Appendix A: Data Preparation

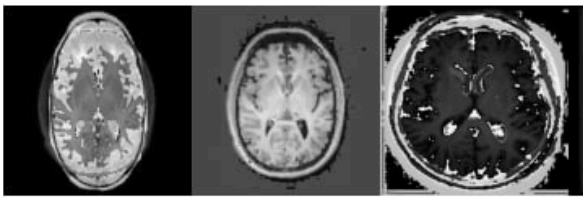
This appendix details our methodology with respect to image augmentation and upsampling methods

For deep learning algorithms like Convolutional Neural Networks (CNN), a large amount of data is commonly required to train the networks and achieve excellent performances without overfitting. In our work, image augmentation was used to deal with limited data and class imbalance. We apply transformations to our images with the given parameters in **Supplementary Table 1**.

Supplementary Table 1. Parameters for image augmentation

Parameter	**Value**
Rotation angle	Random integer (-35,35)
Shear angle	Random float (0, 0.1)
Shift (in both x and y direction)	Random float (0.8, 1.3)
Zoom (in both x and y direction)	Random float (0.8, 1.3)
Horizontal Flip	Always (p = 1)
Vertical Flip	p = 0.5

Also, we found that our dataset has mild class imbalance (62 benign and 129 malignant). To tackle class imbalance, we explored Synthetic Minority Over-sampling Technique (SMOTE) to upsample minority (benign) classes. We observed that SMOTE may produce synthetic images with features across different modalities (**Supplementary Figure 1**). However, the overlapping features in these synthetic images are not representative of typical MRI brain scans.



Supplementary Figure 1*. Synthetic images generated by SMOTE algorithm using the benign dataset.*

Appendix B: CNN Models

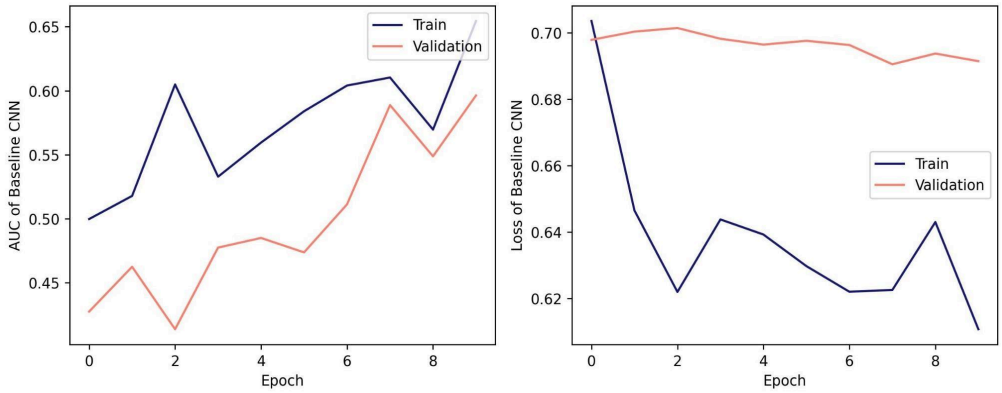
This appendix details our fine-tuning experiments for our CNN models and respective findings.

To fine-tune the hyperparameters for our baseline CNN model, initially we explored GridSearchCV. However, unlike the model, we were unable to feed a generator object into GridSearchCV for model fitting. Hence, we manually iterated through the values described in **Supplementary Table 2** with the best hyperparameters highlighted.

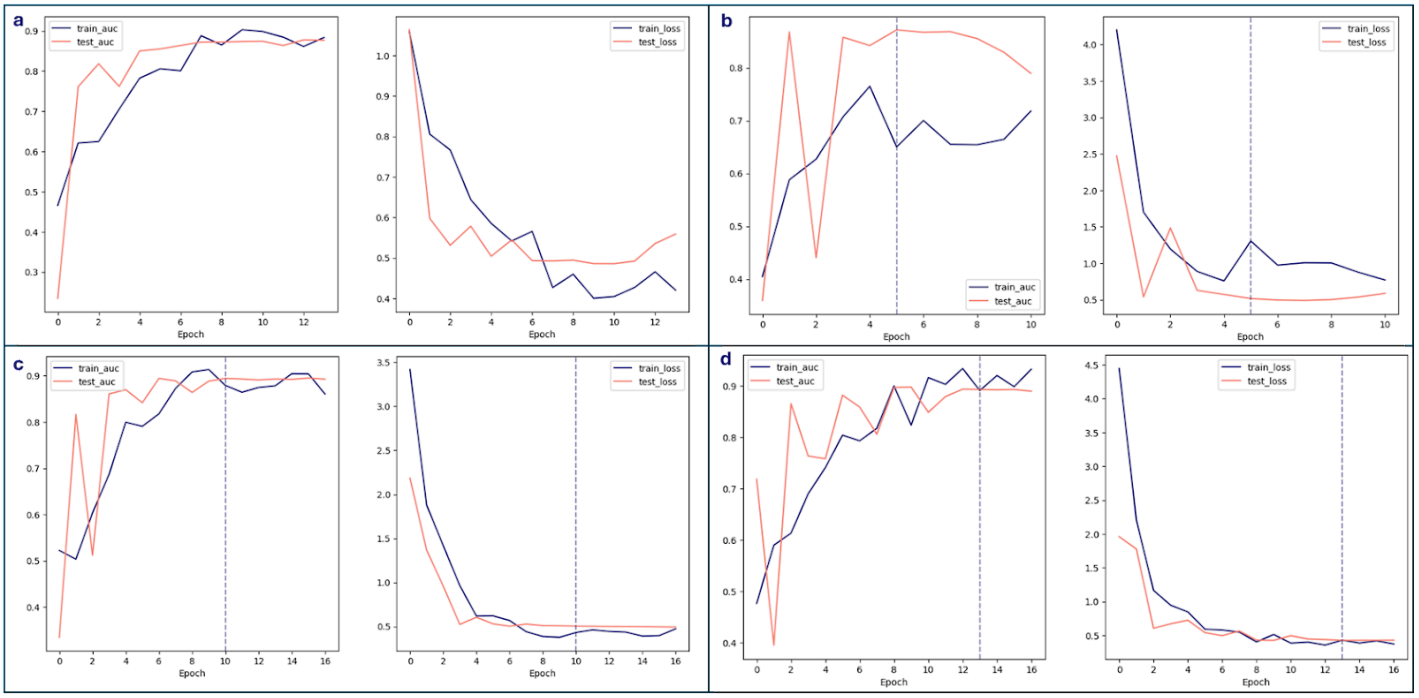
Supplementary Table 2: Manual fine-tuning of baseline CNN model

Hyperparameter	**Values**
Number of Conv2D Layers	2, **3**, 4
Learning Rate	0.0001, 0.001, **0.01**
Optimizer	Adam, **SGD**
Batch size	32, **64**, 128
Dropout Rate	0.2, 0.3, **0.4**

Supplementary Figures 2 and 3 shows the AUC score and loss over epochs during training and validation procedures for baseline CNN model and the various VGG16-ANN models respectively. Generally, the AUC scores are increasing while loss decreases across epochs. This suggests that the model is learning and predicting the class membership more accurately.

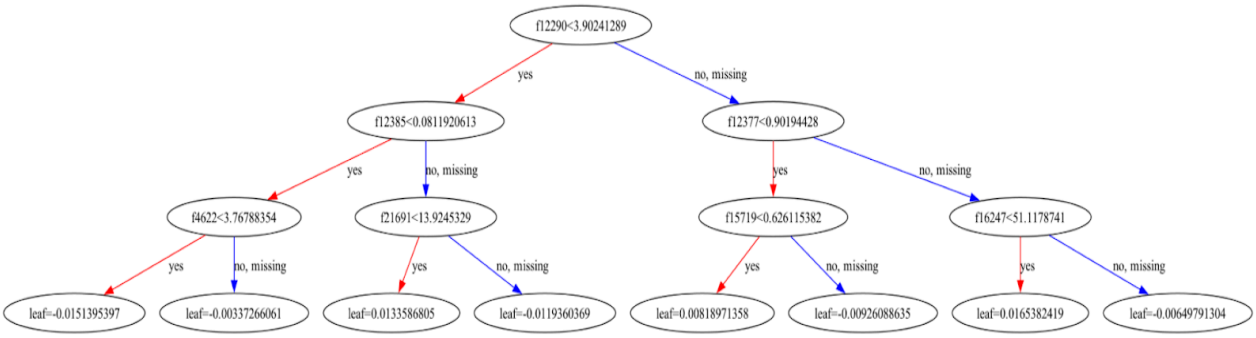


Supplementary Figure 2. *ROC AUC and loss graphs for baseline CNN model.*



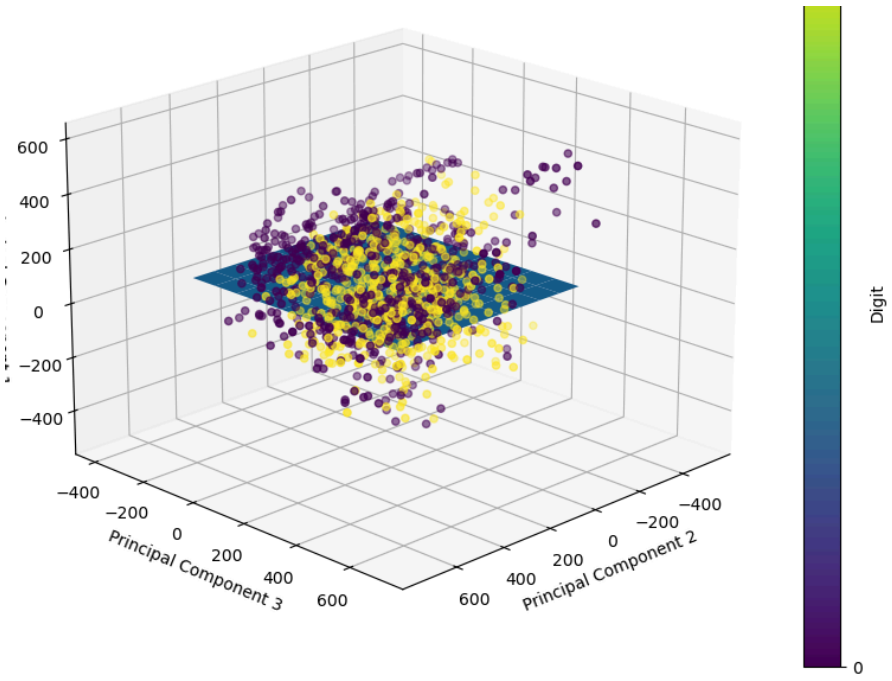
***Supplementary Figure 3**: ROC AUC and loss curve over epochs for VGG16-ANN models - (a) Baseline VGG16-ANN, (b) VGG16TF-11, (c) VGG16TF-15, (d) VGG16TF-18. Blue dotted line separates the initial training and fine-tuning phases.*

Supplementary Figure 4 illustrates the first tree of the base variant of VGG16-XGBoost model.

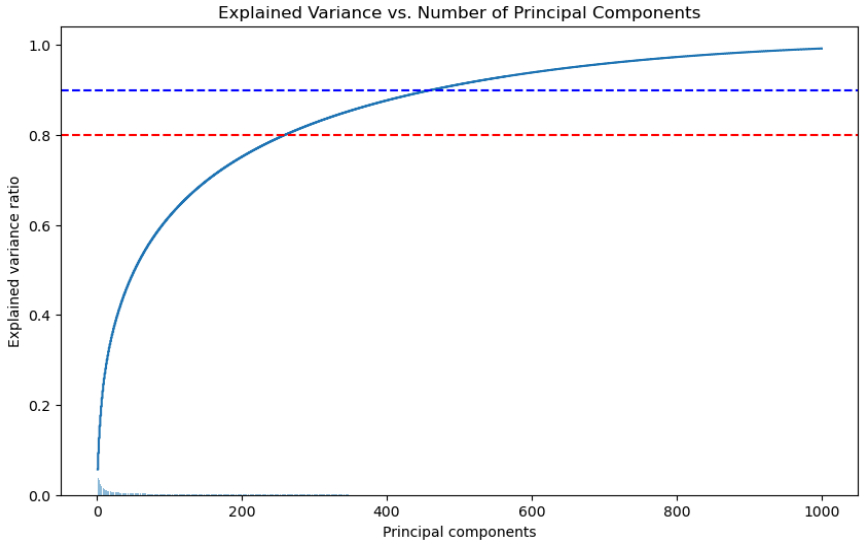


***Supplementary Figure 4**. First tree of the base model of VGG16-XGBoost.*

VGG16 pre-trained model was able to extract 25088 features to be passed into XGBoost Classifier. Principal Component Analysis (PCA) shown in **Supplementary Figure 5** indicates that there is a clear division of data points, hence PCA can be used to reduce the number of features. **Supplementary Figure 6** shows that 250 and 450 principal components are able to explain for 80% and 90% of the variance in data respectively. Thus, both values were used in the proposed model and evaluated using performance metrics.



Supplementary Figure 5. *Clear dissection of distinction made by the first three principal component.*



Supplementary Figure 6. *Number of principal components against explained variance. Red and blue lines show number of principal components explaining 80% and 90% of variance respectively.*