**Assignment 2: Machine learning with Deep Neural Networks**

**Medical Image Analysis and Classification**

**CI7520**

Shairam Ramachandran Sritharan (K2036078)

Nikhil Subramanian (K2218203)

Kamonchanok Promsilp (K194123)

Table of Contents

[Aims and Objectives 3](#_Toc129885418)

[Introduction 3](#_Toc129885419)

[Evaluation of Data 4](#_Toc129885420)

[Analysis of the existing DNNs 5](#_Toc129885421)

[ResNet\_V1\_50 5](#_Toc129885422)

[Inception\_V3 6](#_Toc129885423)

[Inception\_V1 6](#_Toc129885424)

[Efficientnet lite3 6](#_Toc129885425)

[Efficientnet lite0 6](#_Toc129885426)

[Analysis of the designed DNNs 6](#_Toc129885427)

[BreastMnist 6](#_Toc129885428)

[BloodMNist 8](#_Toc129885429)

[Analysis of the training process 10](#_Toc129885430)

[Analysis of the designed interactive application and the related GUI 10](#_Toc129885431)

[Comparative analysis 11](#_Toc129885432)

[Breast Accuracy Score 13](#_Toc129885433)

[Blood Accuracy Score 15](#_Toc129885434)

[Conclusion 17](#_Toc129885435)

[Reference 18](#_Toc129885436)

# Aims and Objectives

Build a Deep Learning based system which aids in classifying images from the dataset available on the MedMNIST2D dataset. The datasets to be used in this coursework is BloodMNIST and BreastMNIST datasets.

* The data needs to verify to have no missing records.
* As the data is multi-dimensional, it needs normalized and formatted for classification and image analysis.
* The data needs to be visualized and ground truth needs to be displayed.
* Due to class imbalance, the data needs to be augmented to improve performance.
* The data needs to be split into train and test.
* A total of 4 models need to be created with Keras, 2 for BloodMNIST and 2 for BreastMNIST.
* The model performance metrics need to be evaluated on the test dataset.
* A website/ mobile app needs to be built that allows the user to upload or capture an image with the camera and use REST SPI to send the data to a server, execute the best deep model and return to the ap the outcome visualizing the class of the uploaded data/image.

# Introduction

A picture containing text, screenshot

Description automatically generatedThis report looks at the medical Images of breasts and blood from the MedMnist dataset. There are 780 breast ultrasound images with different degrees of classifications. The original dataset has 3 classifications: normal, benign, and malignant. However, the dataset we will be using only has 2 classifications, benign and normal, which have been merged into one entity and malignant. The dataset provided uses binary labelling, in which normal and benign are classified as positive while malignant is negative. The original source of the image was changed from 500 x 500 x 1 however dataset was resized to 28 x 28 x 1 which we will be using. (Yang, et al., 2023)

Figure BreastMNist images. (Yang, et al., 2023)

Graphical user interface, application, website

Description automatically generatedAs for blood, there are 17,092 images with 8 different classifications: basophil, eosinophil, erythroblast, immature granulocyte, lymphocyte, monocyte, neutrophil and platelet with different degrees of the dataset available. The original source of the image was changed from 360 x 363 x 3 however dataset was resized to 28 x 28 x 3 which we will be using. (Yang, et al., 2023)

Figure BloodMNist images. (Yang, et al., 2023)

All the models created are run on Google Colab Python 3 Google Compute Engine (GPU).

## Evaluation of Data

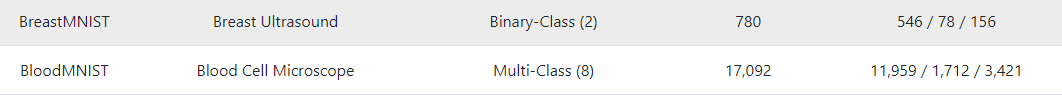


Figure Details on MNIST data. (Yang, et al., 2023)

Breast dataset has no missing values in the dataset with all having the same shape of 28 x 28 x 1. There are varying degrees of data provided to benign and malignant, with benign having 399 data sets and malignant with 147 datasets provided. The data set provided is with the following ratio of 7:1:2, Training, Validation and Test. This can be seen in figure……… Due to the nature of 2 classifications in the dataset, it is labelled as binary classification. (Yang, et al., 2023)

Blood dataset has no missing values in the dataset with all having the same shape of 28 x 28 x 3. There are 8 classifications all with varying degrees of data provided ranging from 122 to 2026 for training and validation datasets. The data set provided is with the following ratio of 7:1:2, Training, Validation and Test. This can be in figure 3. Due to nature of 8 classifications in the dataset, it is labelled as multiclass. (Yang, et al., 2023) (Imbalance-learn, N/A)

# Analysis of the existing DNNs

In this coursework, we have used 5 pre-trained DNN models to develop models for our dataset. With 4 pretrain in BloodMNist dataset and 1 in BreastMNist.

The best performing models for our use case are:

## ResNet\_V1\_50

This model has 4 stages, each stage has a mixture of 1x1 Conv and 3x3 Conv layers all connected to a Average Pooling Layer. The output layer is Fully Connected Layer with 1000 classes. (Ji, et al., 2019)

Diagram, box and whisker chart

Description automatically generatedStructure:

Figure ResNet\_V1\_50 model architecture. (Ji, et al., 2019)

The default input shape this model can take is 224,224,3 with the out of 1001 classes for classification. The model’s input size is changed, and its output flattened. Then it undergoes few dense layers before getting to the output layer. The input is changed to 28,28,3 for our use case, and output into 8 classes.

## Inception\_V3

This model was released in 2015, it has 42 layers with multiple inception modules in it. This model has major modifications from Inception v1 like Factorization into small Convolutions, Spatial Factorization into Asymmetric Convolutions, Utility of Auxiliary Classifiers and Efficient Grid Size Reduction. The default input shape is 299x299x3, with output of 1000 classes for classification. The model’s input size is changed, and its output flattened. Then it undergoes few dense layers before getting to the output layer. The input is changed to 75,75,3 for our use case, and output into 2 classes. (Brital, 2021)

## Inception\_V1

This model is also called Google Net, it has achieved top spot at classification task during ILSVRC (ImageNet Large Scale Visual Recognition Challenge) -2014. It has over 6.7 million parameters. The concept behind this is to use all the operations at the same time and then concatenated into a single output. Such a module is called Inception Module. (machine-learning, 2022)

This approach makes the model computationally less expensive. This model has 9 inception modules where Global average Pooling is used as a FC Layer. It has 2 auxiliary SoftMax layers at the output layer where both give regularization effect. (machine-learning, 2022)

The default input shape this model can take is 224,224,3 with the output of 1000 classes for classification. The model’s input size is changed, and its output flattened. Then it undergoes few dense layers before getting to the output layer. The input is changed to 28,28,3 for our use case, and output into 8 classes.

The other models which performed poorly in our use case are:

## Efficientnet lite3

This model is modified version of Efficientnet-B3 by removing squeeze and excite modules and replaces all swish with ReLU6 and fix the stem and head while scaling models up. The default input shape this model can take is 300x300 with the output of 1000 classes for classification. The model’s input size is changed, and its output flattened. Then it undergoes few dense layers before getting to the output layer. The input is changed to 28,28,3 for our use case, and output into 8 classes. (Tensorflow, N/A)

## Efficientnet lite0

This model is modified version of Efficientnet-B0 by removing squeeze and excite modules and replaces all swish with ReLU6 and fix the stem and head while scaling models up. The default input shape this model can take is 300x300 with the output of 1000 classes for classification. The model’s input size is changed, and its output flattened. Then it undergoes few dense layers before getting to the output layer. The input is changed to 28,28,3 for our use case, and output into 8 classes. (Tensorflow, N/A)

# Analysis of the designed DNNs

## BreastMnist

It is observed that the Breastmnist dataset consists of 546/78/156 in the category of train, validation and test respectively. The size of each image is 28 x 28 in grayscale format. Different combinations of preprocessing techniques are used on the novel model for optimization.

The dataset, as mentioned above, has a positive class considered as ‘Normal/Benign’ - 1, and negative class considered as ‘Malignant’ - 0. This might be misleading in the context of medical diagnosis, since positivity always means the presence of abnormalities/diseases and negative, being normal. Therefore, the authors performed value switching on the train, validation, and test labels mapping 0 -> 1 and 1 -> 0, and this will be more descriptive since now positive (1) means is ‘Malignant’ (CANCEROUS), and negative (0) means is ‘Normal/Benign’ (Healthy/Not to be concerned).

The below diagram shows the skeleton of the novel model used for BreastMnist dataset:

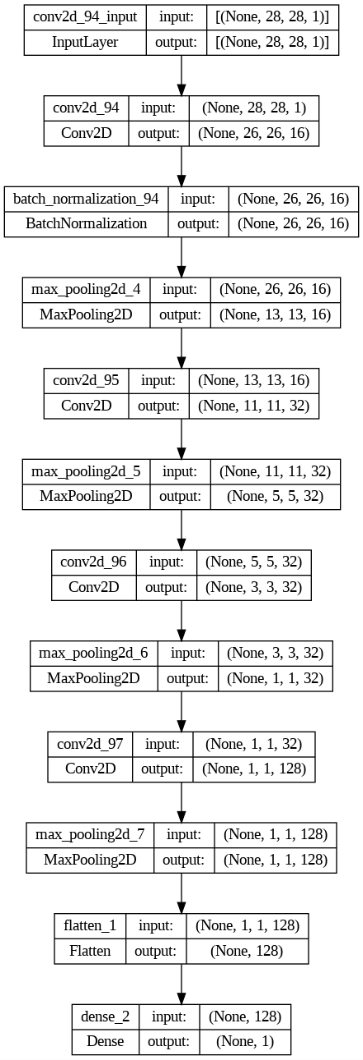


Figure Novel Breast Architecture

The novel model on BreastMnist dataset initially defined an input layer of shape 28,28,1 to relate with the dataset as explained above.

A 2D convolution layer of 16 filters is used in the input layer with the filter shape of 3 by 3 and Relu was used as the activation function to perform the non-linear functionality in each neurons. The 2D convolution layer is used since the images are 2D (Length\*Width). A batch normalization layer alongside with Conv2D is implemented. Then a MaxPooling2D layer with 2,2 pool size and stride of 2 is used for performing down sampling in the process of feature extraction.

A second 2D convolutional layer of 32 filters is used with the filter shape of 3 by 3. Then a MaxPooling2D layer with the same specifications is followed.

A third 2D convolutional layer of 32 filters is used for with the filter shape of 3 by 3 and with the activation function being the Relu. Unlike the first and second, the third Conv2D layer has been regularized using L2 with an alpha value of 0.0005. This helped in recognizing feature weights that significantly impacts the prediction accuracy and F1 score, eventually promoting smaller coefficients for the feature weights.

A similar MaxPooling2D layer was used again for down sampling.

A fourth 2D convolutional layer of 128 layers with the filter shape of 3 by 3. This layer also is regularized using L2 regularization with alpha value of 0.0005 with the activation function being Relu.

Finally, we flatten all the layers used to point to a single fully connected layer which is a Dense layer with 1 neuron, predicting the probability of the binary class between 0 and 1, with non-linear operation using sigmoid function. This layer is also regularized with L2 regularization and an alpha value of 0.5 to mainly consider the coefficients of the significant features that are contributing to do the prediction.

## BloodMNist

It is observed in BloodMNist, with the following categories, Training, Validation and Test: 11,959/1,712/3,421 shown in figure……….Due to the imbalance of data provided for each classifications, we have increase the maximum data for only training and validation data: training to 2330 and validation to 333 by using Random Oversampling, which randomly duplicates data. This ensures that all the classifications have the same amount of data which enables DNN models to increase their overall accuracy.

With the size of image 28 x 28, the model structure is shown below:

Diagram, table

Description automatically generated

Diagram

Description automatically generated

Figure Novel Blood Architecture

The model provided has 5 convolution (2D) layers starting from 16,32,64,128 and 256 and 5 max pooling (2D). Each of the convolution layer has a kernel size 2,2 with the activation function set as relu. As for the max pooling layer all of the layers have kernel size of 2,2, but the strides for the first two layers is 1 and the last three layers is 2. This is done to maximize learning close to input image at its native size.

A Flatten layer is added to bring the 4 dimensional output into 2 dimensional output. (i.e. number of samples and its prediction). After flattening, a dense layer (256) with relu activation is added to reduce the predicted classes from 1024 to 256. At this point a dropout is added to prevent the model overfitting to the training data. Then 2 dense layers (64,32) with relu activation is added to fine train at the last stage before the output layer.

The output layer is also a dense layer(8) with activation function as softmax. We use L1L2 regularization to increase the performance of the model.

# Analysis of the training process

The Image size of the blood dataset is RGB layered square image of 28 pixels (28x28x3). The data is trained after Random Over Sampling the data to make up for the class imbalance. Since the validation data is also used while training, both train and validation data undergoes Random Over Sampling. However, Test Dataset is used only for performance metrics, hence class imbalance won’t affect the model’s performance. The image is given as input to the model and an output of 8 classes. This received output is binarized, hence, to calculate the performance metrics, True test predictions is label binarized before calculation. Finally, when an individual data entry is given the predict, the output is converted from a binarized form to a label which returns the class label of the input.

The BreastMnist dataset has grayscale images with the height and width of 28 by 28 pixels. This dataset is used for binary classification to identify if the cell has Malignant tumor or not. Label 0 represents the presence of Malignant while Label 1 represents Normal cells. But according to the medical concept we always consider a class positive when the disease/abnormality is present. Therefore, the Labels have been transposed such that label 0 represents Normal (Negative) and 1 represents the presence of Malignant (Positive). It is also realized that the dataset is imbalanced with samples of label 1 (Malignant Positive) cases are 147 and label 0 (Malignant Negative) cases are 399. In order to tackle this imbalance issue, RandomOverSampler from imblearn is used to balance the datasets such that each class has equal number of samples.

Also, as a part of data processing, augmentation is performed in order for the CNN model to learn more about the features in different positions.

When performing hyperparameter tuning, GridSearchCV was used to automate the process. Different list of parameters like optimizers, number of epochs, batch sizes were given and GridSearchCV trains all the different combinations.

Training process of the BreastMnist dataset involved 4 separate sub processes:

* Train with no augmentation and over sampler
* Train with augmentation alone
* Train with over sampler alone
* Train with augmentation and over sampler

The Novel model is trained 4 times as above and the relevant performance comparison is given in the next section of the report.

# Analysis of the designed interactive application and the related GUI

A web-based application has been developed for external users to upload images and obtain predictions from our model.

The architecture of the web application is as follows:

* Backend, containing the models, is exposed via FASTAPI, a REST API framework from python.
* Image preprocessing is done using PILLOW and Numpy, libraries available in python
* Frontend, developed with ReactJS framework and used some npm libraries to upload the image file.

The API in the backend receives the images in the form of byte codes and reconstruct it again to the image using PILLOW. The image is then converted to nd array (numpy array). These arrays are then used for Rescaling/Resizing and normalization and then passed into the model for prediction.

The output is then sent back to the client as a response to the API call made. The response consists of similar JSON object as below:

{

Class: Malignant,

Confidence: 0.876

}

React app will then fetch this object and display in a user-friendly manner. A snapshot of the flow is given below. Also the complete application code is available in [[github](https://github.com/KUMedicalAI)].

# Comparative analysis

The performance chart from Jiancheng Yang and his team using pretrain model are shown in Figure 7. With 4 Pretrained models performing well in breast and blood dataset: ResNet-18(28), ResNet-18(224), AutoKeras, Google AutoML Vision with result over 90% in different categories. The worst performing models would have to be auto-sklearn with accuracy score of 80.3% in breast and 87.8% in blood with AUC score of 83.6% in breast and 87.8% in blood. The best model overall is Google AutoML Vision with accuracy score of 86.1% in breast and 0.966 in blood. The ROC AUC score for Google AutoML Vision is 91.9% for breast and 96.6% for blood. (Yang, et al., 2023) (Yang, et al., 2023)

Table

Description automatically generated

Figure Accuracy Matrix for BloodMNist and BreastMNist. (Yang, et al., 2023)

ResNet-18 (28) model score very well for with accuracy rate for blood at 96% followed by breast at 86%. As for the AUC (areas under curves), blood performs better with 99.8% followed by breast at 90.1%. The accuracy rate is higher for blood than breast due to the size of the model architectures layer and the amount of data available for the model to train and validate. This is prevalent when comparing the available data for breast and blood, with breast containing 780 images while blood contains 17,092 images. As well as the model, ResNet 18 to ResNet 50 as the model architecture for ResNet 50 is more complex with 50 deep layers while ResNet only has 18 deep layers. ResNet 18 performs better overall as the model is less complex and the images are not in-detailed (image has shrunk from 224 x 224 to 28 x28). (Boesch, 2023)

We will be mainly focusing on the F1 and AUC scores, Accuracy matrix will not be looked at as much as it can be very misleading if the dataset is imbalanced. (Korstanje, 2021) (Narkhede, 2018)

## Breast Accuracy Score

Table

Description automatically generated

Figure BreastMNist Accuracy Score

In figure 8, all results are compared and the best models produce is Novel Breast with oversampling and no augmentation (NB3) with highest scoring in Accuracy (85.256), Recall (85.256),F1 Score (85.079) and ROC AUC (0.880). The pre-trained model we have used is Inception V3 with an F1 score of 81.591 and ROC AUC score of 0.836 which is the third highest result compared to other Novel breast models. The model performs better with no augmentation and with oversampling, this may be due to the case of not having lots of sample pool to train and validate on when training the model. Therefore, lower F1 and ROC AUC score for NB4. Another reason why it may have achieved lower scores compared to its peers could be in the form of Novel Breast architecture with smaller architecture size. However, this is proven to be wrong due to the size of Inception\_V3 with 42 deep layers.

Chart, treemap chart

Description automatically generated

Figure NB3 Confusion Matrix on Prediction Accuracy

Chart, treemap chart

Description automatically generated

Figure Inception\_V3 Confusion Matrix on Predictive Accuracy

When looking at the confusion matric for NB3 and Inception\_V3 seen in figure 9 and 10, both models have higher predictive accuracy for true negative results with the highest being benign/normal. The predictive accuracy score for NB3 is around 82% while Inception\_V3 is around 78% with the difference at 4%.

Since this is a sensitive context (identifying Malignant cells), the true positive prediction is monitored carefully which will have to be demonstrated appropriately. Also, the prediction threshold is set to 0.4 in the model application, I.e. any prediction probability that is less than 0.4 is considered a Normal case and anything greater is a Malignant case. This threshold value is thus used to reduce any false alarms.

Comparing NB3 and Yang and his team’s pretrain model seen in figure 8, the ROC AUC rate is better in Google AutoMl Vision with scores of 91.9% and the best accuracy score is ResNet-18 (28) with. The best model cannot be definitive due to the insufficient matrix on F1 score, Precision and Recall, however it would be best to use Yang and his teams pretrain ResNet-18(28) with the accuracy score of 86.3%.

Please note that the accuracy score and confusion matrix score can change slightly due to the model randomly running the samples.

## Blood Accuracy Score

Table

Description automatically generated

Figure BloodMNist Accuray Score

Comparing all the results from figure 11, Novel Blood model with random sampling has the best performance with F1 score: 93.727 and ROC AUC: 99.5%, followed by ResNet\_V1\_50 with F1 score: 73.512 and ROC AUC: 95.0% and Inception\_V1 with F1 score: 74.314 and ROC AUC: 94.4%. F1 matrix are the combination of matrix score from Precision and Recall with harmonic means. It is best to use F1 score as a base for accuracy. As for the ROC AUC, it is a probability curve and measures the distinguishment of classification from each other in the model. The worst performing model is Efficientnet lite3 with F1 score of 26.654 and ROC AUC score of 82.3%. (Korstanje, 2021) (Narkhede, 2018)

Comparing the model created with and without random sampling, with sampling performs better than without sampling. This could be due to the availability of the sample has increase which therefore allows the Deep neural networking architecture to have more samples to analyse from. The increase is less than 1% for all matrix categories.

Yang and his team’s pretrained model shown in figure 8 to a similar pretrained model ResNet-50, their models perform better with the AUC of 95.6% while our ResNet model performs at 95%. The accuracy score for Yang and his teams are better with scores of 99.7% as for our pretrained model only performs 70.4%. However, it should be noted that accuracy score can produce a misleading report if the data is imbalanced. The F1 score and ROC AUC should be used to compare to other models.

When comparing Novel Blood model with random sampling to Yang and his teams pretrained model seen in figure 8, the accuracy score for our model performs exceptionally well compared to their pre-trained models with an accuracy score of 93.66% where theres are performing under 92%. The AUC for Novel Blood models performs around the same range of 99% with the difference of -0.3%.

However, comparing the confusion matrix of Novel Blood with random sampling to ResNet\_V1\_50, the Novel Blood can accurately predicts platelet with 100% and second highest prediction of 99% for erythroblast. The lowest prediction score located at basophil with 76%. The overall accuracy rate for prediction is around 93% calculated from the confusion matrix. ResNet\_V1\_50 model can accurately predict platelets at 92% with the lowest prediction accuracy score of 33%.

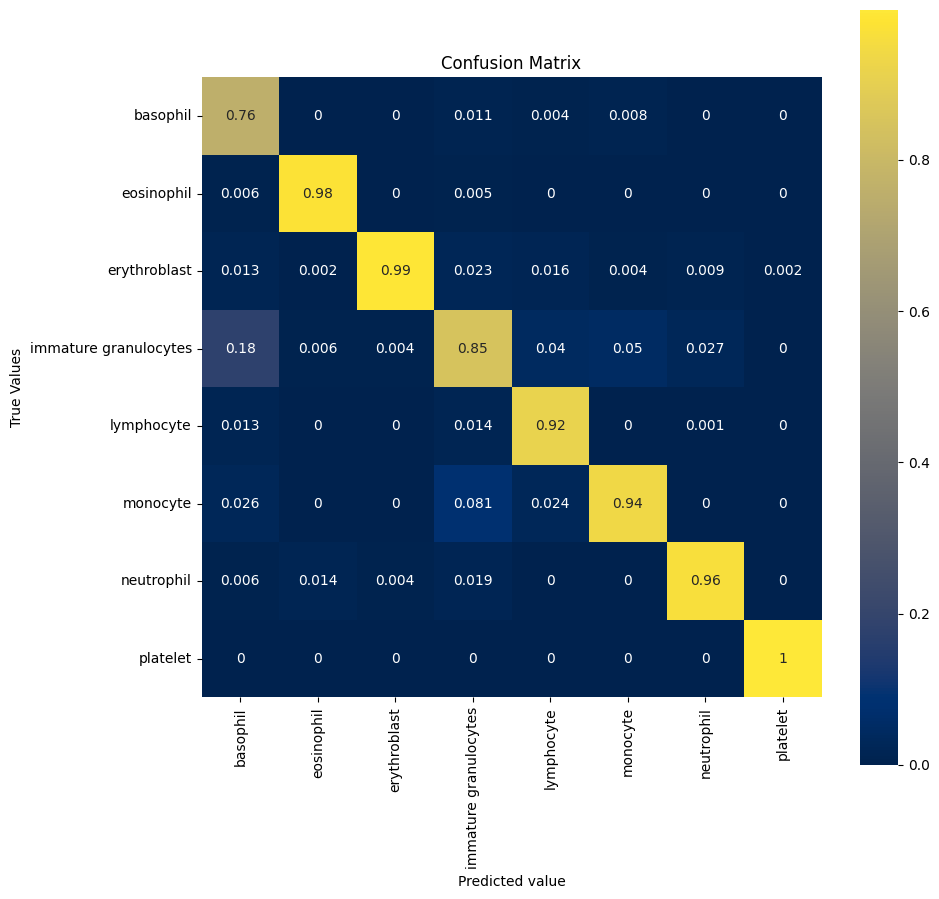
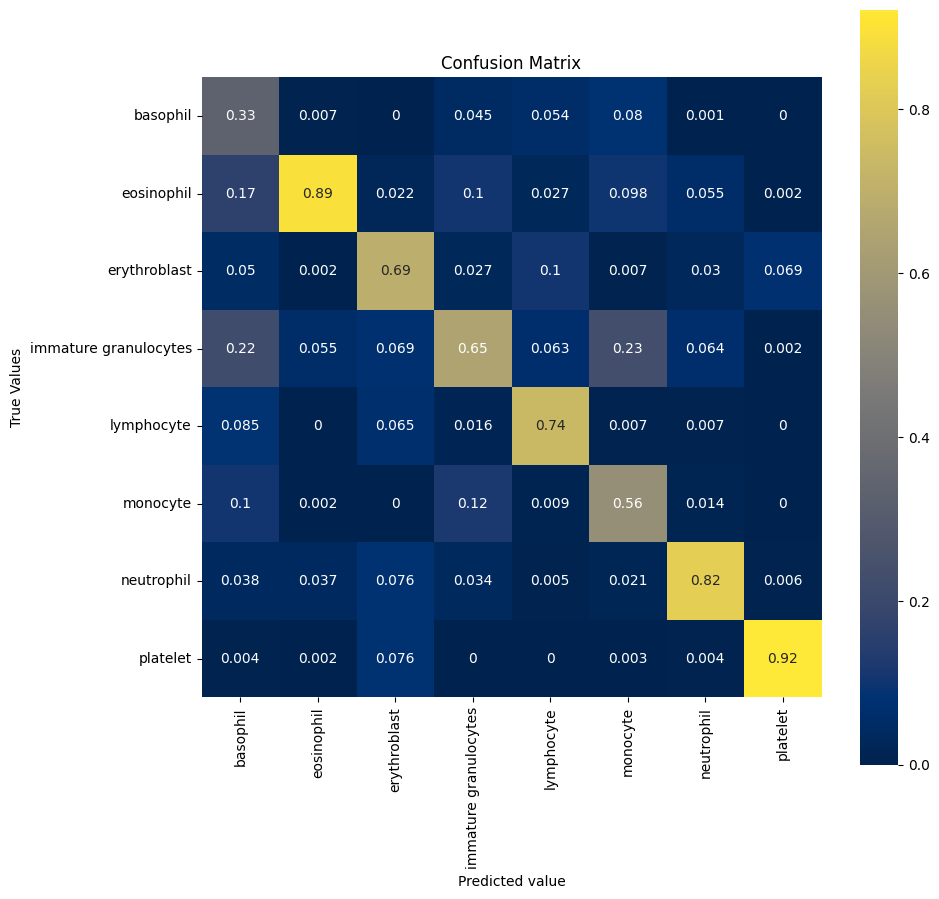


Figure 13 ResNet\_V1\_50 confusion matrix on Predictive Accuracy

Figure 12 Novel Blood with random oversampling confusion matrix on Predictive Accuracy

Any applications with a low predictive score will need to be cautious with the result handed out as they can produce a false positive report. In this case, it would be a serious matter as patient health are on the line. Therefore, it would be best to use models with a high F1 score, ROC AUC score and high confusion matrix. It is difficult to come with the conclusion on whether which model is better due to the insufficient matrix on F1 score, Precision and Recall, however it would be best to use Yang and his teams pretrain Google AutoML Vision image size 224 x 224.

Please note that the accuracy score and confusion matrix score can change slightly due the model randomly running the samples.

# Conclusion

Out of all the models we have created, the best four Deep Learning models have been presented in this report. A novel model and ResNet50 model for Blood dataset; A novel model and InceptionV3 model for Breast dataset. Comparing all models’ performance metrics, we can conclude that in both cases the novel model performs better compared to the pretrained model. This can be explained by the image size used. The novel model is designed to facilitate the classification of a square image of size 28 pixels. Whereas the pre-trained models are designed to work with 224 sized square images, hence when an image of size 28 is used, these models don’t have enough data to train perfectly. But these models are powerful as they are able to produce about 70-73 % of accuracy even with this huge disadvantage of image size.

For the given datasets, it is best to use the pre-train models used by Jiancheng Yang and his team as they give a higher performance metrics compared to our models. This is mostly because the data they trained the models in is a high-resolution data of 360x360 pixels and 500x500 pixels. Hence, the performance metrics are higher as the optimum pre-train model input is the 224 square pixels. But the data given in the dataset is the reduced form of this image, which on expanding to 224x224 causes the model to break and not perform well.

# Reference

Boesch, G. (2023) Deep Residual Networks (ResNet, ResNet50) – 2023 Guide. [Online] Available at: <https://viso.ai/deep-learning/resnet-residual-neural-network/#:~:text=Deep%20residual%20networks%20like%20the,other%20to%20form%20a%20network>. [Accessed 10 February 2023].

Brital, A. (2021) Inception V3 CNN Architecture Explained. [Online] Available at: <https://medium.com/@AnasBrital98/inception-v3-cnn-architecture-explained-691cfb7bba08> [Accessed 10 March 2023].

Brownlee, J. (2016) Use Keras Deep Learning Models with Scikit-Learn in Python. [Online] Available at: <https://machinelearningmastery.com/use-keras-deep-learning-models-scikit-learn-python/> [Accessed 8 March 2023].

Brownlee, J. (2022) How to Calculate Precision, Recall, F1, and More for Deep Learning Models. [Online] Available at: <https://machinelearningmastery.com/how-to-calculate-precision-recall-f1-and-more-for-deep-learning-models/> [Accessed 3 March 2023].

Foo, F. (2014) How to normalize a confusion matrix?. [Online] Available at: <https://stackoverflow.com/questions/20927368/how-to-normalize-a-confusion-matrix> [Accessed 15 March 2023].

Google (N/A) imagenet/inception\_v1/classification. [Online] Available at: <https://tfhub.dev/google/imagenet/inception_v1/classification/5> [Accessed 12 March 2023].

Google (N/A) imagenet/resnet\_v1\_50/classification. [Online] Available at: <https://tfhub.dev/google/imagenet/resnet_v1_50/classification/5> [Accessed 10 March 2023].

Hoorn, R. v. (2022) Ways to Reduce False Positive or False Negatives in Binary Classification (0,1). [Online] Available at: <https://ai.stackexchange.com/questions/38405/ways-to-reduce-false-positive-or-false-negatives-in-binary-classification-0-1> [Accessed 10 March 2023].

Hugging Face (N/A) google/efficientnet-b3. [Online] Available at: <https://huggingface.co/google/efficientnet-b3> [Accessed 10 March 2023].

Imbalance-learn (N/A) imblearn.over\_sampling.RandomOverSampler. [Online] Available at: <http://glemaitre.github.io/imbalanced-learn/generated/imblearn.over_sampling.RandomOverSampler.html> [Accessed 25 February 2023].

Ji, Q., Huang, J., He, W. & Sun, Y. (2019) Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images. [Online] Available at: <https://www.researchgate.net/publication/331364877_Optimized_Deep_Convolutional_Neural_Networks_for_Identification_of_Macular_Diseases_from_Optical_Coherence_Tomography_Images> [Accessed 16 March 2023].

Keras (N/A) Adam. [Online] Available at: <https://keras.io/api/optimizers/adam/> [Accessed 26 February 2023].

Keras (N/A) <https://keras.io/api/applications/xception/>. [Online] Available at: <https://keras.io/api/applications/xception/> [Accessed 27 February 2023].

Keras (N/A) Layer weight regularizers. [Online] Available at: <https://keras.io/api/layers/regularizers/> [Accessed 27 February 2023].

Koech, K. E. (2020) Cross-Entropy Loss Function. [Online] Available at: <https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e#:~:text=Categorical%20cross%2Dentropy%20is%20used,%5D%20for%203%2Dclass%20problem> [Accessed 25 February 2023].

Korstanje, J. (2021) The F1 score. [Online] Available at: <https://towardsdatascience.com/the-f1-score-bec2bbc38aa6> [Accessed 15 March 2023].

Kumar, V. & jeffhale (2018) Inverse Process of Label Binarisation in Python. [Online] Available at: <https://stackoverflow.com/questions/45118093/inverse-process-of-label-binarisation-in-python> [Accessed 15 March 2023].

Liang, X. (N/A) Machine Learning and Artificial Intelligence. [Online] Available at: <https://kingston.app.box.com/s/9us9iwuz2nqm1fpgyf8oyy06zo81y263> [Accessed 5 February 2023].

machine-learning (2022) Inception-V1 (GoogLeNet): Summary and Implementation. [Online] Available at: <https://hackmd.io/@machine-learning/Bk-61Fo8U> [Accessed 14 March 2023].

Narkhede, S. (2018) Understanding AUC - ROC Curve. [Online] Available at: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5> [Accessed 15 March 2023].

NumPy (N/A) Array creation routines. [Online] Available at: <https://numpy.org/doc/stable/reference/routines.array-creation.html> [Accessed 10 March 2023].

Rana, A. (2019) Scikit-learn's LabelBinarizer vs. OneHotEncoder. [Online] Available at: <https://stackoverflow.com/questions/50473381/scikit-learns-labelbinarizer-vs-onehotencoder#:~:text=Scikitlearn%20suggests%20using%20OneHotEncoder%20for%20X%20matrix%20i.e.,you%20won%27t%20really%20need%20that%20in%20y%20labels> [Accessed 10 March 2023].

Sachan, A. (N/A) Detailed Guide to Understand and Implement ResNets. [Online] Available at: <https://cv-tricks.com/keras/understand-implement-resnets/> [Accessed 10 March 2023].

Scikit Learn (N/A) sklearn.metrics.multilabel\_confusion\_matrix. [Online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.multilabel_confusion_matrix.html> [Accessed 5 March 2023].

Scikit Learn (N/A) sklearn.preprocessing.label\_binarize. [Online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.label_binarize.html> [Accessed 3 March 2023].

Seaborn (N/A) seaborn.heatmap. [Online] Available at: <https://seaborn.pydata.org/generated/seaborn.heatmap.html> [Accessed 15 March 2023].

Tang, L. (2021) Draw Confusion Matrix for CNN models. [Online] Available at: <https://leslietj.github.io/2020/06/22/Draw-Confusion-Matrix-for-CNN-models/> [Accessed 15 March 2023].

T, D. (2019) Confusion Matrix Visualization. [Online] Available at: <https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea> [Accessed 28 February 2023].

Tensorflow (N/A) efficientnet/b3/classification. [Online] Available at: <https://tfhub.dev/tensorflow/efficientnet/b3/classification/1> [Accessed 12 March 2023].

Tensorflow (N/A) efficientnet/lite0/classification. [Online] Available at: <https://tfhub.dev/tensorflow/efficientnet/lite0/classification/2> [Accessed 12 March 2023].

Tensorflow (N/A) efficientnet/lite3/classification. [Online] Available at: <https://tfhub.dev/tensorflow/efficientnet/lite3/classification/2> [Accessed 12 March 2023].

University of Michigan Library (2022) All About Images. [Online] Available at: <https://guides.lib.umich.edu/c.php?g=282942&p=1885347#:~:text=Not%20all%20of%20our%20images%20are%20the%20exact,will%20be%20discarded%20by%20the%20photo%20editor%20%28Photoshop%29> [Accessed 16 March 2023].

Verma, S.(2019) A Simple Guide to Using Keras Pretrained Models. [Online] Available at: <https://towardsdatascience.com/step-by-step-guide-to-using-pretrained-models-in-keras-c9097b647b29> [Accessed 1 March 2023].

Yalçın, O. G.(2018) Image Classification in 10 Minutes with MNIST Dataset. [Online] Available at: <https://towardsdatascience.com/image-classification-in-10-minutes-with-mnist-dataset-54c35b77a38d> [Accessed 26 February 2023].

Yang, J. et al. (2023) MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification. [Online] Available at: <https://www.nature.com/articles/s41597-022-01721-8> [Accessed 24 February 2022].

Yang, J. et al.(2023) MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification. [Online] Available at: <https://medmnist.com/> [Accessed 23 February 2023].