Application of Deep Learning Neural Networks in the Identification of Abnormal Bone Marrow Cells

Github link: https://github.com/saejin123/BT5153_Final_Project

Abstract

Classification of hematopoietic stem cells (HSCs) are based on bone marrow biopsy and flow cytometry. Both methods have limitations which range from labor intensive approach to high investment and operating cost. However, given that classification of HSCs is crucial for the diagnosis of the hematological disorders (where the number of cases are increasing globally), this paper will evaluate the impact of adopting convolutional neural networks, transfer learning approach and visual transformers in classifying five classes of HSCs. The analysis is conducted on two data sets where one data set is based on171,374 microscopic cytological images taken from bone marrow smears from 945 patients with expert-annotation of the type of hematological diseases whereas the other is based on the same images but cropped. Our CNN and pre-trained CNN2 models achieve a MCC of 0.68 for non-cropped data set. This study is a set towards an automated evaluation of HSCs using state-of-the-art image classification algorithms.

1. Introduction

1.1 Background

Hematopoiesis refers to the formation of blood cellular component in the bone marrow which occurs during embryonic development and throughout adulthood to produce and replenish the blood system. Studying hematopoiesis can help scientists and clinicians to understand better the processes behind blood disorder like leukemia, MDS-RA, etc. and also hematopoietic stem cells (HSCs) can be used as model system for understanding tissue steam cells and their role in ageing and oncogenesis.

Currently, to analyze the HSCs, flow cytometry is used for comprehensive single-cell analysis method over the traditional method of bone marrow biopsy where the analysis of HSCs is done almost relatively manually by cytotechnologist The flow cytometry will require cells obtained from blood and placed into suspension before staining with dyes. This is followed by three processes of using fluidics system to guide cell sample past the laser for separate measurement of every single cell, the optics system which will emit light to collect signal and lastly detection of signals to digital parameters to analyze the cells using a separate software.

Despite the advantages of flow cytometry of enabling scientists and clinicians to evaluate blood disorders, there are several limitations. Some limitations include (i) fluidics system usually causes blockades which affect the analysis of HSCs, (ii) manual check on the laser alignment, (iii) damage of cells which affects the analysis of HSCs and (iv) the exorbitant cost of purchasing a flow cytometry and the high operating cost of the device.

Furthermore, effective examinations are highly dependent on the availability and experience of cytotechnologists. This problem is exacerbated by the increase in bone marrow biopsies required due to the increase in the number of hematological disorders globally (e.g., the number of newly diagnosed leukemia cases increased from 354,500 in 1990 to 518,500 in 2017 (Dong, 2020)).

1.2 Project objective

One major development in recent years is the application of machine learning techniques to identify abnormal bone marrow cells. However, these models often rely on cell level features in order to make a prediction. These cell level features often require specialized techniques to extract (i.e., flow cytometry). While this is an improvement over the traditional approach of bone marrow biopsy, it would still require some lab work (which is often also the bottlenecks and limitations) in order to obtain the features.

Therefore, in this study, we aim to improve the analysis of HSCs and overcome the challenges faced by both flow cytometry and bone marrow biopsy by using images of single-cell samples to train a neural network that can be used to identify the five classes of HSCs. The reduced samples could be flagged to a trained medical professional for further examination. We believe that adopting this approach will reduce the diagnosis time for certain bone marrow related diseases drastically, which can lead to earlier treatment and overall better prognosis for the patient. Furthermore, it will also reduce the cost of analyzing HSCs which helps to allow support earlier blood abnormalities detection.

1.3 Business applications

Although cytopathologists and cytotechnologists play an important role in precision medicine, there is a global shortage of these professionals (S.J. Robboy, 2013). This imposes challenges on the early diagnosis of abnormalities in the bone marrow. This study can strengthen the basis for the automatic analysis of the cell morphologies to quantify the initial degree of malignancy; on top of the numerous literature papers that have similar findings (Mori, 2020), (Teekaraman, 2021). Furthermore, it also improves the diagnostic accuracy as it reduces the reliance on the experience of cytopathologists and shorten the diagnosis time since it reduces the images that the cytotechnologists are required to review. In addition, the deep neural networks can be further studied to use images taken directly from the microscope from the cell culture flasks; which eliminates the need for a dedicated smearing of single-cell and dedicated imaging platform (Yao, 2019).

Furthermore, the deep neural networks developed for the study can be extended to other medical diagnosis like breast cancer pathology images (Wang H, 2014), lung cancer detection (Cheng J-Z, 2016), skin cancer classification (Rao P, 2016), etc. via deep learning and transfer learning. In addition, the study may propel more hospitals to invest resources to create bigger and highly accurate classification of biomedical images; which will improve the automatic analysis done via deep neural networks in the long run.

2. Data

2.1 Summary of raw data set

The cleaned raw data set contains 171,374 identified, expert-annotated mainly single-celled images from bone marrow smears taken from 945 patients stained using the May-Grünwald-Giemsa/Pappenheim stain between 2011 and 2013 (Matek, 2021). The data source where data set is downloaded is from a Kaggle website (Bone Marrow Cell Classification, 2022).

The images were annotated into 21 classes, including 4 classes that were artificially added (i.e., smudge cell, artefact, other cell and not identifiable) to avoid biasing the annotation for easily classifiable images. Each image is 250 by 250 pixels. There is no overlap between images implying no correlation between different images in the data set.

2.2 Classes in the raw data set

With reference to the ***Figure 1***, the cell types for 21 classes in the raw dataset have highly imbalanced distributions. Such class imbalance is common in medical data because of the unequal prevalence of the hematological diseases and the collection of the annotated single-cell images from patients. Refer to ***Appendix I*** for detailed explanation of each class.

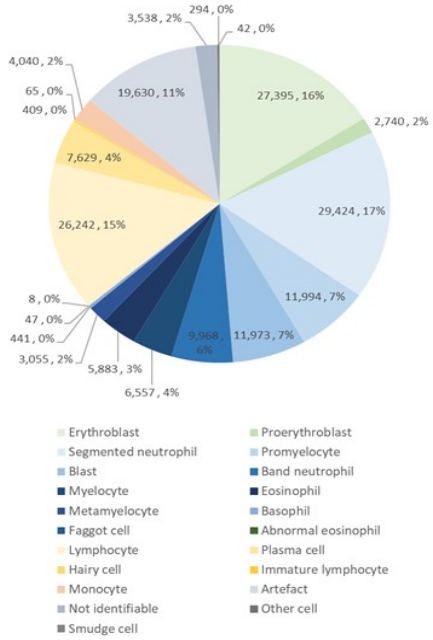


Figure 1*: Distribution of 21 classes in the raw dataset*

2.3 Data pre-processing

2.3.1 AGGREGATION OF SOME CLASSES

Given that some of the morphological classes are challenging to distinguish; even for trained cytotechnologists, there are there is a reduction of the number of classes from 21 to 5 since these classes matches the basic categories of HSCs that define the differentiation pathway in hematopoiesis based on our consultation with a medical doctor. Refer to ***Figure 2***. Despite the challenges to distinguish some morphological classes, this paper will attempt to use 21 sets of CNN models to do one-vs-rest classification (refer to ***Section 3.1.2*** for more details).

- Erythropoiesis refers to the process which generates fully mature erythrocytes and requires the synthesis of vast amounts of hemoglobin along with the ultimate loss of the cell's nucleus and intracellular organelles. “Erythroblast” and “Proerythroblast” are classified under Erythropoiesis.
- Granulopoiesis refers to the process by which mature granulocytes differentiate within the bone marrow. “Segmented neutrophil”, “Promyelocyte”, “Blast”, “Band neutrophil”, “Myelocyte”, “Eosinophil”, “Metamyelocyte”,

"Basophil", "Faggot cell" and "Abnormal eosinophil" are classified under Granulopoiesis.

- Lymphopoiesis refers to the process by which mature lymphocytes differentiate within the bone marrow. "Lymphocyte", "Plasma cell", "Hairy cell" and "Immature lymphocyte" are classified under Lymphopoiesis.
- Monopoiesis refers to the process by which mature monocytes are generated within the bone marrow. Only "Monocyte" is classified under Monopoiesis.
- Others refer to the additional 4 classes (i.e., smudge cell, artefact, other cell and not identifiable) were artificially added to avoid biasing the images.

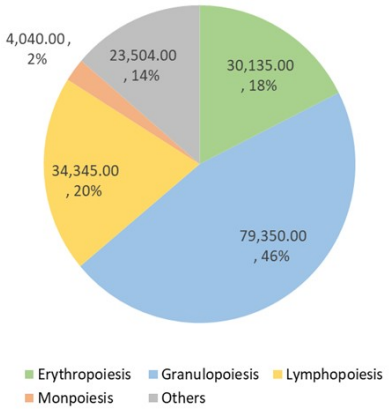


***Figure 2**: Distribution of 5 classes after aggregation*

2.3.2 TRAIN TEST SPLIT FOR DATA SET

The raw data set will be randomly divided into the training-validation data set and testing data set in the ratio of 4:1. Within the training-validation data set, the samples would further be divided into training and validation data validation by the ratio of 4:1 for the purposes of hyperparameter tuning.

2.3.3 CLASS IMBALANCE

With reference to the Figure 2, the cell types for the 5 classes would still have highly imbalanced distributions. To address the class imbalance in the training data set, down-sampling and up-sampling will be adopted to achieve 5,000 images per class. The up-sampling will be based on image augmentation transformation (i.e., rotation by random continuous angle, vertical and horizontal flips, shifts up to 25% of the image weight and height and shears by 5% of the image size). Whereas, the down-sampling will select the first nth images up to the sample size required.

For example, since Lymphopoiesis class has 4 types of cells, each cell type will have 1,250 images. Lymphocyte and plasma cells have training images more than 1,250, they will be down-sampled whereas the hairy cell and immature lymphocyte will be up-sampled.

Refer to ***Table 1*** for the summary of the training, validation and testing datasets.

***Table 1**: Summary of training, validation and test dataset*

Class	Abbreviation	Terminology	Raw data set	Imbalanced Training data set	Validation data set	Test data set	From the python script	Check	Proportion within each class	Balanced Training data set	To up sample	To down sample
Erythropoiesis	EBO	Erythroblast	27,395	17,532	4,384	5,479	27,395	TRUE	50%	2,500	-	15,032
Erythropoiesis	PEB	Proerythroblast	2,740	1,753	439	548	2,740	TRUE	50%	2,500	747	-
Granulopoiesis	NGS	Segmented neutrophil	29,424	18,831	4,708	5,885	29,424	TRUE	10%	500	-	18,331
Granulopoiesis	PMO	Promyelocyte	11,994	7,676	1,919	2,399	11,994	TRUE	10%	500	-	7,176
Granulopoiesis	BLA	Blast	11,973	7,662	1,916	2,395	11,973	TRUE	10%	500	-	7,162
Granulopoiesis	NGB	Band neutrophil	9,968	6,379	1,595	1,994	9,968	TRUE	10%	500	-	5,879
Granulopoiesis	MYB	Myelocyte	6,557	4,196	1,049	1,312	6,557	TRUE	10%	500	-	3,696
Granulopoiesis	EOS	Eosinophil	5,883	3,764	942	1,177	5,883	TRUE	10%	500	-	3,264
Granulopoiesis	MMZ	Metamyelocyte	3,055	1,955	489	611	3,055	TRUE	10%	500	-	1,455
Granulopoiesis	BAS	Basophil	441	281	71	89	441	TRUE	10%	500	219	-
Granulopoiesis	FGC	Faggot cell	47	29	8	10	47	TRUE	10%	500	471	-
Granulopoiesis	ABE	Abnormal eosinophil	8	4	2	2	8	TRUE	10%	500	496	-
Lymphopoiesis	LYT	Lymphocyte	26,242	16,794	4,199	5,249	26,242	TRUE	25%	1,250	-	15,544
Lymphopoiesis	PLM	Plasma cell	7,629	4,882	1,221	1,526	7,629	TRUE	25%	1,250	-	3,632
Lymphopoiesis	HAC	Hairy cell	409	261	66	82	409	TRUE	25%	1,250	989	-
Lymphopoiesis	LYI	Immature lymphocyte	65	41	11	13	65	TRUE	25%	1,250	1,209	-
Monpoiesis	MON	Monocyte	4,040	2,585	647	808	4,040	TRUE	100%	5,000	2,415	-
Others	ART	Artefact	19,630	12,563	3,141	3,926	19,630	TRUE	25%	1,250	-	11,313
Others	NIF	Not identifiable	3,538	2,264	566	708	3,538	TRUE	25%	1,250	-	1,014
Others	OTH	Other cell	294	188	47	59	294	TRUE	25%	1,250	1,062	-
Others	KSC	Smudge cell	42	26	7	9	42	TRUE	25%	1,250	1,224	-
	Total		171,374	109,666	27,427	34,281	171,374		500%	25,000	8,832	93,498

2.3.4 IMAGE CROPPING

Even though the raw data set attempted to focus mainly on single-celled images, it is observed that there are substantial number of images with peripheral cells. Based on our consultation with a medical doctor, the classification of the blood cell type focuses on the appearance of a single cell (eg. the shape of the nucleus, the surface appearance of the cytoplasm, the colour of the nucleus, etc).

As such, there will be an evaluation of the models based on 2 sets of training, validation and test dataset – one without image cropping and one with image cropping. This will enable future research to evaluate if image cropping is useful to improve model performance.

In order to get a good crop of the cells, we first perform edge detection to identify contours of all cells in the image. Next, all contours less than 100 pixels in length are dropped as these would mainly relate to smudges in the cell staining process or incomplete cells. Next, we identify the cell of interest in the image by identifying the center most contour and cropping a square around that contour. A sample of the cropped images can be found in ***Appendix II.***

3. Methodology

3.1 Models

To classify the 5 classes, we have adopted 4 models – convolutional neural network with the Xception model architecture (CNN), transfer learning combined with different number of layers for the CNN ("Pre-CNN1" and Pre-CNN2") and vision transformer. All were trained on a batch size of 32 for the different sets (5C5K, 1vR etc).

3.1.1 CNN (MULTICLASS CLASSIFICATION)

CNNs are at the core of most state-of -the-art computer vision solutions for a wide variety of tasks (Szegedy et al., 2016). For our CNN architecture, we chose an Xception model with depth wise separable convolutions in order to limit the number of training parameters as compared with traditional inception architectures. In this architecture, residual blocks from certain convolutional layers are passed through as separable convolutional layers in the neural network to optimize learning. Empirically, there is evidence to suggest that such networks are easier to optimize as compared to an inception network of similar depth (He, Zhang, Ren, & Sun, 2015). Refer to the ***Figure 3*** for the summary of the model and ***Appendix III*** for detailed model plot.

Model: "model"

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 250, 250, 3)]	0	[]
rescaling (Rescaling)	(None, 250, 250, 3)	0	['input_1[0][0]']
conv2d (Conv2D)	(None, 125, 125, 32)	896	['rescaling[0][0]']
batch_normalization (BatchNormalization)	(None, 125, 125, 32)	128	['conv2d[0][0]']
activation (Activation)	(None, 125, 125, 32)	0	['batch_normalization[0][0]']
conv2d_1 (Conv2D)	(None, 63, 63, 64)	18496	['activation[0][0]']
batch_normalization_1 (BatchNormalization)	(None, 63, 63, 64)	256	['conv2d_1[0][0]']
activation_1 (Activation)	(None, 63, 63, 64)	0	['batch_normalization_1[0][0]']
activation_2 (Activation)	(None, 63, 63, 64)	0	['activation_1[0][0]']
separable_conv2d (SeparableConv2D)	(None, 63, 63, 128)	8896	['activation_2[0][0]']
batch_normalization_2 (BatchNormalization)	(None, 63, 63, 128)	512	['separable_conv2d[0][0]']
activation_3 (Activation)	(None, 63, 63, 128)	0	['batch_normalization_2[0][0]']
separable_conv2d_1 (SeparableConv2D)	(None, 63, 63, 128)	17664	['activation_3[0][0]']
batch_normalization_3 (BatchNormalization)	(None, 63, 63, 128)	512	['separable_conv2d_1[0][0]']
max_pooling2d (MaxPooling2D)	(None, 32, 32, 128)	0	['batch_normalization_3[0][0]']
conv2d_2 (Conv2D)	(None, 32, 32, 128)	8320	['activation_1[0][0]']
add (Add)	(None, 32, 32, 128)	0	['max_pooling2d[0][0]', 'conv2d_2[0][0]']
activation_4 (Activation)	(None, 32, 32, 128)	0	['add[0][0]']
separable_conv2d_2 (SeparableConv2D)	(None, 32, 32, 256)	34176	['activation_4[0][0]']
batch_normalization_4 (BatchNormalization)	(None, 32, 32, 256)	1024	['separable_conv2d_2[0][0]']
activation_5 (Activation)	(None, 32, 32, 256)	0	['batch_normalization_4[0][0]']
separable_conv2d_3 (SeparableConv2D)	(None, 32, 32, 256)	68096	['activation_5[0][0]']
batch_normalization_5 (BatchNormalization)	(None, 32, 32, 256)	1024	['separable_conv2d_3[0][0]']
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 256)	0	['batch_normalization_5[0][0]']
conv2d_3 (Conv2D)	(None, 16, 16, 256)	33024	['add[0][0]']
add_1 (Add)	(None, 16, 16, 256)	0	['max_pooling2d_1[0][0]', 'conv2d_3[0][0]']
activation_6 (Activation)	(None, 16, 16, 256)	0	['add_1[0][0]']
separable_conv2d_4 (SeparableConv2D)	(None, 16, 16, 512)	133888	['activation_6[0][0]']
batch_normalization_6 (BatchNormalization)	(None, 16, 16, 512)	2048	['separable_conv2d_4[0][0]']
activation_7 (Activation)	(None, 16, 16, 512)	0	['batch_normalization_6[0][0]']
separable_conv2d_5 (SeparableConv2D)	(None, 16, 16, 512)	267264	['activation_7[0][0]']
batch_normalization_7 (BatchNormalization)	(None, 16, 16, 512)	2048	['separable_conv2d_5[0][0]']
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 512)	0	['batch_normalization_7[0][0]']
conv2d_4 (Conv2D)	(None, 8, 8, 512)	131584	['add_1[0][0]']
add_2 (Add)	(None, 8, 8, 512)	0	['max_pooling2d_2[0][0]', 'conv2d_4[0][0]']
activation_8 (Activation)	(None, 8, 8, 512)	0	['add_2[0][0]']
separable_conv2d_6 (SeparableConv2D)	(None, 8, 8, 728)	378072	['activation_8[0][0]']
batch_normalization_8 (BatchNormalization)	(None, 8, 8, 728)	2912	['separable_conv2d_6[0][0]']
activation_9 (Activation)	(None, 8, 8, 728)	0	['batch_normalization_8[0][0]']
separable_conv2d_7 (SeparableConv2D)	(None, 8, 8, 728)	537264	['activation_9[0][0]']
batch_normalization_9 (BatchNormalization)	(None, 8, 8, 728)	2912	['separable_conv2d_7[0][0]']
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 728)	0	['batch_normalization_9[0][0]']
conv2d_5 (Conv2D)	(None, 4, 4, 728)	373464	['add_2[0][0]']
add_3 (Add)	(None, 4, 4, 728)	0	['max_pooling2d_3[0][0]', 'conv2d_5[0][0]']
activation_10 (Activation)	(None, 4, 4, 728)	0	['add_3[0][0]']
separable_conv2d_8 (SeparableConv2D)	(None, 4, 4, 1024)	753048	['activation_10[0][0]']
batch_normalization_10 (BatchNormalization)	(None, 4, 4, 1024)	4096	['separable_conv2d_8[0][0]']
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1024)	0	['batch_normalization_10[0][0]']
dropout (Dropout)	(None, 1024)	0	['global_average_pooling2d[0][0]']
dense (Dense)	(None, 5)	5125	['dropout[0][0]']

Total params: 2,786,749
Trainable params: 2,778,013
Non-trainable params: 8,736

Figure 3*: Summary of model architecture for CNN*

3.1.2 CNN (ONE VS REST CLASSIFICATION)

During our exploratory model training phase, we discovered that the CNN appears to be very good at binary classification problems (achieving >99% test accuracy on sample binary classification datasets taken from our main dataset). Therefore, we also explore if we can get better classification results if we train a CNN to identify each class (21 in total)[1]. Samples can then be classified by each of the 21 models and classified as the class whose model outputs the highest probability. For this model, the CNN architecture is the same as that described in 3.1.1, except that the output layer contains only 1 neuron with the "sigmoid" activation function. A detailed flowchart of the 1vR classification process can be found in ***Appendix IV***.

3.1.3 INCEPTTIONV3 AND CNN

Transfer learning approach is adopted to leverage on existing established pre-trained image recognition model for feature extraction from images. For this study, the InceptionV3 is adopted as it is a CNN trained on ImageNet dataset which consists of more than 14 million images and is known to attain accuracy around 78%.

As such, we use this pre-trained model to extract the features vectors from our training data set of 25,000 images before using them as inputs to two different CNN models. The CNN model architecture is based on a subset of the Xception model architecture. The difference between "Pre-CNN1" and Pre-CNN2 models are the number of layers and hence level of complexity.

Refer to ***Figure 4 and 5*** for the summaries of the 2 models.

[1] Due to computational resource constraints, each of the 21 models is only trained on only 1,000 samples each in the positive and negative class. We assemble the positive class by randomly selecting 1,000 samples from the class that the model is supposed to identify. The negative class is assembled by randomly selecting 50 samples from each of the remaining 20 classes (for a total of 1,000 samples in the negative class)

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inception_v3 (Functional)   (None, 6, 6, 2048)        21802784

 rescaling (Rescaling)       (None, 6, 6, 2048)        0

 conv2d_94 (Conv2D)          (None, 3, 3, 32)          589856

 batch_normalization_94 (Bat  (None, 3, 3, 32)         128
 chNormalization)

 activation_94 (Activation)  (None, 3, 3, 32)          0

 global_average_pooling2d (G  (None, 32)               0
 lobalAveragePooling2D)

 dropout (Dropout)           (None, 32)                0

 dense (Dense)               (None, 5)                 165

=================================================================
Total params: 22,392,933
Trainable params: 22,358,437
Non-trainable params: 34,496
_________________________________________________________________
```

***Figure 4**: Summary of model architecture for Pre-CNN1*

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inception_v3 (Functional)   (None, 6, 6, 2048)        21802784

 rescaling (Rescaling)       (None, 6, 6, 2048)        0

 conv2d_94 (Conv2D)          (None, 3, 3, 32)          589856

 batch_normalization_94 (Bat  (None, 3, 3, 32)         128
 chNormalization)

 activation_94 (Activation)  (None, 3, 3, 32)          0

 conv2d_95 (Conv2D)          (None, 2, 2, 64)          18496

 batch_normalization_95 (Bat  (None, 2, 2, 64)         256
 chNormalization)

 activation_95 (Activation)  (None, 2, 2, 64)          0

 global_average_pooling2d (G  (None, 64)               0
 lobalAveragePooling2D)

 dropout (Dropout)           (None, 64)                0

 dense (Dense)               (None, 5)                 325

=================================================================
Total params: 22,411,845
Trainable params: 22,377,221
Non-trainable params: 34,624
_________________________________________________________________
```

***Figure 5** : Summary of model architecture for Pre-CNN2*

3.1.4 VISION TRANSFORMER

Transformers were already well known and utilized in NLP tasks, with some examples being BERT, DeBERTa. However, they had limited applications in images, until Vision Transformers (ViT) came along.While ViTs are not an entirely new concept, they became popular through a revolutionary new way in processing images. Earlier ViT methods tried looking at each pixel, but the revolutionary approach by Dosovitsky, A.et al., (2021) divided each image into patches.

Although the authors showed the remarkable performance of ViTs, surpassing even the long held SOTA CNNs, one downside of ViTs were the large amount of data needing to be fed before they would outperform. This could range in the scale of at least 14 million images.

Given that training large number of images was computationally intensive, we opted for a pre-trained ViT-B/32-224 (Morales, F, 2021), shown in ***Figure 6***.

```
Model: "vision_transformer"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vit-b32 (Functional)        (None, 768)               87429888

 flatten_1 (Flatten)         (None, 768)               0

 batch_normalization_2 (Batc  (None, 768)              3072
 hNormalization)

 dense_2 (Dense)             (None, 11)                8459

 batch_normalization_3 (Batc  (None, 11)               44
 hNormalization)

 dense_3 (Dense)             (None, 5)                 60

=================================================================
Total params: 87,441,523
Trainable params: 87,439,965
Non-trainable params: 1,558
_________________________________________________________________
```

***Figure** 6: Summary of model architecture for pre-trained ViT*

The pre-trained model was just fine-tuned minimally following Momin's (2021) method by adding a BatchNormalization, followed by a small dense layer with GELU activation between BatchNormalizations before the final output of 5 Classes. GELU – which features in transformers such as BERT, can be seen as an improvement over ReLU as they characterize input by value instead of the sign (in ReLU) In addition, they have remarkedly better performance, likely due to the function being curved at all points, allowing for improved approximation (Hendrycks & Gimpel, 2020).

4. Results

We have adopted (i) accuracy, (ii) weighted average recall and (iii) Matthews Correlation Coefficient (MCC) to evaluate the performance of the models.

Accuracy is an overall measure of how much the model is correctly predicting the classification of a single image above the entire set of data. Refer to ***Equation 1***. Even though accuracy is the most famous classification performance indicator, it will be the least important performance metric among the three (K Blagex, 2020) since it is less appropriate when the dataset is imbalanced.

$$\text{Accuracy} = \frac{\text{\# correct prediction}}{\text{\# total prediction}}$$

***Equation 1**: Accuracy Formula*

Whereas, weighted average recall is the weighed mean of recall with weights equal to the class probability. There is a focus on recall (refer to ***Equation 2***) because the detrimental consequence of classifying the HSCs which

results in delays in receiving necessary medical treatments.

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

***Equation 2**: Recall Formula*

MCC is adopted as the most important evaluation metric among the three because (i) it is widely used in biomedical as a performance metric, (ii) it is a more reliable statistical measure which produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (true positives, false negative, true negative and false positive), proportional to both the size of the positive element and the size of negative elements in the dataset and (iii) it can be adopted for imbalanced classification and can even be used for classes that are very different in sizes (Chicco D, 2020), (Ietswaart R, 2020). Refer to ***Equation 3***.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FN)\,(TP + FP)\,(TN + FP)\,(TN + FN)}}$$

***Equation 3**: MCC Formula*

4.1 Performance evaluation metrics for 5 classes and datasets (not cropped vs cropped)

4.1.1 COMPARISON OF COMPUTATION EFFICIENCY

One important fact that cannot be overlooked too would be the computational time which, for a larger dataset, would scale immensely although it provides improved scores & estimates.

Table 2 shows in seconds, the time taken to run the various models on the training & validation dataset. Unless specified otherwise, they were run using Google Colab Pro's Tesla P100 GPU with 16GB Ram. The testing evaluation took around the same time, averaging at approximately 75mins each.

***Table 2**: Comparison Computational Efficiency*

Not Cropped				
CNN	**Pre-CNN1**	**Pre-CNN2**	**Pre-trained ViT**	**Average (excluding *)**
16512	16164	51674*	15640	16106
Cropped				
11294	18933	42633*		15114

*Due to run-time constraints, the following models were run locally on a X510U Asus computer comprising 8GB Ram & NVIDIA MX130 Card

We can see that each model, even at a smaller dataset for training & validation, takes 4.5 hours on average. Initial trials run on the full dataset were attempted but ultimately abandoned as each epoch would take around 4 hours.

4.1.2 COMPARISON OF PERFORMANCE

Overall, all 4 models attained a higher accuracy, weighted average recall and MCC for the test data set that is not cropped, as compared to cropped images. This implies that cropping images result in the loss of critical features for the classification. Refer to ***Table 3*** for the summary of results by evaluation metrics.

***Table 3**: Summary of results by evaluation metrics*

Dataset	Without image cropping				With image cropping		
Metrics\Models	**CNN**	**Pre-CNN1**	**Pre-CNN2**	**Pre-trained ViT**	**CNN**	**Pre-CNN1**	**Pre-CNN2**
Accuracy	0.77	0.74	0.76	0.58	0.64	0.71	0.69
Weight avg recall	0.77	0.74	0.76	0.58	0.64	0.71	0.69
MCC	0.68	0.67	0.68	0.43	0.54	0.62	0.59

Furthermore, the CNN and Pre-CNN2 models attained almost similar performance for the non-cropped test data prediction. Both models outperform the Pre-CNN1 and Pre-trained ViT performed the worst. Based on our analysis, even though both Pre-CNN1, Pre-CNN2 and Pre-trained ViT models were all pre-trained on ImageNet dataset, the Pre-trained ViT model lacks the inductive bias typically present in CNNs such as translational symmetry that gave rise to CNNs outperforming FCNN. This may prevent the model from fully capturing the critical features necessary to differentiate the individual HSC class as it would require vastly more data as compared to the CNN model. all were pre-trained on ImageNet dataset.

In additional, even though the Pre-CNN2 has 4 additional layers (which results in an additional of 18,784 trainable parameters) as compared to Pre-CNN1, it is observed that the MCC differ by only 0.008. This implies that InceptionV3 is relatively effective in extracting the critical features for HSCs classification; which corroborates with the use of InceptionV3 for similar cell classification (Mzurikwao, D., 2020).

In addition, it is observed that the recall for the "Monopoiesis" class is the lowest for all 4 models based on non-cropped images. This may be due to the fact that the training dataset for "Monopoiesis" class has the highest percentage of augmented images (48%). This implies that the image augmentation may distorted the necessary key features required to differentiate the HSCs. Refer to ***Table 4*** and ***5*** for results by classes. Refer to ***Appendix V*** for the confusion matrix.

***Table 4**: Summary of results for non-cropped images*

Without image cropping																
	CNN				Pre-CNN1				Pre-CNN2				Pre-trained ViT			
Class	**precision**	**recall**	**f1-score**	**support**	**precision**	**recall**	**f1-score**	**support**	**precision**	**recall**	**f1-score**	**support**	**precision**	**recall**	**f1-score**	**support**
Erythropoiesis	0.88	0.87	0.87	6,027	0.76	0.92	0.83	6,027	0.83	0.9	0.86	6,027	0.77	0.59	0.67	6,027
Granulopoiesis	0.86	0.78	0.82	15,874	0.97	0.64	0.77	15,874	0.94	0.69	0.8	15,874	0.74	0.62	0.68	15,874
Lymphopoiesis	0.73	0.81	0.77	6,870	0.63	0.88	0.74	6,870	0.61	0.83	0.7	6,870	0.59	0.4	0.48	6,870
Monopoiesis	0.36	0.61	0.45	808	0.36	0.67	0.47	808	0.37	0.64	0.47	808	0.14	0.36	0.20	808
Others	0.58	0.60	0.59	4,702	0.59	0.68	0.64	4,702	0.63	0.70	0.66	4,702	0.35	0.71	0.47	4,702
Accuracy			0.77	34,281			0.74	34,281			0.76	34,281			0.58	34,281
Macro avg	0.68	0.74	0.7	34,281	0.66	0.76	0.69	34,281	0.67	0.75	0.7	34,281	0.52	0.54	0.5	34,281
Weight avg	0.79	0.77	0.78	34,281	0.8	0.74	0.75	34,281	0.8	0.76	0.76	34,281	0.65	0.58	0.6	34,281
MCC			0.68	34,281			0.67	34,281			0.68	34,281			0.43	34,281

***Table 5**: Summary of results for cropped images*

With image cropping	CNN				Pre-CNN1				Pre-CNN2			
Class	precision	recall	f1-score	support	precision	recall	f1-score	support	precision	recall	f1-score	support
Erythropoiesis	0.81	0.74	0.77	6,027	0.70	0.82	0.76	6,027	0.82	0.81	0.81	6,027
Granulopoiesis	0.91	0.54	0.68	15,874	0.92	0.71	0.8	15,874	0.9	0.68	0.77	15,874
Lymphopoiesis	0.54	0.71	0.62	6,870	0.75	0.61	0.67	6,870	0.73	0.62	0.67	6,870
Monpoiesis	0.3	0.47	0.37	808	0.28	0.73	0.4	808	0.16	0.78	0.27	808
Others	0.38	0.75	0.51	4,702	0.47	0.72	0.57	4,702	0.47	0.66	0.55	4,702
Accuracy			0.64	34,281			0.71	34,281			0.69	34,281
Macro avg	0.59	0.64	0.59	34,281	0.62	0.72	0.64	34,281	0.61	0.71	0.61	34,281
Weight avg	0.73	0.64	0.65	34,281	0.722	0.71	0.73	34,281	0.77	0.69	0.72	34,281
MCC			0.54	34,281			0.62	34,281			0.59	34,281

4.2 Performance evaluation metrics for 21 classes

We evaluated the performance of the one-versus-rest CNN classifier on a test dataset containing 884 samples across all 21 classes. The classifier only managed to achieve an accuracy score of 45.48%. In order to understand why this strategy performed poorly, we analyze the confusion matrix of the test set classification, as shown in ***Figure 7*** below.

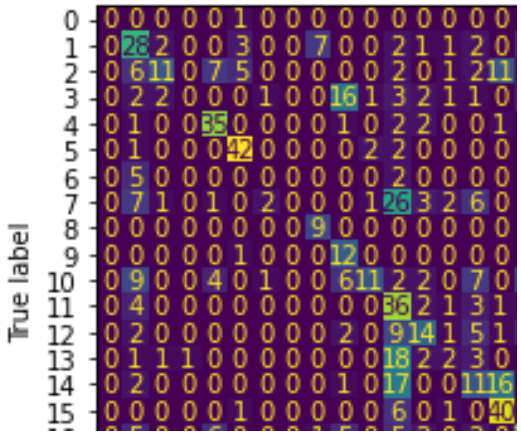


***Figure 7**: Confusion matrix for the 21 classes*

From the confusion matrix we observe that some binary classifiers performed extremely poorly with extremely high false positive rates or false negative rates. For example, classifier 11 obtained a false positive rate of 75.34% while classifier 7 obtained a false negative rate of 100%.

5. Discussions

Neural networks have shown to be successful in various image classification problems. In this study, the results are encouraging with a relatively high MCC of 0.68 given the small training data set that the models are performed due to the limitation on computational resources.

We believe that limitations in the ViT model attempted resulted in a low performance as compared to other models. Given the relatively nascent stage of ViT, although it is becoming more widely adopted, the availability of pre-trained models was primarily offered in PyTorch and/or JAX. An adapted keras version was found however it was limited in the implementation to a ViT-B/32-224 pre-trained on imagenet21k and fine-tuned on imagenet1k

Further improvements could be had by further fine-tuning patch size, transformer layers, projection dimensions and training on larger amounts of data. As mentioned in 4.1.2, although ViTs are extremely powerful, we can see that when applying it on a domain that is remarkedly different from the domain on which it is trained, the performance of ViTs for a smaller data size is unable to match up with a pre-trained CNN.

For the one-versus-rest classifier, we believe that part of the problem was the extremely imbalanced dataset. Even though each binary classifier models were trained on synthetically balanced datasets, the number of original samples for certain cell types (such as ABE) is so few that most of the augmented samples are very similar. This hindered the model learning, which could explain why certain binary classifiers had extremely high false negative and false positive rates.

References

(n.d.). Retrieved from American Society of Hematology Image Bank: https://imagebank.hematology.org/about

Ali Darakhshandeh, M. D. (2017). Faggot cell (leukemic promyelocyte). Retrieved from https://imagebank.hematology.org/image/61053/faggot-cell-leukemic-promyelocyte

Biron. (n.d.). Metamyelocyte - Glossary: Laboratory, Radiology, sleep and genetic. Retrieved from https://www.biron.com/en/glossary/metamyelocyte/

Blagec, Kathrin & Dorffner, Georg & Moradi, Milad & Samwald, Matthias. (2020). A critical analysis of metrics used for measuring progress in artificial intelligence.

Bone Marrow Cell Classification. (2022). Retrieved from https://www.kaggle.com/andrewmvd/bone-marrow-cell-classification?select=abbreviations.csv

Cheng J-Z, N. D.-H.-M.-C. (2016). Computer-aided diagnosis with deep learning architecture: applications to breast lesions in US images and pulmonary nodules in CT scans.

Chicco D, J. G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics.

Choi JW, K. Y. (2017). White blood cell differential count of maturation stages in bone marrow smear using dual-stage convolutional neural networks. Retrieved from https://doi.org/10.1371/journal.pone.0189259

Christian Matek, S. K. (2021). Retrieved from https://ashpublications.org/blood/article/138/20/1917/477932/Highly-accurate-differentiation-of-bone-marrow

CORPath. (n.d.). Blasts. Retrieved from https://www.corpath.net/blasts

Dong, Y. S. (2020). Leukemia incidence trends at the global, regional, and national level between 1990 and 2017. Retrieved from https://doi.org/10.1186/s40164-020-00170-6

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021, June 3). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv.org. Retrieved April 23, 2022, from https://arxiv.org/abs/2010.11929

eClinpath. (n.d.). Normal leukocytes. Retrieved from https://eclinpath.com/hematology/morphologic-features/white-blood-cells/normal-leukocytes/

Encyclopædia Britannica, Inc. (n.d.). Erythroblast. Retrieved from https://www.britannica.com/science/erythroblast

Encyclopædia Britannica, inc. (n.d.). Erythroblast. Encyclopædia Britannica. Retrieved from https://www.britannica.com/science/erythroblast

Hendrycks, D., & Gimpel, K. (2020, July 8). Gaussian error linear units (GELUs). arXiv. Retrieved April 23, 2022, from https://arxiv.org/abs/1606.08415

Ietswaart R, A. S. (2020). Machine learning guided association of adverse drug reactions with in vitro target-based pharmacology. Retrieved from EBioMedicine: https://pubmed.ncbi.nlm.nih.gov/32565027/

KG, K. (2014). Book Review: Deep Learning.

Krizhevsky A, S. I. (2012). Imagenet classification with deep convolutional neural network. Advances in neural information processing systems.

Matek, C. K. (2021). An Expert-Annotated Dataset of Bone Marrow Cytology in Hematologic Malignancies. Retrieved from The Cancer Imaging Archive: https://doi.org/10.1186/s40164-020-00170-6

Momin, R. (2021, February 11). Vision Transformer (VIT) fine-tuning. Kaggle. Retrieved April 20, 2022, from https://www.kaggle.com/code/raufmomin/vision-transformer-vit-fine-tuning

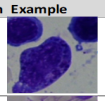
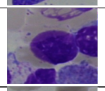
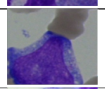
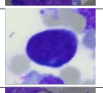
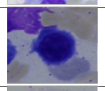
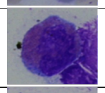
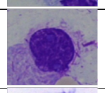
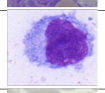
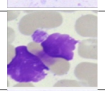
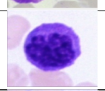
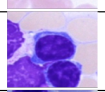
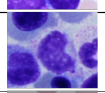
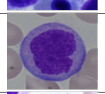
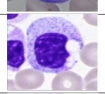
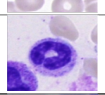
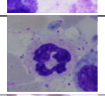
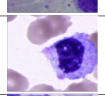
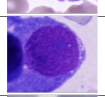
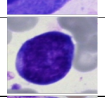
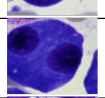
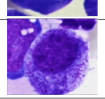
Morales, F. (2021, July). VIT-Keras: Keras implementation of VIT (Vision Transformer). GitHub. Retrieved April 23, 2022, from https://github.com/faustomorales/vit-keras

Mzurikwao, D., Khan, M.U., Samuel, O.W. et al. Towards image-based cancer cell lines authentication using deep neural networks. Sci Rep 10, 19857 (2020). https://doi.org/10.1038/s41598-020-76670-6

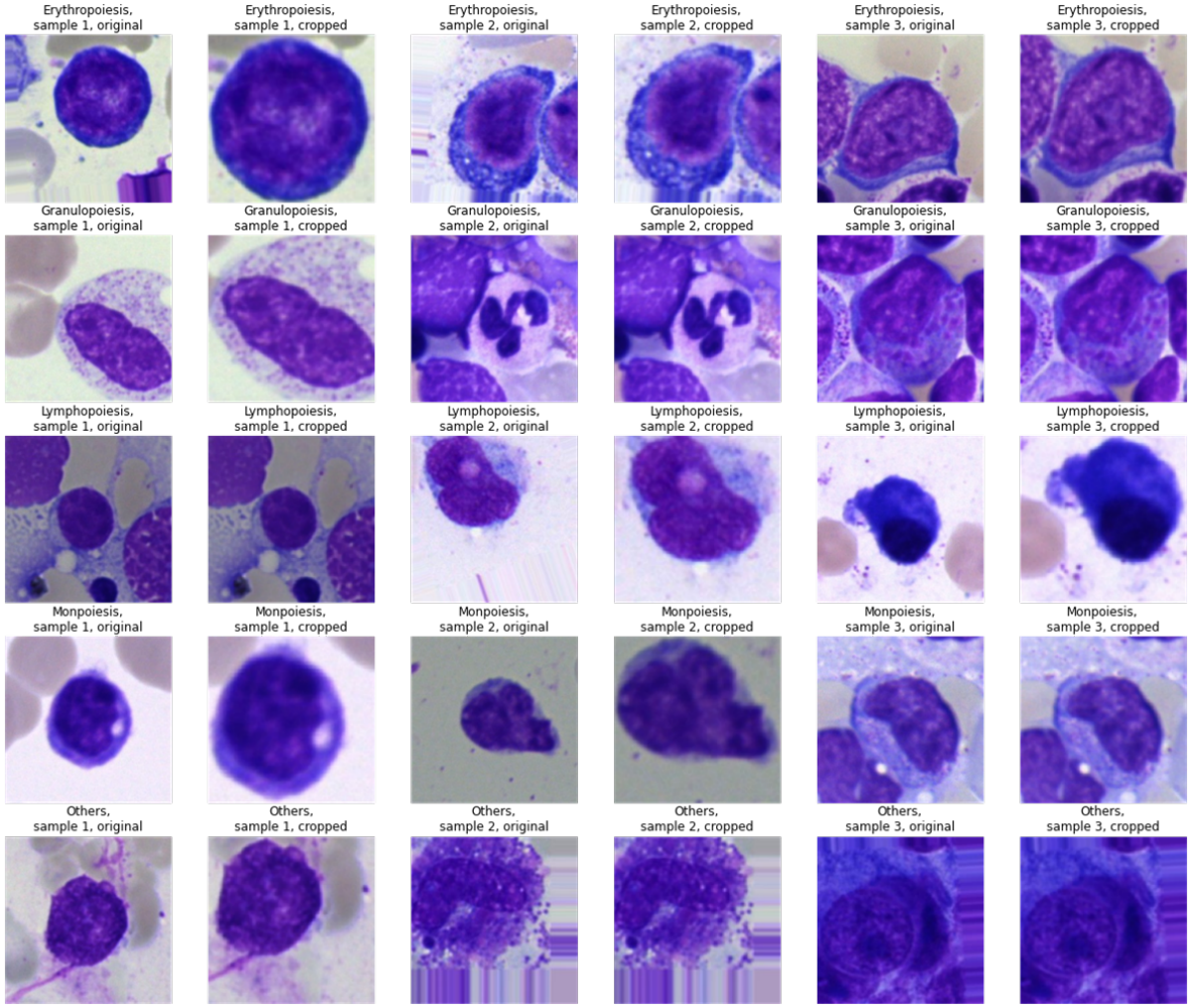
Salama, K. (2021, January 18). Image classification with Vision Transformer. Implementing the Vision Transformer (ViT) model for image classification. Retrieved April 23, 2022, from https://keras.io/examples/vision/image_classification_with_vision_transformer/

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

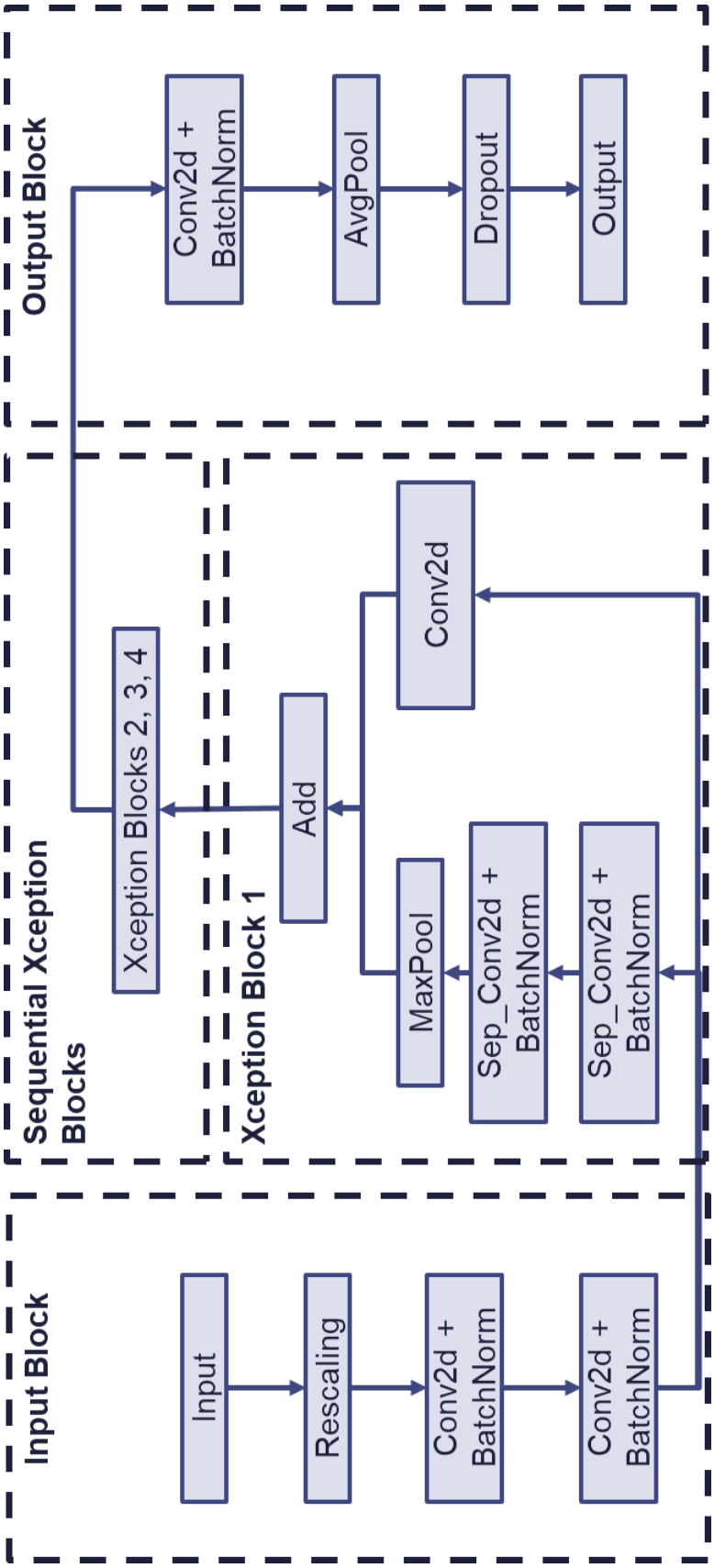
Appendix I: Explanation of each type

No	Abbreviation	Example	Terminology	Class	Explanation
1	ABE		Abnormal eosinophil	Granulopoiesis	Abnormal in Eosinophil occurs when there's a higher-than-normal range; possible causes can be allergies, cancers or infections.
2	ART		Artefact	Others	An artificial or altered structure or tissue due to external factors.
3	BAS		Basophil	Granulopoiesis	One of several white blood cells in body which play in 'immune surveillance' to ward off cancer cells, reacts against allergies and protects body from viruses, etc. However, it is considered irregular if it is more than 1%.
4	BLA		Blast	Granulopoiesis	Precursors to mature blood cells such as EBO, LYT, MON, erythrocyte. It is low in number. If it is more than 20%, it would be a sign of acute leukemia. (CORPath., n.d.)
5	EBO		Erythroblast	Erythropoiesis	Nucleated cell as part of a stage in development of red blood cells (aka erythrocyte), an immature erythrocyte. (Encyclopædia Britannica, Inc., n.d.)
6	EOS		Eosinophil	Granulopoiesis	White blood cell that helps fight disease. It is responsible for curb infections and boost inflammation which aids in fighting disease.
7	FGC		Faggot cell	Granulopoiesis	Normally found in leukemia. (Ali Darakhshandeh, 2017)
8	HAC		Hairy cell	Lymphopoiesis	Usually a cause of a chronic and rare form of Leukaemia – Hairy Cell Leukaemia
9	KSC		Smudge cell	Others	Remnants of cells – typically leukocytes of which Lymphocyte is a part of. (Biron., n.d.)
10	LYI		Immature lymphocyte	Lymphopoiesis	Immature lymphocytes include lymphoblasts and prolymphocytes.
11	LYT		Lymphocyte	Lymphopoiesis	Lymphocytes develop from cells called lymphoblasts to become mature, infection-fighting cells. There are 2 main types of lymphocytes - B lymphocytes (B cells) and T lymphocytes (T cells).
12	MMZ		Metamyelocyte	Granulopoiesis	Together with myelocyte (MYB) and Promyelocyte (PMO) are usually the 'precursors' of neutrophils – the most present white blood cell. (Biron. , n.d.)
13	MON		Monocyte	Monpoiesis	A type of white blood cell like lymphocytes etc
14	MYB		Myelocyte	Granulopoiesis	Young cell, found in bone marrows (comes before metamyelocyte)
15	NGB		Band neutrophil	Granulopoiesis	Less mature than segment neutrophil (NGS) (MD, n.d.)
16	NGS		Segmented neutrophil	Granulopoiesis	Mature form of band neutrophil (NGB) (eClinpath., n.d.)
17	NIF		Not identifiable	Others	Unable to identify the type of cell
18	OTH		Other cell	Others	Other type of cells
19	PEB		Proerythroblast	Erythropoiesis	Precursor cell of erythroblast (EBO)
20	PLM		Plasma cell	Lymphopoiesis	Similar to lymphocyte B cells; (type of activated B cells)
21	PMO		Promyelocyte	Granulopoiesis	Young cell, found in bone marrows (comes before myelocyte)

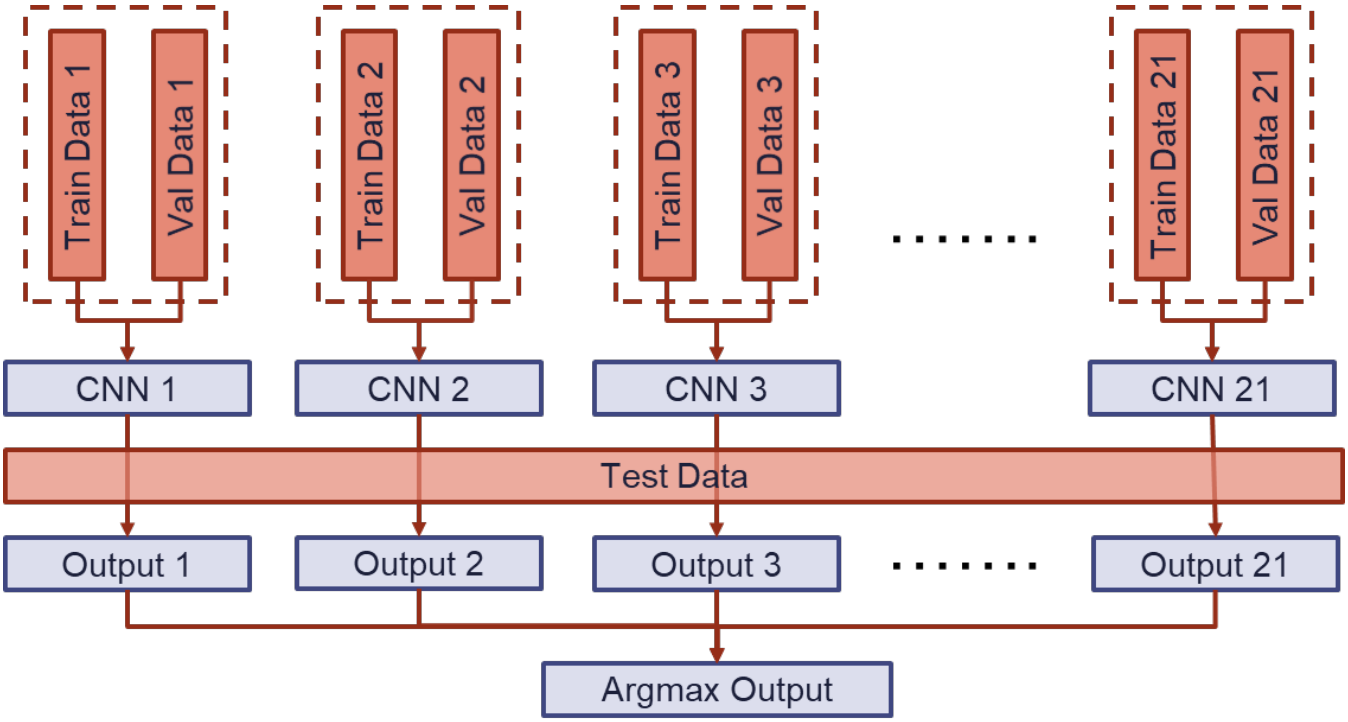
Appendix II: Example of non-cropped against cropped images for 5 classes



Appendix III: Convolutional Neural Network Architecture



Appendix IV: One Versus Rest Classification Strategy



21 binary classifiers are trained to identify one specific cell type out of the 21 cell types. The basic model architecture is the same as the multiclass convolutional neural network with Xception architecture.

Each training dataset contains 1,000 samples from one particular cell type as the positive class, and 50 samples from each of the remaining 20 cell types as the negative class. Each validation dataset contains 100 samples from one particular cell type as the positive class and 5 samples from each of the remaining 20 cell types as the negative class. The training and validation result of each classifier is shown in the table below. The testing dataset contains 778 samples across all 21 classes.

To perform a sample classification, the image is fed to each of the 21 classifiers, which would output a probability that the sample belongs to the class that the classifier is trained to identify. The final classification would be the class of the classifier that outputs the highest probability.

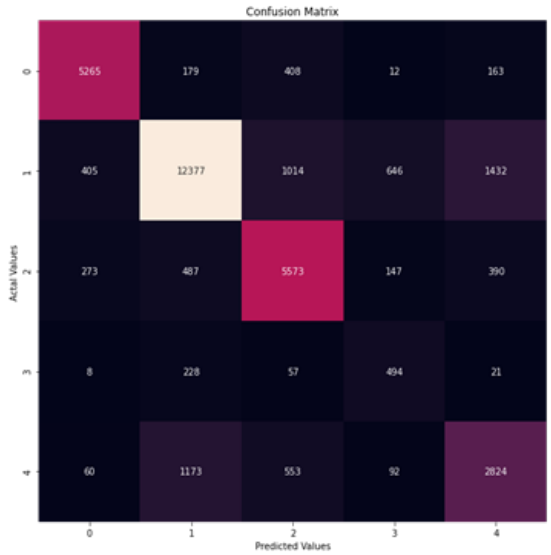
Appendix IV: One Versus Rest Classification Strategy (cont'd)

Classifier	Cell Type to Identify	Val loss	Val Acc
0	ABE	0.7138	0.5000
1	ART	0.0479	0.8350
2	BAS	0.4066	0.8300
3	BLA	0.6396	0.6300
4	EBO	0.3260	0.8800
5	EOS	0.0943	0.9800
6	FGC	0.6027	0.7750
7	HAC	0.7521	0.5000
8	KSC	0.0014	1.0000
9	LYI	0.3425	0.9050
10	LYT	0.5450	0.7500
11	MMZ	0.5196	0.8300
12	MON	0.6866	0.7100
13	MYB	0.5842	0.7650
14	NGB	0.3840	0.8150
15	NGS	0.3369	0.9050
16	NIF	0.5965	0.6800
17	OTH	0.3132	0.8800
18	PEB	0.2728	0.9500
19	PLM	0.4294	0.8150
20	PMO	0.4139	0.8400

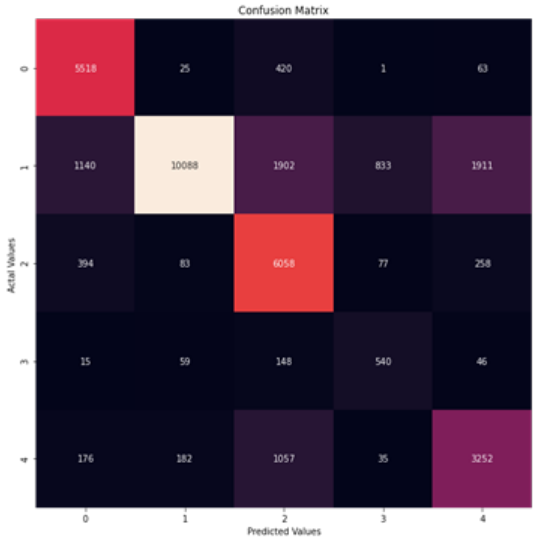
Appendix V: Confusion matrix

(a) Without image cropping

CNN:

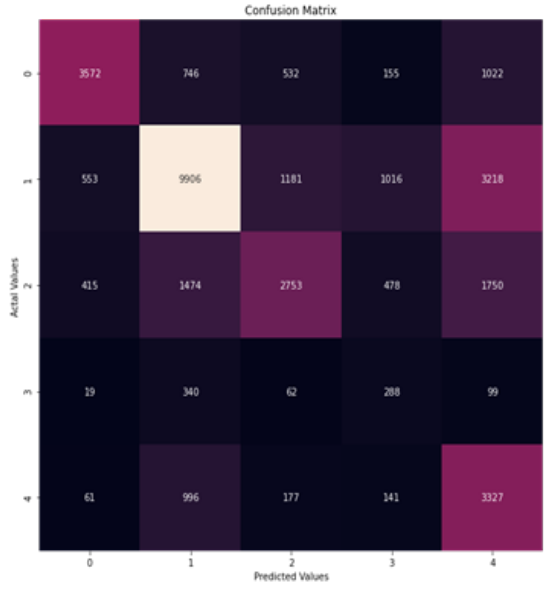


Pre-CNN1:



Pre-CNN2:

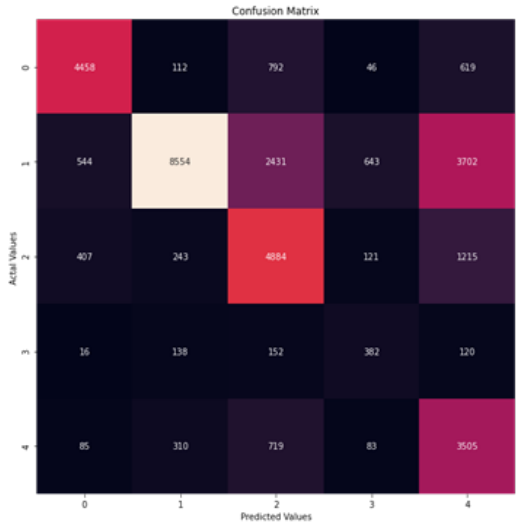
Pre-trained ViT:



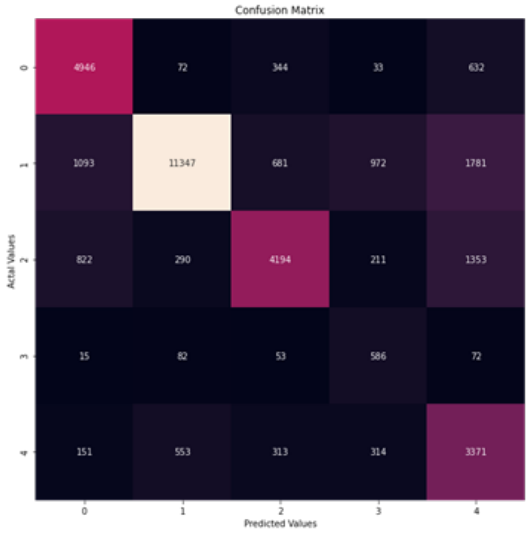
Appendix V: Confusion matrix (cont'd)

(b) With image cropping

CNN:



Pre-CNN1:



Pre-CNN2: