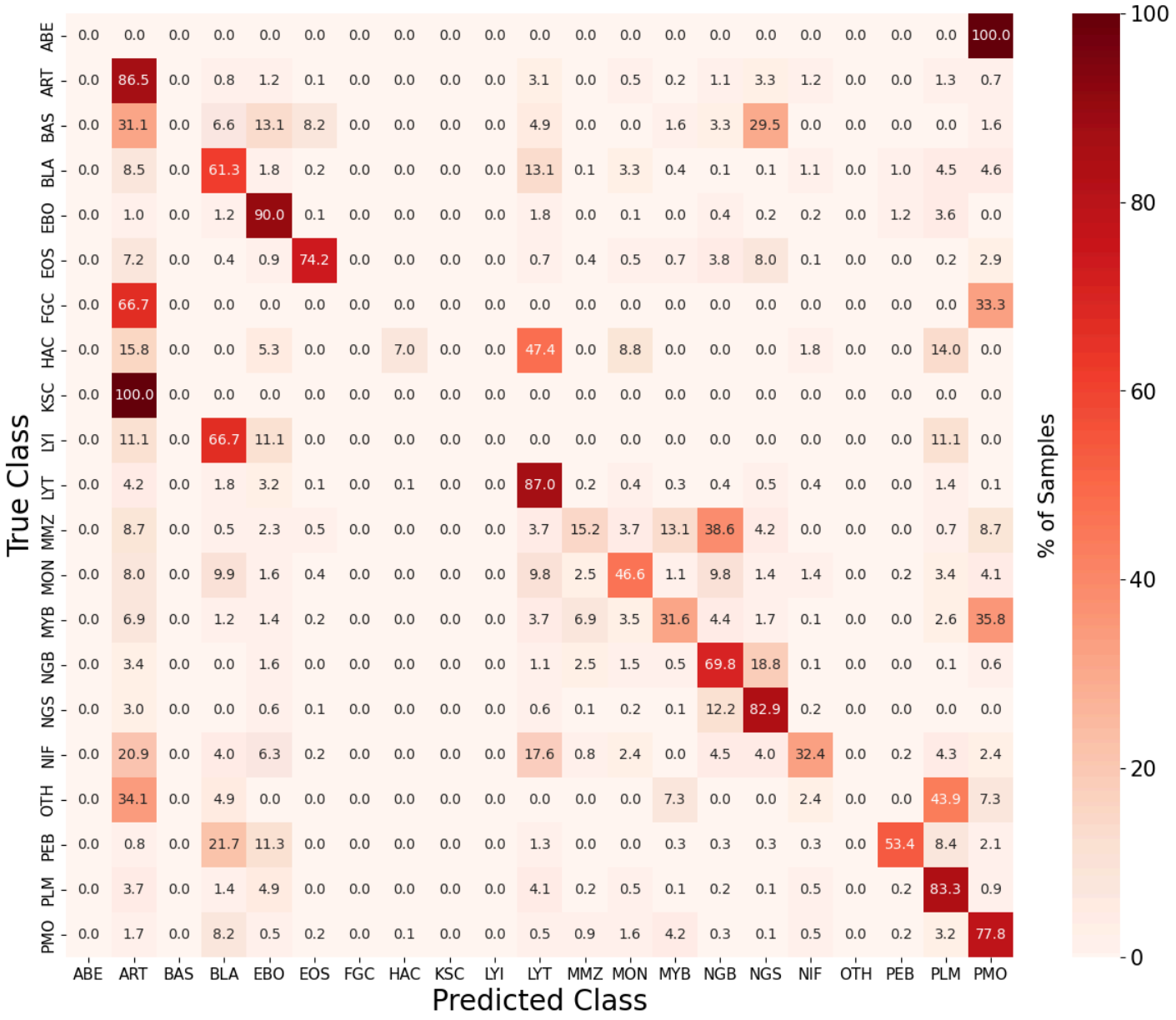


Figure 4.6.2.4: Normalised confusion matrix for the optimised model on the validation set

Class	Precision (%)	Recall (%)	F1-Score (%)	Support
ABE	**0**	**0**	**0**	1
ART	**72**	**86**	**78**	2747
BAS	0	0	0	61
BLA	**68**	61	**65**	1675
EBO	**89**	90	**90**	3834
EOS	**95**	74	84	822
FGC	**0**	**0**	**0**	6
HAC	**50**	7	12	57
KSC	**0**	**0**	**0**	6
LYI	**0**	**0**	**0**	9
LYT	**82**	**87**	**84**	3673
MMZ	**30**	15	20	427
MON	**54**	**47**	**50**	564
MYB	**62**	32	42	917
NGB	52	**70**	**60**	1394
NGS	**87**	83	**85**	4118
NIF	**60**	32	**42**	494
OTH	0	0	0	41
PEB	74	**53**	**62**	382
PLM	65	**83**	**73**	1067
PMO	70	**78**	**74**	1678

Table 4.6.2.5: Classification report showing precision, recall, and F1-score for all classes using the validation set images for the optimised model. Numbers in bold represent an improvement or no change compared to the minimum viable analysis model.

4.7 Data Augmentation

4.7.1 Data Augmentation Method

Despite attempting to resolve generalisability by adding a dropout layer and increasing batch size, the previous sections highlight how imbalanced data can cause a bias towards predicting majority classes like ART. This limited exposure to minority class instances probably hinders the model's ability to learn discriminative features for these classes. Addressing these challenges through techniques like data augmentation - the process of transforming existing images to create new training samples - can mitigate bias whilst improving the model's ability to learn features (Rebuffi et al.).

As identified in Chapter 2, Matek et al. performed data augmentation on the same dataset but did not explain the extent this improved the model's performance. In this section, the same data augmentation method was performed except for stain transformation as the scope of the project only involves the dataset from Chapter 2.2. This means the staining for all test images is consistent. All classes were upsampled to contain roughly 25,000 images in the training data. This was achieved by randomly rotating the image by a random continuous angle in the range of 0° to 180°, vertically and horizontally flipping the image, shifting the image by 25% of the image weight and height, and shearing the image up to 20% of the image size. One aspect Matek et al. did not specify was the fill mode (how blank space was filled) when the images were shifted and sheared. In TensorFlow, fill mode is constant by default. This means missing pixels are filled using a fixed colour. For this project, reflecting the image felt the most appropriate. "Nearest" fill mode chooses the closest pixel value to the border and repeats them, creating strange lines within the image; "Wrap" fills the missing data by copying the values of the known points to the unknown points. However, this creates a hard visible border between 2 sections of the image; "Reflect[ing]" the image over the blank space creates an image that looks the most comparable to an image taken under a microscope (See *Figure 4.7.1.2* to *Figure 4.7.1.5* for a comparison). Reflecting also avoids adding random noisy pixels to the image that can interfere with training the model.

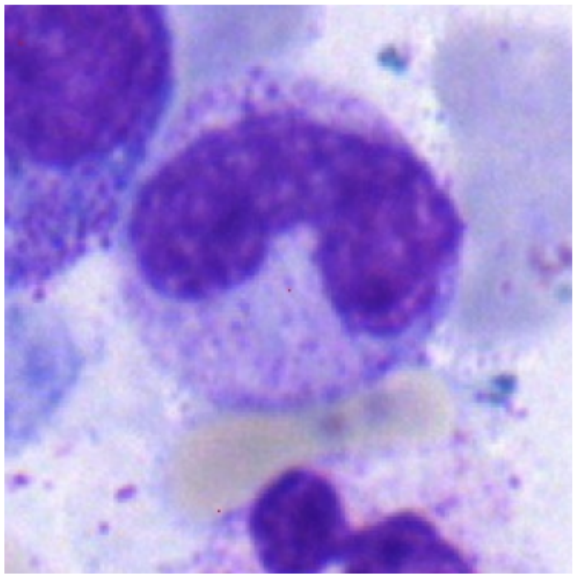


Figure 4.7.1.1: Image showing an NGB cell

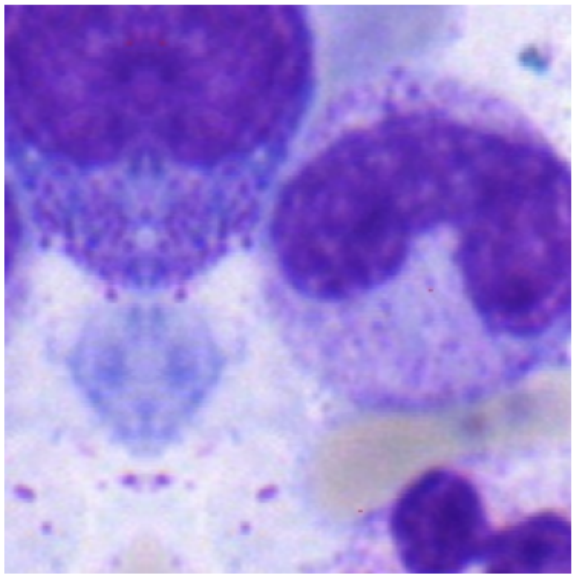


Figure 4.7.1.2: Image showing *Figure 4.7.1.1* shifted right by 25% using **reflect** as the fill mode

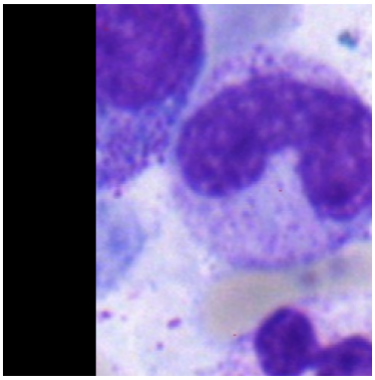


Figure 4.7.1.3: Image showing *Figure 4.7.1.1* shifted right by 25% using **constant** as the fill mode

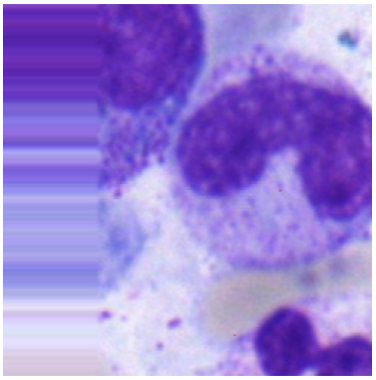


Figure 4.7.1.4: Image showing *Figure 4.7.1.1* shifted right by 25% using **nearest** as the fill mode

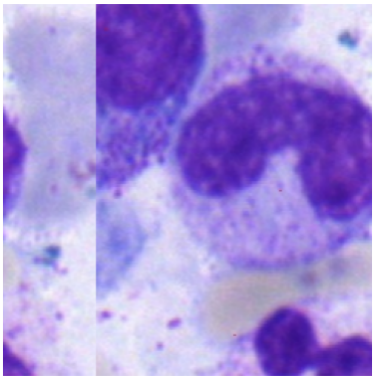


Figure 4.7.1.5: Image showing *Figure 4.7.1.1* shifted right by 25% using **wrap** as the fill mode

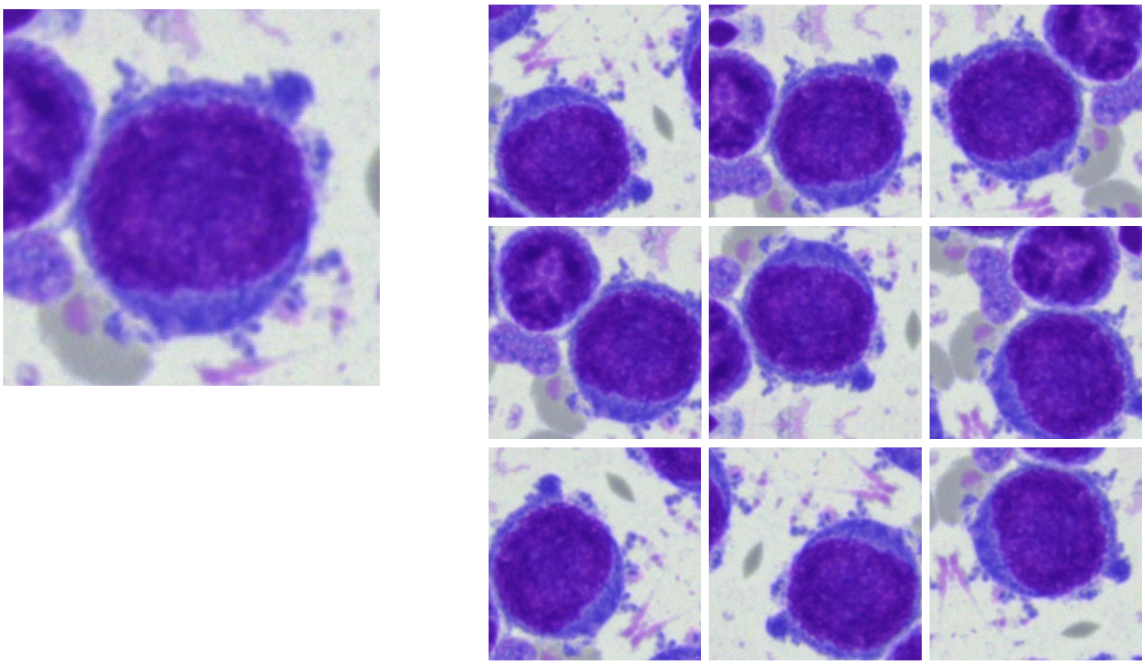


Figure 4.7.1.6: Original image of an LYI cell (Left) augmented using Matek's augmentation method through a combination of 0-180° rotations, vertical and/or horizontal shifts, image shifts of up to 25% for width and height, and sheers of up to 20% with the reflect fill mode to generate 9 example augmented images

Regarding fitting, the same hyperparameters as Chapter 4.6.2 were used except for early stopping's patience level being increased to 15 epochs. This was because artificially adding more images may also inadvertently introduce more noise to the dataset which, causing the model's validation accuracy to fluctuate more across epochs. The initial patience level of 5 was found to be too small to navigate through this fluctuation and thus was increased to 15.

4.7.2 Data Augmentation Results

After over 6 hours and 30 minutes of training, the model was able to reach a validation accuracy of 78.2% after 55 epochs, representing a 2.2% increase in accuracy from 76.0% using the tuned model without data augmentation. Precision, recall, and F-1 score also presented an increase from, 76%, 76%, and 75% to 80%, 78%, and 79% respectively. Balanced accuracy presented the largest increase from 43% to 61% as minority classes were being more accurately predicted (See *Figure 4.7.2.4*). This can be seen clearly through a higher concentration of predictions along the diagonal on the normalised confusion matrix in *Figure 4.7.2.5*.

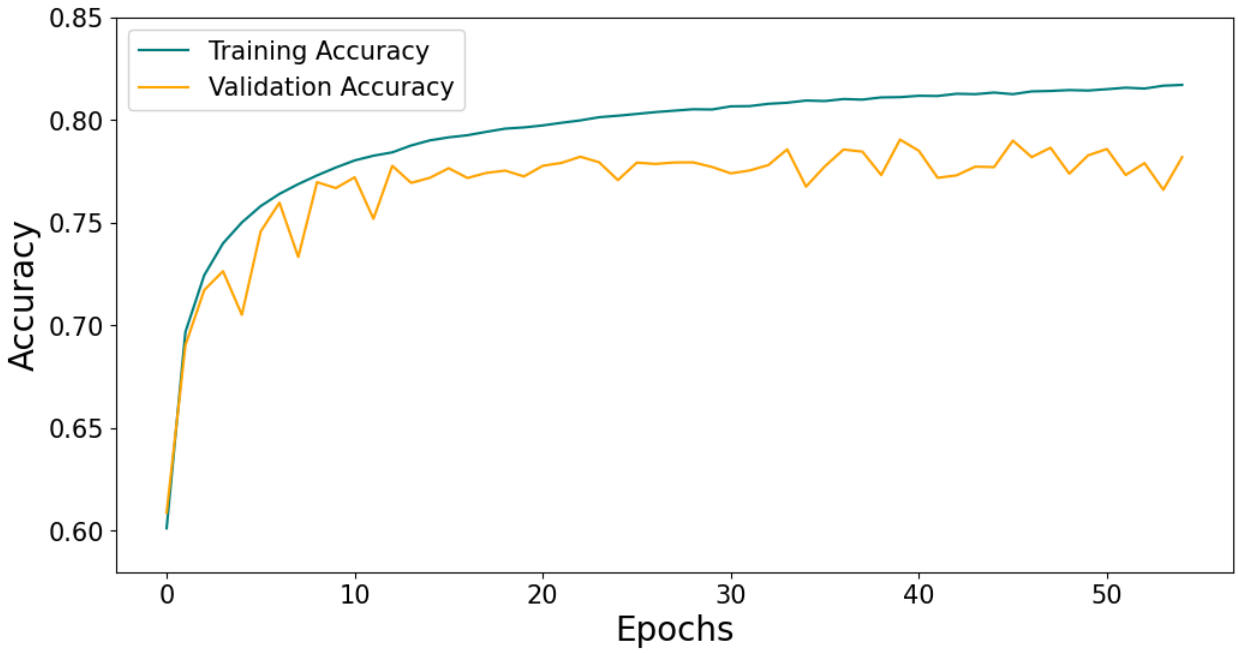


Figure 4.7.2.1: Training and validation accuracy for the model using data augmentation

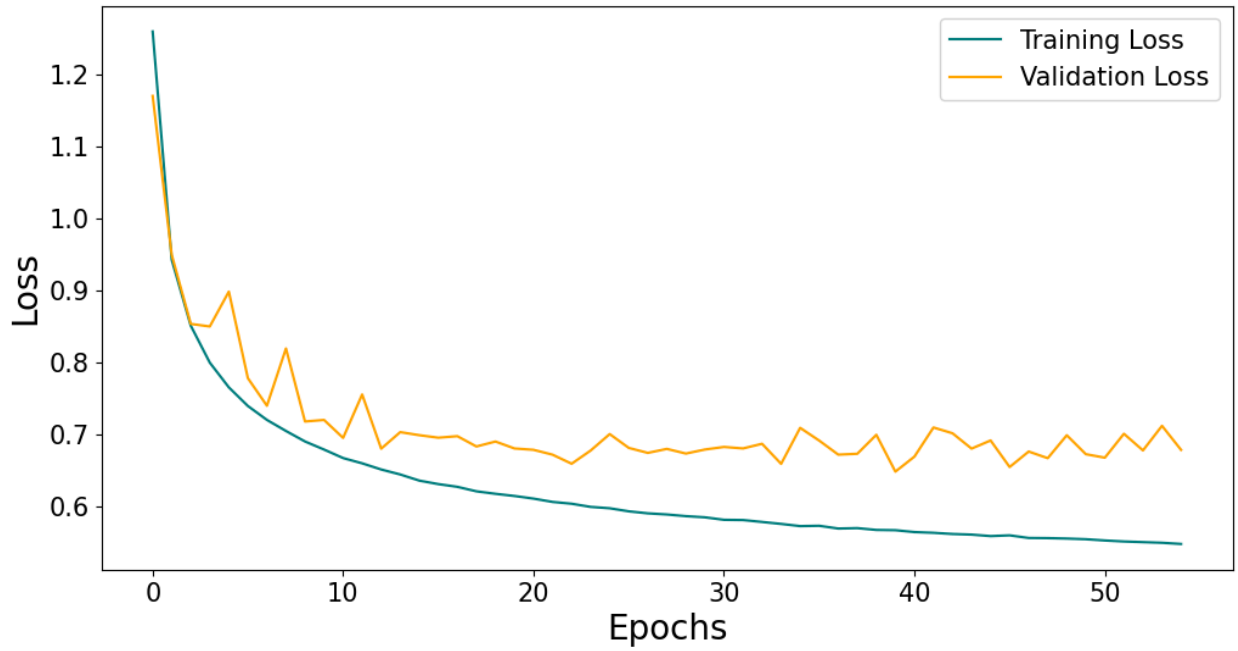


Figure 4.7.2.2: Training and validation loss for the model using data augmentation

Class	Precision (%)	Recall (%)	F1-Score (%)	Support
ABE	**0**	**0**	**0**	1
ART	**86**	78	**82**	2747
BAS	**20**	**61**	**30**	61
BLA	**75**	**70**	**72**	1675
EBO	88	**92**	**90**	3834
EOS	88	**93**	**90**	822
FGC	**14**	**50**	**21**	6
HAC	29	**58**	**39**	57
KSC	**13**	**33**	**19**	6
LYI	**5**	**11**	**7**	9
LYT	**87**	84	**86**	3673
MMZ	**34**	**51**	**41**	427
MON	49	**67**	**57**	564
MYB	59	**56**	**58**	917
NGB	**58**	66	**61**	1394
NGS	**90**	81	**85**	4118
NIF	53	**44**	**48**	494
OTH	**54**	**63**	**58**	41
PEB	61	**70**	**65**	382
PLM	**76**	**84**	**80**	1067
PMO	**83**	70	**76**	1678

Table 4.7.2.3: Precision, recall, F1-Score for the model using data augmentation. Numbers in bold represent an improvement or the exact same value as the tuned model.

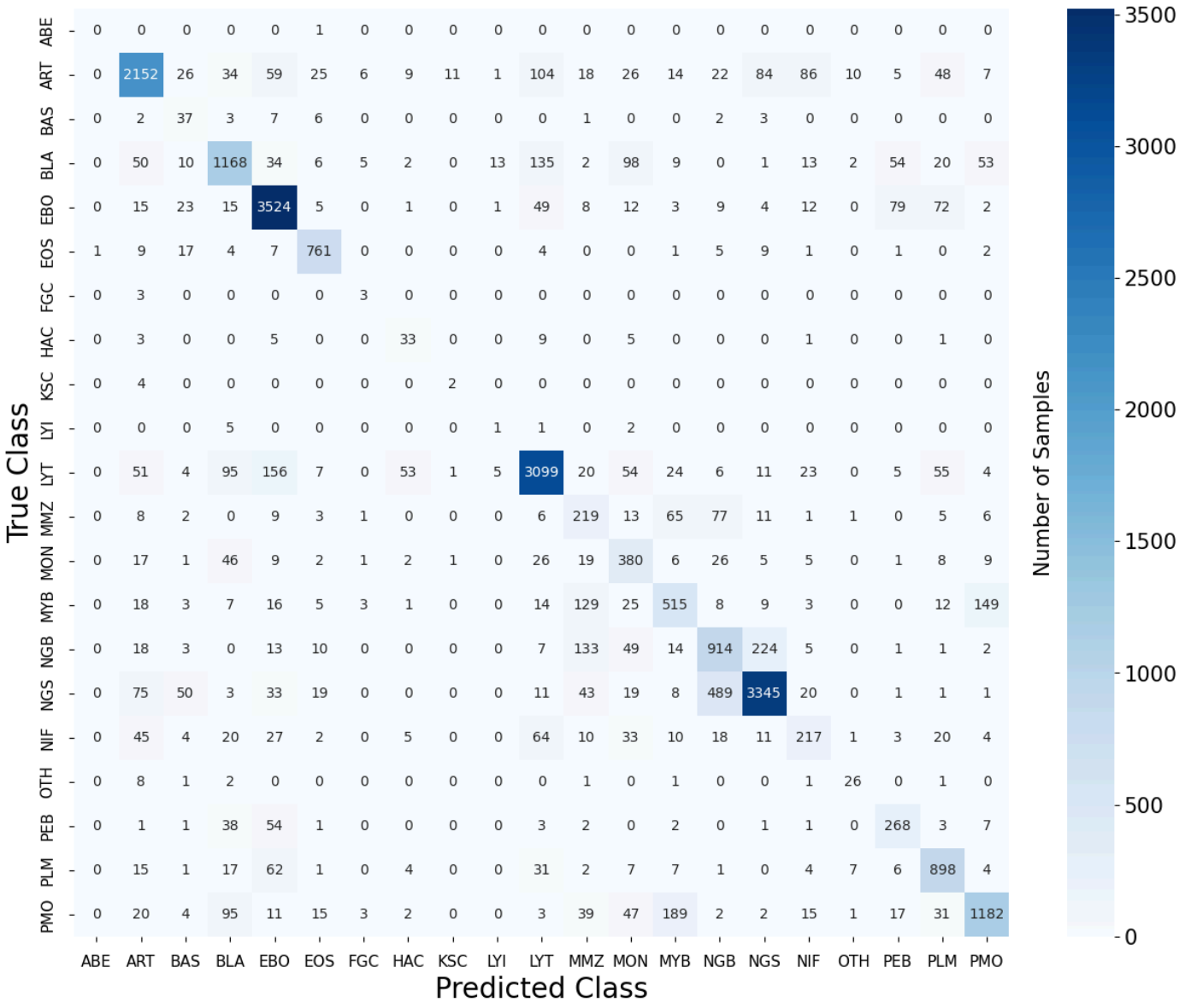


Figure 4.7.2.4: Confusion matrix for the optimised model with data augmentation on the validation set

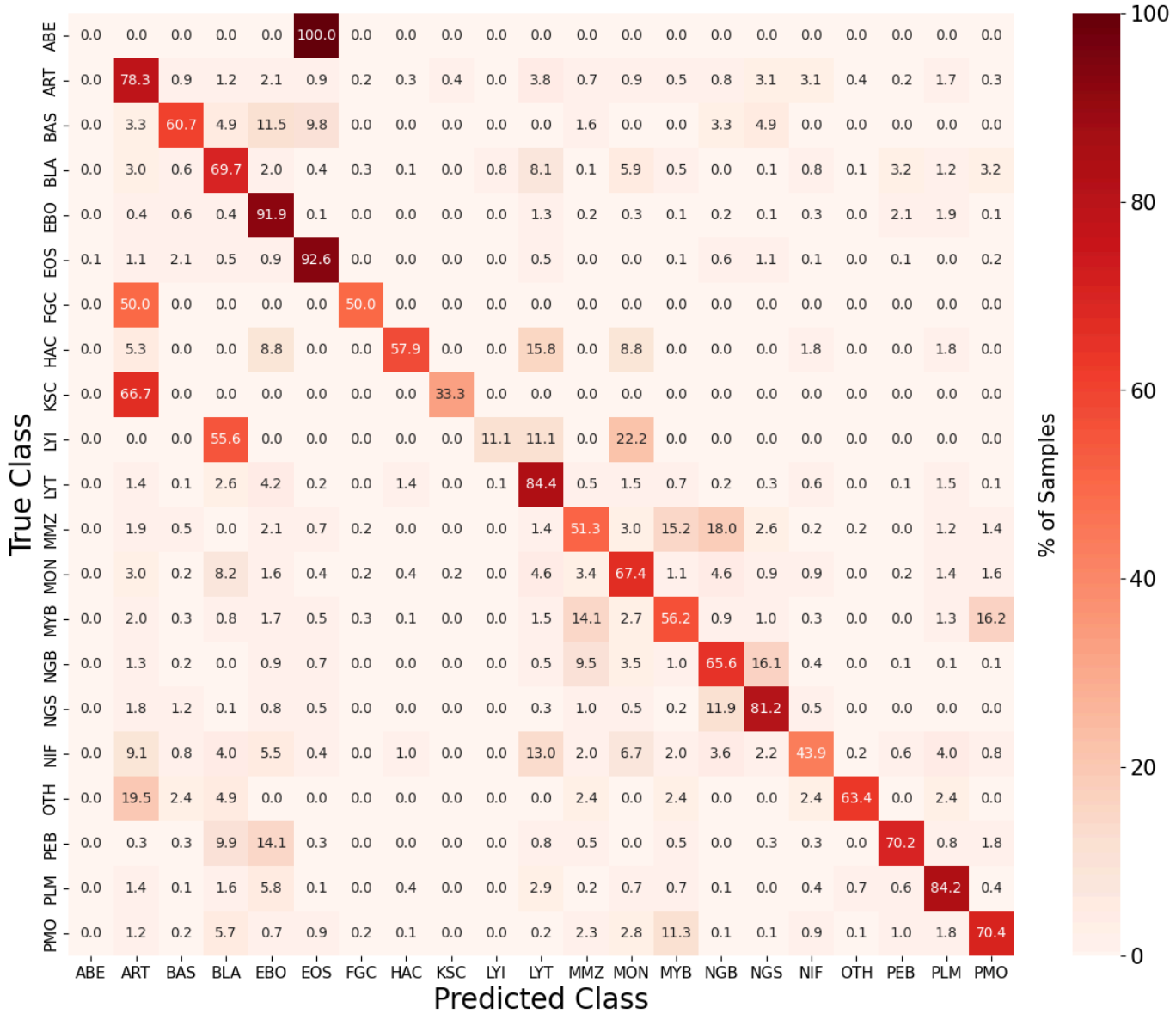


Figure 4.7.2.5: Normalised confusion matrix for the optimised model with data augmentation on the validation set

5. Test Set Results

The test subset was not utilised during the iterative process of improving the models in Chapter 4. It is kept separate and untouched as improvements were made on a validation set. The results presented now provide insight into the model's actual performance when applied to a dataset that it has not encountered or been trained on previously, offering a genuine assessment of its capabilities on unseen data.

The test set contains 51,404 images with the following distribution:

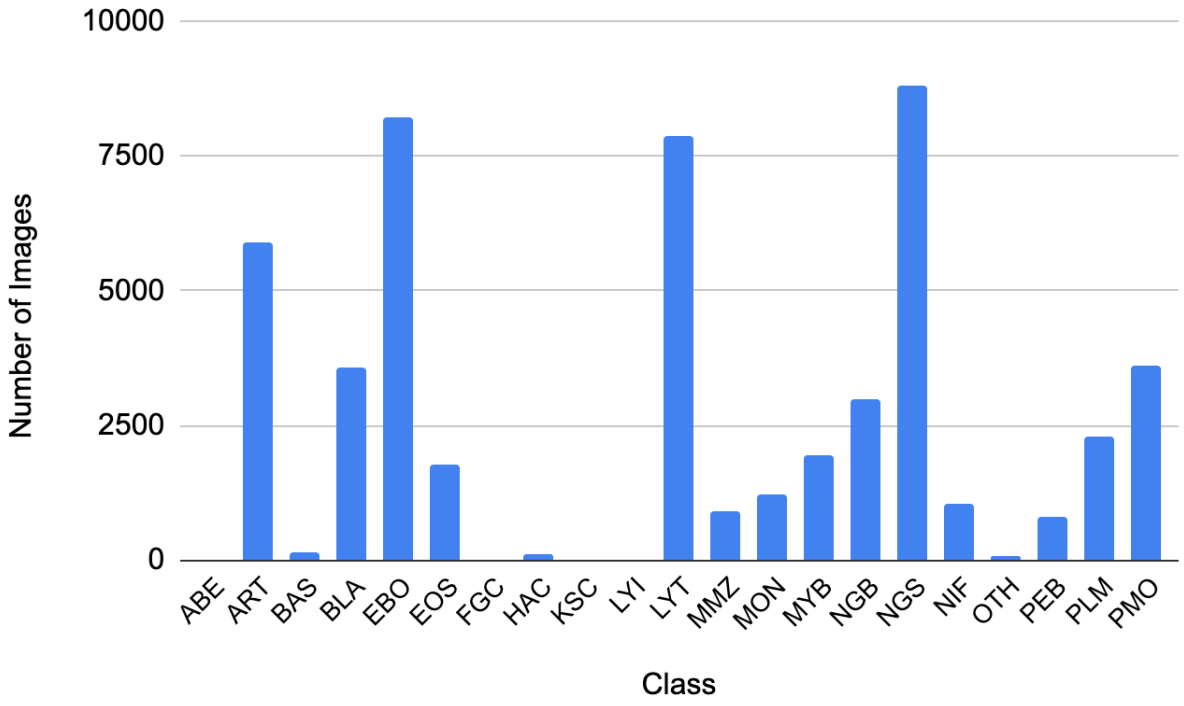


Figure 5.1: Bar chart showing the distribution of images in the test test

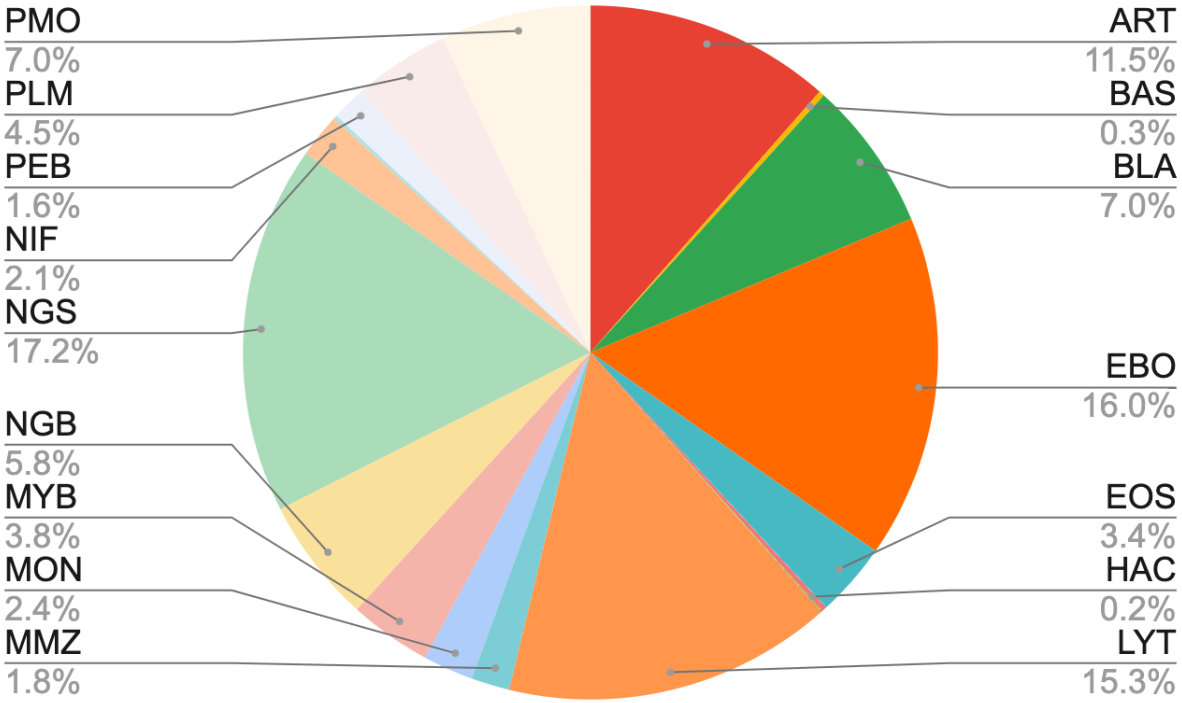


Figure 5.2: Pie chart showing the percentage distribution of images in the test test. This distribution is identical to the dataset itself.

5.1 Minimum Viable Analysis Model

The minimum viable analysis model obtained an accuracy of 74.5% and a balanced accuracy of 44%. Precision, recall, and F1-Score all obtained a score of 74%. The results are nearly identical to the validation results with the minimum viable analysis model obtaining an accuracy of 74.0%, with balanced accuracy, precision, recall, and F1-Score being identical to the 44% and 74% obtained. This shows that this model can generalise well on unseen data. Presented below is the exact breakdown of the distribution of prediction as seen in the confusion matrix and normalised confusion matrix.

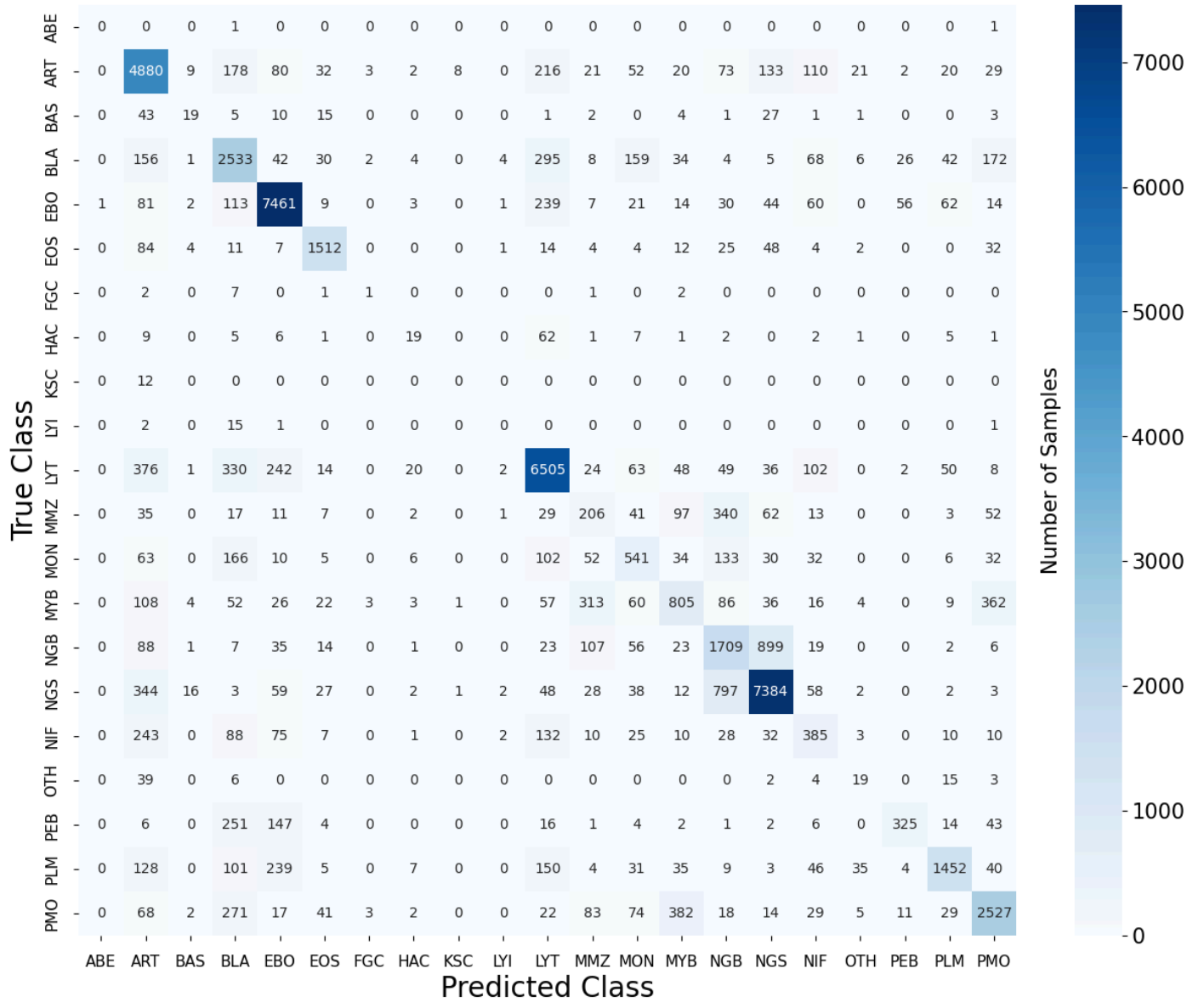


Figure 5.1.1: Confusion matrix for the minimum viable analysis model on the test set

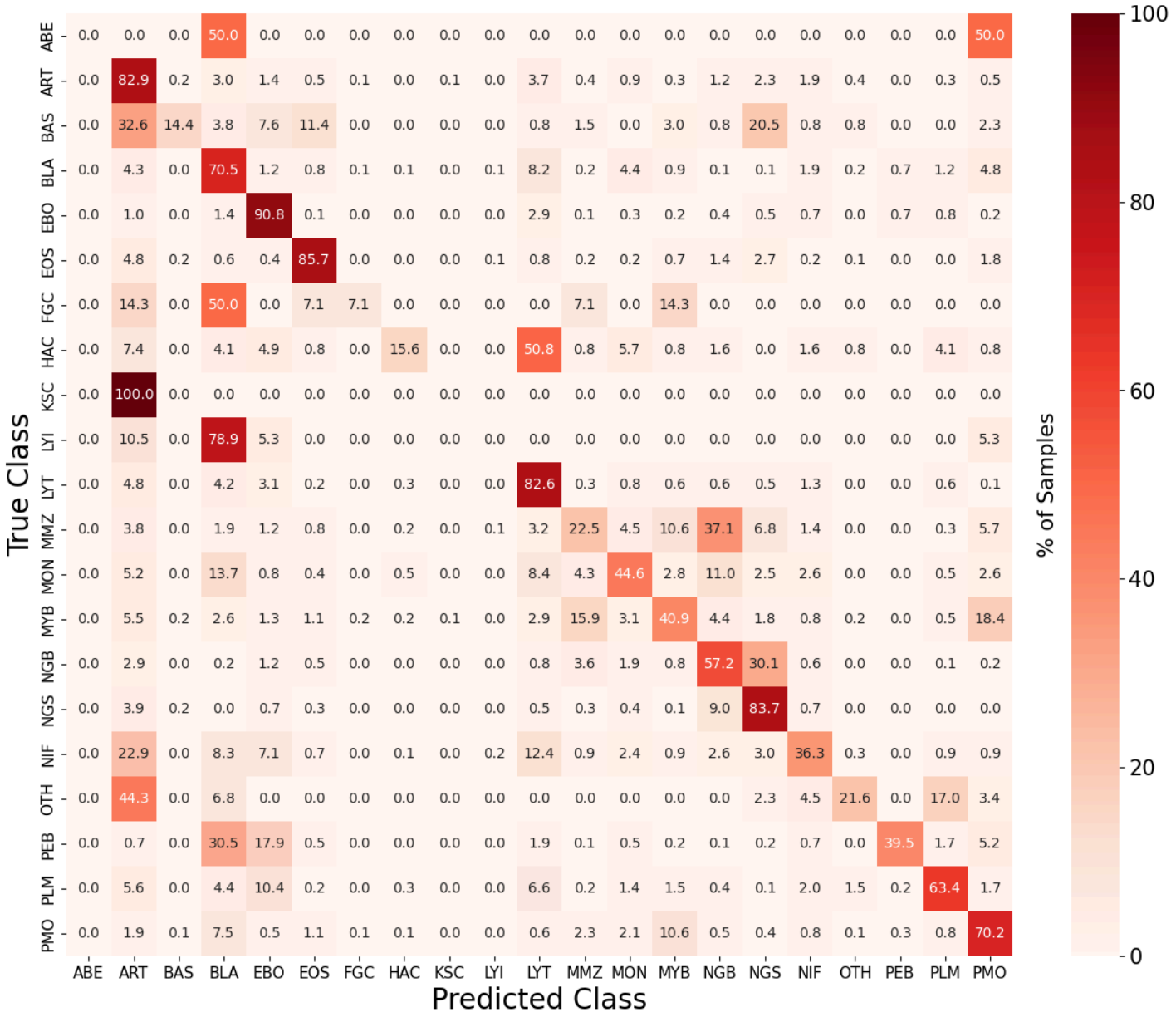


Figure 5.1.2: Normalised confusion matrix for the minimum viable analysis model on the test set

The precision, recall, and F1-score for each class compared to the validation set are extremely similar, with no class differing by more than 8% across all three metrics (See *Table 5.1.3-5.1.5*). Notably, the KSC class saw the most substantial improvement, rising from 0% to 8% in F1-Score, while the HAC class experienced a decline from 24% to 20%. These are minority classes, each comprising less than 0.1% of the dataset samples, meaning variation in accuracy for these classes is expected to be larger due to the weighting each sample has in proportion to the class. Most other classes had a low variation. This highlights the ability of this model to generalise well on unseen data.

Class	Test Set Precision (%)	Validation Set Precision (%)	Difference (%)
ABE	0	0	0
ART	72	72	0
BAS	32	25	7
BLA	61	59	2
EBO	88	88	0
EOS	87	87	0
FGC	8	0	8
HAC	26	29	-3
KSC	0	0	0
LYI	0	0	0
LYT	82	82	0
MMZ	24	24	0
MON	46	41	5
MYB	52	51	1
NGB	52	54	-2
NGS	84	84	0
NIF	40	38	2
OTH	19	20	-1
PEB	76	75	1
PLM	84	82	2
PMO	76	75	1

Table 5.1.3: Table comparing the difference between test set and validation set precision for the minimum viable analysis model

Class	Test Set Recall (%)	Validation Set Recall (%)	Difference (%)
ABE	0	0	0
ART	83	83	0
BAS	14	10	4
BLA	71	71	0
EBO	91	91	0
EOS	86	85	1
FGC	7	0	7
HAC	16	21	-5
KSC	0	0	0
LYI	0	0	0
LYT	83	82	1
MMZ	22	21	1
MON	45	42	3
MYB	41	39	2
NGB	57	57	0
NGS	84	84	0
NIF	36	38	-2
OTH	22	24	-2
PEB	40	38	2
PLM	63	63	0
PMO	70	67	3

Table 5.1.4: Table comparing the difference between test set and validation set recall for the minimum viable analysis model

Class	Test Set F1-Score (%)	Validation Set F1-Score (%)	Difference (%)
ABE	0	0	0
ART	77	77	0
BAS	20	14	6
BLA	65	64	1
EBO	89	90	-1
EOS	86	86	0
FGC	8	0	8
HAC	20	24	-4
KSC	0	0	0
LYI	0	0	0
LYT	82	82	0
MMZ	23	22	1
MON	45	41	4
MYB	46	44	2
NGB	54	55	-1
NGS	84	84	0
NIF	38	38	0
OTH	20	22	-2
PEB	52	51	1
PLM	72	71	1
PMO	73	71	2

Table 5.1.5: Table comparing the difference between test set and validation set F1-Score for the minimum viable analysis model

5.2 Optimised Model

The hyperparameter-tuned optimised model obtained an accuracy of 75.9% and a balanced accuracy of 43%. Precision and recall were 76% each with the F1-score being 75%. Similar to the minimum viable analysis model, the results are nearly identical to the validation results of the optimised model where accuracy was 76.0%, with a balance accuracy, precision, recall, and F1-score being identical to the test results. Presented below is the exact breakdown of the distribution of prediction as seen in the confusion matrix and normalised confusion matrix.

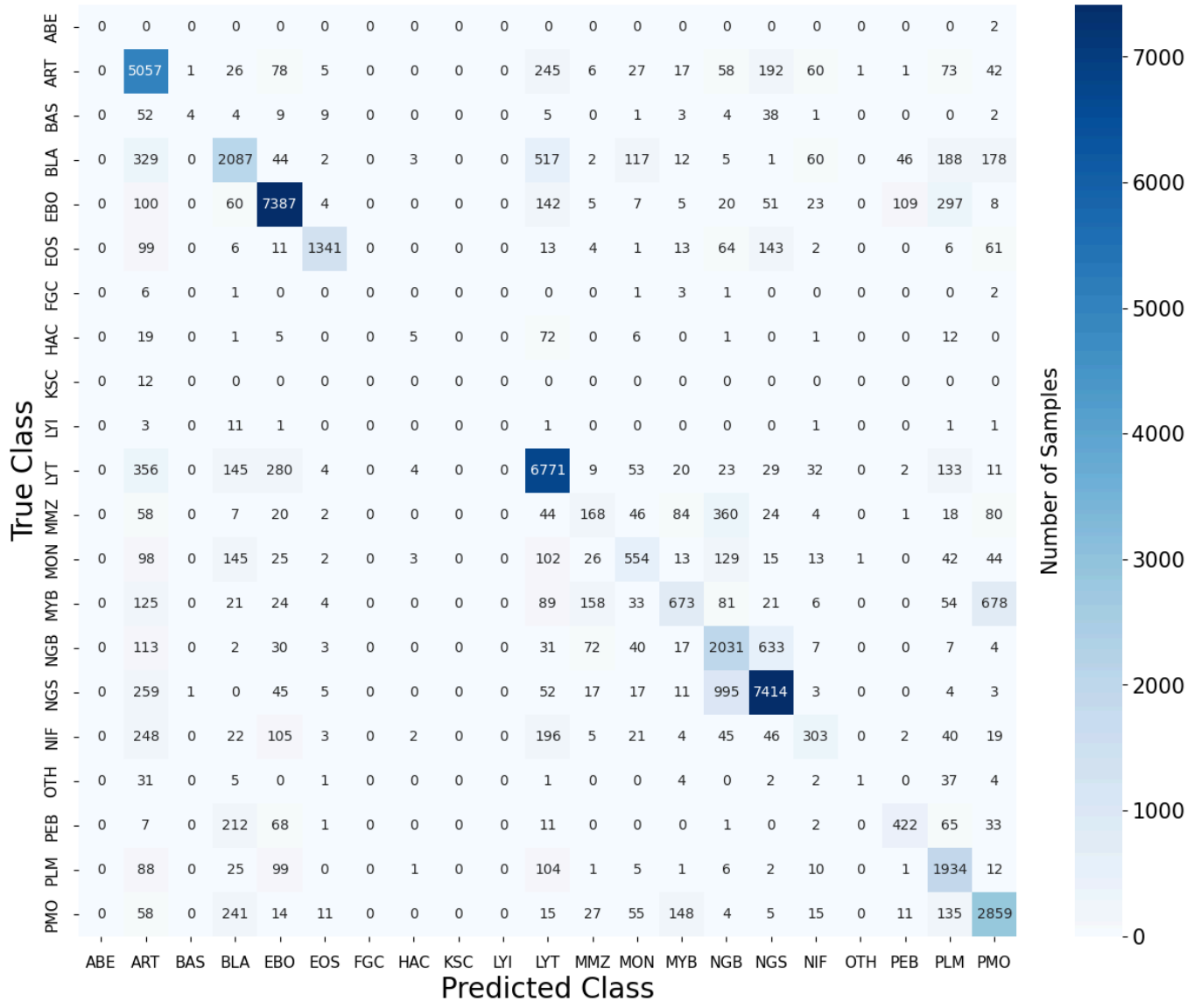


Figure 5.2.1: Confusion matrix for the optimised model on the test set

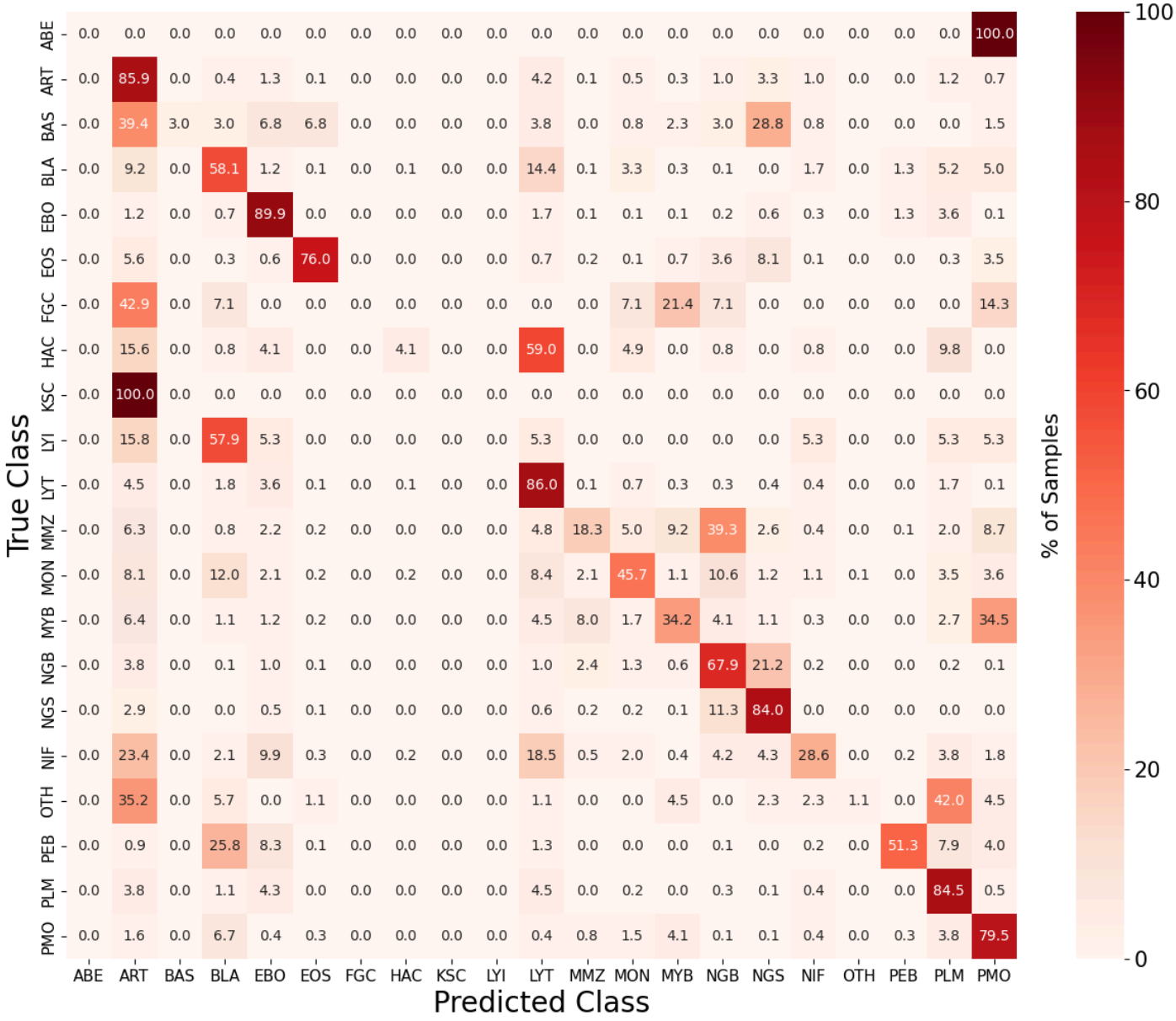


Figure 5.2.2: Normalised confusion matrix for the optimised model on the test set

Unlike the minimum viable analysis test set results, precision within classes had a visible difference between the test and validation set. BAS had a 67% precision in comparison to a 0% precision in the validation set, the OTH class had a 33% precision in comparison to a 0% precision in the validation set, and HAC had a 28% precision in comparison to the 50% precision in the validation set (See *Table 5.2.3*). Similar to the minimum viable analysis test results though, these classes hold less than 0.1% of the dataset each, so variation amongst these classes is expected. Moreover, the recall and F1-scores highlight that

there is no significant difference between the predictions in the test set or validation set (See *Table 5.2.4* and *Table 5.2.5*). This further highlights the model's ability to generalise well.

Class	Test Set Precision %	Validation Set Precision %	Difference %
ABE	0	0	0
ART	71	72	-1
BAS	67	0	67
BLA	69	68	1
EBO	90	89	1
EOS	96	95	1
FGC	0	0	0
HAC	28	50	-22
KSC	0	0	0
LYI	0	0	0
LYT	81	82	-1
MMZ	34	30	4
MON	56	54	2
MYB	65	62	3
NGB	53	52	1
NGS	86	87	-1
NIF	56	60	-4
OTH	33	0	33
PEB	71	74	-3
PLM	63	65	-2
PMO	71	70	1

Table 5.2.3: Table comparing the difference between test set and validation set precision for the optimised model

Class	Test Set Recall (%)	Validation Set Recall (%)	Difference (%)
ABE	0	0	0
ART	86	86	0
BAS	3	0	3
BLA	58	61	-3
EBO	90	90	0
EOS	76	74	2
FGC	0	0	0
HAC	4	7	-3
KSC	0	0	0
LYI	0	0	0
LYT	86	87	-1
MMZ	18	15	3
MON	46	47	-1
MYB	34	32	2
NGB	68	70	-2
NGS	84	83	1
NIF	29	32	-3
OTH	1	0	1
PEB	51	53	-2
PLM	84	83	1
PMO	79	78	1

Table 5.2.4: Table comparing the difference between test set and validation set recall for the optimised model

Class	Test Set F1-Score (%)	Validation Set F1-Score (%)	Difference (%)
ABE	0	0	0
ART	78	78	0
BAS	6	0	6
BLA	63	65	-2
EBO	90	90	0
EOS	85	84	1
FGC	0	0	0
HAC	7	12	-5
KSC	0	0	0
LYI	0	0	0
LYT	83	84	-1
MMZ	24	20	4
MON	50	50	0
MYB	45	42	3
NGB	60	60	0
NGS	85	85	0
NIF	38	42	-4
OTH	2	0	2
PEB	60	62	-2
PLM	73	73	0
PMO	75	74	1

Table 5.2.5: Table comparing the difference between test set and validation set F1-score for the optimised model

5.3 Optimised Model with Data Augmentation

The optimised model obtained with data augmentation achieved an accuracy of 78.3% and a balanced accuracy of 64%. Precision was 80%, recall was 78%, and the F1-score was 79% on the test set. The trend of the test results being extremely similar to the validation results continued as accuracy for the validation set was 78.2%, with a balance accuracy, precision, recall, and F1-score being identical to the test results. Presented in *Figure 5.3.1* and *Figure 5.3.2* is the exact breakdown of the distribution of prediction as seen in the confusion matrix and normalised confusion matrix.

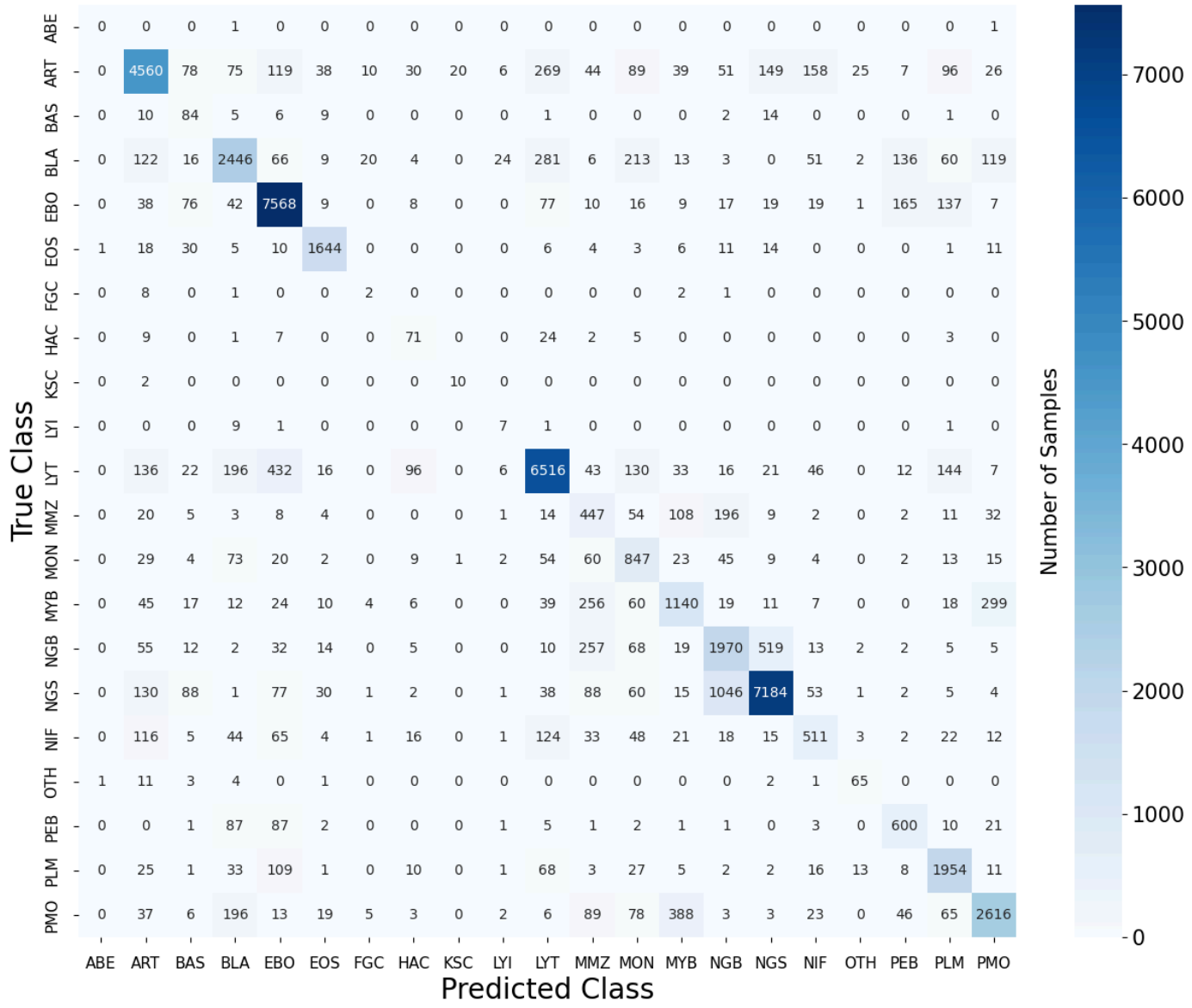


Figure 5.3.1: Confusion matrix for the optimised model using data augmentation on the test set

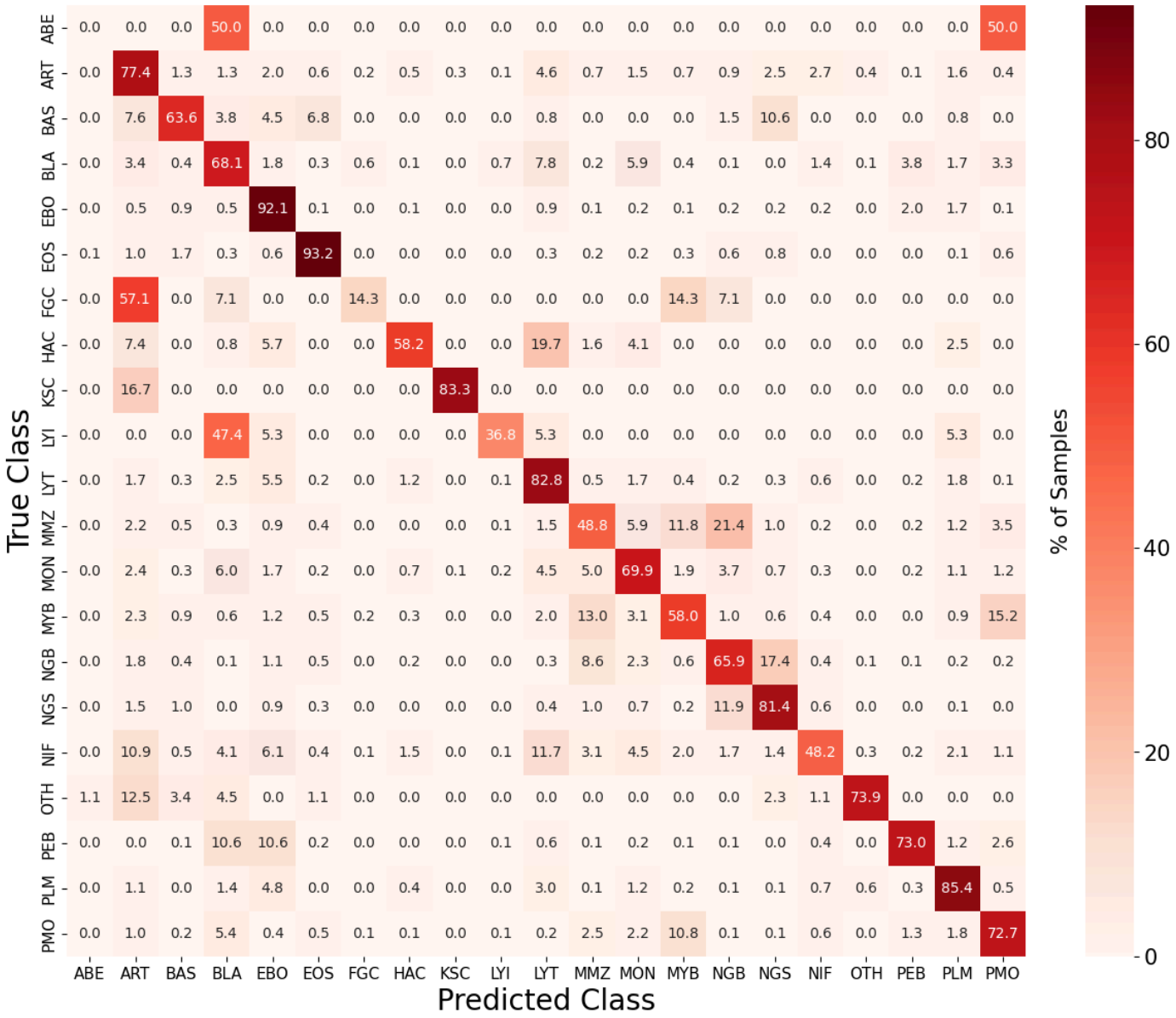


Figure 5.3.2: Normalised confusion matrix for the optimised model using data augmentation on the test set

The precision, recall, and F1-score breakdown for each class is presented in *Tables 5.3.3-5.3.5*, with a small degree of difference between the test and validation set for images that hold a significant proportion of the dataset whilst images with only a few samples presented a larger difference. FGC, KSC, and LYI all had an absolute percentage difference between the test and validation set above 10% in terms of F1-score. This is due to large changes in the precision and recall of these classes. However as described in Chapter 5.1, these minority classes are expected to have a larger variation.

Class	Test Set Precision (%)	Validation Set Precision (%)	Difference (%)
ABE	0	0	0
ART	85	86	-1
BAS	19	20	-1
BLA	76	75	1
EBO	88	88	0
EOS	91	88	3
FGC	5	14	-9
HAC	27	29	-2
KSC	32	13	19
LYI	13	5	8
LYT	86	87	-1
MMZ	33	34	-1
MON	50	49	1
MYB	63	59	4
NGB	58	58	0
NGS	90	90	0
NIF	56	53	3
OTH	58	54	4
PEB	61	61	0
PLM	77	76	1
PMO	82	83	-1

Table 5.3.3: Table comparing the difference between test set and validation set precision for the optimised model using data augmentation

Class	Test Set Recall (%)	Validation Set Recall (%)	Difference (%)
ABE	0	0	0
ART	77	78	-1
BAS	64	61	3
BLA	68	70	-2
EBO	92	92	0
EOS	93	93	0
FGC	14	50	-36
HAC	58	58	0
KSC	83	33	50
LYI	37	11	26
LYT	83	84	-1
MMZ	49	51	-2
MON	70	67	3
MYB	58	56	2
NGB	66	66	0
NGS	81	81	0
NIF	48	44	4
OTH	74	63	11
PEB	73	70	3
PLM	85	84	1
PMO	73	70	3

Table 5.3.4: Table comparing the difference between test set and validation set recall for the optimised model using data augmentation

Class	Test Set F1-Score (%)	Validation Set F1-Score (%)	Difference (%)
ABE	0	0	0
ART	81	82	-1
BAS	29	30	-1
BLA	72	72	0
EBO	90	90	0
EOS	92	90	2
FGC	7	21	-14
HAC	37	39	-2
KSC	47	19	28
LYI	20	7	13
LYT	85	86	-1
MMZ	40	41	-1
MON	58	57	1
MYB	60	58	2
NGB	62	61	1
NGS	86	85	1
NIF	52	48	4
OTH	65	58	7
PEB	66	65	1
PLM	81	80	1
PMO	77	76	1

Table 5.3.5: Table comparing the difference between test set and validation set F1-score for the optimised model using data augmentation

6. Explainable AI Experiment

6.1 Experiment Justification

As mentioned in Chapter 2.1.2.2, deep learning models such as CNNs are black-box models. This is an ethical concern regarding medical diagnoses due to the criticality of incorrect outputs. By applying a technique like LIME, we can help understand which features within an image contribute most to a model's decision-making process. The first step involves segmenting an image into various sections. The most basic method is to segment the image into grids. A more sophisticated method is to use an algorithm like quick shift to maintain the morphological structure of the cell within the image. This algorithm forms clusters of pixels, known as superpixels, based on colour and spatial proximity to generate natural borders (See *Figure 6.1.2*). Quick shift works by initialising each pixel as a superpixel before iteratively merging superpixels that are similar in both the colour and space proximity. This is based on a function known as the kernel. The size of the superpixel is determined by a parameter within the kernel known as the bandwidth, where a smaller bandwidth leads to more segmentations with smaller superpixels, and vice versa for a larger bandwidth.

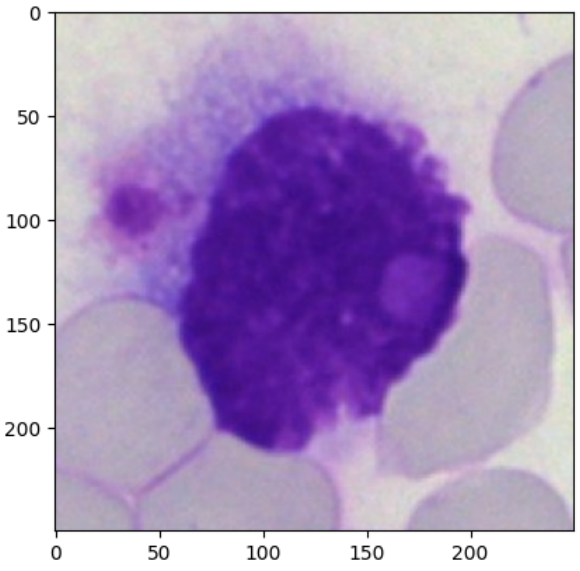


Figure 6.1.1: Image showing a 250x250 pixel artefact (ART class) cell image

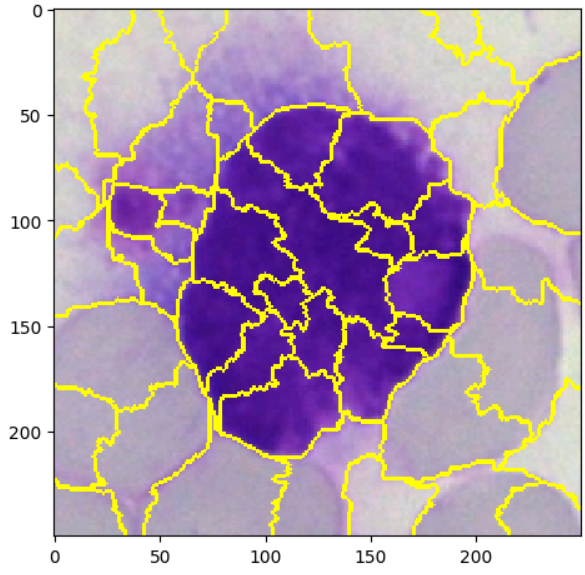


Figure 6.1.2: Image showing *Figure 6.1.1* segmented using the quick shift algorithm

Once the image is segmented, a set of random perturbations is generated by turning on and off certain superpixels within the image (See *Figure 6.1.3*). These images are all fed into the trained model to generate an output probability for a target class. Each perturbed image is then assigned a weight based on how similar it is to the original image, with perturbed images more like the original image assigned a higher weight. Afterwards, a simple linear regression model is fit using the perturbed images, the predictions, and the weights to generate a relationship between the superpixels and the target class in a dimensional space the size of the number of superpixels. The coefficients for this linear model can then be extracted, with each coefficient signifying the importance of the corresponding superpixel for determining the prediction of a target class. The user can then choose the top *n* of superpixels to generate an explanation of what features the model deems most important (See *Figure 6.1.4*).

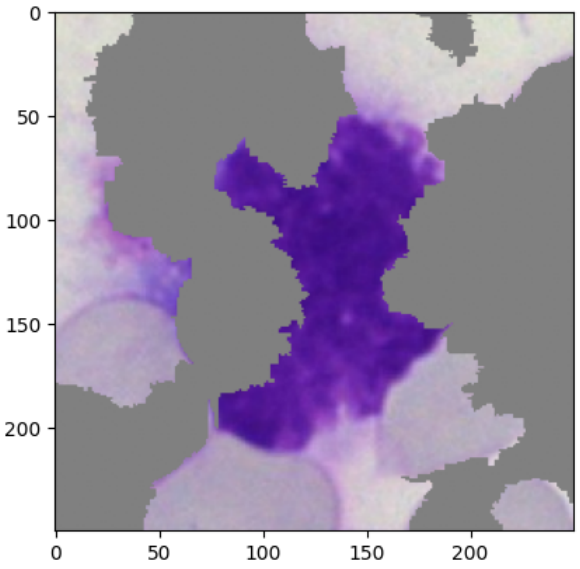


Figure 6.1.3: Image showing a single perturbation of *Figure 6.1.2* with some superpixels occluded

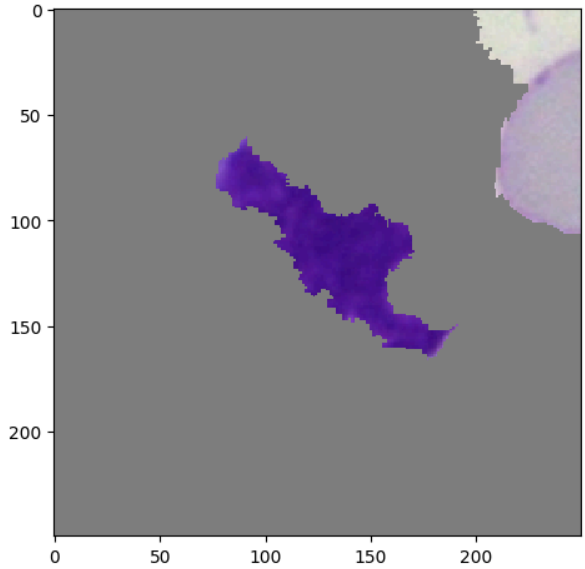


Figure 6.1.4: Image showing the explanation generated for *Figure 6.1.1* where $n = 5$

In the context of medical diagnosis through image classification, this technique can be useful for medical professionals as they can gain trust in machine learning models by examining a set of explanations to verify whether the features the model is picking up are the same as manual examination. Medical professionals would also be able to see whether the model is picking up noise within the image to come up with its prediction instead.

Despite these benefits, this process of validating a model's prediction through LIME remains flawed. This is because even if LIME generates the correct regions of interest within the top *n* superpixels specified by the user, the model itself may require more than the *n* superpixels to correctly classify that class. With the

addition of more superpixels, more noisy pixels are added back into the image. Meaning that the model may still require these noisy pixels to correctly predict a cell's class. With small variations present when taking samples, i.e. the differing process of taking the sample, equipment used, lighting, etc…the model may perform poorly when fed these samples with variations present. This can often be overlooked by the human examiner as oftentimes the background of the image is not considered. However, a machine learning model considers the whole image. Therefore, to further evaluate a model's robustness and the quality of explainable AI methodologies like LIME, any given LIME explanation generated can be fed back into the model to examine whether *n* superpixels is sufficient enough to justify why the model predicted what it predicted. This will therefore help explore the extent the model picks up noise in the background whilst also explaining how susceptible or robust a model can be to adversarial attacks, a type of attack involving making a small change in the image to intentionally mislead the model's prediction. Therefore, this section is dedicated to experimenting to explore the number of superpixels required to sufficiently justify LIME explanations using the best-performing model found in Chapter 5.

6.2 Methodology

The first step is to generate explanations for each image by creating a set of 128 random perturbations with randomly occluded superpixels. Shah and Sheppard occlude the superpixels using Zeiler and Fergus' method of turning these regions grey which will be emulated in this experiment (Zeiler and Fergus 824). Moreover, Shah and Sheppard found that generating over 5,000 explanations took over 10,000 computational hours if 1,000 random perturbations per image were used, which in the context of the senior honours project is not possible. Therefore, only 30% of the validation set images were used whilst 128 perturbations were created per image. This experiment also used the quick shift algorithm to segment the image into superpixels that followed the cell's morphology. This meant that images may therefore contain a differing number of superpixel segments. Overall, the mean number of superpixels found was 43. Explanations with the top *n* superpixels active are then passed onto the image augmentation model created in Chapter 5.7 to validate whether the value of *n* is sufficient enough to justify why the model is predicting what is predicting. Initially, the values of *n* chosen were 1, 5, 15, and 25 as this was a similar range used in Shah and Sheppard's paper. Similarly, 25 superpixels active for this dataset also meant around half the image was also occluded. However as later explained in the following section, 30, 35, 40, and 45 superpixels were also used. All images have at least 25 superpixels but not all images have 25+ superpixels. An image with fewer superpixels than a target *n* will not be used for these test cases. The number of images used per *n* superpixels is described below:

Number of Superpixels	Number of Images
1	7225
5	7225
15	7225
25	7225
30	7224
35	7181
40	6592
45	4162

Table 6.2.1: Table showing the number of images per superpixel group

To measure whether explanations of *n* superpixels contain sufficient justification, precision will serve as the main metric. This is because we want to measure the accuracy of positive predictions in each class to identify whether the model can pick up the relevant features and patterns for that class. As stated in Shah and Sheppard's version of this experiment, if the explanations for *n* superpixels activated perform at least ⅔ as well as the original images in precision, then the explanations generated by the LIME process create images that are sufficient justification for why the model predicted that class.

6.3 Results

Originally, the experiment ranged over 1, 5, 15, and 25 superpixels. However, it quickly became apparent that the explanations generated did not provide sufficient justification and performed poorly. Therefore slowly increasing the number of superpixels with a step of 5 was used to see if precision was able to recover before all superpixels were activated (See *Figure 6.3.1*). From this, 40 activated superpixels were required for sufficient justification to be reached at the ⅔ point threshold. Accuracy also followed a similar recovery trend to precision as the activated superpixels increased (See *Figure 6.3.2*).

Given that the average number of superpixels per image is 43, 40 superpixels would indicate nearly all of the image has to be present for the explanations to be sufficient enough to justify the prediction of a class. This may be because all cell classes were stained using the same method, meaning the texturing of the

cells will appear homogeneous under the microscope. When parts of the cell image are occluded, the morphological features of the cell also become less apparent, causing all images with occlusions to look extremely similar to each other. This possibly creates confusion within the model's prediction as it becomes unsure of what it is seeing. A comparison of a differing number of superpixels active can be visualised in *Figure 6.3.3* to *Figure 6.3.8*.

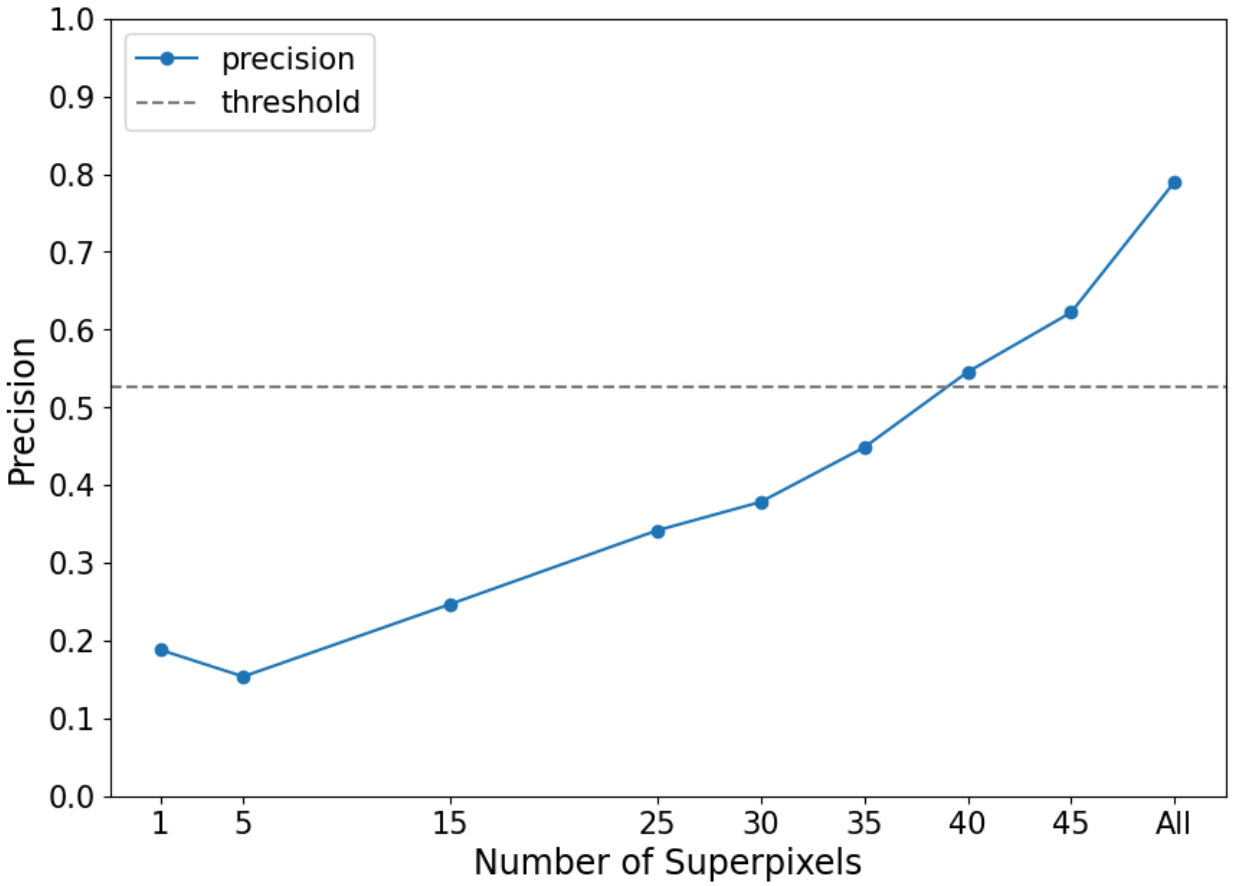


Figure 6.3.1: Graph showing the effect of increasing the number of active superpixels on the average global precision. Threshold is ⅔ precision of all superpixels active

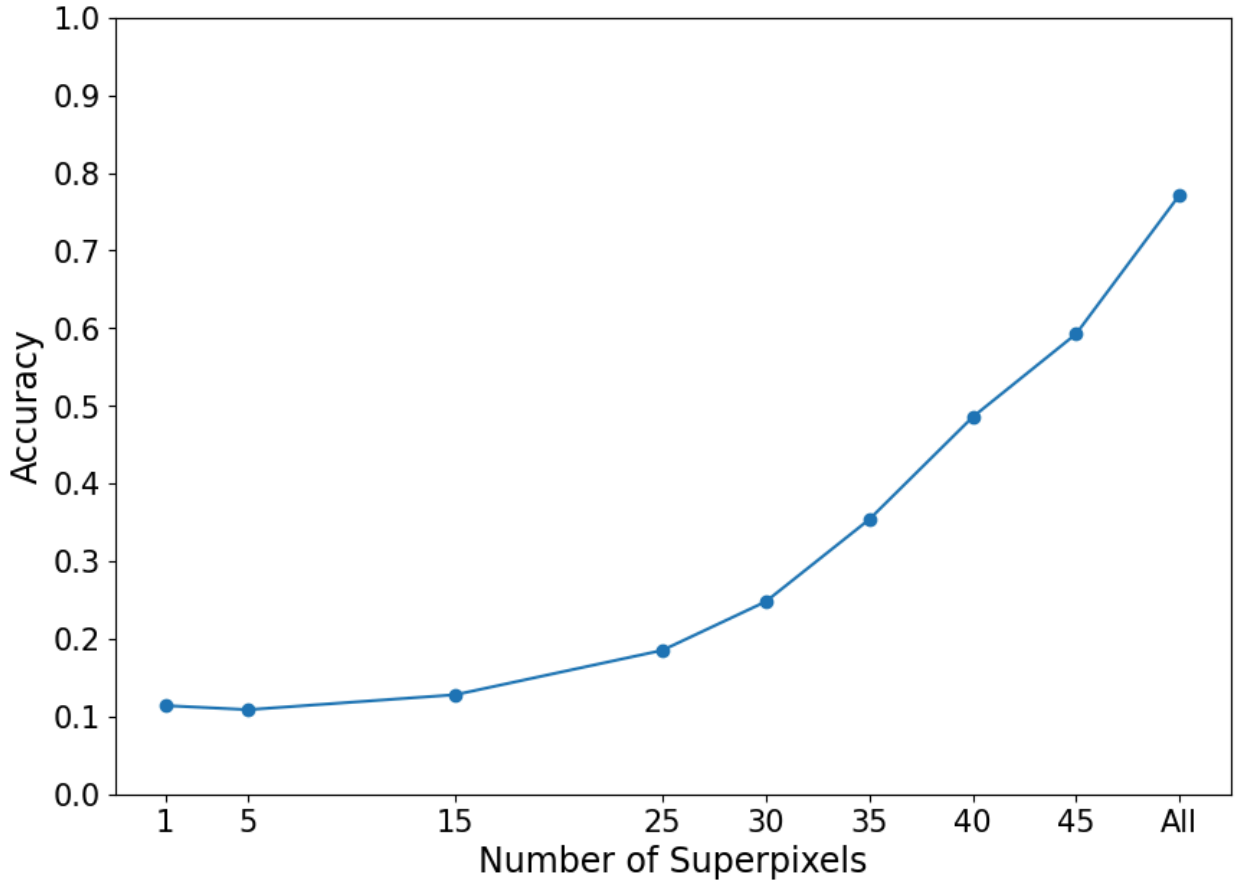


Figure 6.3.2: Graph showing the effect of increasing the number of active superpixels on accuracy

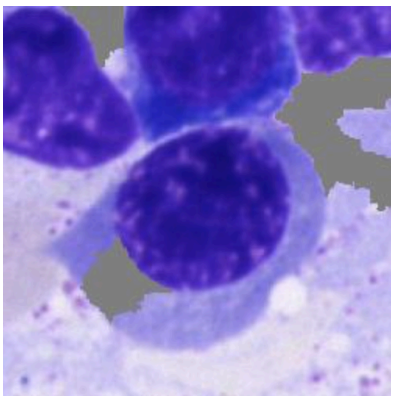


Figure 6.3.3: EBO image with 40 superpixels activated

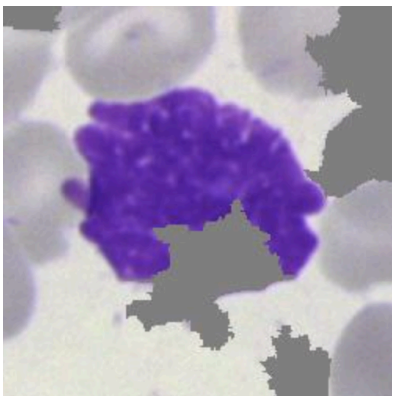


Figure 6.3.4: KSC image with 40 superpixels activated

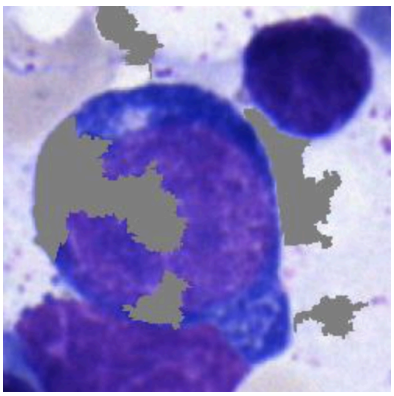


Figure 6.3.5: PEB image with 40 superpixels activated

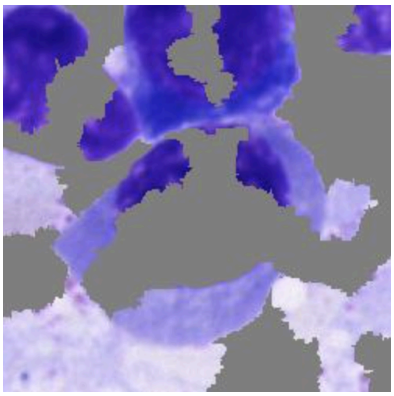


Figure 6.3.6: *Figure 6.3.3* with 25 superpixels activated

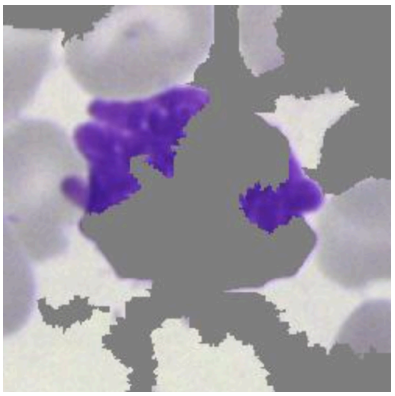


Figure 6.3.7: *Figure 6.3.4* with 25 superpixels activated

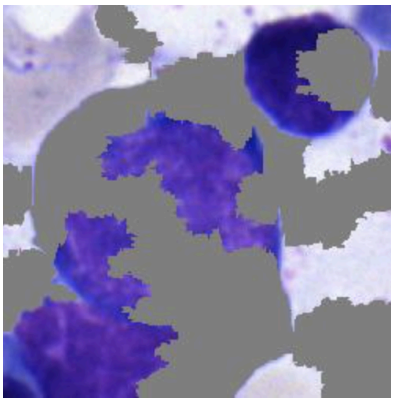


Figure 6.3.8: *Figure 6.3.5* with 25 superpixels activated

For the PLM cells, sufficient justification is reached at 15 superpixels and 35+ superpixels, with 15 superpixels reaching a 100% precision score (See *Figure 6.3.9*). However, if we look more carefully at the confusion matrices in *Figure 6.3.10*, it is observed that only 2 PLM images were correctly classified. This represents 0.6% of PLM images and shows that the model misclassified 99.4% of all PLM images (See *Figure 6.3.11*).

One pattern that is apparent in *Figure 6.3.10* and *Figure 6.3.11* is that a significant majority of images are classified under ART and NIF with 15 active superpixels, contributing to the low accuracy as seen in *Figure 6.3.2*. Although achieving high accuracy with as few superpixels as possible to provide an explanation containing sufficient justification is the best-case scenario, predicting ART or NIF when the model is unsure of what the image is is the next best scenario. This is because ARTs are artefacts, where

according to McInnes are slides that contain defects or abnormalities in tissue sections as a result of the faulty processing of the tissue specimens (100). NIF on the other hand represents not identifiable cells. Both classes signify that the samples cannot be used in identifying the cell class and thus, cannot be used to help in aiding diagnosis. This highlights the model's resilience showing that if it is unsure, it will not randomly classify unknown images as some clinically significant class for cancer diagnosis. This result therefore could reduce the risk of false diagnosis.

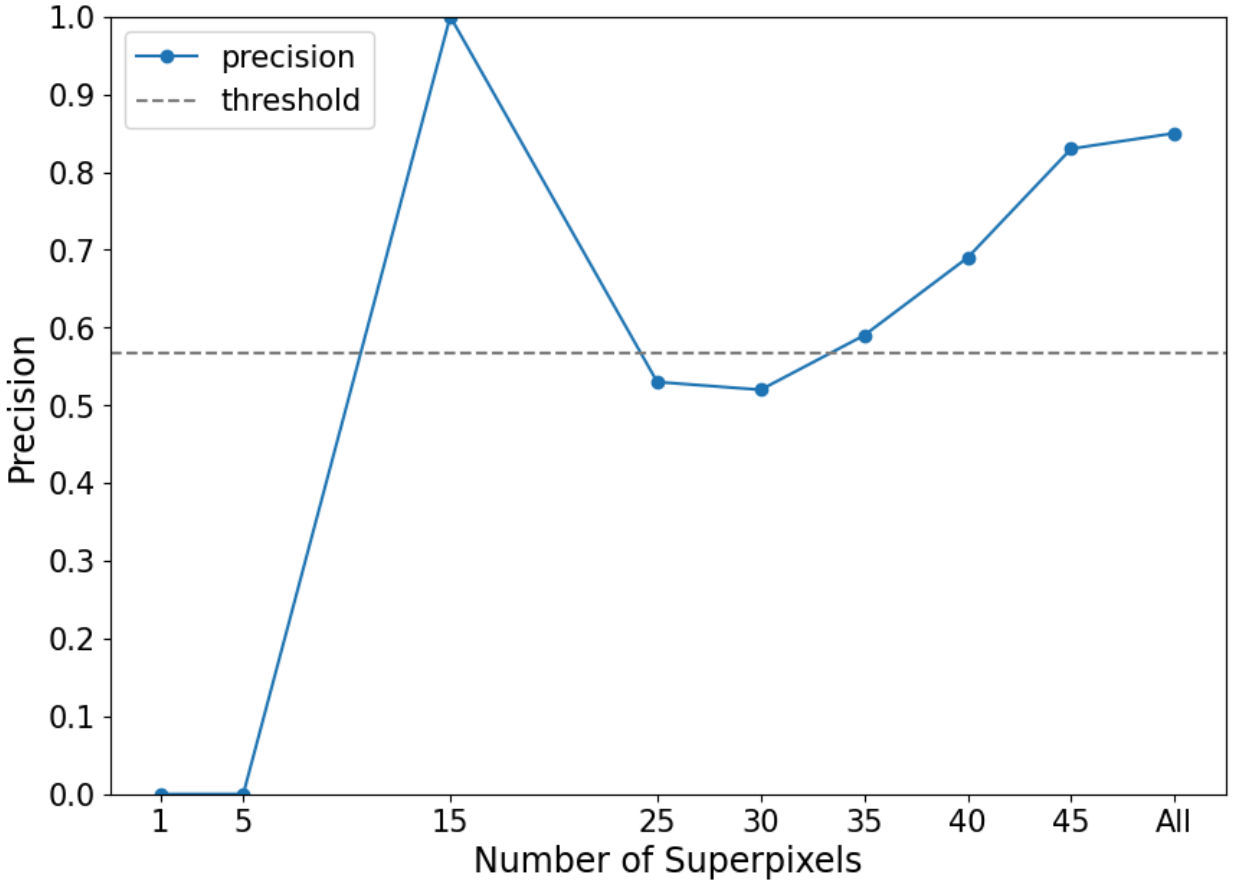


Figure 6.3.9: Graph showing the effect of increasing the number of active superpixels on precision for plasma cells. Threshold is ⅔ precision of all superpixels active

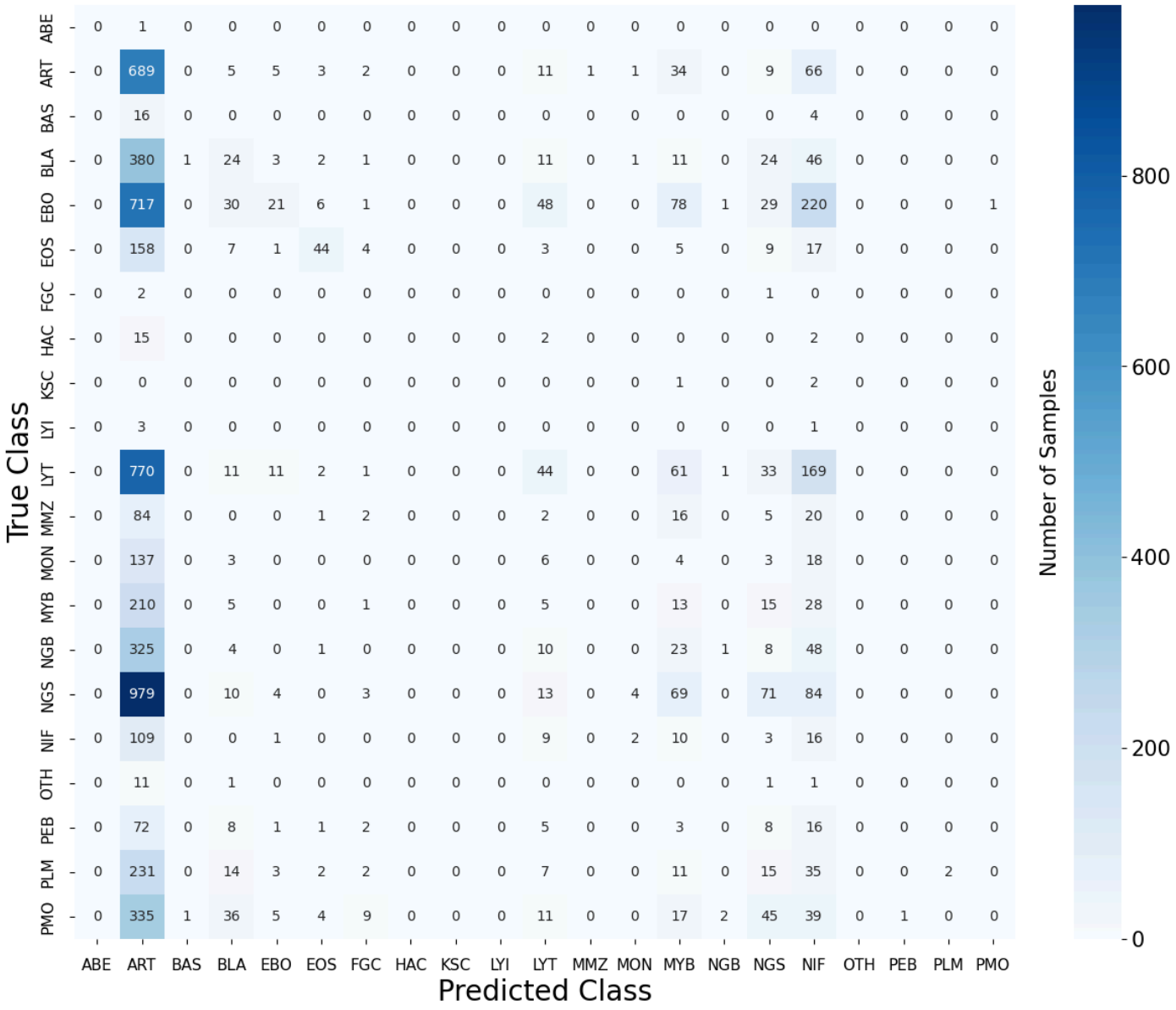


Figure 6.3.10: Confusion matrix for 15 active superpixels

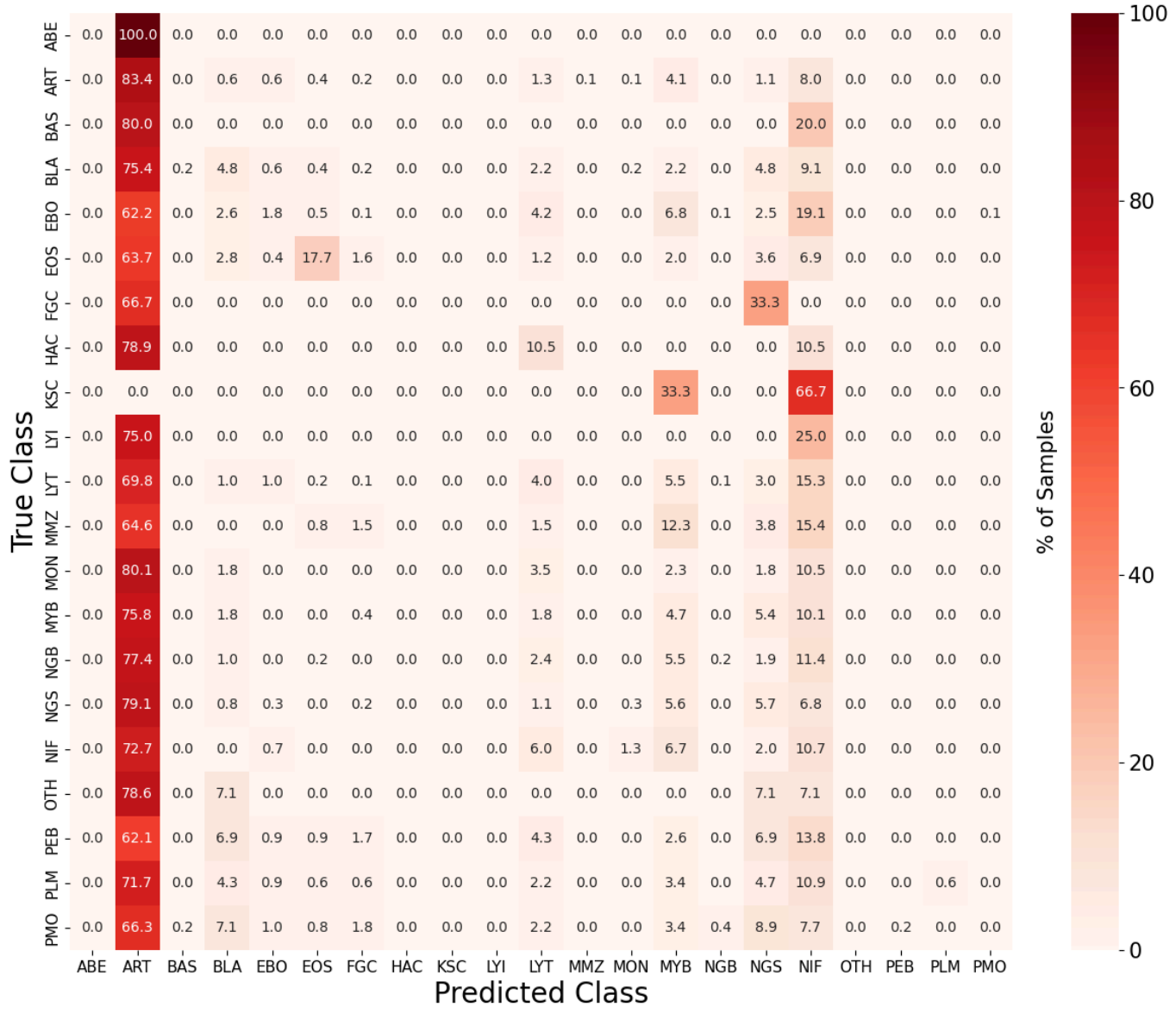


Figure 6.3.11: Normalised Confusion matrix for 15 active superpixels

From *Figures 6.3.12* to *6.3.14*, increasing the active superpixels continues to bring predictions closer to the true estimates of the model. Despite this, however, there is still a significant 17.9% drop off in accuracy from the full image's 77.2% accuracy to 45 superpixels's 59.3% accuracy. This might indicate that the model may still struggle to generalise a given cell's morphological patterns because an increase in superpixels introduces background image noise.

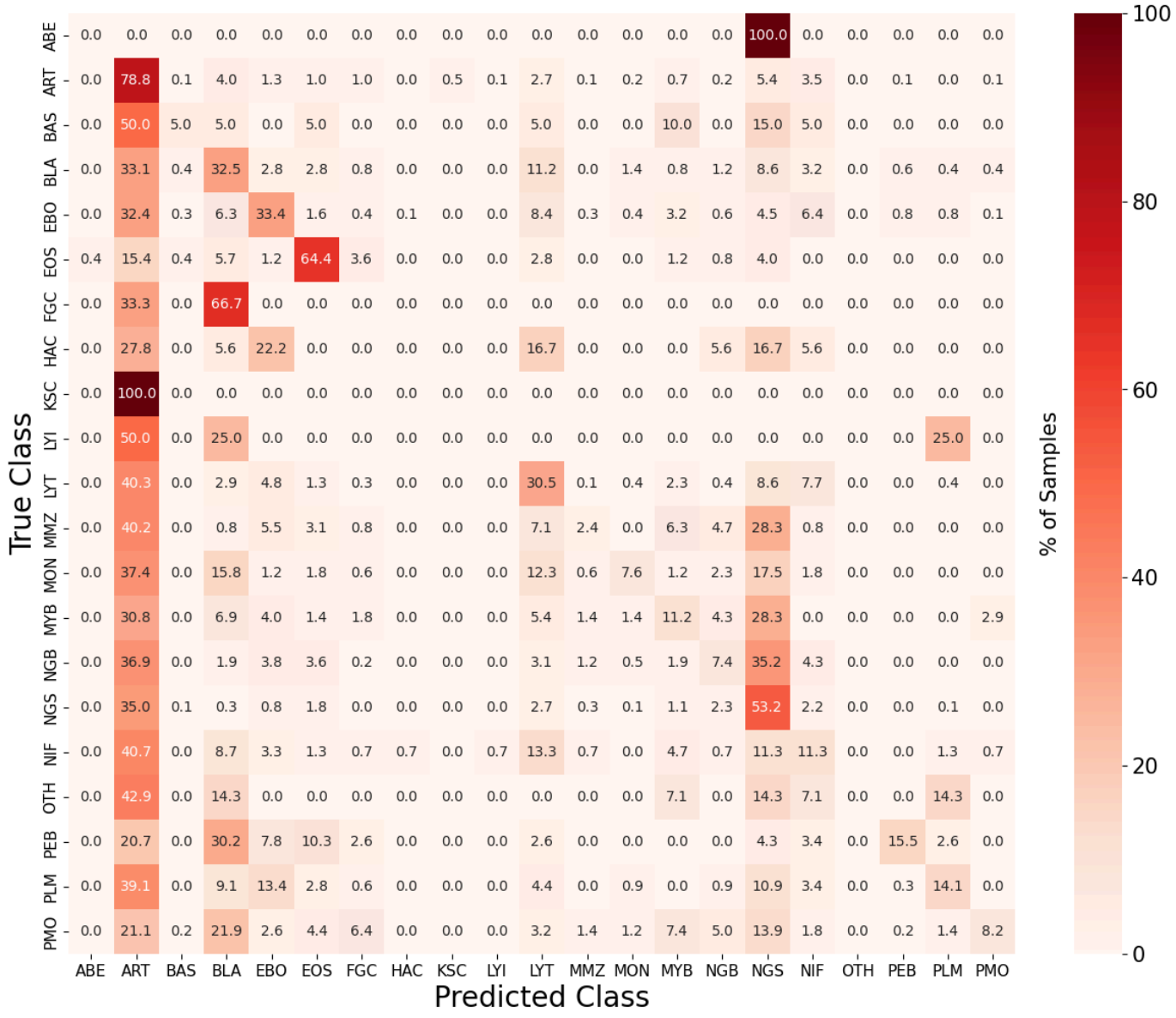


Figure 6.3.12: Normalised Confusion matrix for 35 active superpixels

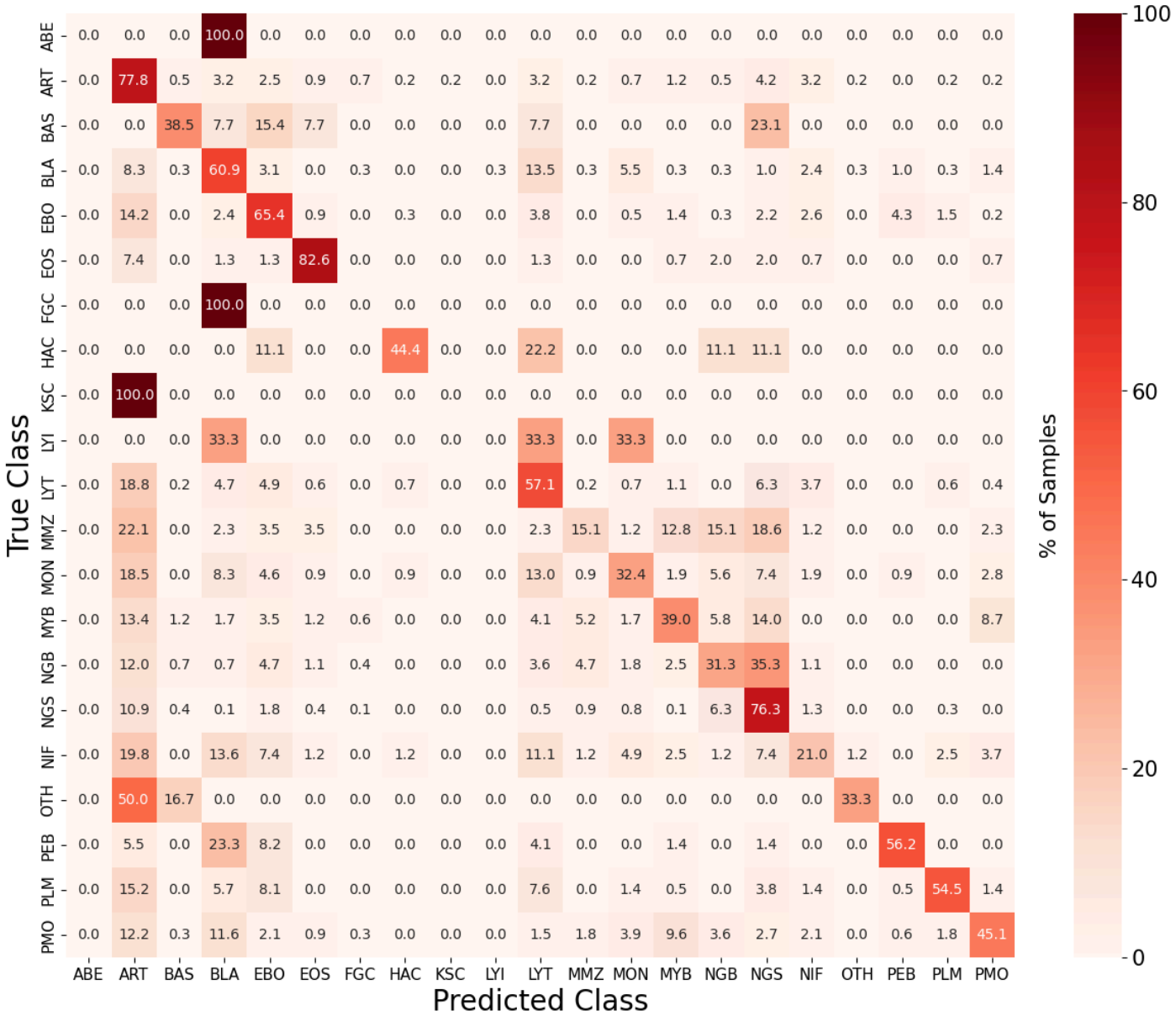


Figure 6.3.13: Normalised Confusion matrix for 45 active superpixels

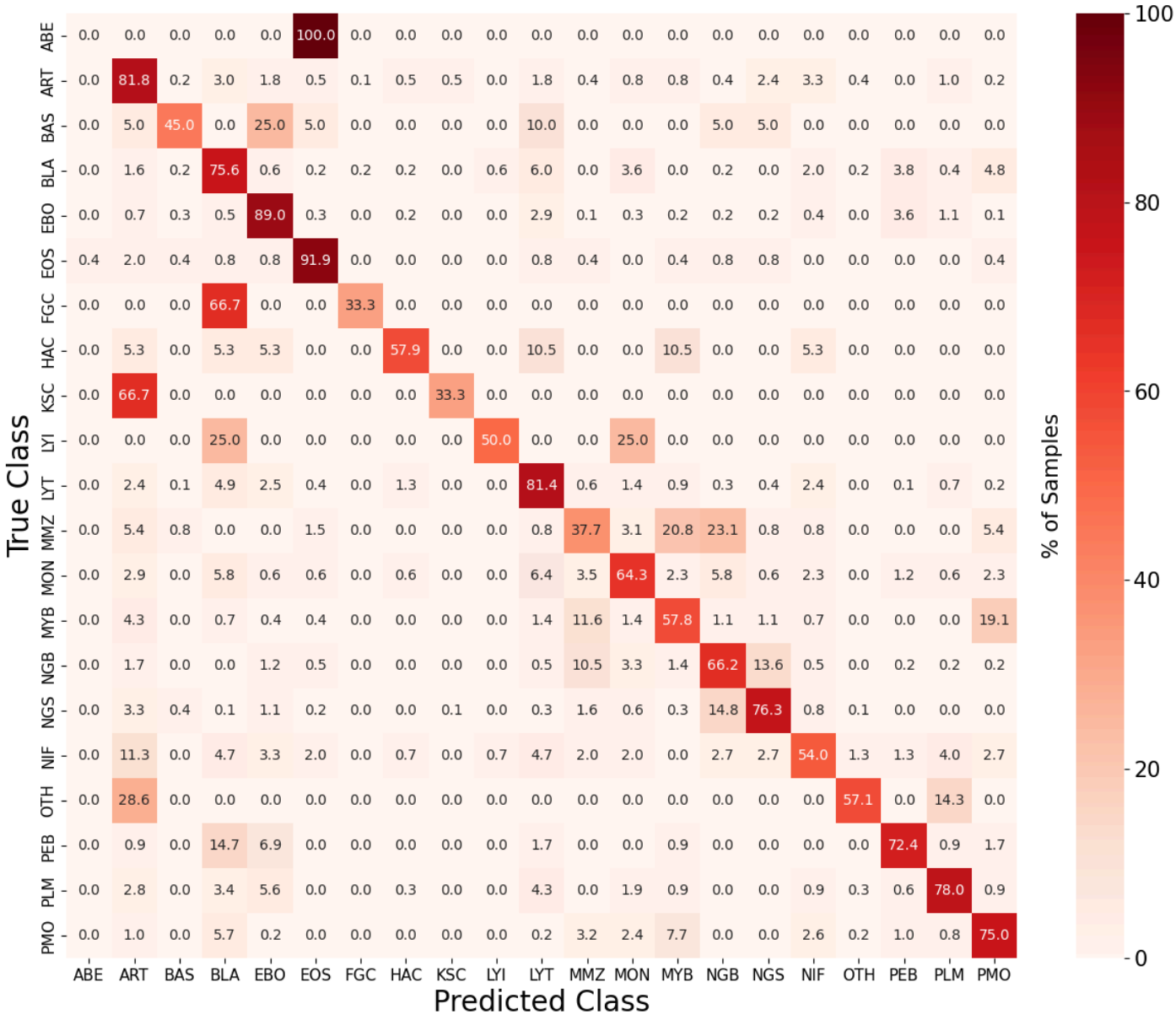


Figure 6.3.14: Normalised Confusion matrix for all superpixels activated (full image)

7. Discussion and Evaluation

7.1 Best Performing Model

The best-identified model in Chapter 4 was the hyperparameter-tuned model that utilised image augmentation. It achieved a validation accuracy of 78.2% and a balanced accuracy of 61%. This model also proved to be effective with generalising on unseen data as the model achieved a 78.3% accuracy and 64% balanced accuracy against the test data. All other models proved to also generalise well against unseen data as the validation accuracy remained comparable to the test accuracy. The baseline model achieved a 74.5% test accuracy against a 74% validation accuracy whilst the hyperparameter-tuned model achieved a 75.9% test accuracy against a 76.0% validation accuracy. These other models however were significantly worse at predicting minority classes which can be reflected by the 44% and 43% balanced accuracy for the baseline model and hyperparameter-tuned model respectively.

7.2 Critical Appraisal

7.2.1 Methodology and Models

In terms of the methodology, a train-validation-test split approach was used to partition the dataset into three distinct subsets. In general, this method may suffer from variability in performance estimates due to randomness in data splitting, particularly when data is limited. The cross-validation strategy involves partitioning the dataset into k subsets and training is performed k-1 times. With each iteration, one of the k subsets becomes a validation set. The mean accuracy score is then used as a final performance metric. This provides more reliable estimates of performance by mitigating the impact of variability in data splitting. However, not only is this dataset considered large for medical imaging, but variability is accounted for by the method as I ensure that splitting is done to preserve the distribution of classes in each of the splits. Moreover, cross-validation has an increased risk of overfitting to the validation sets, especially when hyperparameter tuning is involved, making the evaluation not fully representative of how the model may perform on truly unseen data.

Regarding the models used, despite being able to see significant gains in accuracy to 78.2% on completely unseen data, this may be too low for use in clinical applications. This is especially true for minority classes like HAC where not only does their presence in blood guarantee cancer, their hair-like

structure makes it distinct to categorise compared to most other cells. Having an accuracy of 58.2% for this class would be unacceptable. However, if we consider that manual examinations can sometimes have an error rate of 30-40%, this model may still provide some benefit to medical practitioners in reducing the time needed to classify most cell types.

In terms of predictive capability, all models implemented often confuse MMZ, MON, MYB, NGB, NGS, and PMO with each other. This is most likely due to the fact these cells fall under the same physiological class and will often appear to share similar features (See *Figure 2.2.4*). Moreover, PMO, MYB, MMZ, NGB, and NGS all refer to the different stages of neutrophil development, further highlighting their shared morphological nature (McKenna et al. 5). The data augmentation model significantly reduces this confusion as the hyperparameter-tuned model without augmentation was only correctly predicting:

1. MMZ images 18.3% of the time whilst falsely predicting NGB 39.3% of the time and MYB 9.2% of the time
2. MON images 45.7% of the time whilst falsely predicting NGB 10.6% of the time
3. MYB images 34.0% of the time whilst falsely predicting MMZ 8.0% of the time and PMO 34.5% of the time

The best-performing model that utilised image augmentation reduces this by correctly predicting:

1. MMZ images 48.8% of the time whilst falsely predicting NGB 21.4% of the time and MYB 11.8% of the time
2. MON images 69.9% of the time whilst falsely predicting NGB only 3.7% of the time
3. MYB images 58.0% of the time whilst falsely predicting MMZ 13.0% of the time and PMO 15.2% of the time

Overall however, the best-performing model still often confuses NGB and NGS with each other. 17.4% of NGB images are being predicted as NGS whilst 11.9% of NGS images are being predicted as NGB. This is understandable as NGB and NGS are the last two stages of neutrophil development where the difference is only apparent through the nucleus of the cell, which also can appear extremely similar morphologically (See *Figure 7.2.1.1* and *Figure 7.2.1.2*) (McKenna et al. 5). PMO also performed worse after image augmentation as only 72.7% of images were being correctly identified in the best-performing model with 10.8% of images being confused with MYB cells. In the hyperparameter-tuned model without image augmentation, PMO had an accuracy of 79.5% with MYB only being confused with MYB 4.1% of

the time. This is most likely because PMO has more samples present in the dataset and increasing the samples artificially through data augmentation allowed the model to learn MYB features more accurately.

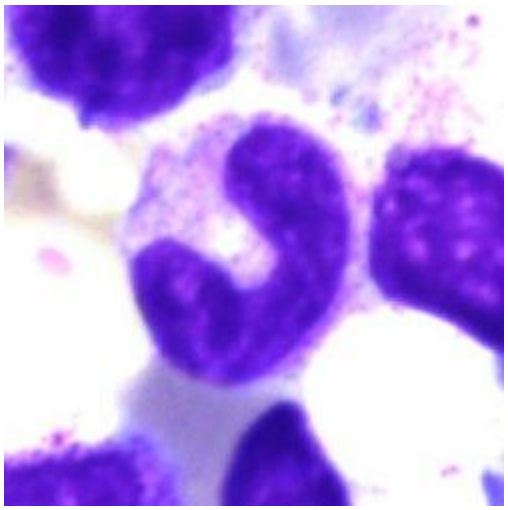


Figure 7.2.1.1: Image of a Band Neutrophil (NGB)

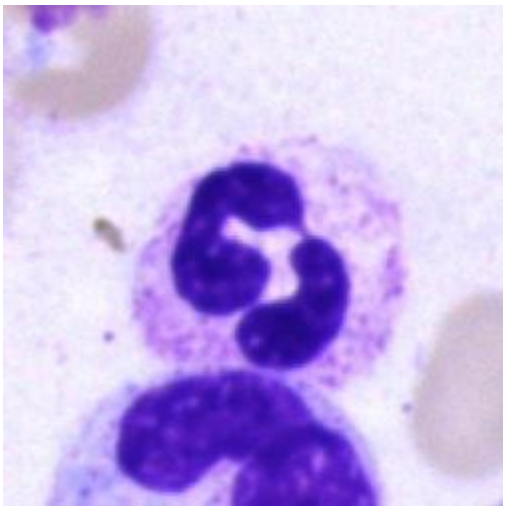


Figure 7.2.1.2: Image of a Segmented Neutrophil (NGS)

Despite the different stages of neutrophil development often being incorrectly predicted with each other, measuring the accuracy of this larger group of cells is still useful. This is because the neutrophil-lymphocyte ratio can be used to determine a patient's outcome in a variety of cancers (Guthrie et al. 219), allowing practitioners to determine an appropriate course of action. *Figure 7.2.1.3* and *Figure 7.2.1.4* group the neutrophils (PMO, MYB, MMZ, NGB, and NGS) together into a new class called NEU using the test set confusion matrices for the best-performing model seen in *Figure 5.3.1* and *Figure 5.3.2*. These figures reveal that despite this model not being able to accurately identify the individual stages of neutrophil development well, by creating a single neutrophil group that encapsulates all these subclasses, the model can correctly predict 16,723 images. This represents a 91.4% accuracy for NEU, making it the third most accurate class behind BLA and EBO. This result proves promising as it leads me to believe the model was able to identify shared features of these cells well.

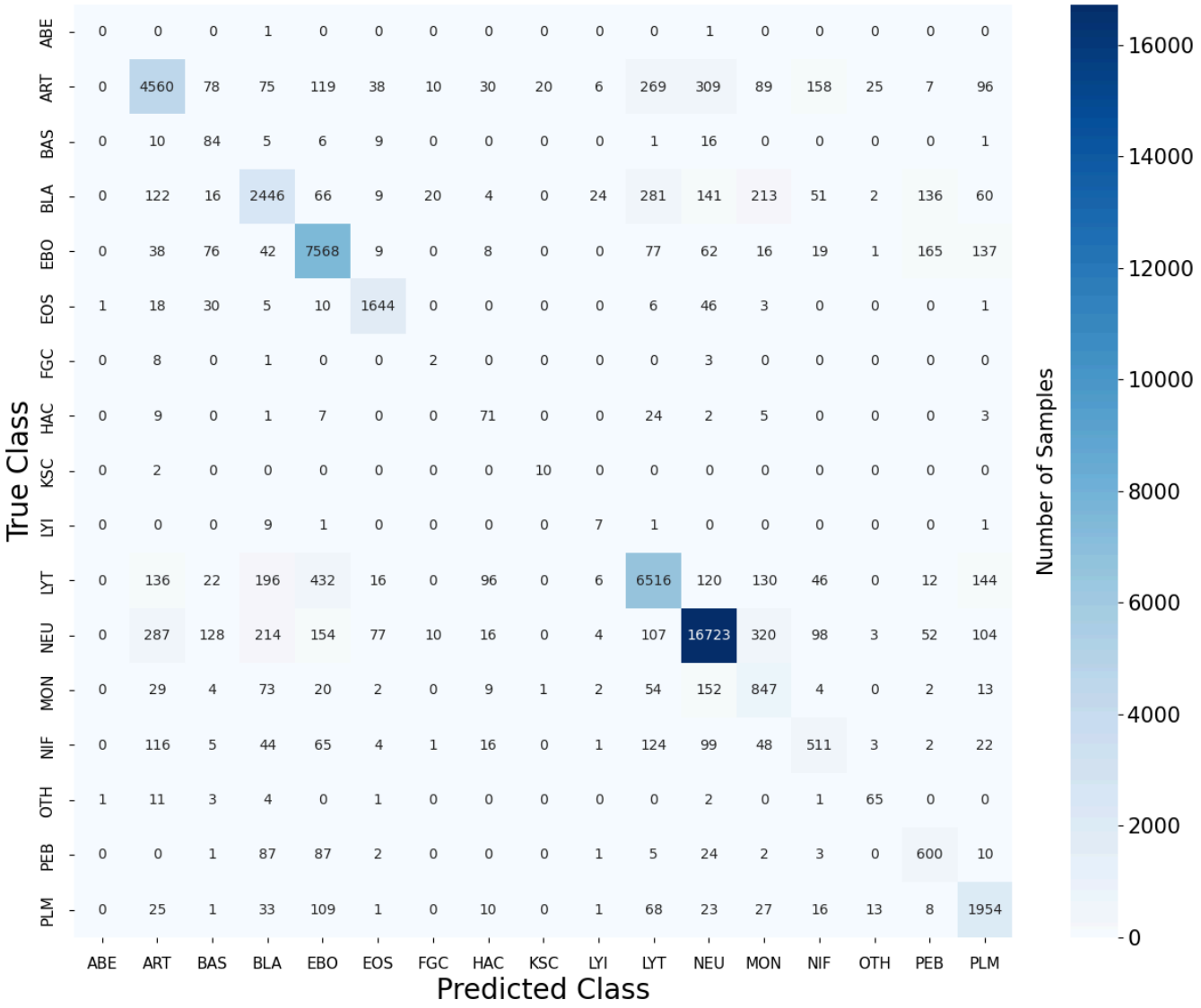


Figure 7.2.1.3: Confusion matrix for the optimised model using data augmentation on the test set where the neutrophils (PMO, MYB, MMZ, NGB, and NGS) are grouped into one category NEU

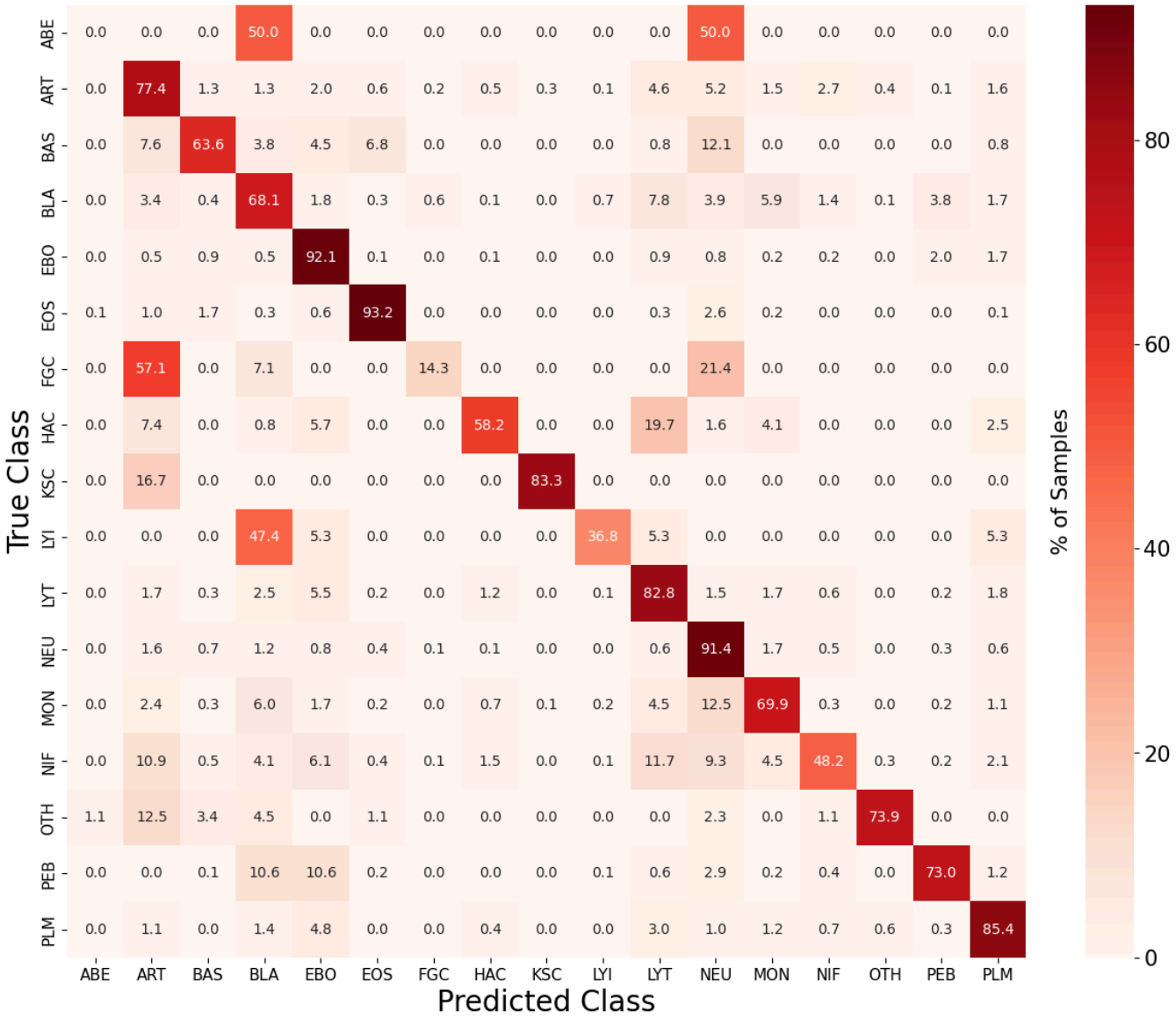


Figure 7.2.1.4: Normalised confusion matrix for the optimised model using data augmentation on the test set where the neutrophils (PMO, MYB, MMZ, NGB, and NGS) are grouped into one category NEU

One aspect that should be considered, however, is that the models employed in the study were not reflective of the state-of-the-art architectures prevalent in the field due to the technology limitations. By comparing the best-performing model with Tripathi's CoAtNet model used on the same dataset, it can be seen that CoAtNet can classify images more accurately in all classes except for EOS (See *Table 7.2.1.5*).

Class	CoAtNet (Tripathi et al.) Accuracy (%)	Best Performing Model Accuracy (%)	Distribution of Images in the Dataset (%)
ABE	**43**	0	<0.1
ART		**77**	11.5
BAS	**64**	63	0.3
BLA	**96**	68	7.0
EBO	**98**	92	16.0
EOS	85	**93**	3.4
FGC	**77**	14	<0.1
HAC	**93**	58	0.2
KSC		**83**	<0.1
LYI	**63**	37	<0.1
LYT	**91**	82	15.3
MMZ	**88**	49	1.8
MON	**77**	70	2.4
MYB	**85**	58	3.8
NGB	**96**	66	5.8
NGS	**97**	81	17.2
NIF		**48**	2.1
OTH		**74**	0.2
PEB	**84**	73	1.6
PLM	**94**	85	4.5
PMO	**97**	73	7.0

Table 7.2.1.5: Table showing comparison of accuracy (%) for each class between the CoAtNet architecture and the best performing model. Numbers in bold represent the best accuracy between the two models.

However, a significant distinction arises from the fact that Tripathi et al. did not utilise all classes such as ART, KSC, NIF, and OTH. Moreover, Tripathi et al. do not make it clear in their methodology whether a train-validation-test split or the mean of cross-validation was used to obtain these metrics. If cross-validation scores were used, the hyperparameters could be tuned to maximise accuracy on data that is not truly unseen, thereby making a one-to-one comparison unfair. Regardless, CoAtNet was able to achieve over a 90% accuracy for all classes that make up more than 5% of the original dataset whilst for all other classes was generally able to achieve over 80% accuracy. Cells like HAC where cancer is guaranteed also show a high level of accuracy with a 93% using CoAtNet.

Engström and Koutakis reduced the problem to only classify myelopoiesis class cells. This reduced MMZ, MON, MYB, NGB, and NGS misclassification as it has a higher F1-score than the best-performing model in these classes. This model may, however, be less applicable in the application of bone marrow classification as this drastically reduces the number of classes predicted from 21 to 9. Moreover, in terms of F1-score it still did not perform the best as the CoAtNet architecture used by Tripathi et al. obtained the highest F1-score for nearly all the classes it used. The Siamese Neural Network by Ananthakrishnan et al. also performed well, especially with minority classes.

As the best-performing model implemented in this investigation was based on Matek's sequential neural network, it is surprising to see that it achieved a higher F1-score across a majority of classes compared to the ResNeXt-50 architecture. This is because they claimed that the sequential model had worse performance than ResNeXt-50 (See *Table 7.2.1.6*). Moreover, Matek's F1-score is based on the mean five-fold cross-validation score as opposed to using a testing set, meaning this score is not reflective of how accurately the model performs on truly unseen data. This difference in scores between the best-performing model and ResNeXt-50 could show the lack of attention to tuning the hyperparameters to achieve the highest F1-score results possible.

Class	Ananthakrishnan et al.	Matek et al. (strict)	Engström (Gaussian)	Engström (Bernoulli)	Tripathi et al.	Best Performing Model
ABE	**70**	4			40	0
ART	**93**	78				82
BAS	34	23			**72**	30
BLA	89	70	85	84	**96**	72
EBO	88	85	96	96	**98**	90

EOS	87	88	**96**	**96**	89	90
FGC	59	27			**85**	22
HAC	68	49			**90**	39
KSC	77	43				19
LYI	**82**	14			65	7
LYT	75	79			**93**	85
MMZ	71	64	51	56	**90**	41
MON	54	63	66	70	**80**	57
MYB	84	55			**87**	57
NGB	86	65	68	69	**96**	62
NGS	**96**	80	91	91	**96**	85
NIF	**76**	40				48
OTH	32	35				**58**
PEB	72	60	74	71	**87**	65
PLM	89	82			**94**	80
PMO	91	74	89	89	**98**	76

Table 7.2.1.6: Table showing comparison of F1-score (%) for each class between the Siamese Neural Network used by Ananthakrishnan, ResNeXt-50 used by Matek et al, the sequential architecture used by Engström and Koutakis, CoAtNet used by Tripathi, and the best performing model from this project. Numbers in **bold** represent the best F1-Score for that class, numbers in red highlight a better F1-Score for the best performing model compared to ResNeXt-50.

The hyperparameter tuning took a total of 6 days, 5 hours, and 9 minutes without the use of data augmentation. However, data augmentation presented the most significant improvement in accuracy for minority classes. This means that hyperparameter tuning without data augmentation becomes a notable gap in the methodology as the best hyperparameters found in this study may not be reflective of the best hyperparameters found with the help of data augmentation. However, performing hyperparameter tuning with the use of data augmentation would have significantly increased the total time for random search to complete which is not feasible during this study. Moreover, the decision to limit the maximum number of epochs to 200 during random search inadvertently restricted the convergence of the models using the SGD optimiser with a learning rate of 0.00001. These combinations of hyperparameters took roughly 5

hours and 30 minutes to reach 200 epochs reaching a validation accuracy of roughly 60%. However, the SGD optimiser with a learning rate of 0.0001 was able to converge within 200 epochs and was not able to achieve a better validation accuracy than the ADAM optimiser with the same learning rate. This discovery should be enough to show that employing ADAM was more suitable for this type of problem as convergence can be achieved much faster. Another aspect was that a small set of hyperparameters were chosen to fine-tune. Tuning different hyperparameters may have also presented an increase in accuracy. However, the choice of hyperparameters was justified by gaps seen in the minimum viable analysis model.

7.2.2 Explainable AI Experiment

In the context of medical imaging classification, LIME is used to generate explanations that help validate that a network is correctly identifying the regions of interest to generate its prediction. For example, Ahsan et al. use LIME to view the most significant features predicted by the MobileNetV2 architecture on the detection of COVID-19 (1). Similarly, Schneider et al. also use LIME and MobileNetV2 in a classification task involving viral pneumonia in the same manner (6). These papers however fail to evaluate the explanations generated as they do not address whether these most significant features are sufficient enough to classify the correct output or not. Some papers do however evaluate whether the LIME explanations justify a model's output. An example of this is Cervantes and Chan's paper where COVID-19 chest x-rays contain a segmented region signifying the region of interest. LIME explanations were then generated on four different models and were compared to the segmented regions to evaluate whether these models were identifying the correct regions. However, this form of evaluation is not possible with this dataset as images contained no labels concerning regions of interest.

Outside of medical image classification, Shah and Sheppard perform a sufficient justification experiment on two different models - one that classified dogs and cats, and the other with different types of flowers. This was done by measuring the precision of each class at differing numbers of superpixels. If the precision for that class at a given number of superpixels n was ⅔ of the precision for all superpixels activated, then an explanation with n superpixels was enough to sufficiently justify that class. From this, they discovered that for the cats and dogs problem, sufficient justification could be achieved with 1 and 15 superpixels for each class respectively. For the flower problem, three out of five of the classes achieved sufficient justification in 5 or fewer superpixels activated. The reasoning for this is most likely due to the network's ability to utilise texture as a distinguishing factor in determining the classes. By

looking at *Figure 7.2.2.1*, with as few as 1 superpixels, it is obvious that the feature generated is the nose of a dog. Incrementally increasing the number of superpixels further clarifies that the image is a dog.

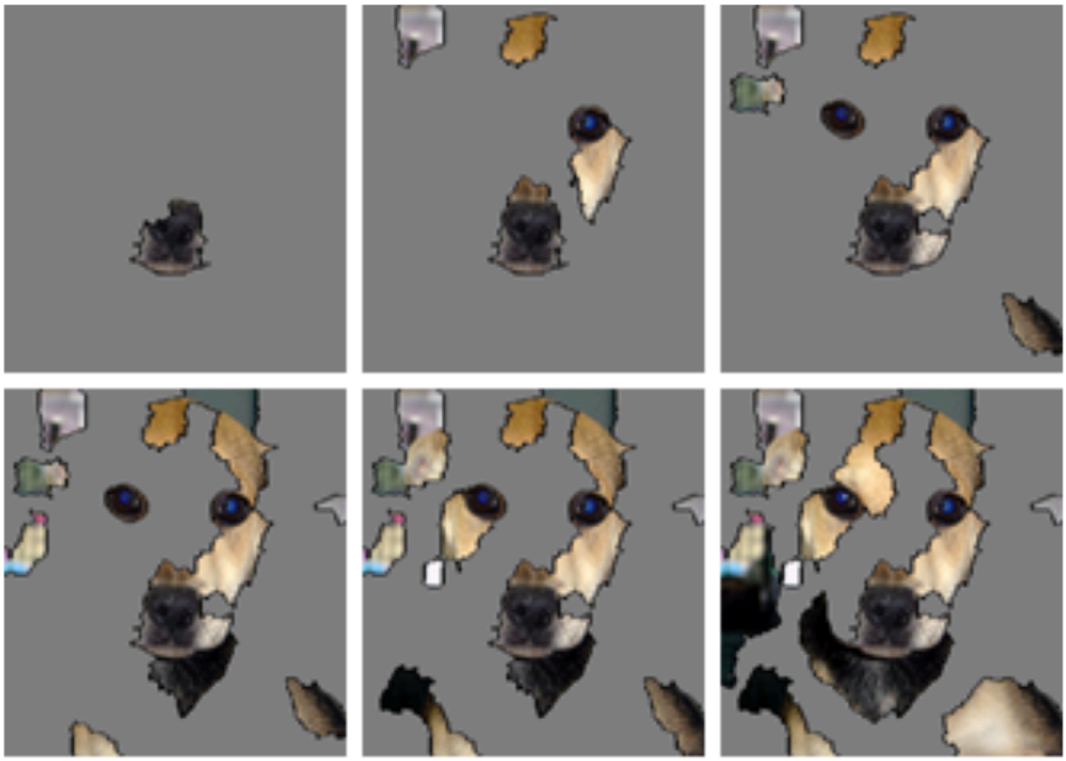


Figure 7.2.2.1: Image showing an example dog explanation generated by LIME using 1, 5, 10, 15, and 25 superpixels on the Inception-V3 architecture (Shah and Sheppard)

Referring back to the cell explanations generated by LIME in *Figures 7.2.2.2* to *7.2.2.4*. We can see that with 25 superpixels activated, the occlusions present make the images appear homogeneous in texture. This not only makes it difficult for the model to classify but makes the explanations difficult for humans to interpret as well. This is because as mentioned by Shah and Sheppard, increasing the number of superpixels included in the explanations also increases the number of noise superpixels presented. Despite the use of superpixels in an attempt to keep morphological structure present in the images, *Figures 7.2.2.2* to *7.2.2.4* highlight that the explanations often occlude large sections of the cell the model is trying to identify. This could be due to several differing factors. The first reason is that the model identifies the noisy superpixels as a key factor in coming up with its prediction as opposed to the cell itself. This would mean the model should not be deployed in a real-world scenario as the reasoning behind the model's prediction is incorrect. Another reason that the explanations may occlude large sections of the cell is that the number of perturbations generated for each image was not enough to come up with a good explanation

for that image. The perturbations are fed back into the original model to generate a set of predictions which are then used to come up with the most significant superpixels based on a linear regression model. With more perturbations per image, the more accurate the explanations. In Shah and Sheppard's experiment, they generated 1,000 perturbed images for each image. This took a total of 10,000 computational hours. Due to the time constraint of the senior honours project, only 128 perturbations per image were generated for each image. This is just over a tenth of the perturbations generated per image. This means that the explanations generated are probably not as accurate as they could be. Further studies should be done to see if increasing the number of perturbations per image would see an improvement in precision and accuracy results.

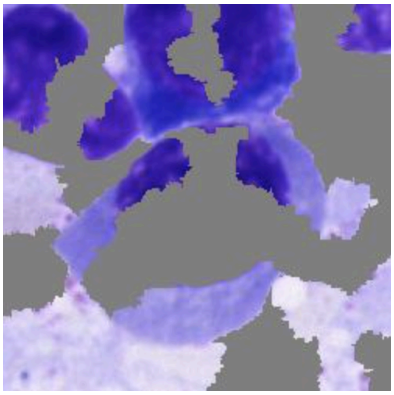


Figure 7.2.2.2: EBO with 25 superpixels activated

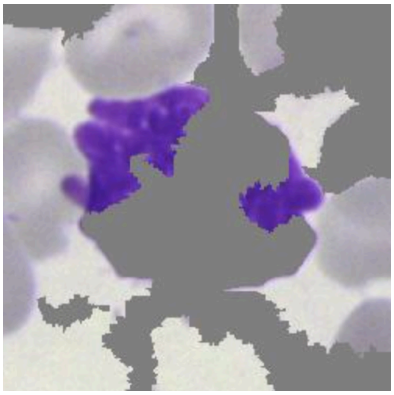


Figure 7.2.2.3: KSC with 25 superpixels activated

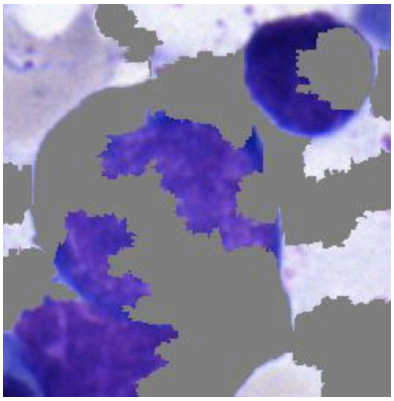


Figure 7.2.2.4: PEB with 25 superpixels activated

One issue identified regarding the methodology of sufficient justification is the use of the ⅔ precision threshold. Shah and Sheppard state that "if the explanations perform at least ⅔ as well as the original images. We hypothesise that explanations generated by the LIME framework for image classifications made by a CNN will provide sufficient justification for the classifications made and will therefore convey a complete representation of the information used to make classification decisions" (Shah and Sheppard). The use of the ⅔ threshold is arbitrary and lacks empirical or theoretical grounding. While Shah and Sheppard propose this threshold as a measure of sufficient justification, there is no clear rationale provided for why precisely ⅔ precision should signify an adequate level of explanation. Moreover, the effectiveness of the LIME framework in providing justifications for image classifications may not necessarily correlate with a specific threshold of performance. Therefore, further investigation and validation are warranted to determine the appropriateness of this threshold in evaluating the sufficiency of explanations provided by the LIME framework.

7.3 Evaluation

This project met all the primary and secondary objectives identified in Chapter 1.2, below is a summary table describing the work completed in meeting the objective.

Objective	Work Completed to Meet the Objective
P1: Perform a literature review on existing machine learning classification models using the Bone Marrow Cytology in Hematologic Malignancies dataset (and other datasets) to identify their state-of-the-art, advantages and disadvantages, as well as potential research gaps	The literature review in Chapter 2.1 explores the use of various machine learning models that use the Bone Marrow Cytology in the Hematologic dataset. This includes work by Matek et al., Ananthakrishnan et al., and Engström and Koutakis. Other classification models utilising other datasets were also explored in this Chapter. These methods were compared and contrasted to obtain their advantages and disadvantages. Chapter 2.1.2 also explores the challenges present in the field and identifies a few research gaps present.
P2: Create an unoptimised minimum viable analysis model that can classify the type of bone marrow cell in a given image	Chapter 4.5 presents a minimum viable analysis model based on Matek et al.'s sequential CNN.
P3: Create extension(s) of the minimum viable analysis model using different methods and parameters to optimise the model for the accuracy of predictions	Chapter 4.6 explores hyperparameter tuning to trial 60 different combinations of the model to find the best combination of hyperparameters to maximise accuracy.
P4: Perform an in-depth investigation around a research question informed by the gaps identified by the literature review.	For the medical field to trust the use of AI technology, having explainable and interpretable AI is necessary. Through the literature review, it became clear that this was often not considered by many researchers in this area. Even when explainable AI was considered, it was not properly evaluated. Chapter 6 explored an evaluation strategy for LIME explanations generated.
S1: Explore the impact data augmentation has on the accuracy of the model	Chapter 4.7 explores the use of data augmentation to improve the accuracy of the resulting model generated in Chapter 4.6.

S2: Compare and contrast the models implemented throughout the investigation	The progression of Chapter 4 justifies the methodology of improving upon the previous model. The following results are then presented and compared against the previous model based on the validation set. Chapter 5 also presents the test results on the model which are then later interpreted and compared in the Evaluation section.

Table 7.3.1: Evaluation table describing how the project objectives were met

8. Conclusion

One in every 16 men and one in every 22 women will develop blood cancer at some point in their lives (Blood Cancer UK). Therefore, one of the aims of this project was to investigate whether deep learning could assist hematologists in classifying bone marrow cells as these help diagnose various types of blood cancers. This was achieved by implementing a convolutional neural network (CNN) used by Matek et al. to train on images from the "Bone Marrow Cytology in Hematological Malignancy" dataset. The literature review questioned the efficacy of data augmentation in enhancing the model's performance as this was not conclusively examined in Matek's research. However, through iterative refinement involving hyperparameter tuning and data augmentation techniques, it was observed that the best-performing model with data augmentation achieved a validation accuracy of 78.2%, surpassing the optimised model without augmentation and minimum viable analysis model which achieved 76.0% and 74.0% accuracy respectively. Furthermore, the test results using the augmented model obtained an accuracy of 78.3%, indicating its ability to generalise effectively to truly unseen data. With manual examinations being costly and prone to a high level of error of 30-40% in some instances (Reta et al. 2), this model could greatly improve the speed and accuracy at which hematologists classify these cells to then come up with a diagnosis, regardless of it being out-performed by the existing state-of-the-art architectures. Moreover, if we consider that the model often confuses cells that are related to each other, we can be confident that the model's predictions are at least accurate in terms of a larger subgrouping of cells. However as CNNs are black-box models, there are clear ethical concerns in the application of these models as medical practitioners using them will not understand why a prediction for a certain cell's class was made. Furthermore, the consequence of false predictions can drastically change the course of action taken for patients who are incorrectly misdiagnosed. Therefore, it is of utmost importance to provide interpretable results to build trust and understanding of these technologies before deployment.

Many papers involving the classification of medical images fail to address the interpretability issue. Papers that do attempt to address these issues through the use of local interpretable model-agnostic explanations (LIME) however, fail to question whether these explanations provide a sufficient enough justification for why the model is predicting certain classes. This project addresses this issue as a sufficient justification experiment based on Shah and Sheppard's research was performed to find the number of superpixels required to meet the ⅔ precision threshold. It was discovered that 40 superpixels were required before sufficient justification was reached, meaning this model could only handle a small number of occlusions within the image to come up with a good prediction. This may signify two

conclusions: either the model requires an image's background to correctly predict cells or that LIME is not the most appropriate method to justify a model's behaviour to classify single-celled images. If the former is true, this signals that the model should not be deployed in practice as the reasoning behind its classification of cells is flawed. If the latter was true, other explainable AI tools should be investigated and compared against LIME. However, this sufficient justification discovery may be limited due to the time constraints of the senior honours project. In Shah and Sheppard's experiment, 1,000 perturbations per image were created, taking over 10,000 computational hours. This was not feasible to replicate and thus, only 128 perturbations per image were created in this experiment. Overall, this may impact the quality of explanations generated and therefore skew the results towards needing a larger number of superpixels to achieve sufficient justification. Future research should explore the effect of increasing the number of perturbations per image to see whether the same trend holds. Future research could also use this technique on state-of-the-art architectures to compare whether fewer superpixels are needed to achieve sufficient justification for an explanation. Exploring the effect of removing artefacts (ART) and unidentifiable cells (NIF) from the dataset to see what the model would predict in the absence of these classes is also a possible future avenue. Moreover, the ⅔ threshold proposed by Shah and Sheppard was arbitrarily chosen without a clear rationale. Future work aimed at devising a more robust metric could offer deeper insights into the notion of sufficient justification facilitated by LIME, consequently assisting in mitigating the interpretability gap present in CNNs.

Appendices

A. User Manual

The code for this project is oben and publicly accessible through GitHub:
https://github.com/ejml1/Deep-Learning-for-Cancer-Detection

A.1 Setup

The dataset has not been included in the submission as it is too large (6.8 GB). It can be downloaded using the IBM Aspera Connect plugin from the following link:
https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=101941770

The downloaded dataset should be named "BM_cytomorphology_data" and placed in the submission folder for the following steps.

The submission contains a Dockerfile used to create a container containing all the required dependencies. The following commands can be used to build and run the container. The following code should then be all run on the docker container's command line:

```
docker build -t model .

docker run -v
<replace-with-path-to-the-following-directory>/Deep-Learning-for-Cancer-D
etection:/Deep-Learning-for-Cancer-Detection -w
/Deep-Learning-for-Cancer-Detection --gpus 1 --shm-size=1g -it -p 8888:8888
--rm model
```

In the Preprocess directory, run the following command to remove the identified corrupted images from the dataset:

```
python DeleteCorrupted.py
```

The dataset can then be split into the train, validation, and test subsets by creating 2 directories for the validation and test subsets and running the following command. For the purpose of training and testing the model for execution, the 2 directories should be name “Validation” and “Test”:

```
python Split.py
```

This script will then ask for the train (the BM_cytomorphology_data directory), validation, and test directories to be input.

To create reproducible results, data augmentation was not performed on the fly. To augment images, copy or rename the “BM_cytomorphology_data” directory to “BM_cytomorphology_data_augmented” should be created. The following command can then be run (not this will take a long time):

```
python AugmentImages.py
```

To generate explanations to perform the LIME experiment, the following command can be run after creating a Explanations/Validation directory:

```
./CreatePerturbations.sh
```

A.2 Execution

Before training the optimised model, the following directories should be created. This is used to save the model itself and its training history as a pickle file:

```
pickle/augmented
```

The following script can then be run in the docker container’s command line to train the model, note that the expected training directory is “BM_cytomorphology_data_augmented”. Therefore, if the data has not been augmented, it should be renamed to “BM_cytomorphology_data_augmented” anyways.

```
python OptimisedModel.py
```

The LIMEResults notebook can be run through JupyterLab in the docker container to produce the LIME experiment results.

B. Ethics Approval

University of St Andrews | FOUNDED 1413

School of Computer Science Ethics Committee

11 October 2023

Dear Ethan,

Thank you for submitting your ethical application which was considered by the School Ethics Committee.

The School of Computer Science Ethics Committee, acting on behalf of the University Teaching and Research Ethics Committee (UTREC), has approved this application:

Approval Code:	CS17271	**Approved on:**	11.10.23	**Approval Expiry:**	11.10.28
Project Title:	Deep Learning for Cancer Detection (and or Segmentation) in Medical Imaging				
Researcher(s):	Ethan Joseph Medina Li				
Supervisor(s):	David Harris-Birtill				

The following supporting documents are also acknowledged and approved:

1. Application Form

Approval is awarded for 5 years, see the approval expiry data above.

If your project has not commenced within 2 years of approval, you must submit a new and updated ethical application to your School Ethics Committee.

If you are unable to complete your research by the approval expiry date you must request an extension to the approval period. You can write to your School Ethics Committee who may grant a discretionary extension of up to 6 months. For longer extensions, or for any other changes, you must submit an ethical amendment application.

You must report any serious adverse events, or significant changes not covered by this approval, related to this study immediately to the School Ethics Committee.

Approval is given on the following conditions:

- that you conduct your research in line with:
  - the details provided in your ethical application
  - the University's Principles of Good Research Conduct
  - the conditions of any funding associated with your work
- that you obtain all applicable additional documents (see the 'additional documents' webpage for guidance) before research commences.

You should retain this approval letter with your study paperwork.

Yours sincerely,

Wendy Boyter

SEC Administrator

School of Computer Science Ethics Committee
Dr Olexandr Konovalov/Convenor, Jack Cole Building, North Haugh, St Andrews, Fife, KY16 9SX
Telephone: 01334 463273 Email: ethics-cs@st-andrews.ac.uk
The University of St Andrews is a charity registered in Scotland: No SC013532

C. Practice Analysis Model

Before the main minimum viable analysis model, a practice model was created but was not used for the purpose of analysis (See BasicModel.ipynb). Below is the information gathered from that model:

C.1 Data Split

Class	Number of Images	Distribution %
ABE	5	0.030
ART	1,120	6.823
BAS	248	1.511
BLA	1,120	6.823
EBO	1,120	6.823
EOS	1,120	6.823
FGC	27	0.164
HAC	230	1.401
KSC	24	0.146
LYI	37	0.225
LYT	1,120	6.823
MMZ	1,120	6.823
MON	1,120	6.823
MYB	1,120	6.823
NGB	1,120	6.823
NGS	1,120	6.823
NIF	1,120	6.823
OTH	165	1.005
PEB	1,120	6.823
PLM	1,120	6.823
PMO	1,120	6.823

C.2 Model Architecture

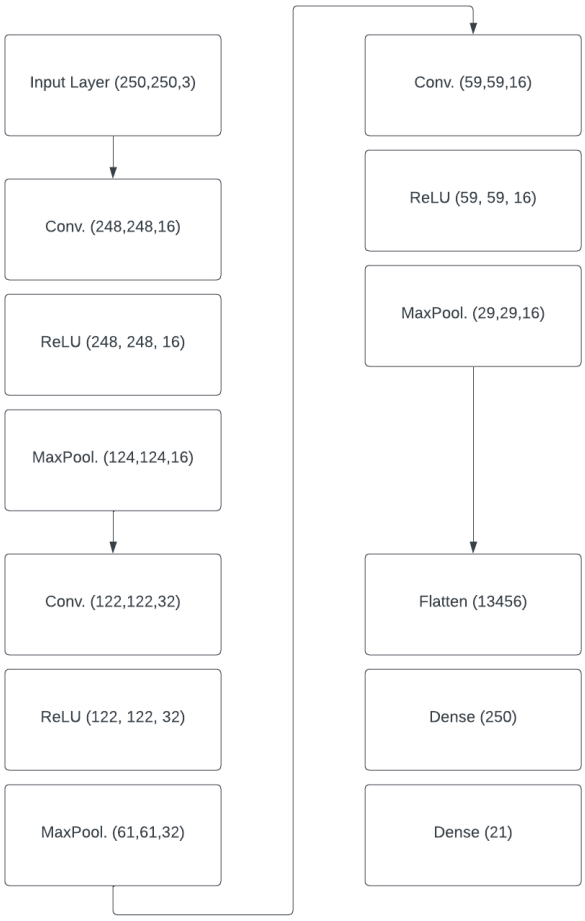


Figure C.2.1: The general architecture of the practice model with the parentheses indicating (width, height, and feature map channels). The matrix becomes a vector after the Flatten layer; the number of nodes is indicated by the parentheses.

C.3 Training Results

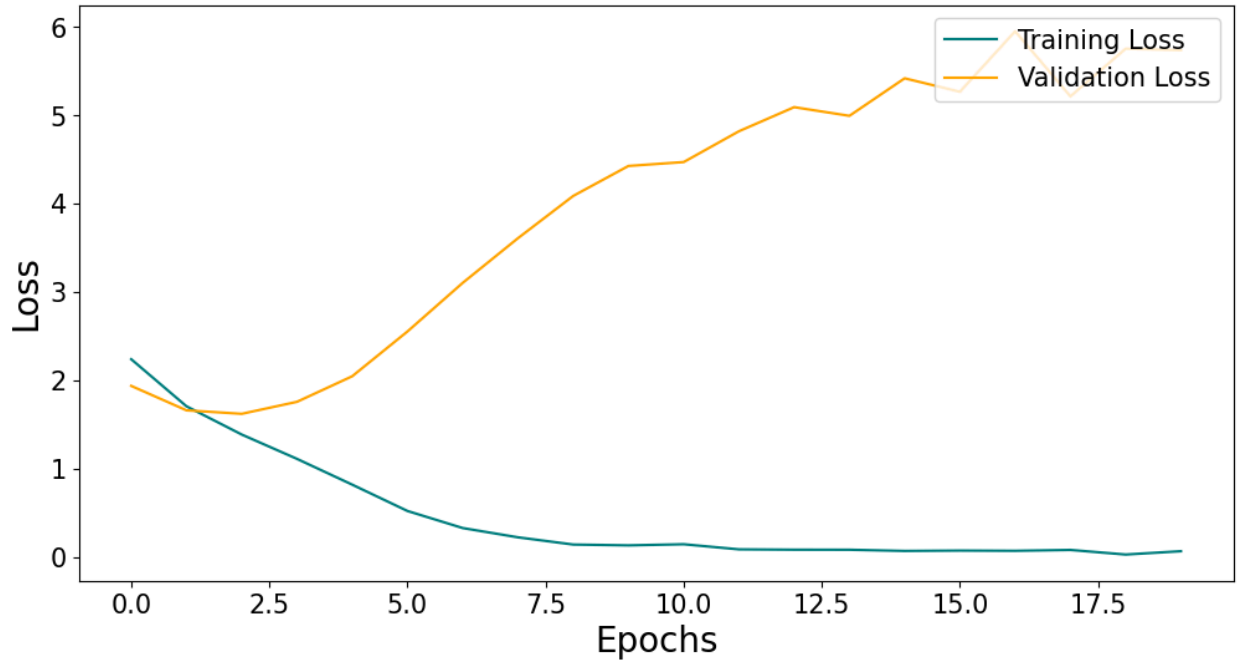


Figure C.3.1: Training and validation loss for practice model

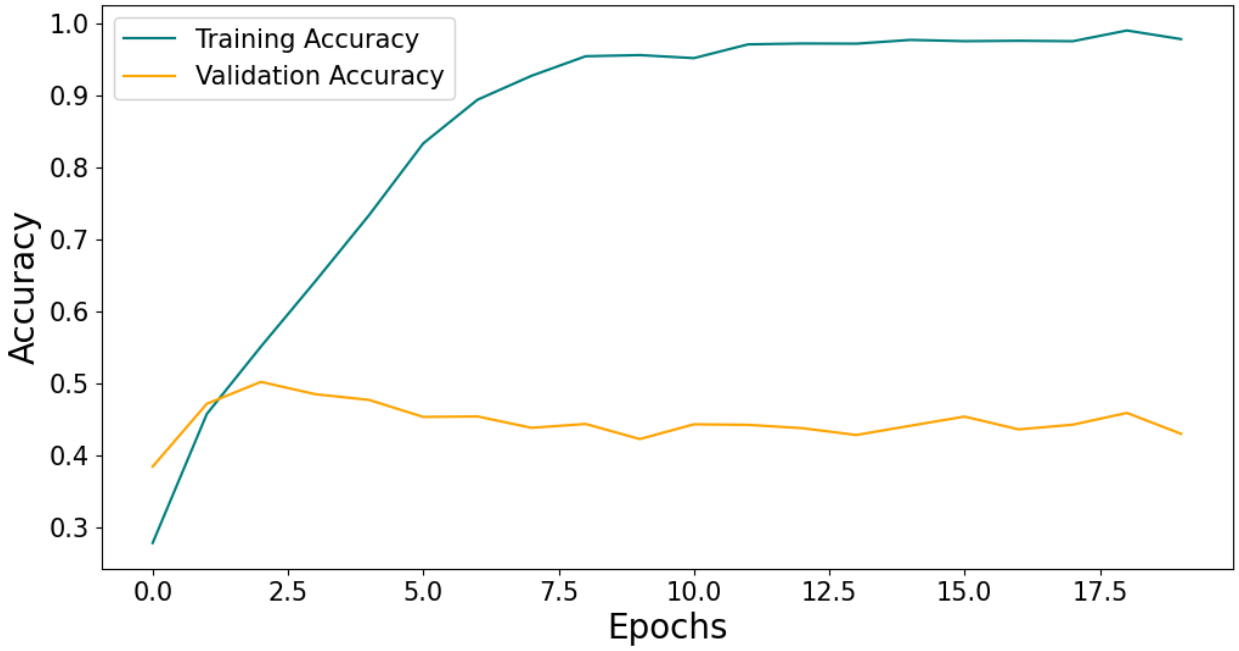


Figure C.3.2: Training and validation accuracy for practice model

C.4 Validation Results

Performance Metric	Result %
Accuracy	42.9
Precision	43.8
Recall	42.1
F1-Score	42.9

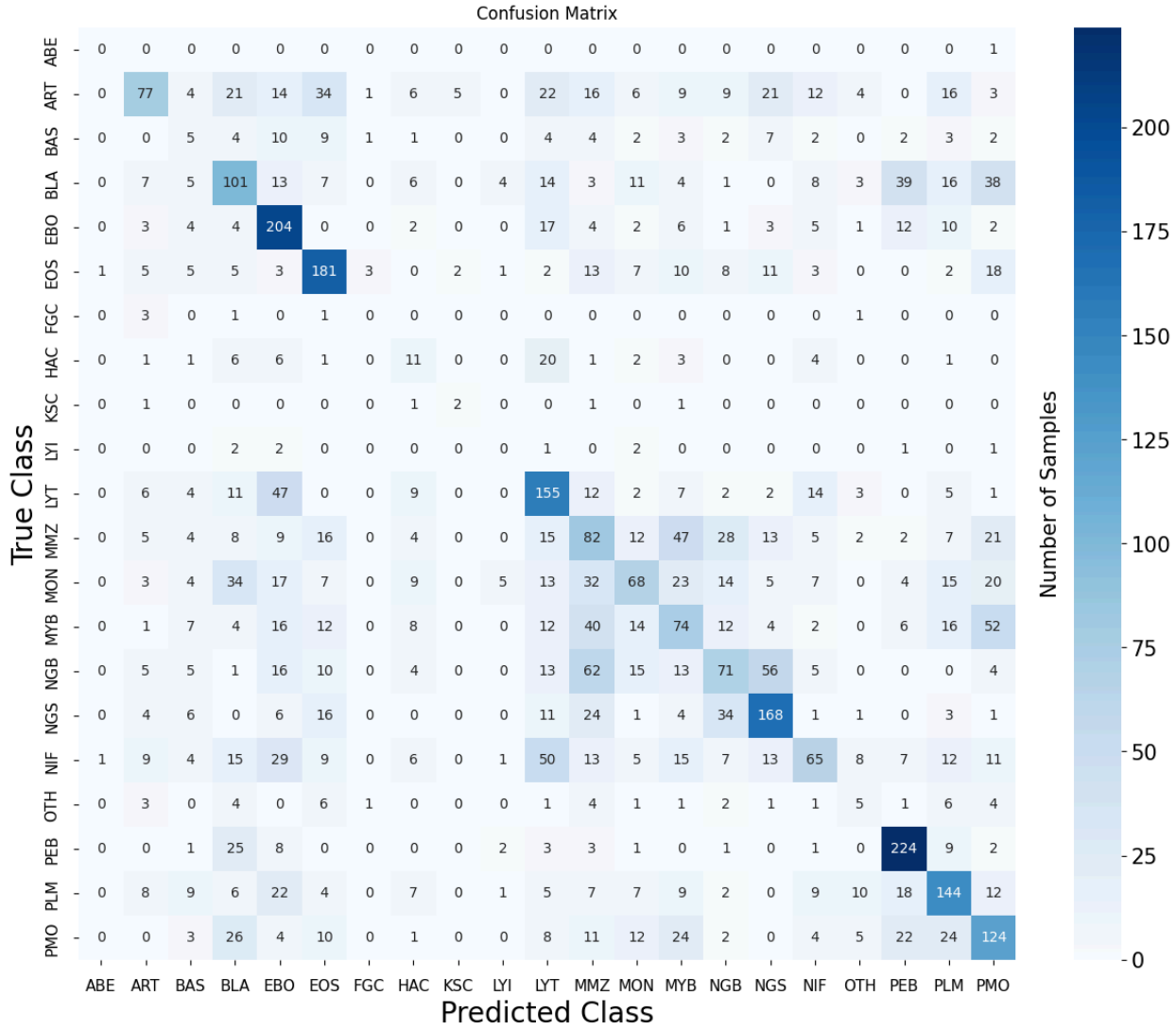


Figure C.4.1: Confusion matrix for the practice model

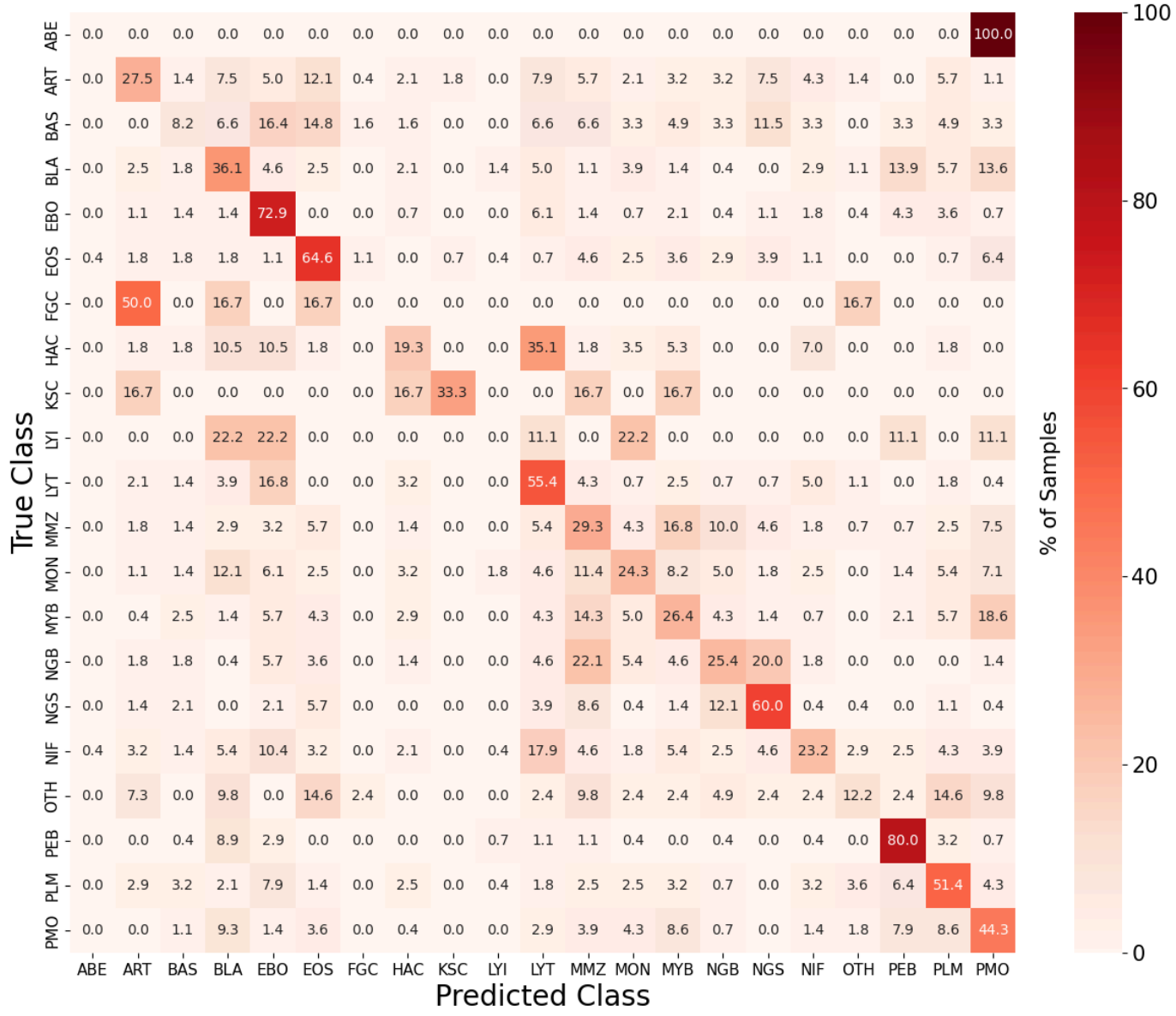


Figure C.4.2: Normalised confusion matrix for the practice model

Classification Report:

	precision	recall	f1-score	support
ABE	0.00	0.00	0.00	1
ART	0.55	0.28	0.37	280
BAS	0.07	0.08	0.08	61
BLA	0.36	0.36	0.36	280
EBO	0.48	0.73	0.58	280
EOS	0.56	0.65	0.60	280
FGC	0.00	0.00	0.00	6
HAC	0.15	0.19	0.17	57
KSC	0.22	0.33	0.27	6
LYI	0.00	0.00	0.00	9
LYT	0.42	0.55	0.48	280
MMZ	0.25	0.29	0.27	280
MON	0.40	0.24	0.30	280
MYB	0.29	0.26	0.28	280
NGB	0.36	0.25	0.30	280
NGS	0.55	0.60	0.58	280
NIF	0.44	0.23	0.30	280
OTH	0.12	0.12	0.12	41
PEB	0.66	0.80	0.72	280
PLM	0.50	0.51	0.51	280
PMO	0.39	0.44	0.42	280
micro avg	0.43	0.43	0.43	4101
macro avg	0.32	0.33	0.32	4101
weighted avg	0.43	0.43	0.42	4101

Figure C.4.3: Classification report showing precision, recall, and F1-score for the practice model

Works Cited

Matek, Christian, et al. "Highly accurate differentiation of bone marrow cell morphologies using deep neural networks on a large image data set." *Blood, The Journal of the American Society of Hematology* 138.20 (2021): 1917-1927.

Matek, C., Krappe, S., Münzenmayer, C., Haferlach, T., & Marr, C. (2021). An Expert-Annotated Dataset of Bone Marrow Cytology in Hematologic Malignancies [Data set]. The Cancer Imaging Archive. https://doi.org/10.7937/TCIA.AXH3-T579

Amin, Morteza Moradi et al. “Recognition of acute lymphoblastic leukemia cells in microscopic images using k-means clustering and support vector machine classifier.” Journal of medical signals and sensors vol. 5,1 (2015): 49-58.

Reta, Carolina, et al. "Segmentation of bone marrow cell images for morphological classification of acute leukemia." *Twenty-Third International FLAIRS Conference*. 2010.

Swati, Zar Nawab Khan, et al. "Brain tumor classification for MR images using transfer learning and fine-tuning." *Computerized Medical Imaging and Graphics* 75 (2019): 34-46.

Raghu, Maithra, et al. "Transfusion: Understanding transfer learning for medical imaging." *Advances in neural information processing systems* 32 (2019).

Abunadi, Ibrahim, and Ebrahim Mohammed Senan. "Multi-method diagnosis of blood microscopic sample for early detection of acute lymphoblastic leukemia based on deep learning and hybrid techniques." *Sensors* 22.4 (2022): 1629.

Rahman, M. Mostafizur, and Darryl N. Davis. "Addressing the class imbalance problem in medical datasets." *International Journal of Machine Learning and Computing* 3.2 (2013): 224.

Ananthakrishnan, Balasundaram, et al. "Automated Bone Marrow Cell Classification for Haematological Disease Diagnosis Using Siamese Neural Network." *Diagnostics* 13.1 (2022): 112.

Abir, Wahidul Hasan et al. “Explainable AI in Diagnosing and Anticipating Leukemia Using Transfer Learning Method.” Computational intelligence and neuroscience vol. 2022 5140148. 27 Apr. 2022, doi:10.1155/2022/5140148. **Retracted**.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 618-626, doi: 10.1109/ICCV.2017.74.

Alzubaidi, Laith et al. “Novel Transfer Learning Approach for Medical Imaging with Limited Labeled Data.” Cancers vol. 13,7 1590. 30 Mar. 2021, doi:10.3390/cancers13071590

Fallah, Mahdi. *Cancer Incidence in Five Provinces of Iran: Ardebil, Gilan, Mazandaran, Golestan and Kerman, 1996-2000*. Tampere University Press, 2007.

Zhu, Jiaxin et al. “Trends in mortality and causes of death among Chinese adolescents aged 10-19 years from 1990 to 2019.” Frontiers in public health vol. 11 1075858. 7 Feb. 2023, doi:10.3389/fpubh.2023.1075858

Basymeleh, Aiman Muhamad, Bagus Esa Pramudya, and Reinato Teguh Santoso. "Acute Lymphoblastic Leukemia Image Classification Performance with Transfer Learning Using CNN Architecture." 2022 4th International Conference on Biomedical Engineering (IBIOMED). IEEE, 2022.

Ahmed, Nizar A., et al. “Identification of Leukemia Subtypes From Microscopic Images Using Convolutional Neural Network.” *Diagnostics*, vol. 9, no. 3, Multidisciplinary Digital Publishing Institute, Aug. 2019, p. 104. https://doi.org/10.3390/diagnostics9030104.

Amin, Morteza Moradi. “Recognition of Acute Lymphoblastic Leukemia Cells in Microscopic Images Using K-Means Clustering and Support Vector Machine Classifier.” *PubMed Central (PMC)*, 1 Mar. 2015, www.ncbi.nlm.nih.gov/pmc/articles/PMC4335145.

Engström, Koutakis. “Comparing Dropout Regularization Algorithms for Convolutional Neural Networks Identifying Malignant Cells for Diagnosis of Leukemia.” *DIVA*, 2023, www.diva-portal.org/smash/record.jsf?dswid=-3420&pid=diva2%3A1763398.

Guennec, Arthur Le. *Data Augmentation for Time Series Classification Using Convolutional Neural Networks*. 19 Sept. 2016, shs.hal.science/halshs-01357973.

Loey, Mohamed, et al. “Deep Transfer Learning in Diagnosing Leukemia in Blood Cells.” *Computers*, vol. 9, no. 2, Multidisciplinary Digital Publishing Institute, Apr. 2020, p. 29. https://doi.org/10.3390/computers9020029.

Meem, Raisa Fairooz, and Khandaker Tabin Hasan. “Bone Marrow Cytomorphology Cell Detection Using InceptionResNetV2.” *arXiv (Cornell University)*, Cornell University, May 2023, https://doi.org/10.48550/arxiv.2305.05430.

Nirthika, Rajendran, et al. “Pooling in Convolutional Neural Networks for Medical Image Analysis: A Survey and an Empirical Study.” *Neural Computing and Applications*, vol. 34, no. 7, Feb. 2022, pp. 5321–47. https://doi.org/10.1007/s00521-022-06953-8.

Phung, Van Hiep, and Eun Joo Rhee. “A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small

Datasets." *Applied Sciences*, vol. 9, no. 21, Oct. 2019, p. 4500. https://doi.org/10.3390/app9214500.

Rajpurohit, Subhash, et al. "Identification of Acute Lymphoblastic Leukemia in Microscopic Blood Image Using Image Processing and Machine Learning Algorithms." *IEEE Conference Publication | IEEE Xplore*, 1 Sept. 2018, ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8554576. Accessed 10 Oct. 2023.

Rehman, Amjad, et al. "Classification of Acute Lymphoblastic Leukemia Using Deep Learning." *Microscopy Research and Technique*, vol. 81, no. 11, Wiley-Blackwell, Oct. 2018, pp. 1310–17. https://doi.org/10.1002/jemt.23139.

Shin, Younghak, and Ilangko Balasingham. "Comparison of Hand-craft Feature Based SVM and CNN Based Deep Learning Framework for Automatic Polyp Classification." *Comparison of Hand-craft Feature Based SVM and CNN Based Deep Learning Framework for Automatic Polyp Classification*, July 2017, https://doi.org/10.1109/embc.2017.8037556.

'Blood Cancer UK | Facts and Information about Blood Cancer'. Blood Cancer UK, https://bloodcancer.org.uk/news/blood-cancer-facts/. Accessed 18 Mar. 2024.

Uyl-de Groot, C. A., et al. 'Costs of Diagnosis, Treatment, and Follow up of Patients with Acute Myeloid Leukemia in the Netherlands'. Journal of Hematotherapy & Stem Cell Research, vol. 10, no. 1, Feb. 2001, pp. 187–92. PubMed, https://doi.org/10.1089/152581601750098499.

Kolarik, Michal, et al. 'Explainability of Deep Learning Models in Medical Video Analysis: A Survey'. PeerJ Computer Science, vol. 9, Mar. 2023, p. e1253. PubMed Central, https://doi.org/10.7717/peerj-cs.1253.

Ying, Xue. ‘An Overview of Overfitting and Its Solutions’. Journal of Physics: Conference Series, vol. 1168, no. 2, Feb. 2019, p. 022022. Institute of Physics, https://doi.org/10.1088/1742-6596/1168/2/022022.

“Updated on Blood Cancer.” *Lukemia & Lymphoma Society*, www.lls.org/sites/default/files/2023-08/PS80_Facts_2022_2023.pdf. Accessed 22 Mar. 2024.

Hairy Cell Leukemia Treatment - NCI. https://www.cancer.gov/types/leukemia/patient/hairy-cell-treatment-pdq. Accessed 19 Feb. 2024.

Higuchi, Toru, et al. ‘Smudge Cells Due to Infectious Mononucleosis’. IDCases, vol. 23, Jan. 2021, p. e01057. PubMed Central, https://doi.org/10.1016/j.idcr.2021.e01057.

Https://Www.Lls.Org/Myelodysplastic-Syndromes/Diagnosis. https://www.lls.org/myelodysplastic-syndromes/diagnosis. Accessed 19 Feb. 2024.

Plasma Cell Neoplasms (Including Multiple Myeloma) Treatment - NCI. 19 Jan. 2024, https://www.cancer.gov/types/myeloma/patient/myeloma-treatment-pdq. nciglobal,ncienterprise.

Taqi, Syed Ahmed, et al. ‘A Review of Artifacts in Histopathology’. Journal of Oral and Maxillofacial Pathology : JOMFP, vol. 22, no. 2, 2018, p. 279. PubMed Central, https://doi.org/10.4103/jomfp.JOMFP_125_15.

Grandini, Margherita, et al. Metrics for Multi-Class Classification: An Overview. arXiv:2008.05756, arXiv, 13 Aug. 2020. arXiv.org, http://arxiv.org/abs/2008.05756.

Krappe, Sebastian, et al. ‘Automated Morphological Analysis of Bone Marrow Cells in Microscopic Images for Diagnosis of Leukemia: Nucleus-Plasma Separation and Cell

Classification Using a Hierarchical Tree Model of Hematopoesis'. Medical Imaging 2016: Computer-Aided Diagnosis, vol. 9785, SPIE, 2016, pp. 856–61. www.spiedigitallibrary.org, https://doi.org/10.1117/12.2216037.

Akosa, Josephine. Predictive Accuracy: A Misleading Performance Measure for Highly Imbalanced Data.

Dalianis, Hercules. 'Evaluation Metrics and Evaluation'. Clinical Text Mining: Secondary Use of Electronic Patient Records, edited by Hercules Dalianis, Springer International Publishing, 2018, pp. 45–53. Springer Link, https://doi.org/10.1007/978-3-319-78503-5_6.

Rebuffi, Sylvestre-Alvise, et al. 'Data Augmentation Can Improve Robustness'. Advances in Neural Information Processing Systems, vol. 34, Curran Associates, Inc., 2021, pp. 29935–48. Neural Information Processing Systems, https://proceedings.neurips.cc/paper/2021/hash/fb4c48608ce8825b558ccf07169a3421-Abstract.html.

Wu, Yanzhao, et al. Demystifying Learning Rate Policies for High Accuracy Training of Deep Neural Networks. arXiv:1908.06477, arXiv, 26 Oct. 2019. arXiv.org, http://arxiv.org/abs/1908.06477.

You, Kaichao, et al. How Does Learning Rate Decay Help Modern Neural Networks? arXiv:1908.01878, arXiv, 26 Sept. 2019. arXiv.org, http://arxiv.org/abs/1908.01878.

Baldi, Pierre, and Peter J. Sadowski. 'Understanding Dropout'. Advances in Neural Information Processing Systems, vol. 26, Curran Associates, Inc., 2013. Neural Information Processing Systems,

https://proceedings.neurips.cc/paper_files/paper/2013/hash/71f6278d140af599e06ad9bf1ba03cb0-Abstract.html.

Bergstra, James, and Yoshua Bengio. Random Search for Hyper-Parameter Optimization.

Kandel, Ibrahem, and Mauro Castelli. 'The Effect of Batch Size on the Generalizability of the Convolutional Neural Networks on a Histopathology Dataset'. ICT Express, vol. 6, no. 4, Dec. 2020, pp. 312–15. ScienceDirect, https://doi.org/10.1016/j.icte.2020.04.010.

Lauriola, Ivano. 'On the Impact of Early Stopping in Multiple Kernel Learning'. Proceedings of the Future Technologies Conference (FTC) 2020, Volume 1, edited by Kohei Arai et al., Springer International Publishing, 2021, pp. 205–15. Springer Link, https://doi.org/10.1007/978-3-030-63128-4_16.

Raitoharju, Jenni. 'Chapter 3 - Convolutional Neural Networks'. Deep Learning for Robot Perception and Cognition, edited by Alexandros Iosifidis and Anastasios Tefas, Academic Press, 2022, pp. 35–69. ScienceDirect, https://doi.org/10.1016/B978-0-32-385787-1.00008-7.

Smith, Samuel L., et al. Don't Decay the Learning Rate, Increase the Batch Size. arXiv:1711.00489, arXiv, 23 Feb. 2018. arXiv.org, http://arxiv.org/abs/1711.00489.

Srivastava, Nitish, et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting.

Wu, Haibing, and Xiaodong Gu. 'Towards Dropout Training for Convolutional Neural Networks'. Neural Networks, vol. 71, Nov. 2015, pp. 1–10. ScienceDirect, https://doi.org/10.1016/j.neunet.2015.07.007.

Zheng, Alice. How to Evaluate Machine Learning Models: Hyperparameter Tuning. 27 May 2015,

https://web.archive.org/web/20160701182750/http://blog.dato.com/how-to-evaluate-machine-learning-models-part-4-hyperparameter-tuning.

Zhou, Pan, et al. 'Towards Theoretically Understanding Why Sgd Generalizes Better Than Adam in Deep Learning'. Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 21285–96. Neural Information Processing Systems, https://proceedings.neurips.cc/paper/2020/hash/f3f27a324736617f20abbf2ffd806f6d-Abstract.html.

Ahsan, Md Manjurul, et al. 'Detection of COVID-19 Patients from CT Scan and Chest X-Ray Data Using Modified MobileNetV2 and LIME'. Healthcare, vol. 9, no. 9, 9, Sept. 2021, p. 1099. www.mdpi.com, https://doi.org/10.3390/healthcare9091099.

Cervantes, Eduardo Gasca, and Wai-Yip Chan. 'LIME-Enabled Investigation of Convolutional Neural Network Performances in COVID-19 Chest X-Ray Detection'. 2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), IEEE, 2021, pp. 1–6. DOI.org (Crossref), https://doi.org/10.1109/CCECE53047.2021.9569029.

McInnes, Elizabeth. 'Artefacts in Histopathology'. Comparative Clinical Pathology, vol. 13, no. 3, Mar. 2005, pp. 100–08. Springer Link, https://doi.org/10.1007/s00580-004-0532-4.

Schneider, Pia, et al. Classification of Viral Pneumonia X-Ray Images with the Aucmedi Framework. arXiv:2110.01017, arXiv, 3 Oct. 2021. arXiv.org, http://arxiv.org/abs/2110.01017.

Shah, Sumeet S., and John W. Sheppard. 'Evaluating Explanations of Convolutional Neural Network Image Classifications'. 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–8. IEEE Xplore, https://doi.org/10.1109/IJCNN48605.2020.9207129.

Zeiler, Matthew D., and Rob Fergus. ‘Visualizing and Understanding Convolutional Networks’. Computer Vision – ECCV 2014, edited by David Fleet et al., Springer International Publishing, 2014, pp. 818–33. Springer Link, https://doi.org/10.1007/978-3-319-10590-1_53.

Guthrie, Graeme J. K., et al. ‘The Systemic Inflammation-Based Neutrophil–Lymphocyte Ratio: Experience in Patients with Cancer’. Critical Reviews in Oncology/Hematology, vol. 88, no. 1, Oct. 2013, pp. 218–30. ScienceDirect, https://doi.org/10.1016/j.critrevonc.2013.03.010.

McKenna, Ellen, et al. ‘Neutrophils: Need for Standardized Nomenclature’. Frontiers in Immunology, vol. 12, Apr. 2021, p. 602963. PubMed Central, https://doi.org/10.3389/fimmu.2021.602963.

Tripathi, Satvik, et al. ‘HematoNet: Expert Level Classification of Bone Marrow Cytology Morphology in Hematological Malignancy with Deep Learning’. Artificial Intelligence in the Life Sciences, vol. 2, Dec. 2022, p. 100043. ScienceDirect, https://doi.org/10.1016/j.ailsci.2022.100043.

Ahmed, Nizar, et al. ‘Identification of Leukemia Subtypes from Microscopic Images Using Convolutional Neural Network’. Diagnostics, vol. 9, no. 3, Aug. 2019, p. 104. PubMed Central, https://doi.org/10.3390/diagnostics9030104.