Analysing brain tumour scans with CNN

Andong, Chuan Zheng, Rover, Woh Jon

National University of Singapore (NUS)

Introduction

This report evaluates the effectiveness of different Convolutional Neural Network (CNN) architectures in predicting the malignancy of brain tumours using brain scans. Precise CNN models with minimal false negatives are crucial for promptly identifying potentially harmful brain tumours, aiding medical professionals in early intervention and treatment planning.

There has been ongoing research in the field of brain tumour classification that attempts to employ CNN methods. Deshpande et al. (2021) employed resolution enhancement techniques (Discrete Cosine Transform-based image fusion) along with CNN, in order to facilitate more accurate classification between scans of brain tissue with and without tumour. The paper displays high accuracy of this method, positioning it as a promising method for brain tumour classification. However, there is a lack of comparison with other CNN architectures.

Narayanasamy et al. (2023) utilized RMSprop, an optimization algorithm designed to tackle slow convergence in Stochastic Gradient Descent (SGD). By adaptively scaling the learning rate for each parameter, RMSprop was combined with Convolutional Neural Networks (CNNs) to enhance model performance. This led to a very high accuracy level of 99.76%.

We plan to provide an extension to the available research on brain tumour classification by:

- Comparing different CNN architectures
- Incorporating additional features such as symmetry into our model.
- Testing out the effect of having different trainable layers on model performance.

We outline our methods below, with Appendix Figure 5 presenting a flowchart schematic of our process.

Dataset

The dataset comprises 231 RGB images categorized into benign and malignant, with 154 malignant and 77 benign images. Among the benign images, approximately 66 show no tumor, while 11 exhibit a benign tumor. Due to the dataset's limited size, especially for benign images showing tumors, we employ data augmentation techniques to expand the training dataset. We also explore symmetry metrics for feature extraction, integrating them into the CNN architecture to enhance model performance. Our data handling process is depicted in Figure 5.

Train - Validation - Test split

Using OpenCV, we resized the images to 128x128 pixels, categorized them as benign or malignant based on labels, and split them into training, validation, and test sets in a 63:27:10 ratio. The validation data assesses the model's fit to the training data, aiding in hyperparameter tuning, while the testing data provides an impartial evaluation of the final model, as it remains unseen by the user during hyperparameter tuning (Shah, 2020).

Data augmentation

We utilised data augmentation techniques to generate 8 augmented images per sample. This enriches the training dataset and increase the volume of training data for our Convolutional Neural Network (CNN) models. These techniques includes:

- Intensity level inversion
- Flip operations
- Random contrast adjustments
- Random zooms.
- Random rotation
- Random brightness level adjustments

Inversion involves subtracting each pixel value from 255 based on a specified probability p, introducing additional variability to the dataset.

Data augmentation increases the generality of the data as well. This allows the model to learn more invariant patterns in the input data, reducing the likelihood of overfitting. Some examples of augmented images are shown in Figure 7 (Appendix).

Standardisation of pixel values and feature data

Before feeding the data into the model, we processed the image and feature data. Regarding the pixel values of the images, we had several options to consider:

- Normalisation

$$X_{normalised} = \frac{X - min(X)}{max(X) - min(X)}$$

- Per-image standardisation

$$X_{standardised} = \frac{X - mean(X)}{sd(X)}$$

- Standardisation relative to dataset (dataset standardisation)

$$X_{standardised} = \frac{X - mean(D)}{sd(D)}$$

Here, X represents the array of pixel values of an image, D denotes the set of all image arrays and sd computes the population standard deviation.

Standardisation vs normalisation Standardization was preferred over normalization because standardization handles outliers better and prevents a few inputs from disproportionately affecting the training process. Moreover, standardization improves the convergence, accuracy, and efficiency of the model (Géron, 2019, p. 333-335).

Per-image vs dataset standardisation Per-image standardization was chosen over dataset standardisation to enable the model to capture important visual cues relative to the rest of the pixel values in each image. Dataset standardization, on the other hand, might de-emphasize critical features such as the tumor boundary, particularly for images that are darker overall.

The results of our comparison across the 3 methods are shown below:

Method	Accuracy	Recall	F1-score
Normalisation	0.714	0.881	0.789
Per-image standardisation	0.857	0.929	0.891
Dataset standardisation	0.872	0.905	0.872

Table 1: Per-image standardisation performs the best in F1-score and recall

Feature data Regarding the feature data, we performed dataset standardisation for our feature variables (consisting of only the symmetry metric). This prevents numerical overflow or underflow, improving the performance of the model.

Methods

Symmetry metric

Symmetry evaluation is crucial in medical image analysis as asymmetry can signal pathology (Choi et al., 2023), particularly in symmetrical organs such as the brain. Malignant tumor brain scans typically exhibit lower levels of symmetry compared to those of benign tumors.

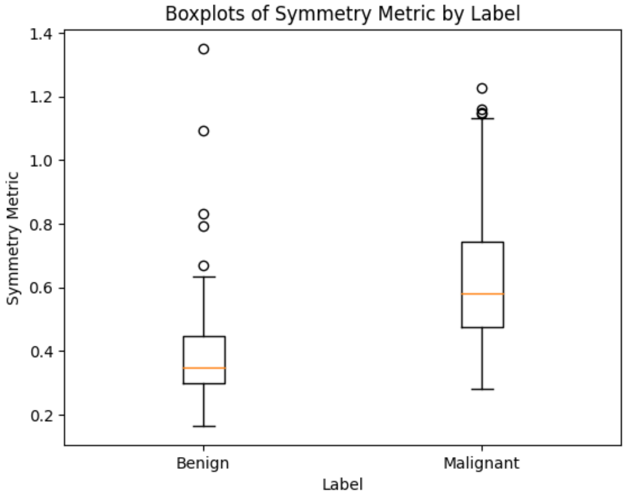


Figure 1: Symmetry metric of benign vs malignant tumour

Computing the metric The symmetry metric is found through these two steps:

1. Finding the line of symmetry
2. Evaluating the metric

1. Jing Yiran's (2020) code was used to identify the line of symmetry of the image. His code uses SIFT (a feature detecting algorithm to find distinctive points) to determine the most appropriate line of symmetry.
2. After reflecting the original image about the line of symmetry. Our metric is derived from the MAE (mean absolute error) :

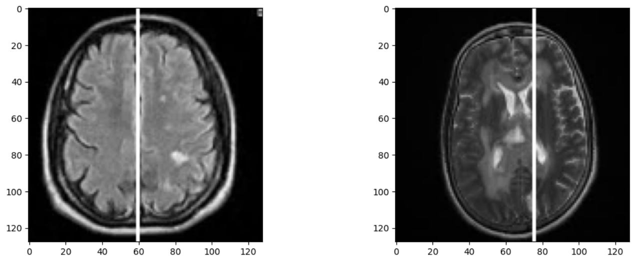
$$\frac{1}{n}\sum_{i=1}^{n} |std(x_i) - std(y_i)|$$

where x_i and y_i are the pixel values for the original and reflected image respectively and std computes the per-image standardised pixel values. The standardisation prevents brighter or darker images from reporting a larger or smaller symmetry metric.

Results From the given dataset, it is found that brain scans of malignant tumours tend to have a higher symmetry metric due to their higher levels of asymmetry (Fig. 1).

The symmetry metric will be subsequently included in the neural network through concatenation with other inputs. This strategic integration aims to extend the model's capability beyond discerning patterns solely from pixel values and image features, to encompass broader contextual cues such as symmetry.

Outliers The current method, as implemented by Jing Yiran, may sometimes inaccurately determine the line of symmetry, leading to a lack of precision in assessing brain shape symmetry (Fig. 2b). Consequently, the computed Mean Absolute Error (MAE) fails to provide a comprehensive reflection of the overall symmetry of brain shapes. This inaccuracy contributes to the presence of outliers and an increase



(a) Successful (b) Unsuccessful

Figure 2: Identification of the line of symmetry

in the variance of distributions for both malignant and benign labels (Fig. 1).

CNN architectures

> "You either do it yourself, or ask someone else to do it for you"
>
> - Prof Chin Chee Whye

In our classification task, we applied transfer learning with convolutional layers from pre-trained VGG16, InceptionResNetV2, and ResNet50 models trained on ImageNet.

We chose to test these 3 models as they performed better than most models on the ImageNet dataset. VGG16 neural network architecture by Simonyan and Zisserman (2014), comprises 13 convolutional layers and 5 max-pooling layers (Tamina, 2019). Unlike VGG16, InceptionResNetV2 is made up of residual connections, which involves adding shortcut connections that skip one or more layers in a neural network, and inception blocks, which apply multiple filter sizes to the previous layer and concatenating their output (Ferreira et al., 2018). Residual connections alleviates the vanishing gradient problem while inception blocks help the model capture features at different spatial scales. ResNet50, comprising 50 layers of residual networks (Mukti and Biswas, 2019), is characterized by its absence of inception blocks, setting it apart from InceptionResNetV2. ResNet50 also has a shallower structure compared to InceptionResNetv2, potentially reducing the likelihood of overfitting.

Transfer learning Transfer learning allows us to leverge on pre-trained data to improve the results of our classification model. Features learned by the model after training on ImageNet can be used to decipher patterns belonging to benign and malignant tumours respectively.

These steps were followed to apply transfer learning:

1. Load a pre-trained model in `keras` while excluding its final dense layers used for classification.
2. Freeze all layers in the base model by setting `trainable = False`.
3. Add custom layers on top of the model and concatenate the feature input (symmetry metric)
4. Train the model on the dataset

For the case of ResNet50, these were the custom layers added:

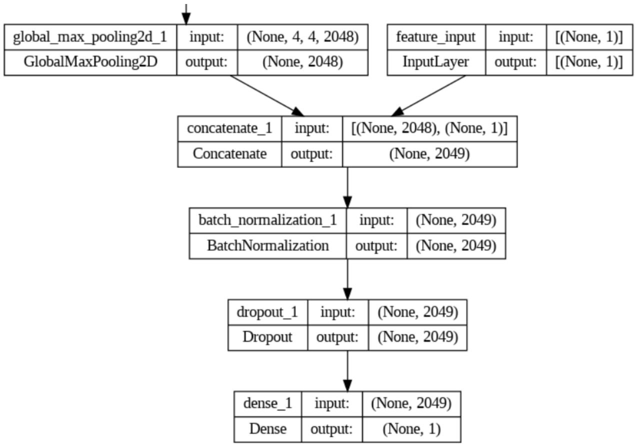

global_max_pooling2d_1 | input: | (None, 4, 4, 2048)
GlobalMaxPooling2D | output: | (None, 2048)
feature_input | input: | [(None, 1)]
InputLayer | output: | [(None, 1)]
concatenate_1 | input: | [(None, 2048), (None, 1)]
Concatenate | output: | (None, 2049)
batch_normalization_1 | input: | (None, 2049)
BatchNormalization | output: | (None, 2049)
dropout_1 | input: | (None, 2049)
Dropout | output: | (None, 2049)
dense_1 | input: | (None, 2049)
Dense | output: | (None, 1)

Figure 3: Final layers added to the base ResNet50 architecture

Batch normalisation Batch normalisation addresses vanishing and exploding gradient problems, increasing the convergence rate of the model (Géron, 2019, p. 333-335).

Dropout Dropout prevents overfitting and improves the performance of the model. According to Géron, "neurons trained with dropout cannot co-adapt with their neighboring neurons; they have to be as useful as possible on their own" resulting in a "robust network that generalizes better" (2019, p. 358).

Fine tuning Fine tuning was done to provide incremental improvements of the model by unfreezing certain convolutional layers and training the model again, albeit at a slower learning rate to prevent overfitting. We deliberately trained the final layers of the model rather than the initial ones. This preserves the basic features learned during pretraining, such as the recognition of shapes or lines, while allowing for slight adjustments to the higher-level features learned at the end of the base model. Therefore, the model retains its ability to recognize fundamental patterns common to various images, while also adapting to the specific characteristics of brain MRI scans during fine-tuning. For ResNet50, layers including and succeeding the reference layer: `conv4_block1_1_conv` were now set to trainable and the model was now trained at a slower learning rate of `1e-5`. These are the reference layers we used for the other models, with approximately 20% of the base model being set to trainable:

Architecture	Reference layer
VGG16	`block4_conv1`
InceptionResNet V2	`block8_10_mixed`
ResNet50	`conv4_block1_1_conv`

Table 2

Model	Metric	Before Fine Tuning	After Fine Tuning
VGG16	Accuracy	0.827	0.850
	Recall	0.882	0.928
	F1 Score	0.854	0.887
InceptionResNetV2	Accuracy	0.842	0.848
	Recall	0.899	0.857
	F1 Score	0.869	0.852
ResNet50	Accuracy	0.865	0.882
	Recall	0.928	0.901
	F1 Score	0.895	0.891

Table 3: Model Performance Before and After Fine Tuning, averaged across 4 seed values

Results and discussion

Of the performance metrics that we can use, accuracy and recall and F1 scores are the most important.

Recall A model performs better if it maximises recall, minimising the false negative (miss) rate. This is because the consequence of missing out on a malignant brain tumour can be extremely serious.

Accuracy On the other hand, we would not want a model that simply classifies almost everything as positive to reduce the false negative rate. If this were the case, more resources have to be dedicated to further diagnose a larger group of false positives. Thus, the accuracy metric ensures that medical diagnosis is efficient and effective.

F1 score The F1 score is chosen to be the main metric alongside accuracy and recall as it is a compromise between accuracy and recall. It is defined to be the harmonic mean of accuracy and recall:

$$f1 = \frac{2}{\frac{1}{accuracy} + \frac{1}{recall}}$$

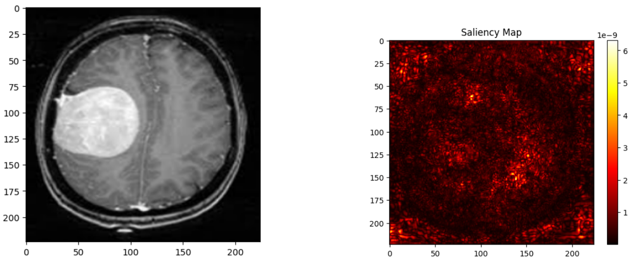
Model performance and number of trainable layers

For VGG16, the model performance (as determined by the 3 metrics) seems to be improved after fine tuning. However, fine tuning seems to have a negligible and even negative effect on the model performance for InceptionResNetV2 and ResNet50.

While fine-tuning enhances model accuracy for specific datasets, it risks overfitting. VGG16's simpler architecture benefits from additional training with lower overfitting risk, whereas the relatively higher complexity of InceptionResNetV2 and ResNet50 may exacerbate overfitting during fine-tuning, especially when dealing with a small dataset. The risk of overfitting in fine tuning ResNet50 is further examplified in Fig. 6b and Fig. 6d, where the decrease in loss is minute and the accuracy increases after only 2 epochs.

VGG16 vs InceptionResNetV2 vs ResNet50

Comparing both Recall and F1 scores (Table 2), ResNet50 without fine tuning seems to be the best performing, which agrees with available literature comparing the performance between VGG16 and ResNet50 (Mascarenhas & Agarwal, 2021).



(a) Sample image

(b) Saliency map

Figure 4: Saliency map generated by passing the image through the pre-trained VGG16 model

We hypothesize that InceptionResNetV2 underperforms in terms of F1 score due to its complexity, leading to overfitting. ResNet50, while less complex than InceptionResNetV2, benefits from residual connections compared to VGG16, ensuring better preservation of important features across layers.

Symmetry metric vs no symmetry metric

	Accuracy	Recall	F1-score
Without	0.839	0.117	0.861
With	0.885	0.084	0.900

Table 4: Metrics obtained, with and without applying symmetry metric for ResNet50

By integrating the symmetry metric alongside other inputs in the neural network, we expand the model's capacity to leverage contextual cues beyond pixel values and image features, leading to improved model performance across all metrics (Table 4).

Saliency maps

We utilized saliency maps, which highlighted crucial pixels for accurate image classification in a heat map format (Figure 4). They offer insight into the model's decision-making process, emphasizing influential pixels.

Observations:

- Reduced emphasis on the skull and tumour boundaries
- Increased emphasis within tumour, white matter region and the corners

Regarding the tumour region, it can be seen that the VGG16 model was be able to pick up important features such as the size and shape of the tumour.

References

Chhabra, M., & Kumar, R. (2022). An advanced VGG16 architecture-based deep learning model to detect pneumonia from medical images. *Emergent Converging Technologies and Biomedical Systems*, 457–471. https://doi.org/10.1007/978-981-16-8774-7_37

Choi, D., Sunwoo, L., You, S.-H., Lee, K. J., & Ryoo, I. (2023). Application of symmetry evaluation to deep learning algorithm in detection of mastoiditis on mastoid radiographs. *Scientific Reports*, *13*(1). https://doi.org/10.1038/s41598-023-32147-w

Ferreira, C. A., Melo, T., Sousa, P., Meyer, M. I., Shakibapour, E., Costa, P., & Campilho, A. (2018). Classification of breast cancer histology images through transfer learning using a pre-trained inception resnet V2. *Lecture Notes in Computer Science*, 763–770. https://doi.org/10.1007/978-3-319-93000-8_86

Jing, Y. (2020, January). *Detecting Mirror Symmetry*. GitHub. https://github.com/YiranJing/MirrorSymmetry

Deshpande, A., Estrela, V. V., & Patavardhan, P. (2021). The DCT-CNN-resnet50 architecture to classify brain tumors with super-resolution, convolutional neural network, and the resnet50. *Neuroscience Informatics*, *1*(4), 100013. https://doi.org/10.1016/j.neuri.2021.100013

Mascarenhas, S., & Agarwal, M. (2021). A comparison between VGG16, VGG19 and Resnet50 architecture frameworks for Image Classification. *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*. https://doi.org/10.1109/centcon52345.2021.9687944

Mukti, I. Z., & Biswas, D. (2019). Transfer learning based plant diseases detection using RESNET50. *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*. https://doi.org/10.1109/eict48899.2019.9068805

Tammina, S. (2019). Transfer learning using VGG-16 with deep convolutional neural network for classifying images. *International Journal of Scientific and Research Publications (IJSRP)*, *9*(10). https://doi.org/10.29322/ijsrp.9.10.2019.p9420

Shah, T. (2020, July 10). *About train, validation and test sets in machine learning*. Medium. https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

Narayanasamy, K., Elangovan, L, S. K., Maragatharajan, M., & Deepa, D. (2023). CNN-based deep learning approach for MRI-based brain tumor detection. *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*. https://doi.org/10.1109/icosec58147.2023.10276086

Géron, A. (2019). *Hands-on machine learning with scikit-learn, keras, and tensorflow*.

Appendix

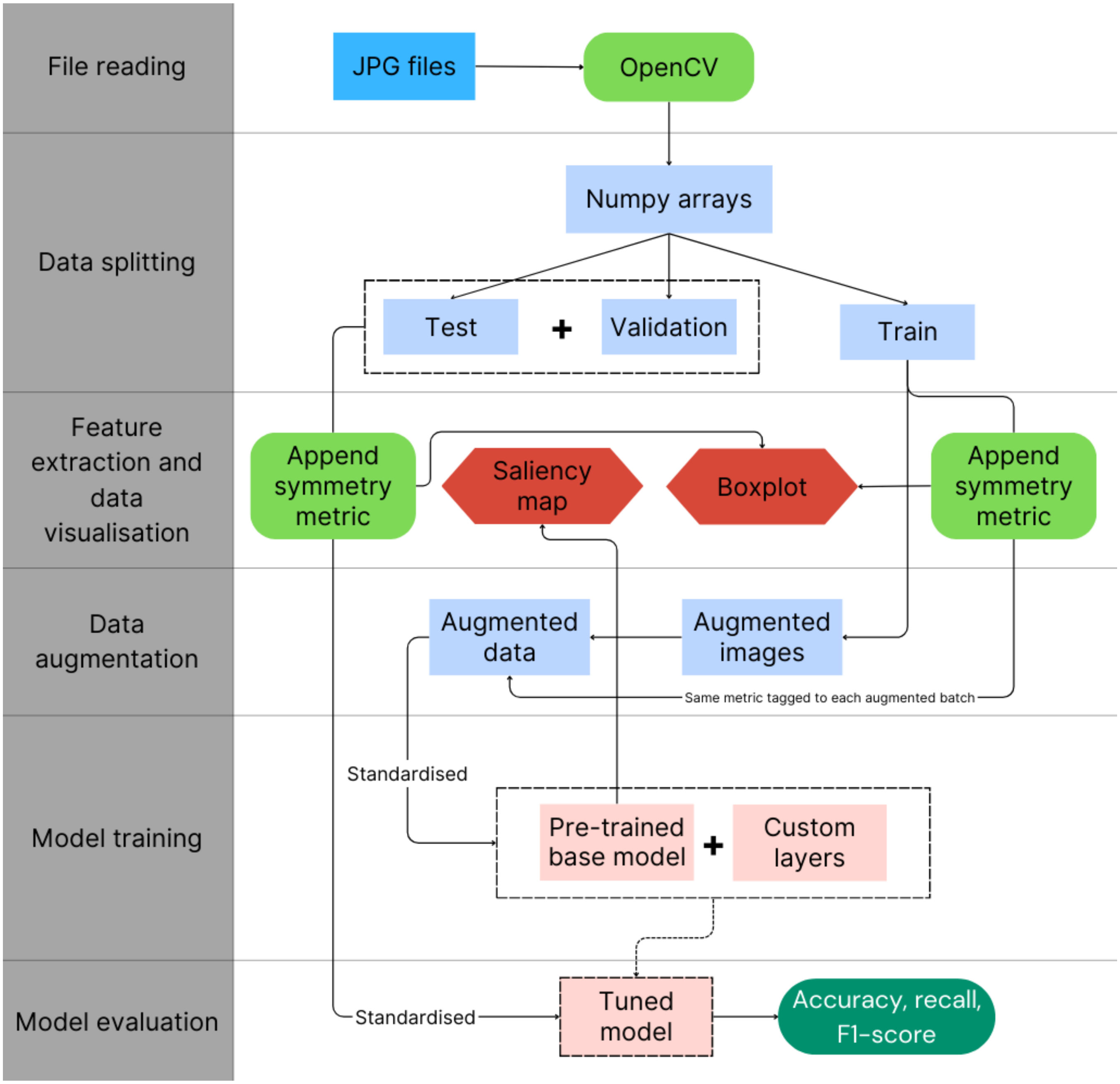
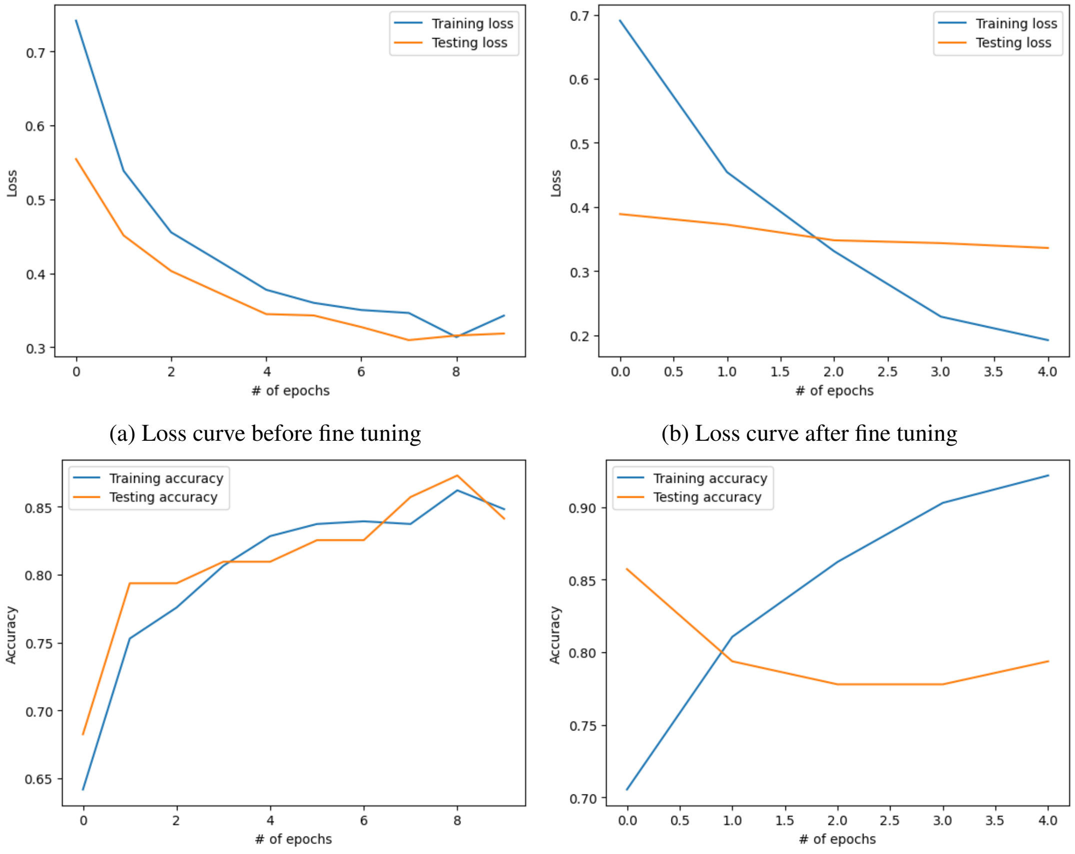


Figure 5: Data handling



(a) Loss curve before fine tuning

(b) Loss curve after fine tuning

(c) Accuracy curve before fine tuning

(d) Accuracy curve after fine tuning

Figure 6: Loss and accuracy curves for ResNet50 before and after fine tuning

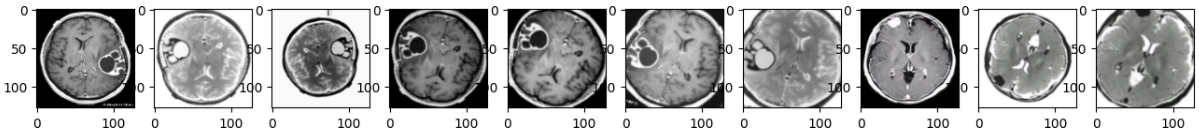


Figure 7: Augmented brain tumour images