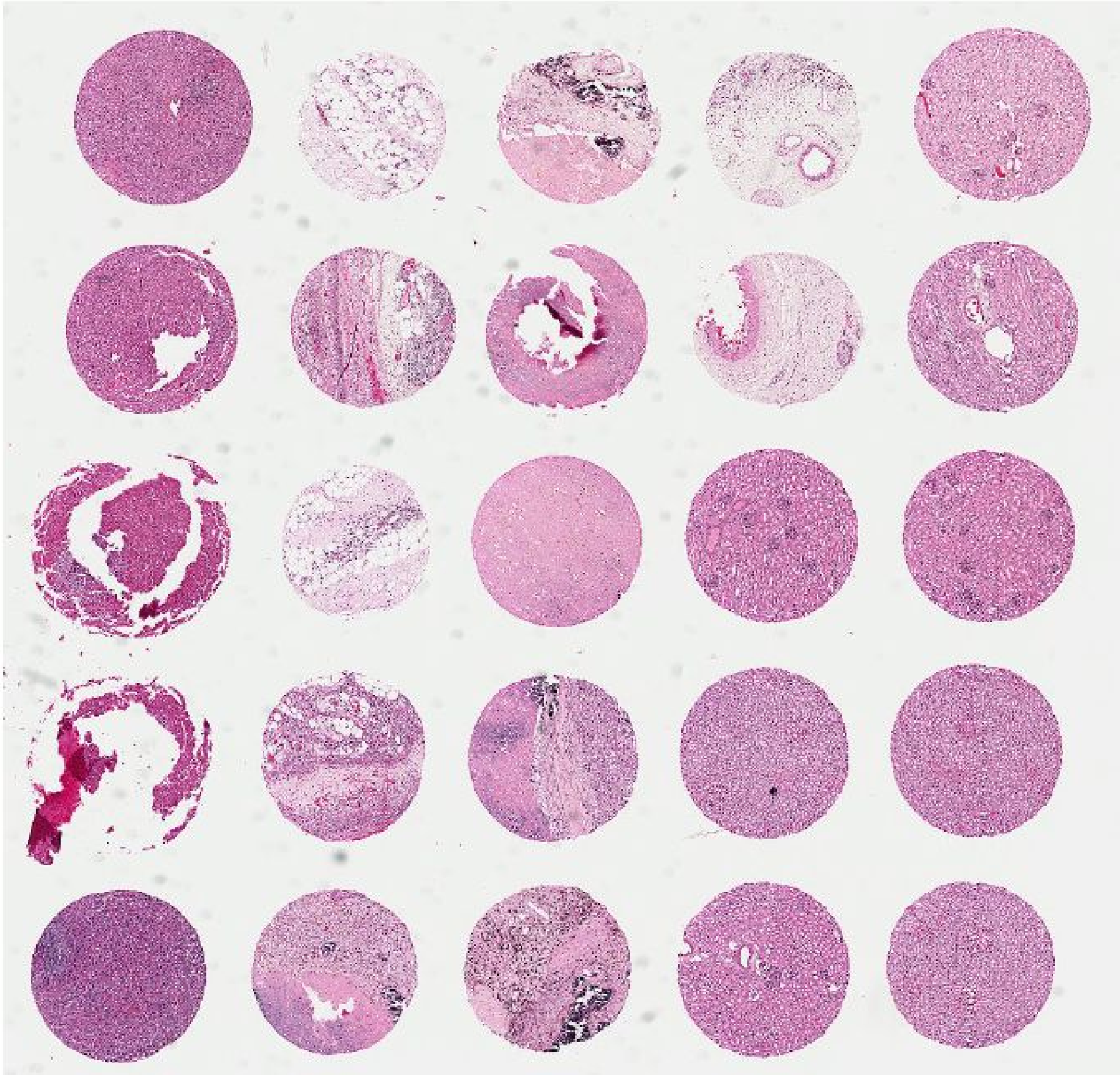


Classification of cancer tissue images using CNNs

Final project

ZiHao Yang	13019738
Tetiana Kraay	12605109
Khannika Chanachai	12184673

Table of Contents

1.Problem description **4**

2.Challenges **7**

3. Methodology **7**
- 3.1. Convolutional Neural Network architecture 7
- 3.2. Data exploration 11
  - 3.2.1. Data description 11
  - 3.2.2. Dataset issues 12
  - 3.2.3. Data splitting 12
  - 3.2.4. Data augmentation 13
- 3.3. ResNet50 CNN Model 13
  - 3.3.1. ResNet model description and structure 13
  - 3.3.2. Transfer learning 15
  - 3.3.3. Fine-tuning 16
- 3.4. ResNet34 16
  - 3.4.1. ResNet34 Description 16
  - 3.4.2. Number of Epochs 17
  - 3.4.3. Dropout 18
  - 3.4.4. Optimizer selection 19
  - 3.4.5. Regularization 20
  - 3.4.6. Learning Rate 21

4. Achievements and Findings **22**
- 4.1. Confusion matrices 22
- 4.2. Classification Report 24
- 4.3. Evaluation 24

5. Recommendations **26**

6. Difficulties encountered **27**
  - 6.1. High performance outcome require 27
  - 6.2. Optimization 28
  - 6.3. The application of transfer learning 28
  - 6.4. Dataset and time limitation 28

7. Conclusions **29**

Table of individual contribution **30**

Appendices **30**
- Original Dataset 30
- Sources documentation 30
- Designed Model scheme 31
- Github link 31

References **32**

Executive summary

Convolutional Neural Network (CNN) is one of the state-of-art and latest deep learning approaches to image classification. CNN is designed to extract high-level characteristics from histological images even if the images are shrunken, shifted or rotated. As CNN is powerful tool in image classification, in this project TAZ Data Science & Co proposed the method for Neuroblastoma cancer images classification for data provided by our client The Children's Hospital at Westmead. The original dataset is comprised of 1043 images of 5 labelled types of Neuroblastoma that have been primarily classified and diagnosed visually using Shimada microscopic grading system. According to this system, 5 types of Neuroblastoma are assigned to Undifferentiated Neuroblastoma, Poorly Differentiated Neuroblastoma, Differentiating Neuroblastoma, Ganglioneuroma and Ganglioneuroblastoma. Applying Convolutional Neural Networks to given dataset, our team is aiming to demonstrate high artificial intelligence capability of CNN classifier that would achieve convincing performance in classifying Neuroblastoma cancer images. The dataset underwent through some preparation and preprocessing stages considering its small size and imbalanced structure. The initial findings are summarised and presented in the paper. Encountered difficulties while feeding and running a model with preprocessed dataset are described.

1.Problem description

Neuroblastoma is the most common solid extracranial tumor in infants and children and the second most frequent cause of mortality in children with cancer. Neuroblastoma is a type of tumor composed of "neuroblasts", specifically neural crest cells, which are cells involved in the development of the sympathetic nervous system.

There are five types of neuroblastoma (NT), and they are categorised based on the size and shape of the tumor cells: undifferentiated (UDNB), poorly differentiated (PDNB), differentiating (DNB), ganglioneuroma (GN) and Ganglioneuroblastoma (GNB) (Figure 1).

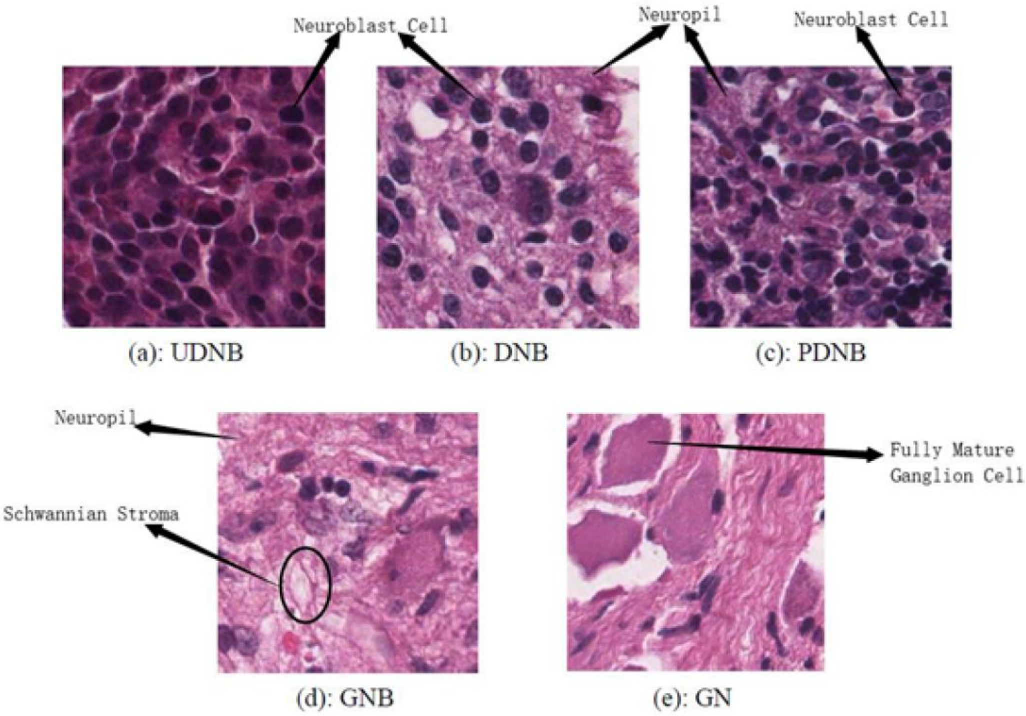


Figure 1. Representative images of five categories of neuroblastoma: (a) Undifferentiated neuroblastoma, (b) Differentiated neuroblastoma, (c) Poorly differentiated neuroblastoma, (d) Ganglioneuroblastoma, (e) Ganglioneuroma. All collected samples are classified according to Shimada classification system (Shimada 2003).

An undifferentiated neuroblastoma (Figure 1, a) is mostly comprised of neural crest cells, which are sometimes called small round blue cells, due to a lack of cytoplasm, and a large blue nuclei. A poorly differentiated neuroblastoma (Figure 1, c) is composed of slightly more differentiated cells, with smaller nuclei and more cytoplasm. The cells of poorly differentiated neuroblastoma may be surrounded by a substance called neuropil, which is a dense network of interwoven nerve fibers, like axon and dendrite branches. Differentiating neuroblastoma (Figure 1, b) is composed of cells with lots of cytoplasm, and is often surrounded by substance called Schwannian stroma. Schwannian stroma is a connective tissue from non-neuronal cells that forms myelin, which insulates the axons of the mature neurons and present in Ganglioneuroma and Ganglioneuroblastoma as well. When any type of NT forms, cells in the surrounding tissue release chemokines called CXCL12 and that stimulates nearby immune cells. CXCL12 is normally produced by some organs like the lymph nodes, liver, bones, and especially the bone marrow, but gets produced in higher amounts when tissue is damaged.
Due to its extremely variable clinical behavior, the most efficient way to diagnose this disease is to identify certain morphological features from the tumor tissues. The neuroblastic tumors are commonly analysed and characterised via histopathological examinations of stained tissue sections which is

performed by pathologists using optical microscope. The slices of tumor tissue undergo examining and further classification by Shimada system (Shimada 2003).

To achieve higher performance in the classification of histological images of NT, researches have expressed the high interest in developing approaches for computer assisted classification. Computer based classification of NT images can be implemented by two methods: segmented morphological features such as size and shape of tumor, and feature extraction method where the features that need to be classified are not always seen by human eyes. Even though NT cells belong to the same class, the morphological characteristics of those cells are very different. This becomes the major issue for the segmentation method while the existing feature extraction methods are based on the features which are not robust to scale variations.

With the wave of deep learning, the prediction of this kind of diseases has reached a higher level. Applying deep learning algorithms in biomedical applications where the labeled images are in scarcity, is becoming more prevalent over other methods of computing assisted image classification.

A deep Convolutional Neural Network have been previously used for classifying brain and breast cancer tissues (Zhang 2015, Alom et al. 2018). With proved capability and great performance to classify the tissues based on the cytological components in a multi-resolution manner, CNN based method has been chosen by our team to develop the classification of the grade of differentiation on the histological images of Neuroblastoma. As it is kept in mind that a large dataset of Neuroblastoma tissue samples can not be obtained, and transfer learning and fine-tuning of an ImageNet pre-trained CNN model on medical image datasets has not yet been exploited in previous researches and there is also no data augmentation method has been used on the classification of neuroblastoma histological images, for our project we explore the effectiveness of implementing these methods on neuroblastoma images classification and further develop their implementation in diagnosis establishing. The objective of the project is to use pre-trained CNN model from high-level neural network library (Keras) with transfer learning and fine-tuning methods applied to the model or construct an unique model with its own configuration, structure and parameters.

Our client is Children's Hospital at Westmead, Sydney, Australia. The dataset of 1043 histological images of neuroblastic tumors collected from 125 patients was provided by Tumour Bank of the Kid's Research Institute at the hospital. Tumor access is compliant with local policy, national legislation, and ethical mandates to use the human tissue in research. All patient-specific details were removed, and a de-identified dataset was used for this research.

The project focuses on using an image classification model to extract high-level images of Neuroblastoma tissues and develop the diagnosis implementation. During experimentation, our team will use a Convolutional Neural Network (CNN) via the high-level neural network library - Keras to build the high quality models and minimal number of coding. There are two models are being used during the experiment:

- ResNet50 - high-level pre-trained CNN model (ImageNet dataset);
- ResNet34 - customised CNN model with 34 layers trained from scratch.

2.Challenges

As earlier mentioned in the report, Neuroblastoma is one of the most frequently occurring cancerous tumors in children and finding the modern solution is very crucial. The project is very challenging for our team in terms of employing our knowledge and skills to find or develop the most appropriate CNN model. There are no known CNN approaches for classification of neuroblastoma histological images. The data collected from patients is very limited not only in a number of patients but also in a number of images. Therefore, we predict that certain techniques for data preprocessing and augmentation are going to be implemented during the experiment.

In case if using the pre-trained CNN model will not give desirable results and will show overfitting, our team will have to design and evaluate CNN without using transfer learning and constructing its unique structure of convolutional neural network with training the model from scratch where the model will have to learn complex, high-dimensional, non-linear mappings from large collections of samples images. This is a big task for our team as we strive to provide our clients with the robust and reliable high-performing CNN model in order to increase the accuracy of the classification of neuroblastoma histological images compared to state-of-the-art methods.

3. Methodology

3.1. Convolutional Neural Network architecture

The Convolutional neural networks as a neural network consists of layers of neurons and was designed similarly to a deeply complex hierarchical structure of neurons and connections in the human brain. Neurons can be thought as numbers' holders, specifically the numbers from 0 to 1. The figure below (Figure 2) illustrates the image 28x28 pixels of size where there are 784 neurons corresponding to each intersection of all 28 pixels in rows and all 28 pixels in columns. 784 neurons hold a number that represents the grayscale value (as this image is in grayscale) of the corresponding pixel ranging from 0 for black pixels up to 1 for white pixels. The number inside the neuron is called its activation. All 784 neurons make up the first layer of a neural network (Figure 3).

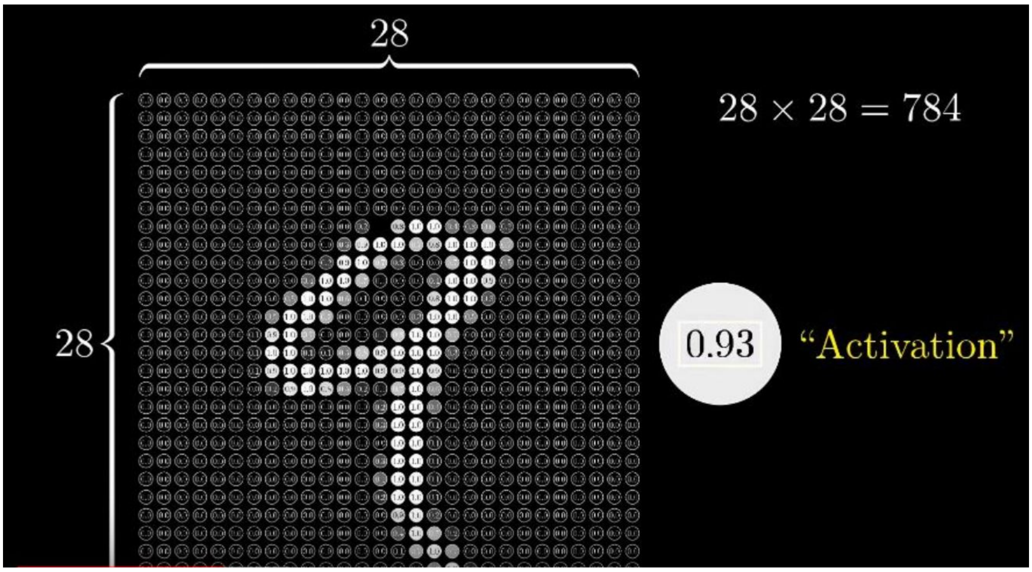


Figure 2. Image representation as 2-dimensional array of pixels.

The ultimate goal of the neural network is to produce the output where the image will fall in one of the categories the researches need to classify the dataset in (Figure 3). In the case of hand-written numbers dataset it is a top layer with 10 neurons.

There are two or more hidden layers in between. Every neuron in the next layer is fully connected to all neurons in previous layer. This is how neural network is organised to be able to classify digits from hand-written numbers.

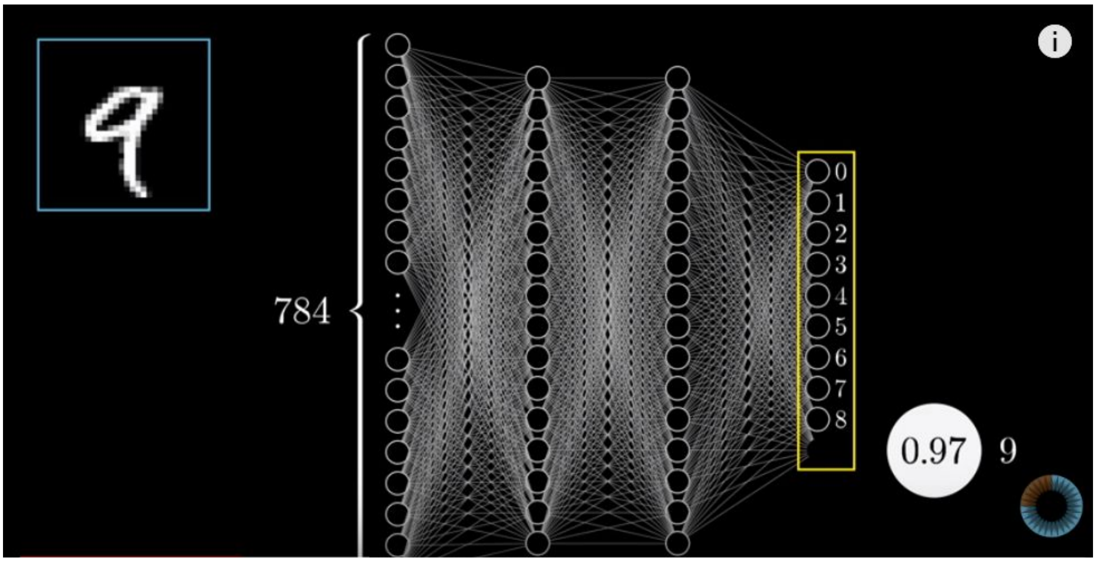


Figure 3. Neural network representation.

Convolutional neural network has some distinctive differences from neural network. Firstly, the layers are organised in 3 dimensions: width, height and depth. Secondly, the neurons in one layer do not connect to all the neurons in the next layer but only to small region of it. Lastly, the final output will be reduced to a single vector of probability scores, organized along the depth dimension.

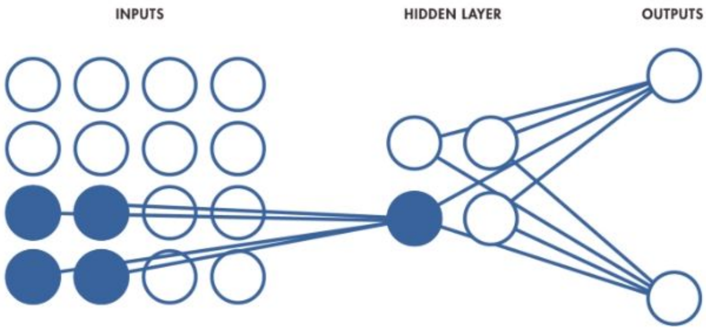


Figure 4. Convolutional neural network representation.

In the convolutional neural network only a small region of input layer neurons is connected to neurons in a hidden layer. These regions are referred as local receptive fields. The local receptive field is translated across an image to create a feature map from the input layer to the hidden layer neurons. In the feature map every hidden layer increases the complexity of the learnt image features. The first layer, for instance, learns how to detect edges, the last layer learns how to detect more complex shapes. The final layer connects every neuron from the last hidden layer to the output layer and is called fully connected layer.

The convolution is used in this method to implement the process efficiently. The neurons in hidden layer are extracting the same feature in different regions of the image. This makes network tolerant to translation of objects in an image, i.e. recognising the same feature in different images.

The next steps in classification are activation and pooling. The activation step applies transformation to the output of each neuron by using activation functions. Rectified linear unit or ReLu is an example of commonly used activation function (Figure 5). It takes the output of neuron and maps it to the highest positive value. If the output is negative the function maps it to zero.

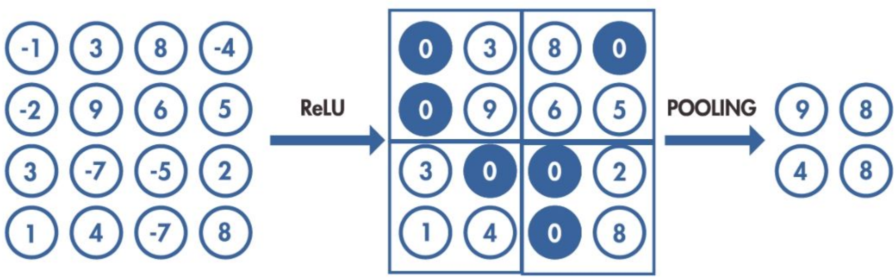


Figure 5. Activation function and pooling in CNN method.

The pooling step is further transformation of the output of activation function. Pooling reduces the dimensionality of the feature map like condensing the output of small regions of neurons into a single output. The function of pooling is to continuously reduce the dimensionality to reduce the number of parameters and computation in the network. This shortens the training time for the model to learn features.

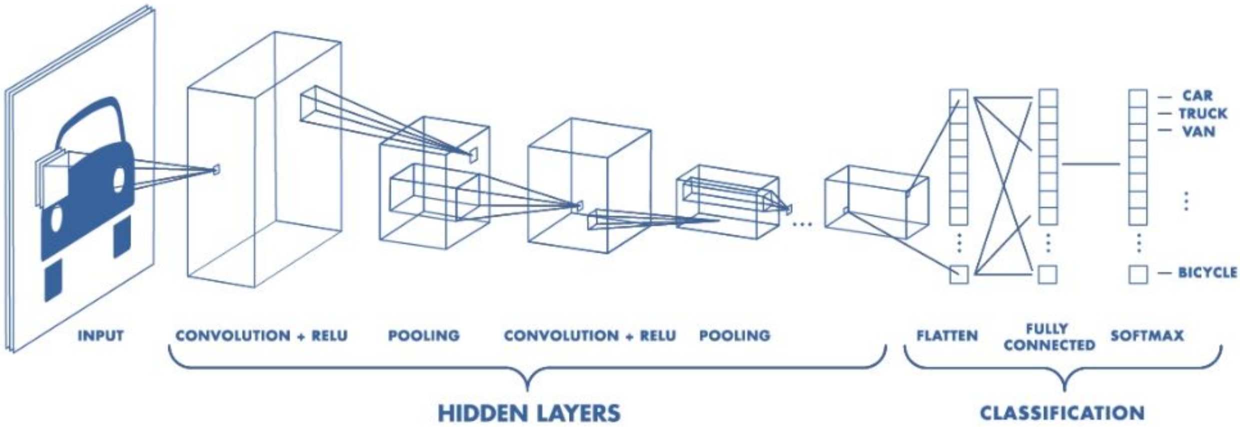


Figure 6. Convolutional neural network for image processing.

There are three three techniques for using CNN for image classification:

1. Train the CNN from scratch (Figure 7). This method is highly accurate and the most challenging due to the need of great number of labelled images and significant computational resources.

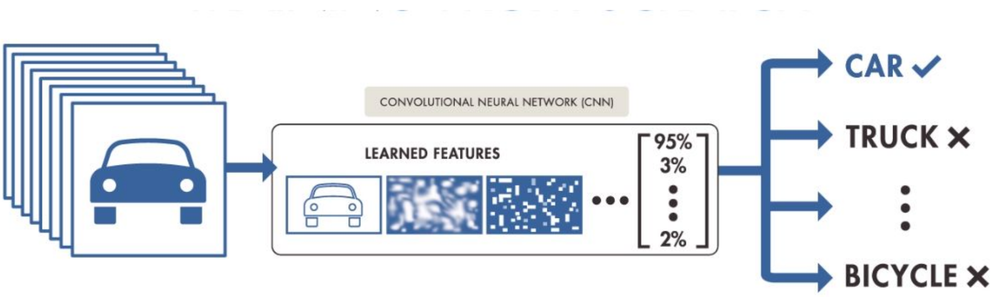


Figure 7. Training CNN from scratch.

2. Transfer learning when pre-trained from natural image dataset model is used for classifying images of other dataset by transferring the parameters of the trained model to the newly

constructed model to help its training (Figure 8). Fine-tuning method as a part of transfer learning is used here too. It enables the modification in certain hidden layers with changing parameters such as dropout rate, learning rate, number of epochs and optimizer selection.

3. Feature extraction method is used in pre-trained CNN to extract features for training the machine learning model (supported vector machine, decision tree).

.

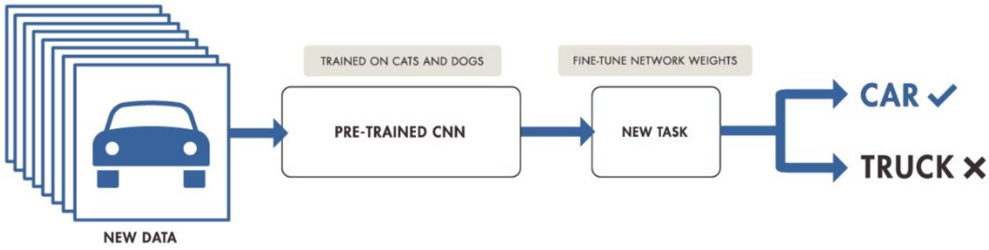


Figure 8. Transfer learning.

3.2. Data exploration

3.2.1. Data description

The original dataset contains 1043 histological images of neuroblastic tumors collected from 125 patients. These images have been gathered from the The Tumour Bank of the Kid's Research Institute at The Children's Hospital at Westmead, Sydney, Australia. Tumor access is compliant with local policy, national legislation, and ethical mandates to use the human tissue in research. All patient-specific details were removed, and a de-identified dataset was used for this research.
For each tissue sample of neuroblastic tumors, there are 20 to 40 1.2mm diameter coreson it. To show the histological structures of these samples clearly, the neuroblastic tumor tissues have been cut as the microscopical sections or microarrays with 3μm thickness. After that, they have been stained with Hematoxylin and Eosin (H&E). Then, the Aperio Scan Scope system and Image Scope software have been used for extracting and viewing the high-resolution images, and the initial dataset that we used has been obtained. Areas with the best representative of each category were cropped from tissue core at x40 zoom magnification. The size of cropped images is 300 x 300 pixels. Figure 9 illustrates the actual size of tissue spots and cropped images.

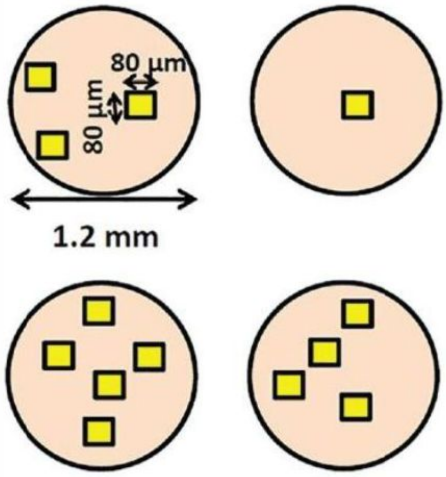


Figure 9. Actual size of cropped images of Neuroblastoma from tissue core. Adapted from Gheisari et al (2018).

According to the Shimada classification system, these images have been classified and labeled into 5 classes, which are undifferentiated neuroblastoma (UDNB), differentiating neuroblastoma (DNB), poorly-differentiated neuroblastoma (PDNB), ganglioneuroblastoma (GNB), and ganglioneuroma (GN) (see Appendices - Original Dataset).

3.2.2. Dataset issues

There are two major issues with dataset have emerged while studying and analysing it. Firstly, the dataset is small and has not enough data for training and building a model. This can possibly lead to overfitting the model and have further negative impact on model performance when testing the model. Secondly, the dataset is highly imbalanced. The number of images in each class are highly imbalanced and can result in bias in examining classes with more samples which is quite undesirable. The number of images for PDNB class is more than ten times larger than for GN class, three times larger than UDNB class.

3.2.3. Data splitting

For training and testing purposes the original dataset was split into separate training and testing subset. Training set is a dataset that used to train the model. During each epoch our model will be trained over and over again and will continue to learn about the features of the data. When we deploy the model later it will make predictions accurately on data never seen before. It will be making the predictions based on what it learnt about the training data. For current business case we split the original dataset on training/testing subset in proportion 80/20.The major difference between two subsets that test set is not labelled so we can see the metrics give during training like the loss in the accuracy from each epoch.

3.2.4. Data augmentation

To resolve the imbalance issue we have augmented training dataset to 600 images in each one out of 5 classes. The images in original dataset were augmented via ImageDataGenerator using processes (arguments in Keras library) such as rotation on 50 degrees, height shift range, width shift range and horizontal flip (randomly flips input horizontally).

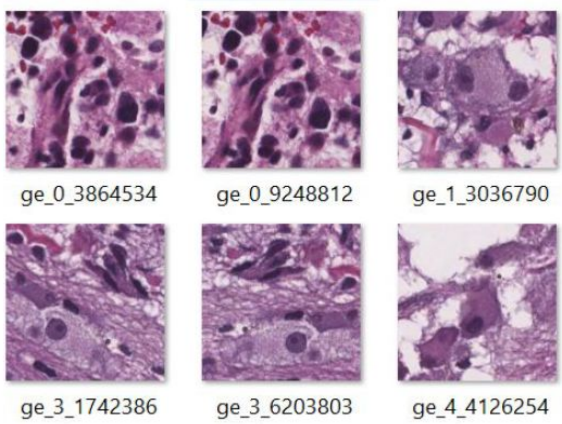


Figure 9. Augmented images of Differentiating Neuroblastoma (DNB).

3.3. ResNet50 CNN Model

3.3.1. ResNet model description and structure

After thorough exploration of dataset and understanding the problem of image recognition and classification, our team have found some interesting point that would support us to make our decision in developing the CNN model:

- Extract images' features by using high-level feature learning approach
- Explore the factors which affect performance of the model
- Evaluate the model

AT the beginning of the experiment the team decided to build Convolutional Neural Network in the Keras library by using the ResNet50 with weights trained on ImageNet to build the Convolution Neural Network (CNN). The RestNest is an improved version of the CNN that will assist for degradation problem of the CNN, by taking a shortcuts between the layer to help the network get more deeper. The RestNest allowed us to train extremely deep neural networks with 150+ layers successfully. Instead of 2 layered (3x3) convolutions, the RestNest 50 will use (1x1),(3x3), and (1x1) convolutions.

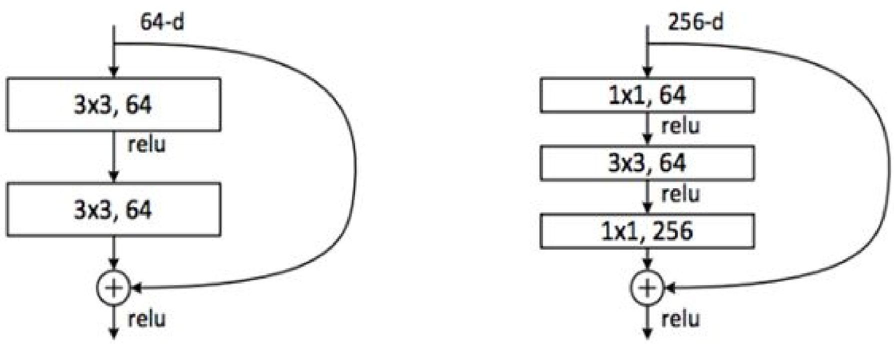


Figure 10. Residual Learning : Building blocks. Adapted from He et al (2015).

The strength of RestNest is skipping connection that helps to mitigate the problem of vanishing gradient, it allows the alternate shortcut path for gradient to flow through. Moreover, it supports the CNN model in understanding an identity function to ensure the higher layer that will perform at least as good as the lower layer, and not worse.

Here is the example of RestNest model which illustrated in figure 11.

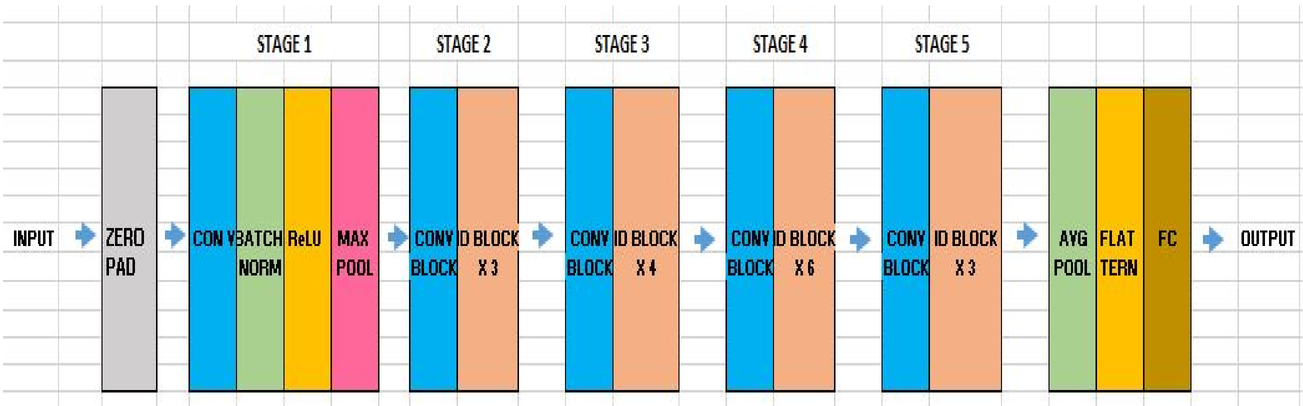


Figure 11. Example of RestNet Model.

Figure 11 shows that ResNet-50 model has 5 stages which is a convolution and identity block. There are 3 convolution layer in each each convolution block, also, each identity block consists of 3 convolution layers.

The details of this ResNet50 model are:

- **Zero-padding** : When the feature map is smaller than the input, padding will help to prevent the feature map from shrinking. At layer zero, the pixel value is added to surround the input, so the feature will not shrink. At the above figure the team pads the input with pad of 3,3 to keep the spatial size constant when performing the convolution.
- **Stage 1**: The 2D Convolution has filtering of shape as 64, then BatchNorm is applied to the channels axis of the input. After that the MaxPooling uses a (3,3) window and a stride 2,2.
- **Stage 2:** The convolutional block uses one set of filters of size 64 and the 3 identity blocks use three set of filters of size 64.
- **Stage 3:** The convolutional block uses one set of filters of size and the 4 identity blocks use three set of filters of size 128.
- **Stage 4:** The convolutional block uses one set of filters of size and the 6 identity blocks use three set of filters of size 256.
- **Stage 5:** The convolutional block uses one set of filters of size and the 3 identity blocks use three set of filters of size 512.
- The 2D Average Pooling uses a window of shape 2,2.
- The flatten doesn't have any hyperparameters or name.
- The Fully Connected (Dense) layer reduces its input to the number of classes using a softmax activation. Its name should be 'fc' + str(classes).

When using the RestNet for experimentation and training data set, there is a significant decreasing for validation loss value and a lower training loss. This trend shows the transfer learning under Convolutional Neural Network showing a higher performance from begining. However, to reach higher performance of the Convolution Network , our team will take the transfer learning and fine-tuning to develop the model.

3.3.2. Transfer learning

As it is kept in mind that a large dataset of Neuroblastoma tissue samples can not be obtained, and transfer learning and fine-tuning of an ImageNet pre-trained CNN model on medical image datasets has not yet been exploited in previous researches and there is also no data augmentation method has been used on the classification of neuroblastoma histological images, for our project we are going to explore the effectiveness of implementing these methods on neuroblastoma images classification and further develop their implementation in diagnosis establishing. ResNet50 is used in the experiment as a pre-trained CNN model from high-level neural network library (Keras) with transfer learning and fine-tuning methods applied to the model. As in the case of transfer learning we do not have to train the model from scratch, the less training time and therefore faster output is obtained.

When we use this pre-trained model and retrain it for our dataset in CNNs, the middle layers usually learn the higher-level features and the final layers learn more specific features. Then the final layer will remove and add other classifier layer, which is how the the ResNet50 model is trained to classify our dataset.

3.3.3. Fine-tuning

ResNet50 is trained to classify the ImageNet dataset. Therefore, it already knows how to classify a specific set of images. We'll use it as a baseline to train our image classifier. Keras has a built-in function for ResNet50 pre-trained models. In this process, the team will define the shape of image as an input and then freeze the layers of the ResNet model. The customized will add into classification layer there, then average-pooling and softmax layers will add into the ResNet model respectively.

3.4. ResNet34

3.4.1. ResNet34 Description

There is similar network structure between Resnet 34 and Resnet 50 (Figure 11), According to figure 12, Instead of 2 layered (3x3) convolutions, the RestNest 34 is using use (3x3), and (3x3) convolutions which is different with Resnet 50 in (1x1),(3x3) and (1x1) convolutions.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 12. Resnet network with different layers.

Based on Resnet 34 structure, we trained our own model for classification of neuroblastoma into 5 classes. Several steps were taken for training process.

3.4.2. Number of Epochs

The epoch number should be decided first because it relatives to the computing cost that the model take, if the epoch number not enough, the model will have limit on learning the features from original data. In addition, if there is a large number of epochs, it may cause the waste of computing resource.

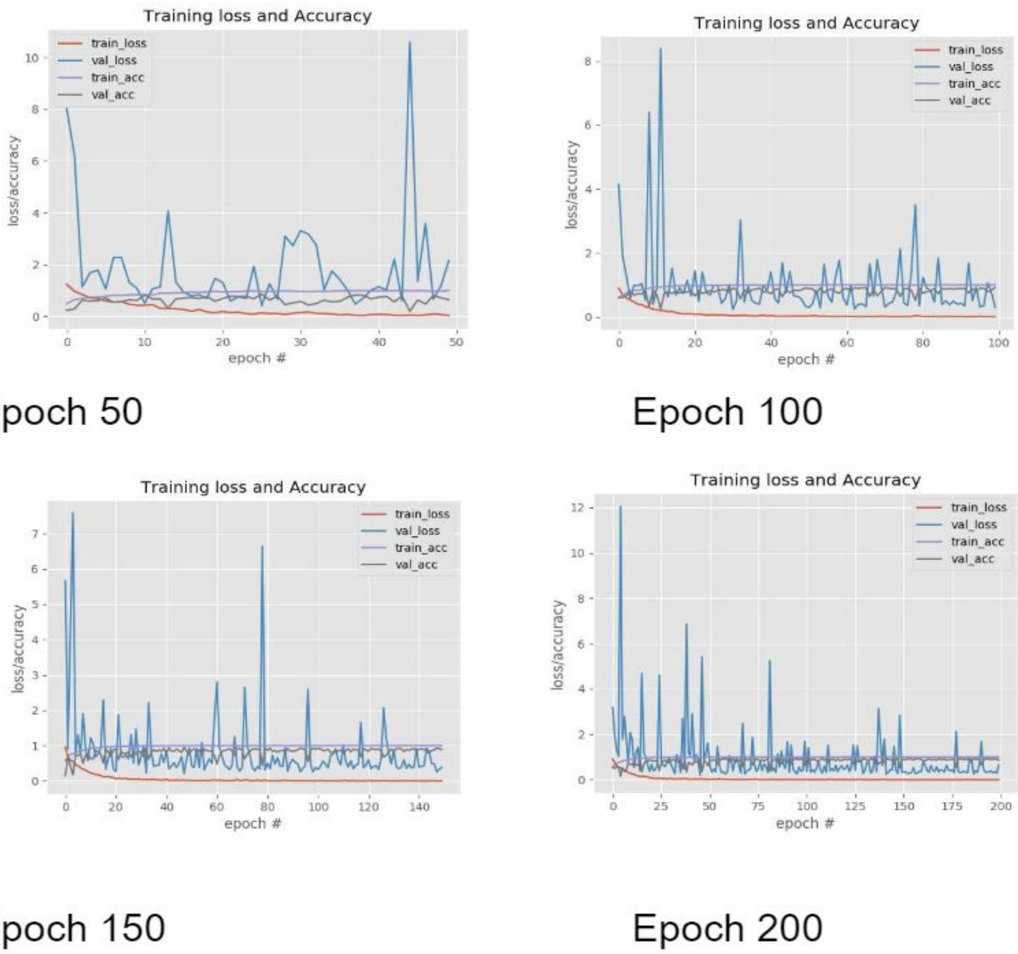


Figure 13. Loss Value in different epochs where learning rate=0.001 base on Resnet 34.

We have tried running the model within 50, 100, 150 and 200 epochs and plot the loss value of them. Loss value refers to the attributes which can be used to evaluate the performance of a model, it normally showing in a diagram for better understanding the performance of a model, a lower loss normally means there is a great performance for relatives model. Figure 13 shows the first steps which is to explore the required epoch number, after 50 epoch, both train and validation loss is getting smoothed and will not decrease. Therefore, 50 epoch is meeting the minimum requirements of epoch number.

3.4.3. Dropout

Dropout layer is widely used to increase the generalization ability of a model, base on dropout rate, a fix number of neuron in relatives layer will be removed randomly each time. By adding dropout layers, it is easily to gain a smaller model with same structure by compair with no dropout.

Different rate of drop rate which are 0, 0.25, 0.5 and 0.75 were selected in this experiment, both f-score and confusion matrix will be used for evaluating the changes that Dropout bring.

Dropout	**F1 score**
0	0.90
0.25	0.84
0.5	0.87
0.75	0.88

Table 1. F1 score under different Dropout rate where learning rate = 0.001.

Table 1 shows F1 score under different Dropout rate, where there is no dropout, the f-score is highest which is 0.9, and the lowest f-score is 0.84 where dropout is 0.75.

Confusion matrix illustrates the classification of model, vertical labels show the actually class that validation image belong, horizontal labels show the class that the image is classified. According to Table 2, where no drop, most class get correct classified, but for specific class which are GNB, there is only 5 of 9 images be classified correctly. Where dropout equals 0.25 which is lowest F1 score one, for specific class GNB have highest accuracy of all 4 different dropout which is 8 image be classified correctly in 9 images.

	DNB	GN	GNB_3_SP	PDNB	UDNB
DNB	28	0	1	8	0
GN	0	15	1	0	0
GNB_3_SP	0	4	5	0	0
PDNB	2	0	0	110	2
UDNB	0	0	0	2	29

DP=0

	DNB	GN	GNB_3_SP	PDNB	UDNB
DNB	33	0	0	4	0
GN	0	15	1	0	0
GNB_3_SP	0	1	8	0	0
PDNB	17	0	0	87	10
UDNB	0	0	0	1	30

DP=0.25

	DNB	GN	GNB_3_SP	PDNB	UDNB
DNB	21	1	0	15	0
GN	0	15	1	0	0
GNB_3_SP	0	3	6	0	0
PDNB	2	0	0	110	2
UDNB	0	0	0	3	28

DP=0.5

	DNB	GN	GNB_3_SP	PDNB	UDNB
DNB	28	0	2	7	0
GN	0	15	1	0	0
GNB_3_SP	0	3	6	0	0
PDNB	2	0	0	105	3
UDNB	0	0	0	2	29

DP=0.75

Table 2. Confusion matrix with various dropout rates.

3.4.4. Optimizer selection

The optimizer is another important factor which impacts the performance of the model, the time cost of training and convergence. Base on Keras documentation, Stochastic Gradient Descent (SGD), RMSprop, Adaptive Moment Estimation (Adam) and ADAGRAD are selected for achieving the higher performance.

OPTIMIZER	F1 score
SGD	0.90
RMS_GROUP	0.77
ADAGRAD	0.69
ADAM	0.77

Table 3. F1 score under different optimizer where learning rate=0.001.

Table 3 shows F1 score under different optimizer, where the time cost will not be considered for reaching higher performance, SGD represents the highest F1 score which is 0.87.

3.4.5. Regularization

To keep working with the generalization ability of the model, regularization is a method which is available to resolve the issues of imbalanced classification. There are two different functions which can be used for regularization, L1 Regularization is able to give model sparsity and more useful for a sensitive data set which means it is able to accept extreme difference in row data. L2 is base on normal distribution and good for working with outlier data.

	Predicted					
Actual		**DNB**	**GN**	**GNB**	**PDNB**	**UDNB**
	DNB	**32**	**0**	**1**	**4**	**0**
	GN	**0**	**15**	**1**	**0**	**0**
	GNB	**1**	**0**	**8**	**0**	**0**
	PDNB	**14**	**0**	**0**	**85**	**15**
	UDNB	**0**	**0**	**0**	**0**	**31**

Table 4. L1 Regularization = 0.0001.

	Predicted					
Actual		**DNB**	**GN**	**GNB**	**PDNB**	**UDNB**
	DNB	**36**	**0**	**1**	**0**	**0**
	GN	**0**	**13**	**3**	**0**	**0**
	GNB	**0**	**0**	**9**	**0**	**0**
	PDNB	**20**	**0**	**0**	**83**	**11**
	UDNB	**0**	**0**	**0**	**0**	**31**

Table 5. L2 Regularization = 0.0001.

According to both tables 4 and 5, GNB has been resolved by both function, PDNB have an increased error prediction and both have same F1 score which is 0.83. By comparing both, L1 regularization is better to classify PDNB, and most error of PDNB is from DNB and UDNB, it may caused by they have same features such as Neuropil and Neuroblast cell, L2 regularization is better to classify GNB and UDNB images.

Based on the case above, no dropout and no regularization provide the best f-score and both dropout and regularization increased the generalization ability of the model in different way.

3.4.6. Learning Rate

Learning rate is used for control the speed of updating the weights of the models. The function for calculating weights is:

$$\textit{New weight} = \textit{existing weight} - \textit{learning rate} \times \textit{gradient}$$

If there is a low Learning rate, the update speed of weights will be slow, which means it is hard to learn. If there is a high learning rate, the update speed of weight will be high and easy to jump the lowest loss which is the best weights for getting higher performance of model. In this experiment, Learning rate 0.01, 0.001, 0.005 and 0.0001 will be used for calculating the lowest loss value.

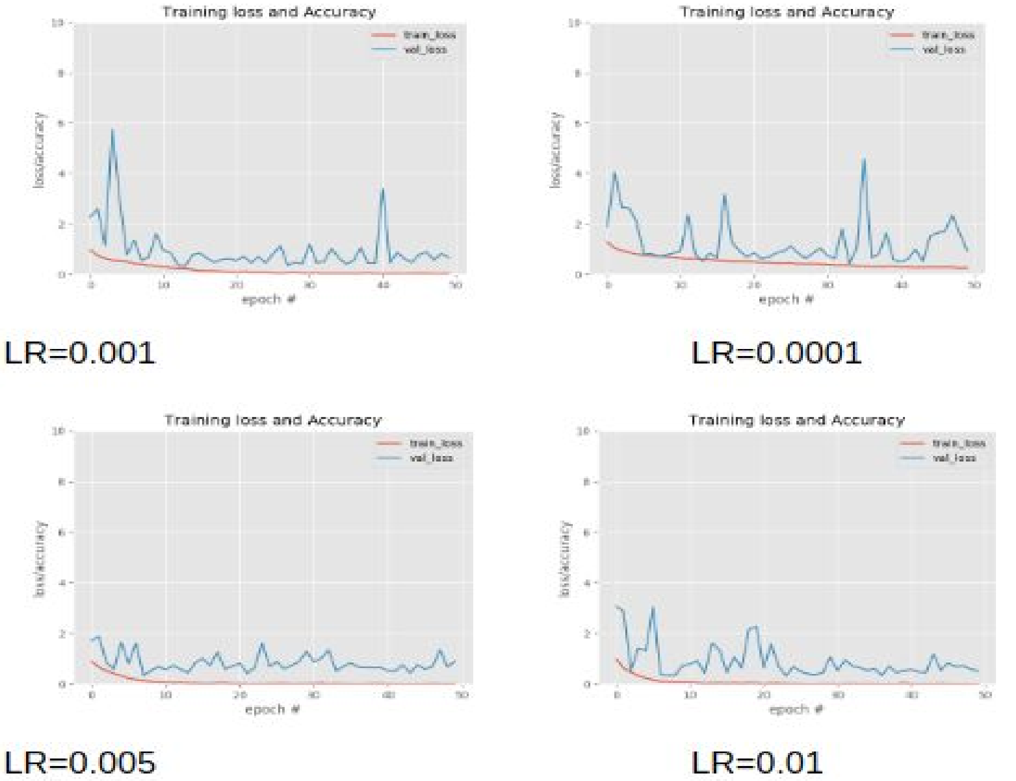


Figure 14. Loss value base on different Learning rate.

According to figure 14, where learning rate equals 0.01 have getting Smoothest curve and lowest loss value in four plot. Which means there is a reliable model which have best performance where learning rate is 0.01 in four different learning rate.

4. Achievements and Findings

4.1. Confusion matrices

The confusion matrix shows the summary of the results of images classification by the model and provides details about how each image can be classified, the number of images be classified into a correct class and others.

	Predicted					
Actual		DNB	GN	GNB	PDNB	UDNB
	DNB	33	0	0	4	0
	GN	0	15	1	0	0
	GNB	0	2	7	0	0
	PDNB	10	0	0	92	12
	UDNB	0	0	0	0	31

Table 6. Confusion matrix of Resnet 34.

Table 6 shows the confusion matrix of the final model evaluation, there are 33 of 37 DNB images have been classified correctly which is 89%, 15 of 16 GN images have been classified correctly which is 93%, 7 of 9 GNB images have been classified correctly which is 77%, 92 of 114 images have been classify correctly which is 80% and 31 of 31 UDNB images have been correctly which is 100%.

The limitation of confusion matrix is that it is hard to use for evaluation of the performance of models even if it is helpful for understanding the model's performance by each class. For example, as table 7 illustrates, both models have better performance for specific class, the first model has better performance on DNB and PDNB, but the second model is better for classifying the other three.

	DNB	GN	GNB_3_SP	PDNB	UDNB
DNB	28	0	1	8	0
GN	0	15	1	0	0
GNB_3_SP	0	4	5	0	0
PDNB	2	0	0	110	2
UDNB	0	0	0	2	29

DP=0

	DNB	GN	GNB_3_SP	PDNB	UDNB
DNB	33	0	0	4	0
GN	0	15	1	0	0
GNB_3_SP	0	1	8	0	0
PDNB	17	0	0	87	10
UDNB	0	0	0	1	30

DP=0.25

Table 7. Confusion matrix.

4.2. Classification Report

Classification report is design to demonstrate the summary of results based on statistic calculation of confusion matrix, table 8 shows the classification report based on no dropout equals and learning rate equal 0.01.

	precision	recall	f1-score	support
DNB	0.93	0.76	0.84	37
GN	0.79	0.94	0.86	16
GNB_3_SP	0.71	0.56	0.63	9
PDNB	0.92	0.96	0.94	114
UDNB	0.94	0.94	0.94	31
micro avg	0.90	0.90	0.90	207
macro avg	0.86	0.83	0.84	207
weighted avg	0.90	0.90	0.90	207

Table 8. Classification Report of training process(dropout=0.25, learning rate = 0.01).

4.3. Evaluation

In this experiment, our model are able to reach highest f-score which is 0.91 in our test set, but in each training there could be a different result, for evaluation of the models, we are planning to train the model more than 10 times by using different training set and test set randomly and gain an interval of the summarized F1 score.

Resnet 34 Data Augmentation = True dropout=0 Learning rate = 0.01 Epochs = 50	ID	F-Score
	1	0.86
	2	0.89
	3	0.87
	4	0.86
	5	0.91
	6	0.87
	7	0.87
	8	0.87
	9	0.88
	10	0.90
interval	#	0.88.5±2.5

Table 9. F1 score by training 10 times.

According to table 9, by training 10 times model with random sets, there is a interval F1 score in the interval of 0.86 to 0.91, the interval of F1 score in GNB is between 0.63 and 0.75.

Table 10 shows one the confusion matrix where no dropout and learning rate in 0.01, it is able to show there is also able to show there is still are difficult in classify the image under DNB and PDNB.

By evaluating the model, DNB and PDNB have similar features which may caused that it is difficult to classify even in the view of human, it also happens in this model. This problem can be resolved by spending more time on it such as regularization for ignoring outlier or increase the performance for specific class, but since it is a CNN model which may cost long time.

	Predicted					
Actual		**DNB**	**GN**	**GNB**	**PDNB**	**UDNB**
	DNB	**25**	**0**	**0**	**12**	**0**
	GN	**0**	**15**	**1**	**0**	**0**

	GNB	**0**	**3**	**6**	**0**	**0**
	PDNB	**4**	**0**	**0**	**109**	**1**
	UDNB	**0**	**0**	**0**	**0**	**31**

Table 10. Confusion matrix (dropout=0, Learning rate = 0.01, Epochs = 50).

5. Recommendations

Due to the achievements and findings that our team have done, the classification report showed the results as the confusion matrix. As the figure below represents the classification report of training process where the dropout equal 0.25 and the learning rate equal 0.01.

To achieve the accuracy statistics, our team will recommend the data worker to consider the single class between the single performance or generalization. Also, after evaluation it might lead to more than one results from the confusion matrix. Moreover, it is better to do the cross validation if there are more resource or dataset available for implementing in the future.

	precision	recall	f1-score	support
DNB	0.49	0.95	0.64	37
GN	0.60	0.38	0.46	16
GNB_3_SP	0.00	0.00	0.00	9
PDNB	0.96	0.77	0.85	114
UDNB	0.90	0.90	0.90	31
micro avg	0.76	0.76	0.76	207
macro avg	0.59	0.60	0.57	207
weighted avg	0.80	0.76	0.76	207

Figure 15. Confusion Matrix.

In order to work with the received dataset, the real work experience is very important. Most of the difficulties that our team has found at the beginning were resulted by insufficient knowledge and of the subject and lack of experience in the deep learning area by our team members. However, due to the whole team contribution, the project was fully completed and deployed in time. The performance of the Convolutional Neural Network ResNet34 model for image classification is acceptable, shows the high accuracy of image classification in the test dataset and feasibility for small or medium-sized organisations who can not gather or access large datasets, or for areas such as our clients which do not have enough data available.

It is extremely crucial for our clients to get the correct classification in order to diagnose the cancer type and further treatment, also avoid false diagnosing of cancer with treatment prescribed because it can harm the patient's health. The model that was designed by our team provides automatic diagnosis method based on image classification which can support medical workers to obtain a quick result to see whether the further diagnosis is required. Also it can be used for double check or verification of the results of manual diagnosis.

6. Difficulties encountered

The issues that our team encountered due to insufficient knowledge, expertise level and limited time, the next limitations to the model performance and experiment flow are listed below. They encouraged each member to gather more information, knowledge and skills on building CNN model and become more professional in the area.

6.1. High performance outcome require

Due to the fact that the model selection is based on successful running and high-performance outcome, our team experimented with the various models, modify and run them for many times to get better results. The team tried to improve the model based on the following criteria:

- Analyse the model error with the dataset
- Check the layer normalization whether it activate or not
- Monitor the percentage of dead nodes.
- Apply gradient clipping (in particular NLP) to control exploding gradients
- Balance the dataset by each class need to have the similar dataset

Moreover, we have to monitor the activation histogram before the activation functions closely. Then we took the best model after running by using the validation dataset. Also, when we do every change in the model's parameters entails significant changes in the other part such as splitting dataset, Learning rate, the epochs, optimizer. This change is very impacting the result for each class which is generating new class is required for each model.

6.2. Optimization

Optimization problems are difficulties encourage step that our team has found. It will contain uncertain parameters with the challenging process of the modeling methods. Our team needs to add a robust method for solving the optimization problems with uncertain parameters. Uncertainty data set will utilize the modeling of problems with robust optimization to determine the probable problem. To optimize the hyperparameter, our team has defined the parameter when begin the learning process by transferring the data size to acceptable for the model. We found that the default parameters and not performing Hyperparameter Optimization can significantly affect the model execution. In addition, having a couple of hyperparameters and hand tuning them as opposed to upgrading through demonstrated techniques is additionally a presentation driving aspect.

6.3. The application of transfer learning

Once the training data set has been started, some slight error occurred and also encountered for the whole data set. The models are too complicated because of many relevance parameters to the large observation. Normally the model capacity is evaluated by itself about the appear performance on the training data set. Our team needs to maximize the training model to generate the new performance of the model. To classify the cancer image, the Deep learning required a big dataset to on training processing to resolve the actual problem, it is very crucial for the model to assigned with sufficient implementing power. The team switched to high performing GPU and corresponding procedure to assure the model quality, hence, this processing requires much power and very time-consuming.

6.4. Dataset and time limitation

As the data collected from the patient in the Children's Hospital at Westmead, the dataset is very privacy sensitive and very limited which is only 1043 images of 5 labelled types. Then, once our team communicate for the project via the online chatting, it is very important to avoid uploading the dataset on iCloud platform. In order to apply the CNN, it is essential to have enough size of the dataset because if the data set is very small the CNN model will suffer from the overfitting, that will give very good performances with the training set but very low performance with the test set.

However, because of the professional of teamwork, we finally managed to reduce the overfitting problem for our dataset.

Due to the selected model require the high performance outcome of the dataset, we need to use more complex model for implement the dataset to receive the sufficient results for our classification. So, we did more on the transfer learning by using the ResNet50, which is one of model that having high artificial capability for the CNN classifiers. However, the selected model has very large memory requirements which is unacceptable for some laptop, also, when running the code it is very time consuming and is about 10 minute for each epoch.

7. Conclusions

This project represents a computer-aided method for classification of neuroblastoma images with Convolutional Neural Network CNN ResNet34 which was developed and trained specifically for our clients - Westmead Children Hospital in Sydney. The model improved the performance of neuroblastoma diagnosis through extracting and classifying the high-level characteristics from histological images of the neuroblastoma tumor. Firstly, we implemented ResNet50 Keras model for our experimenting. It produced high overfitting, that is why we decided to follow the method of building our own model from scratch with no weights and transfer learning applied.

As our dataset is small and imbalanced, the data augmentation method have been employed for ResNet34. The effectiveness of the augmentation has been proved by experiments results and significantly enhance the model's performance in dealing with small datasets.

Furthermore, multiple metrics such as confusion matrix, classification report, accuracy and loss functions, F1 score have been used for evaluating the performance of constructed CNN model. The optimal number of epochs, optimizer, learning rate and dropout rate have been determined, the integrated training and testing experiments have been conducted multiple times thus the entire process of classification has been performed.

As mentioned in the previous section, several limitations were encountered and need to be resolved and improved for future work and designing CNN models for medical images classification. The most important one is to collect more real data or images to make ResNet34 model as an industry-applicable system.

Table of individual contribution

Name	Student ID	Contribution Percentage
Tetiana Kraay	12605109	33.33%
Khannika Chanachai	12184673	33.33%
ZiHao Yang	13019738	33.33%
	Total	100%

Appendices

Original Dataset

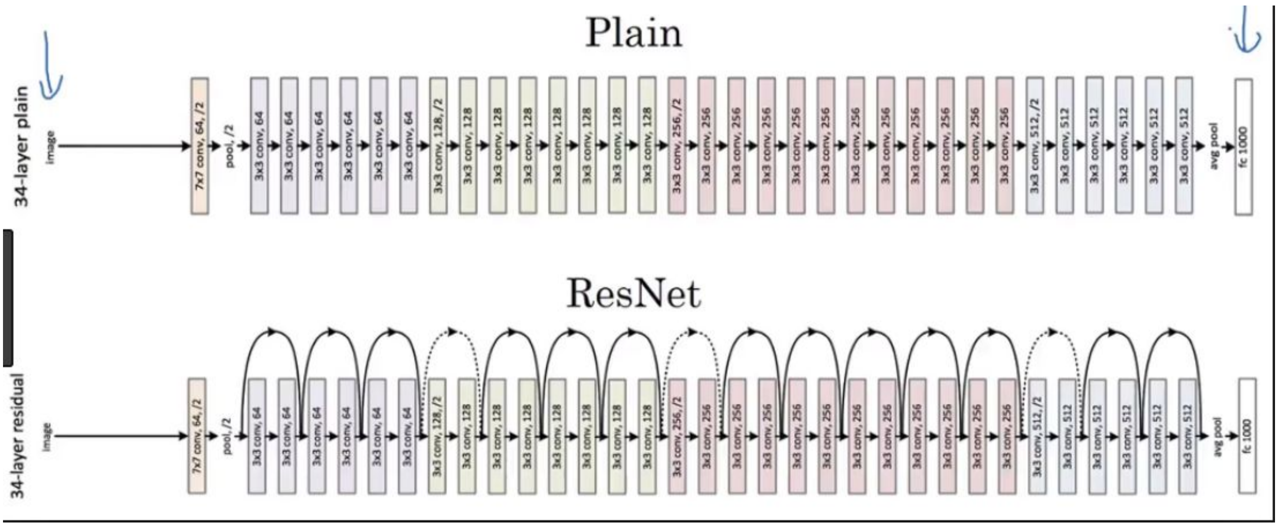
Category of Neuroblastoma	Number of images	Number of patients
Poorly-differentiated (PD)	571	77
Differentiating (D)	187	12
Undifferentiated (UD)	155	10
Ganglioneuroma (GN)	84	18
Ganglioneuroblastoma (GN)	46	8

Sources documentation

Keras Documentation <https://keras.io/>

Scikit-learn Documentation <https://scikit-learn.org/stable/documentation.html>.

Designed Model scheme



Resnet 34

Github link

(Analytics-Capstone) <https://github.com/tanya072/Analytics-Capstone>.

References

Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network, <http://arxiv.org/abs/1811.04241>.

Gheisari, S ., Catchpoole, D ., Charlton, A. & Kennedy, P. 2018, “Convolutional Deep Belief Network
with Feature Encoding for Classification of Neuroblastoma Histological Images”, Journal of Pathology Informatics , vol. 9, pp. 1-28.

He, K., Zhang, X., Ren, S., & Sun, J. 2016, Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778.

Shimada, H. 2003, ‘The International Neuroblastoma Pathology Classification’, Pathologica vol. 95, no. 5, pp. 240–1.

Zhang, W., Li, R., Deng, H., Wang, L., Lin, W., Ji, S. & Shen, D. 2015, “Deep convolutional neural networks for multi-modality isointense infant brain image segmentation”, *NeuroImage*, vol. 108, pp. 214–24.