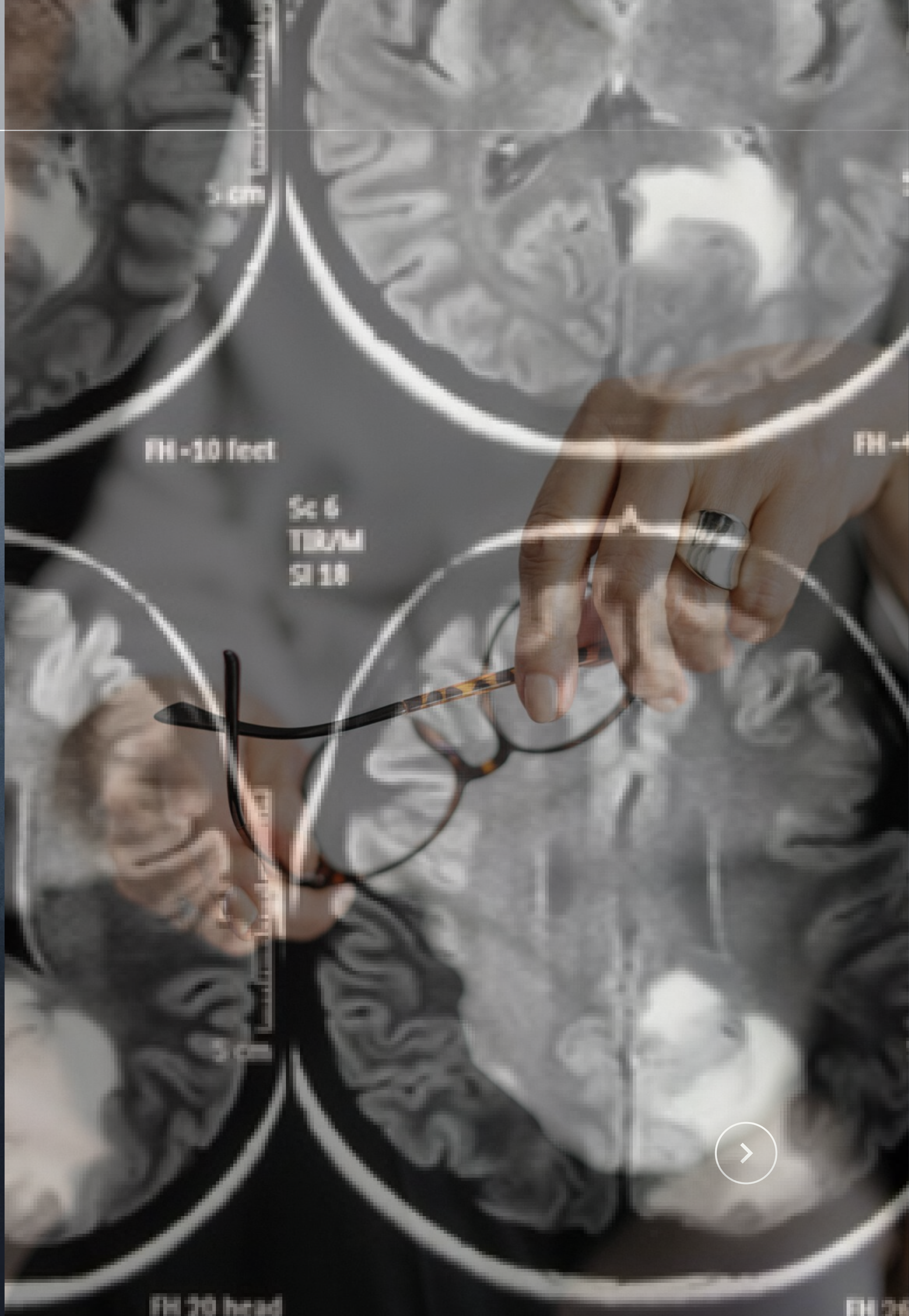
AML MINI PROJECT

Prof. Jayshree Aher

PC44 Palak Mallawat
PC46 Rashi Madnani
PC47 Dhanashree Lodhe



01

Problem Statement

02

Objectives

03

Research contribution

04

Tools used

05

Dataset Description

06

Data Pre-processing

07

System architecture

08

Model architecture

09

Algorithm/Methods /Techniques

10

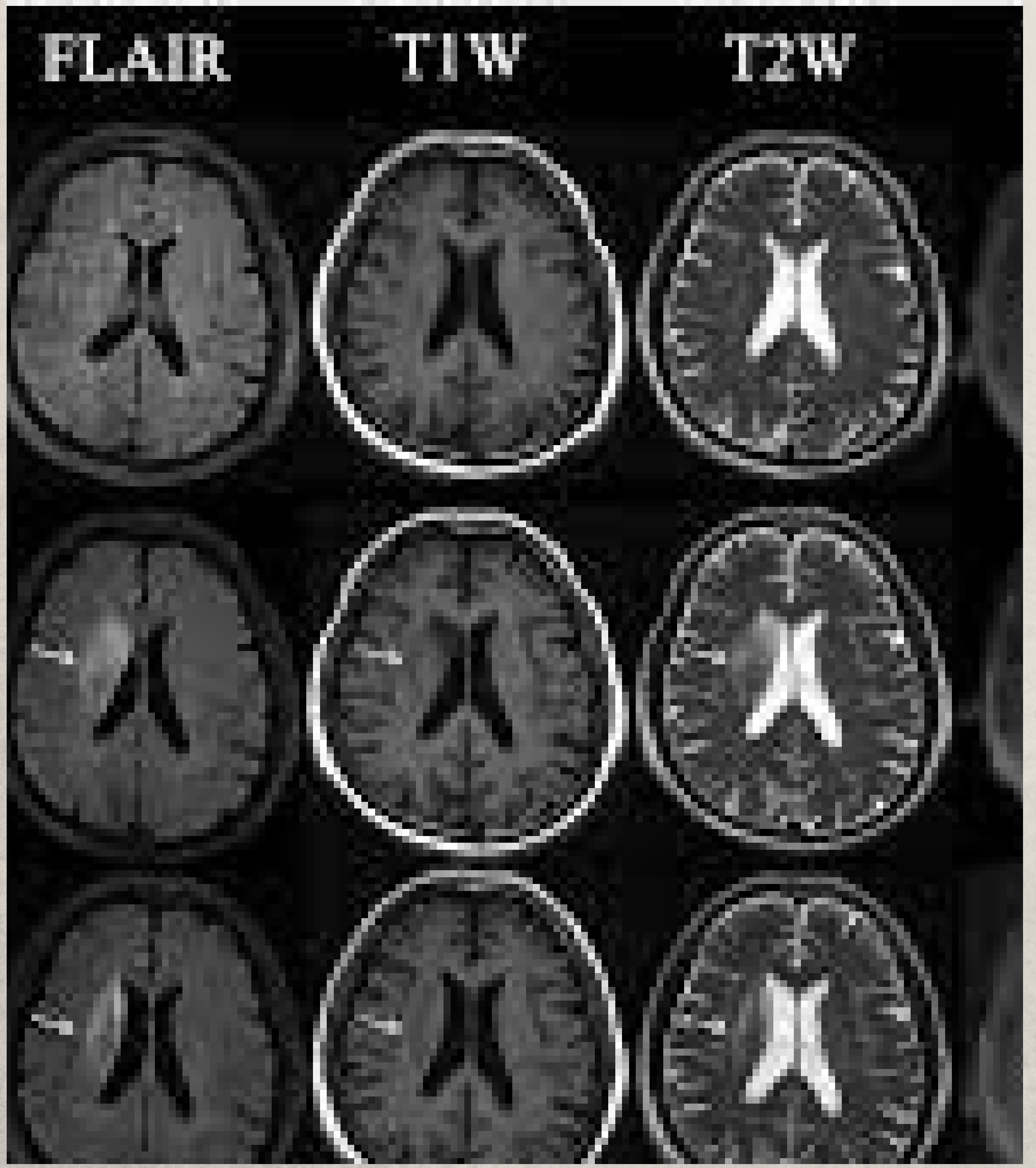
Output and Visualization

11

Conclusion

12

References



Problem Statement

To perform image classification on MRI scans of brain using transfer learning method: VGG19 to classify them into tumor or non-tumor.

OBJECTIVES

Brain tumor detection and segmentation is one of the most challenging and time consuming task in medical image processing.

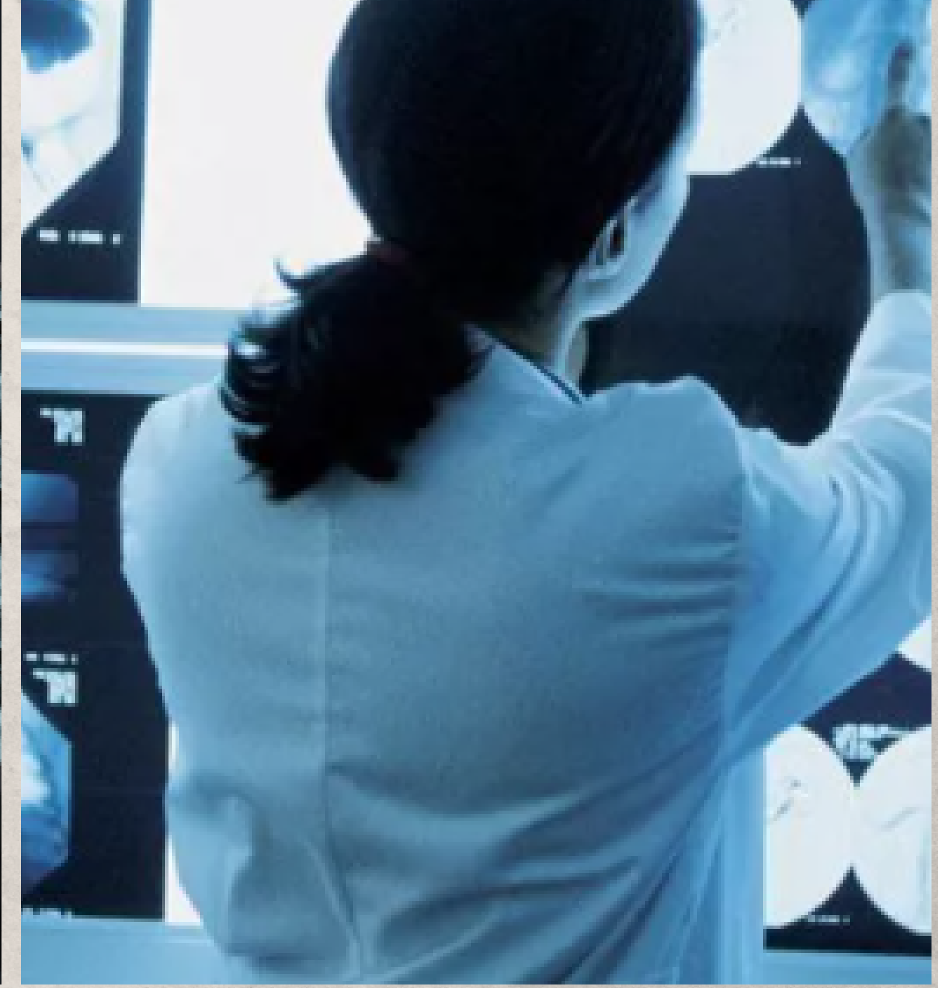
The seriousness of brain tumor is very big among all the variety of cancers, so to save a life immediate detection and proper treatment to must be done.

It is very difficult to have vision about the abnormal structures of human brain using simple imaging techniques, thus MRI images are used which provides plentiful information about the human soft tissue, which helps in the diagnosis of brain tumor.

Research contribution

We tried implementing a transfer learning method, that is, using a pre-trained VGG 19 model. After training our dataset, we were successful in building a less complex model with a fairly better accuracy of 96.4%.

Tools /Technology



Tool

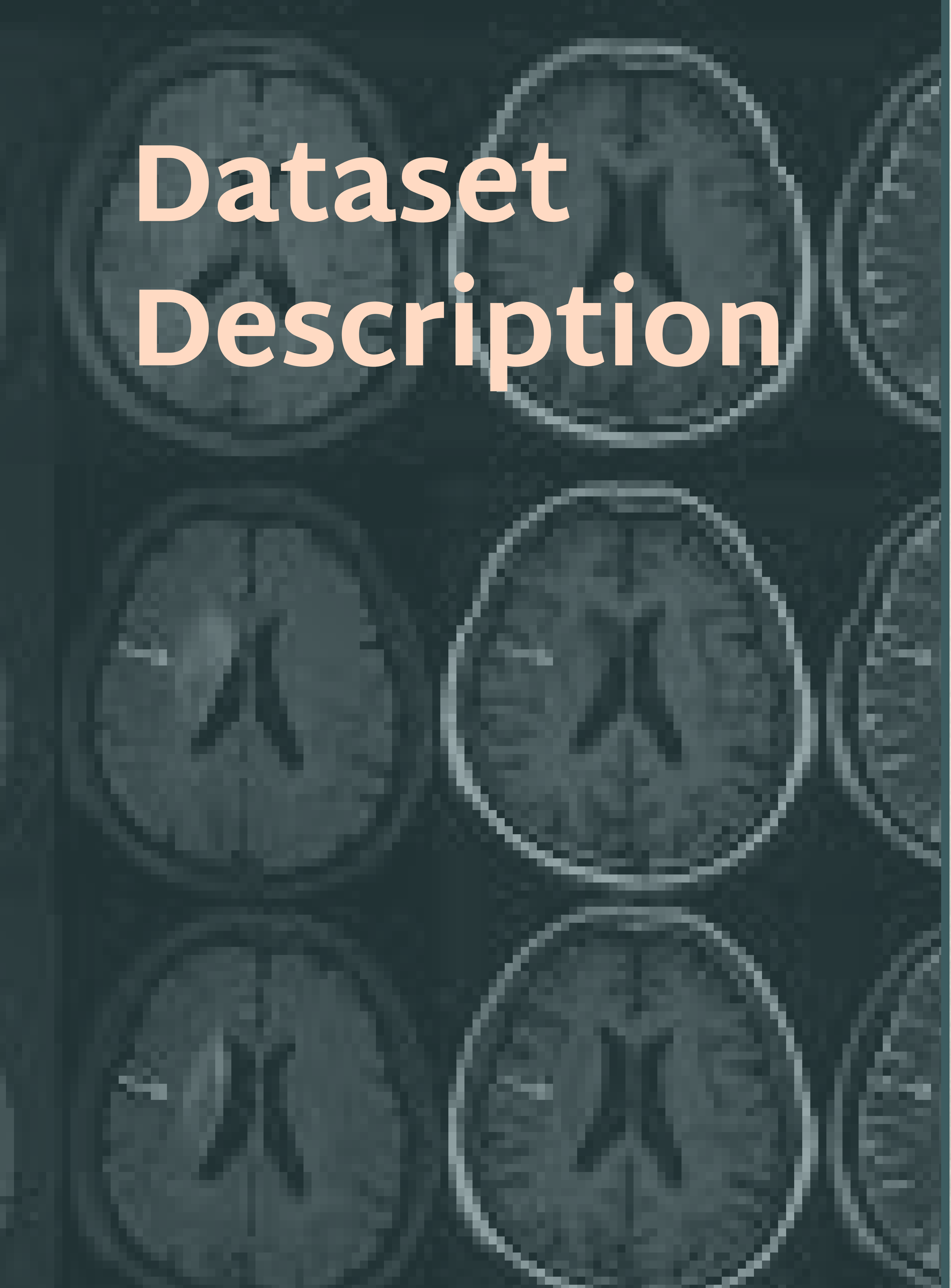
Jupyter Notebook

Library

- Keras
- Tensorflow
- Pandas
- Matplotlib

Model

VGG19



Dataset Description

The data set consists of two different folders that are Yes or No.

Yes folder has patients that have brain tumors whereas No folder has MRI images of patients with no brain tumor.

There are a total of 155 images of positive patients of brain tumor and 98 images of other patients having no brain tumor.

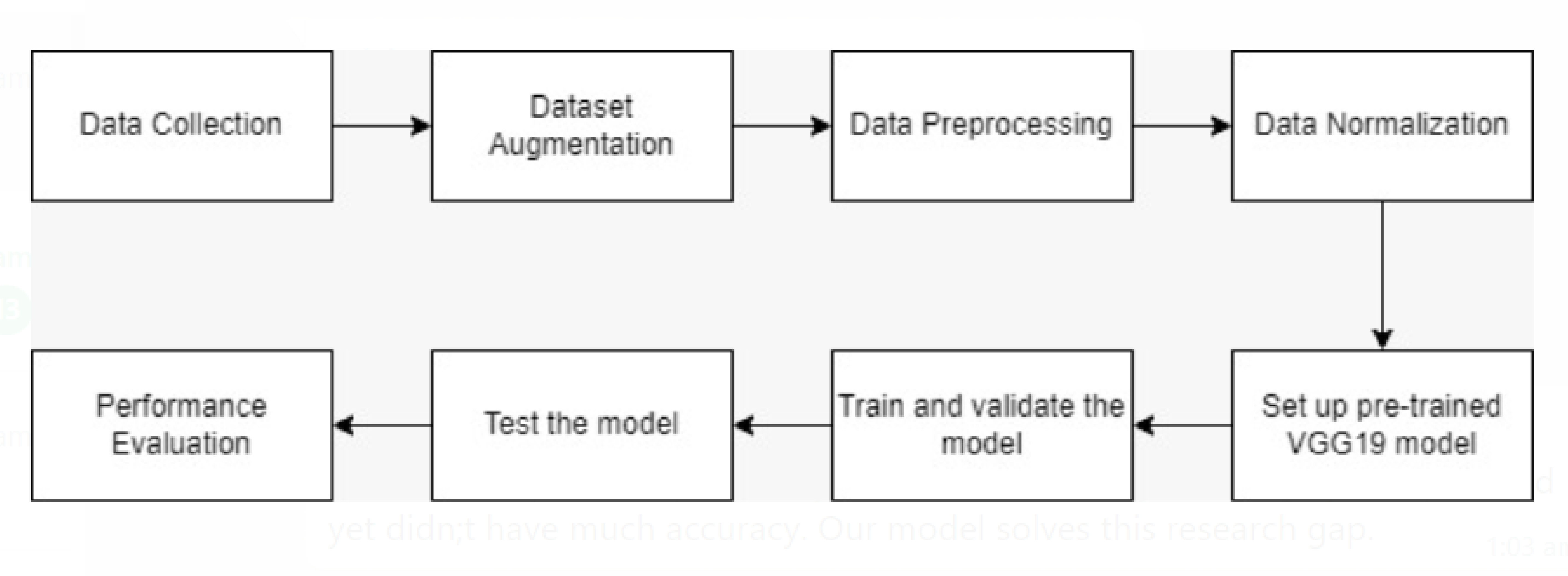
All the images are of 240X240 pixels.

After augmentation there are 1084 images which are positive and 1078 images which are negative

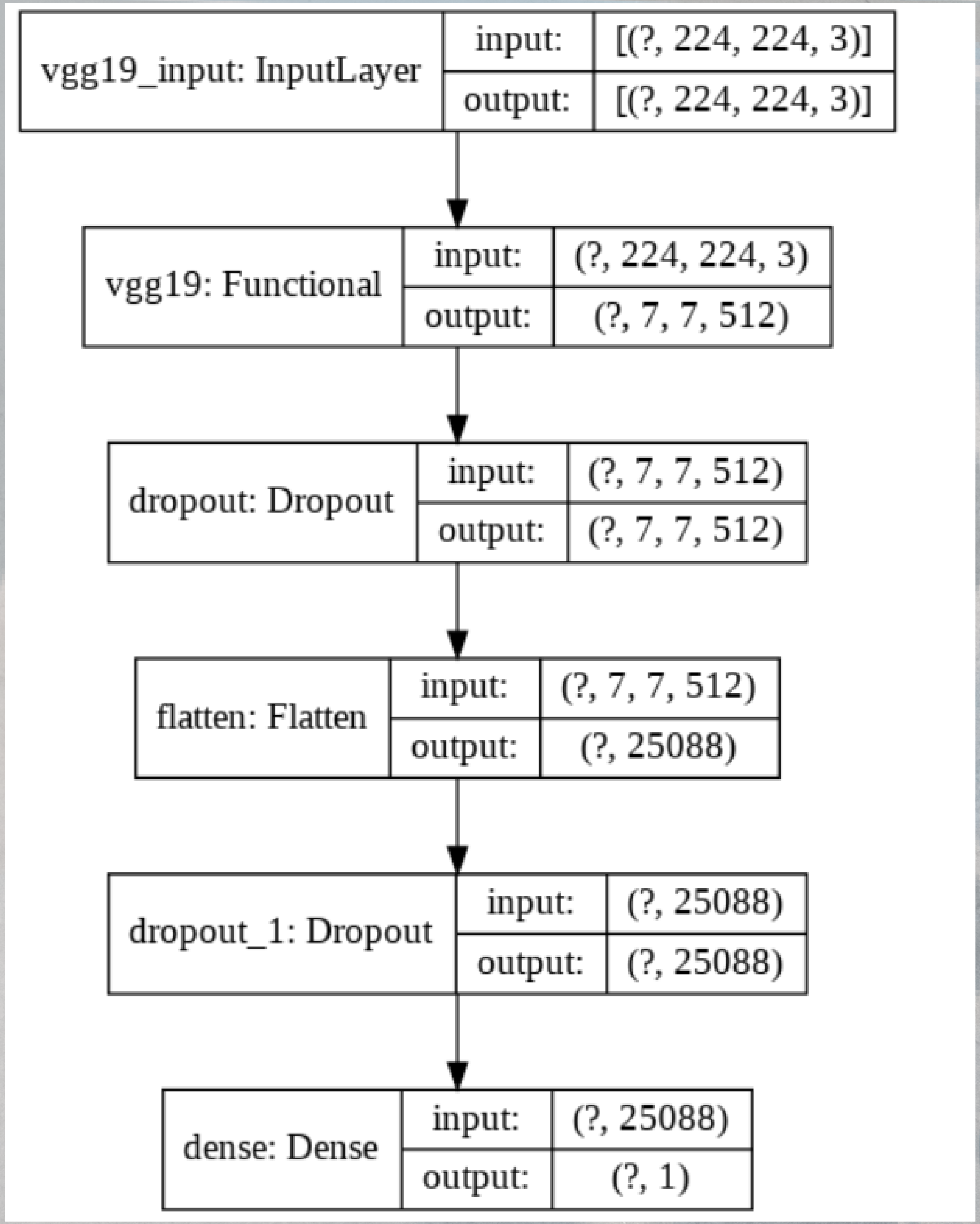
Data Pre-processing

- Augment the dataset
- Resize the images
- Crop the excess in the images

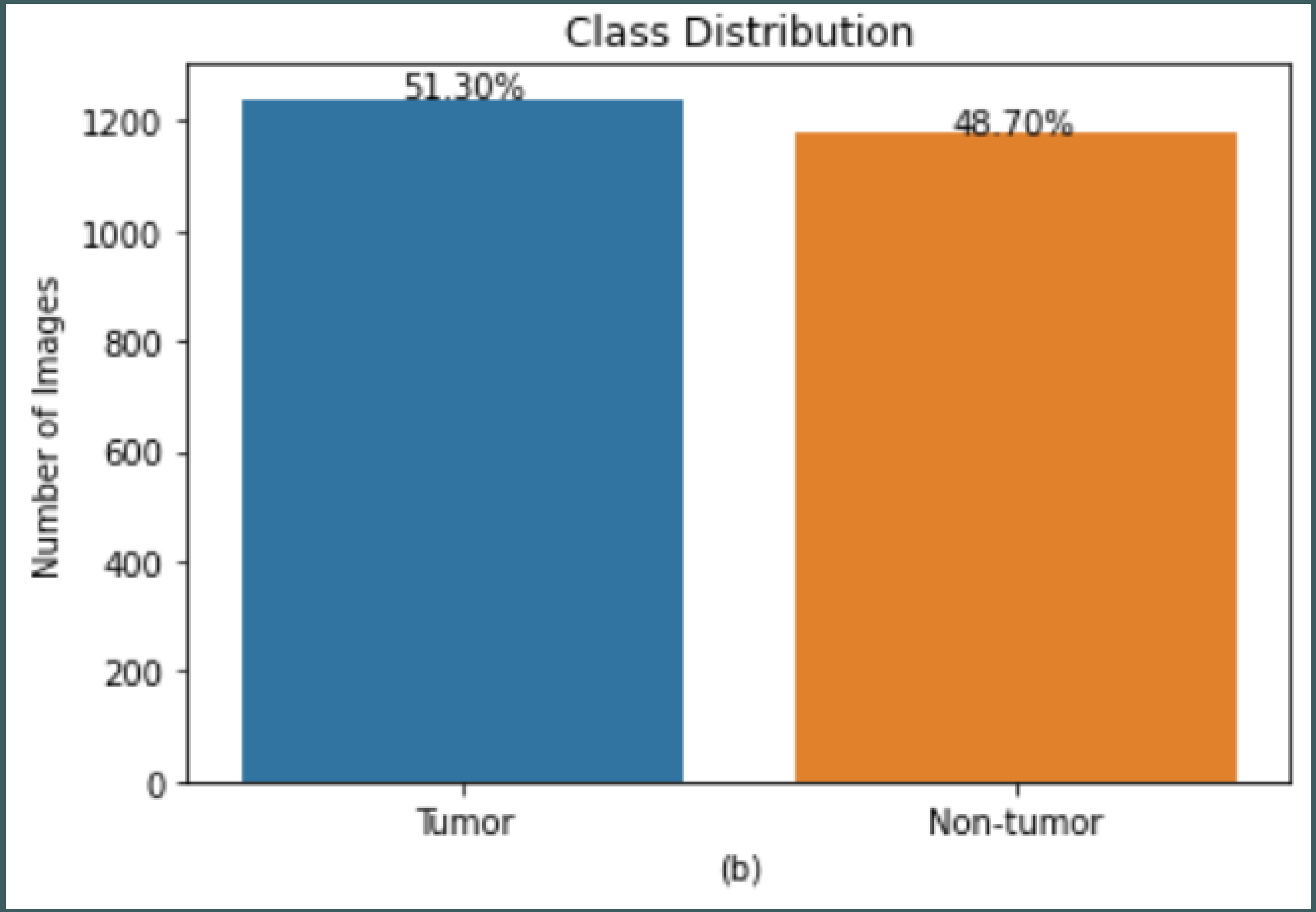
System architecture



Model architecture



Visualization



Training the VGG19 Model

```
Downloading data from https://storage.googleapis.com/tensorflow/keras
ls_notop.h5
80142336/80134624 [==============================] - 17s 0us/step
80150528/80134624 [==============================] - 17s 0us/step
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg19 (Functional)          (None, 7, 7, 512)         20024384

 dropout (Dropout)           (None, 7, 7, 512)         0

 flatten (Flatten)           (None, 25088)             0

 dropout_1 (Dropout)         (None, 25088)             0

 dense (Dense)               (None, 1)                 25089

=================================================================
Total params: 20,049,473
Trainable params: 25,089
Non-trainable params: 20,024,384
_________________________________________________________________
```

Training 5 epochs

```
Epoch 1/5
27/27 [==============================] - 571s 21s/step - loss: 1.6904 - acc: 0.7307 - val_loss: 0.4781 - val_acc: 0.8432
Epoch 2/5
27/27 [==============================] - 639s 24s/step - loss: 0.4379 - acc: 0.8716 - val_loss: 0.3344 - val_acc: 0.9027
Epoch 3/5
27/27 [==============================] - 643s 24s/step - loss: 0.2645 - acc: 0.9049 - val_loss: 0.1602 - val_acc: 0.9351
Epoch 4/5
27/27 [==============================] - 659s 25s/step - loss: 0.1963 - acc: 0.9340 - val_loss: 0.1752 - val_acc: 0.9351
Epoch 5/5
27/27 [==============================] - 629s 23s/step - loss: 0.1292 - acc: 0.9530 - val_loss: 0.1206 - val_acc: 0.9568
```

Output

```
# Evaluate the model on test set
score = vgg19.evaluate(X_test, y_test, verbose=0)

# Print test accuracy
print('\n', 'Test accuracy:', score[1])
```

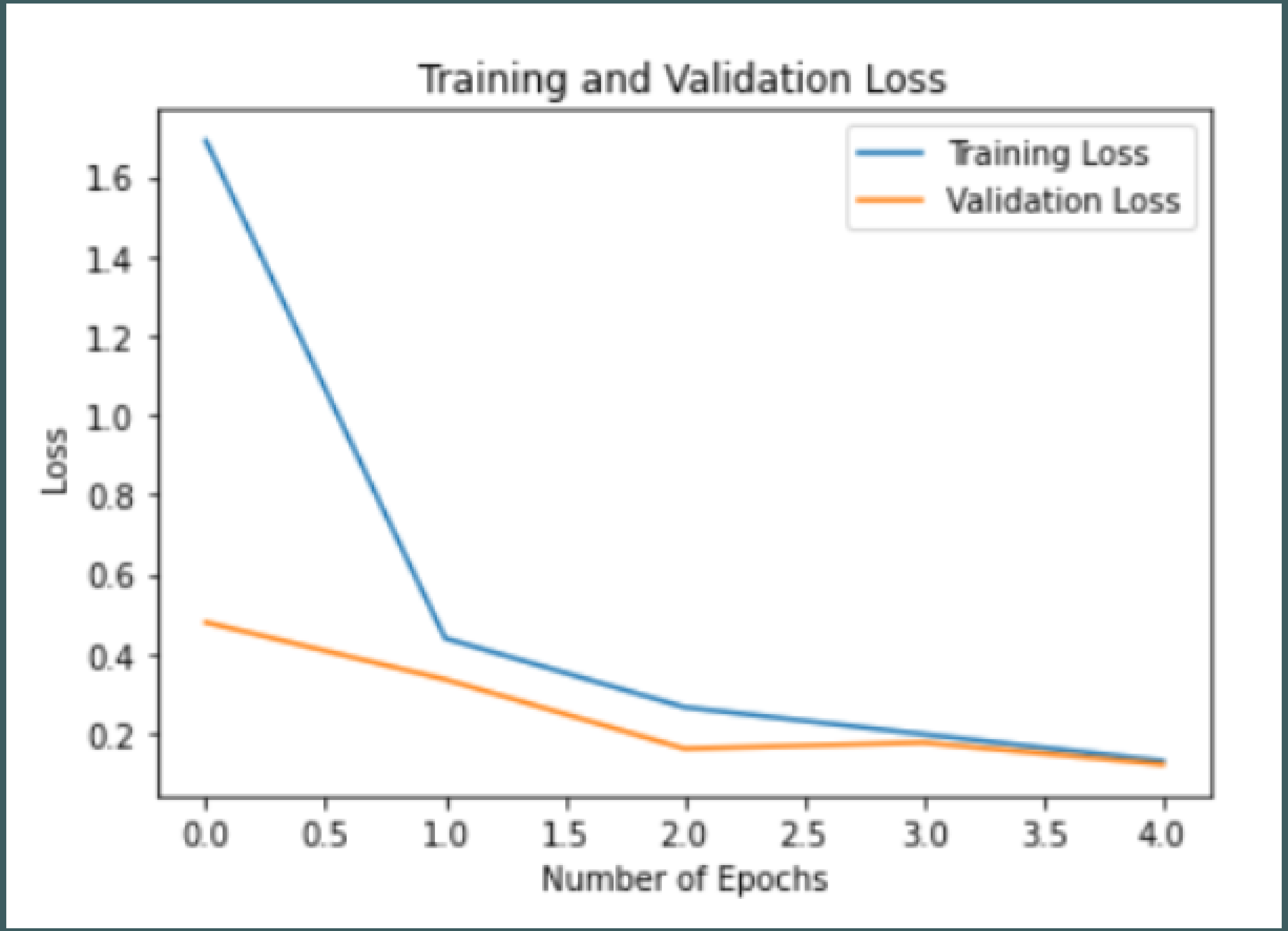
```
 Test accuracy: 0.9641873240470886
```

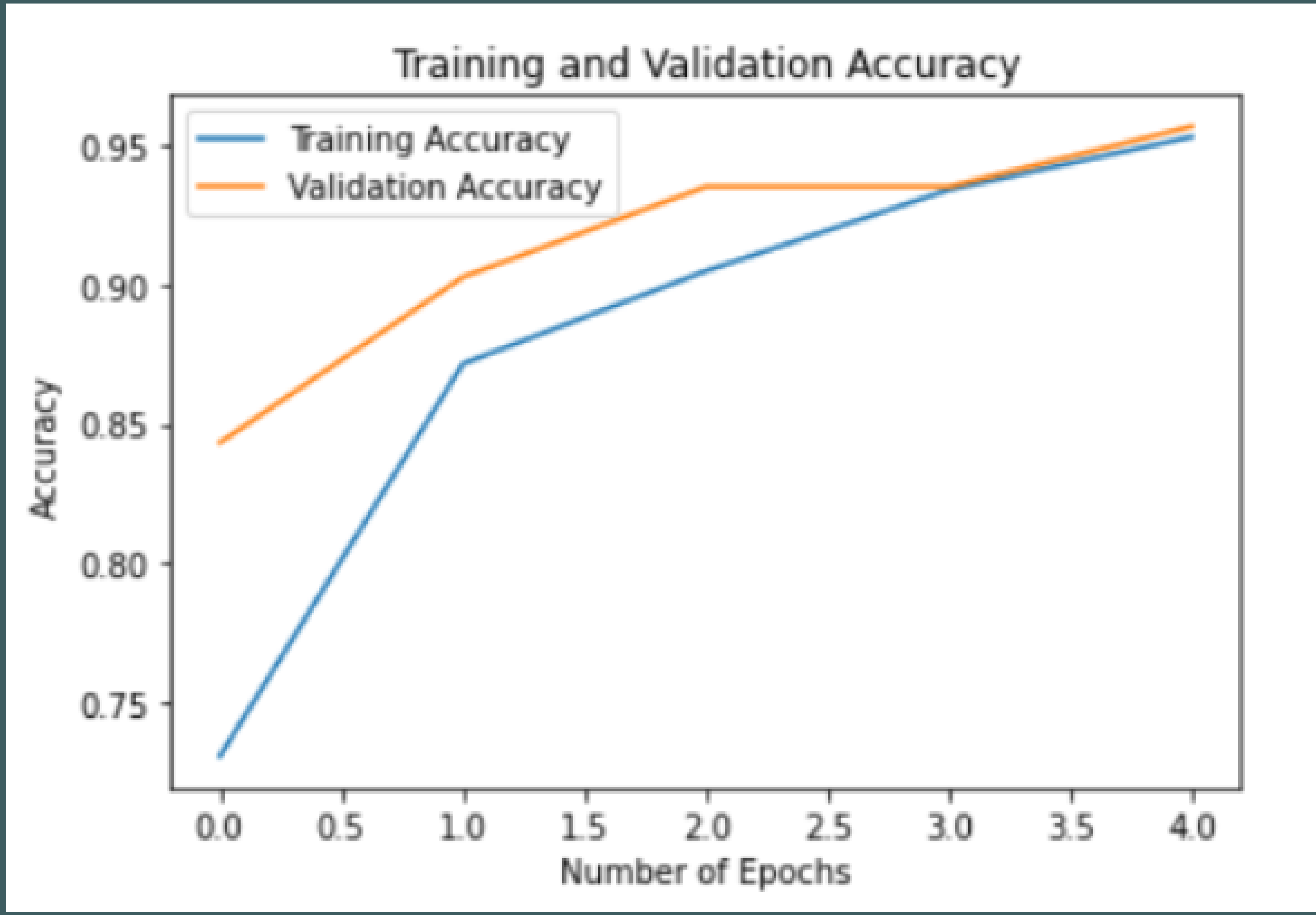
Accuracy of VGG19 model

Accuracy, Precision, Recall, Support

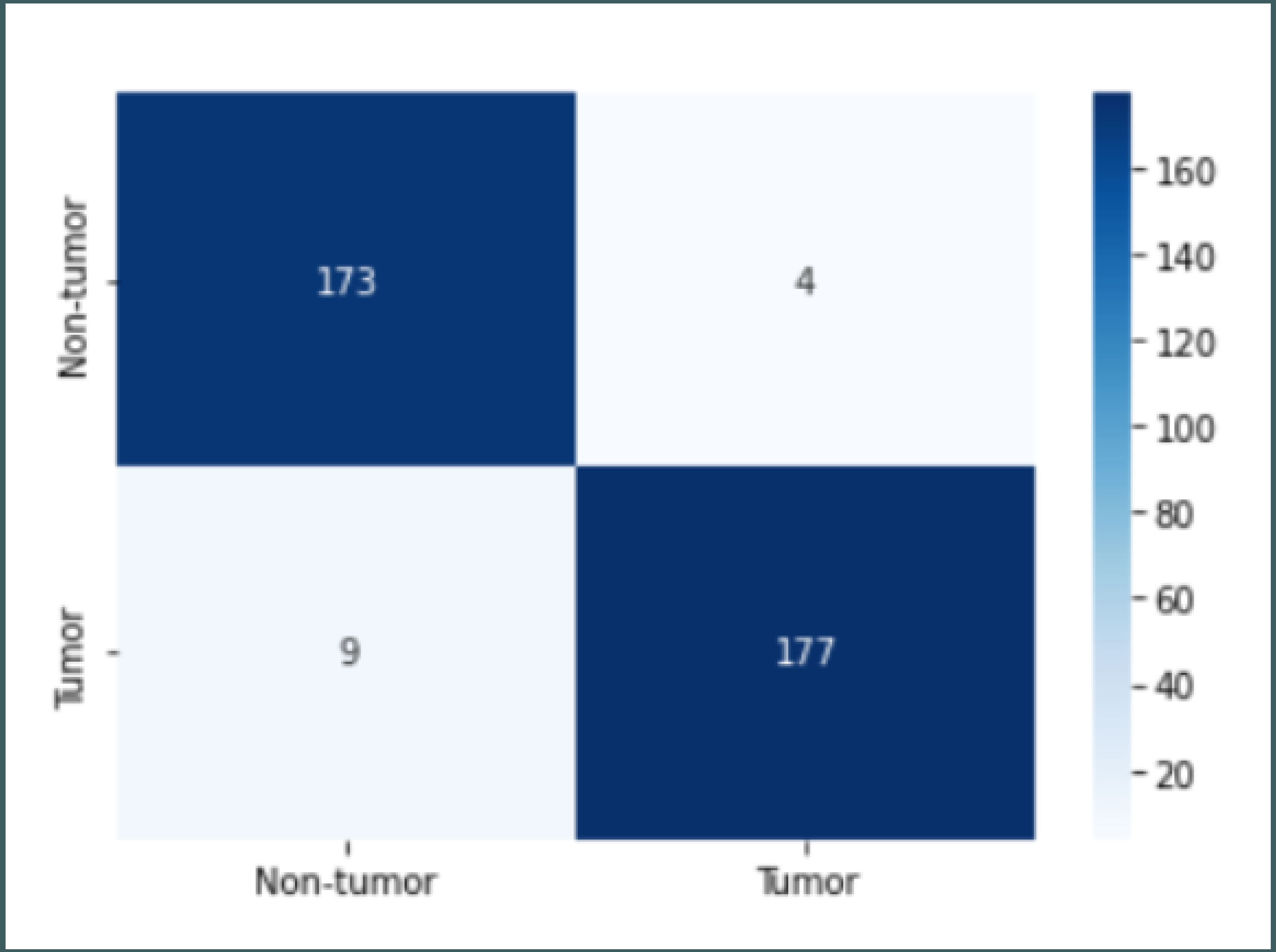
```
Accuracy: 0.964187
Precision: 0.977901
Recall: 0.951613
F1 score: 0.964578
Cohens kappa: 0.928380
ROC AUC: 0.964507
[[173   4]
 [  9 177]]
Specificity:  0.9774011299435028
              precision    recall  f1-score   support

           0       0.95      0.98      0.96       177
           1       0.98      0.95      0.96       186

    accuracy                           0.96       363
   macro avg       0.96      0.96      0.96       363
weighted avg       0.96      0.96      0.96       363
```

Training and validation loss over 5 epochs

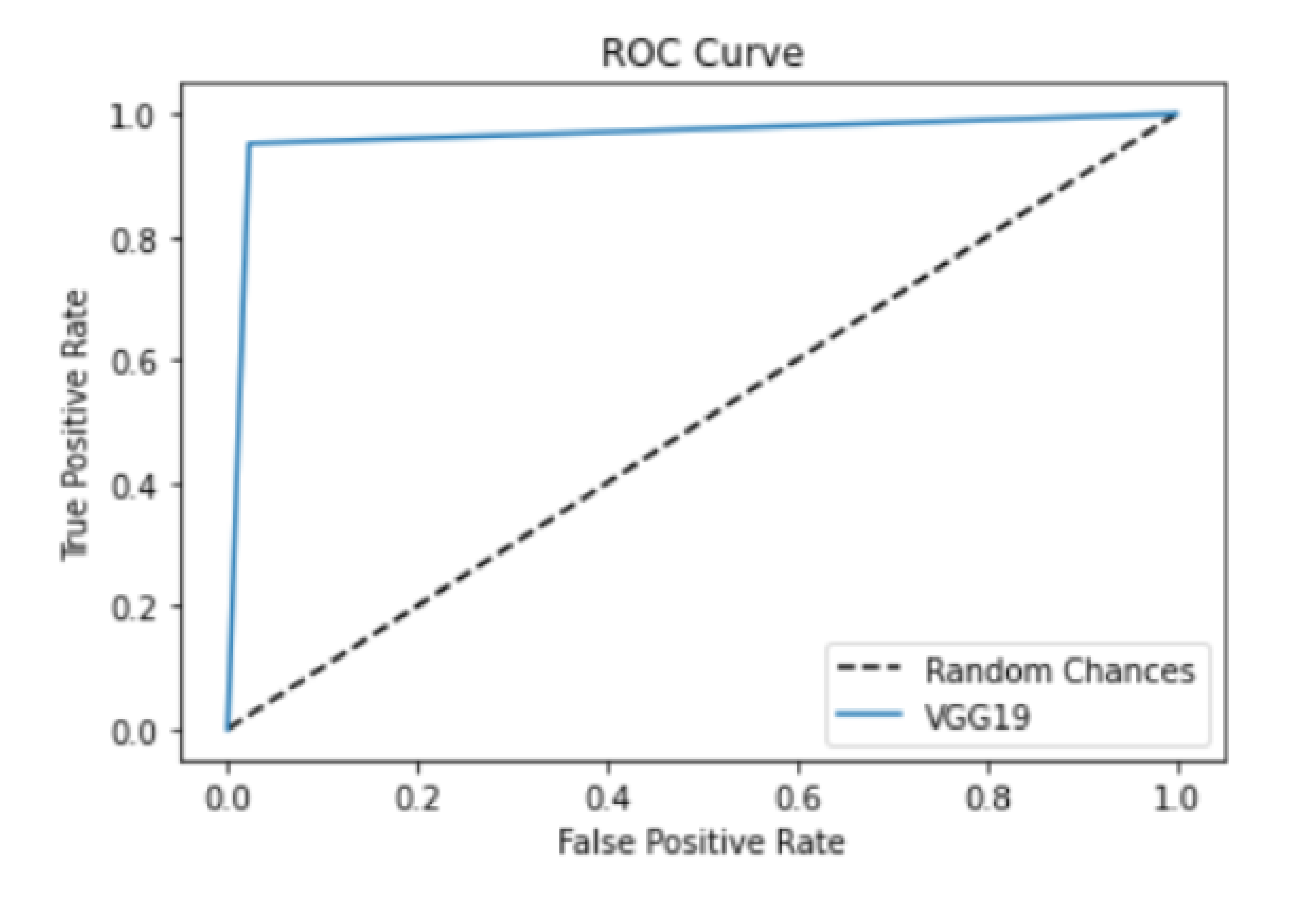




Evaluation metric: Confusion matrix



Evaluation metric: ROC Curve



Conclusion

We have successfully created a model that classifies MRI Scan Images. The accuracy of our model came out to be 96.4%. We employed a transfer learning-based approach, that is, VGG-19. VGG here stands for Visual Geometry Group. It is a standard deep Convolutional Neural Network (CNN) architecture with multiple layers. Our dataset initially did not have many images to work on. Thus, during pre-processing, we augmented the images in order to create a larger dataset and further cropped and resized the images for better accuracy of the model.

References

[1] T. Kalaiselvi, S. T. Padmapriya, P. Sriramakrishnan, and K. Somasundaram, ‘Deriving tumor detection models using convolutional neural networks from MRI of human brain scans’, Int. J. Inf. Technol., pp. 2–7, 2020, doi: 10.1007/s41870-020-00438-4.

[2]Roy, S. S., Rodrigues, N., & Taguchi, Y. (2020). Incremental Dilations Using CNN for Brain Tumor Classification. Applied Sciences, 10(14), 4915. doi:10.3390/app10144915

[3]Gokila Brindha, P., Kavinraj, M., Manivasakam, P., & Prasanth, P. (2021). Brain tumor detection from MRI images using deep learning techniques. IOP Conference Series: Materials Science and Engineering, 1055(1), 012115. doi:10.1088/1757-899x/1055/1/012115

[4]Amin, J., Sharif, M., Gul, N., Yasmin, M., & Shad, S. A. (2019). Brain Tumor Classification based on DWT Fusion of MRI Sequences using Convolutional Neural Network. Pattern Recognition Letters. doi:10.1016/j.patrec.2019.11.016

[5]Kalaiselvi, T., Padmapriya, S. T., Sriramakrishnan, P., & Somasundaram, K. (2020). Deriving tumor detection models using convolutional neural networks from MRI of human brain scans. International Journal of Information Technology. doi:10.1007/s41870-020-00438-4

[6]Toğaçar, M., Cömert, Z., & Ergen, B. (2020). Classification of Brain MRI Using Hyper Column Technique with Convolutional Neural Network and Feature Selection Method. Expert Systems with Applications, 113274. doi:10.1016/j.eswa.2020.113274

[7]J. Amin, M. Sharif, N. Gul, M. Yasmin, and S. Ali, 'Brain tumor classification based on DWT fusion of MRI sequences using convolutional neural network ✩', vol. 129, pp. 115–122, 2020, doi: 10.1016/j.patrec.2019.11.016

[8]A. M. Sarhan, 'Brain Tumor Classification in Magnetic Resonance Images Using Deep Learning and Wavelet Transform', J. Biomed. Sci. Eng., vol. 13, no. 06, pp. 102–112, 2020, doi: 10.4236/jbise.2020.136010.

[9]N. Poonguzhali, K. R. Rajendra, T. Mageswari, and T. Pavithra, 'Heterogeneous deep neural network for healthcare using metric learning', 2019 IEEE Int. Conf. Syst. Comput. Autom. Networking, ICSCAN 2019, pp. 1–4, 2019, doi: 10.1109/ICSCAN.2019.8878728.

[10]K. Adu, Y. Yu, J. Cai, and N. Tashi, 'Dilated Capsule Network for Brain Tumor Type Classification Via MRI Segmented Tumor Region', in 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2019, no. December, pp. 942–947, doi: 10.1109/ROBIO49542.2019.8961610.

THANK YOU