NEW YORK INSTITUTE
OF TECHNOLOGY

New York Institute of Technology
Current Procedure of CNN and SVM

DTSC 870 Project I

Advisor: Houwei Cao

Members:

Michael Trzaskoma	1202901	mtrzasko@nyit.edu
Hui (Henry) Chen	1242445	hchen60@nyit.edu

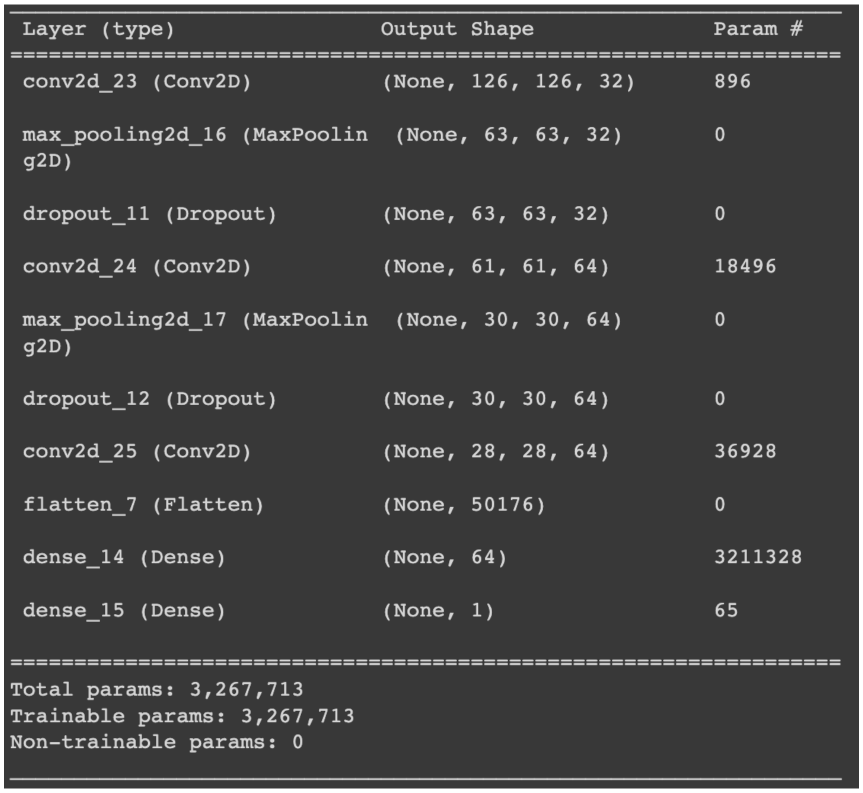
Table Of Contents

I. Classification Model Procedure

A. CNN Model

First step in regards to model creation is selecting the dimensions of the image to be utilized for classification, while the MRI tumor image is a 256x256 dimension, the dimension was reduced to 128x128 due to small sample size to reduce the amount of parameters to be tuned, resulting in reduced computation size. The images are also regularized from a 255 scale to a scale between 1 and 0 for pixel values.

Following is the creation of the initial structure of the cnn:



```
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_23 (Conv2D)          (None, 126, 126, 32)      896

 max_pooling2d_16 (MaxPoolin  (None, 63, 63, 32)       0
 g2D)

 dropout_11 (Dropout)        (None, 63, 63, 32)        0

 conv2d_24 (Conv2D)          (None, 61, 61, 64)        18496

 max_pooling2d_17 (MaxPoolin  (None, 30, 30, 64)       0
 g2D)

 dropout_12 (Dropout)        (None, 30, 30, 64)        0

 conv2d_25 (Conv2D)          (None, 28, 28, 64)        36928

 flatten_7 (Flatten)         (None, 50176)             0

 dense_14 (Dense)            (None, 64)                3211328

 dense_15 (Dense)            (None, 1)                 65

=================================================================
Total params: 3,267,713
Trainable params: 3,267,713
Non-trainable params: 0
_________________________________________________________________
```

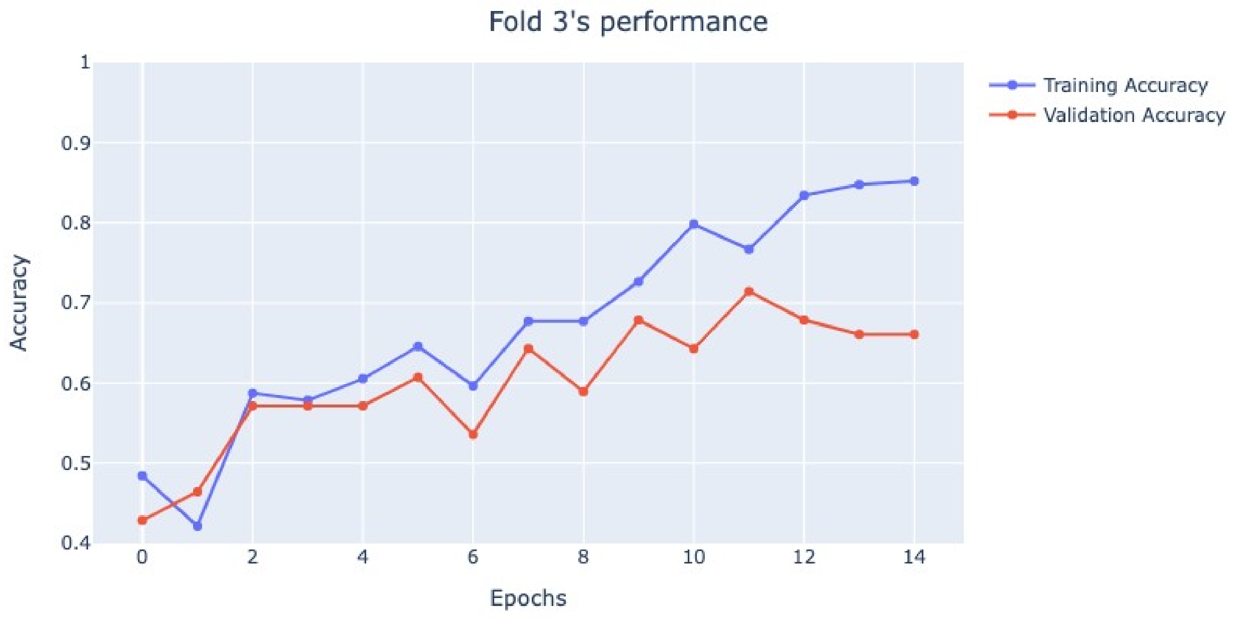
After creation of the model, before we begin training the model with the training data, the weights of the classes are calculated. The purpose of this is when training the mode, we would not want the model to be biased towards the majority class when training is done. The calculation of the weights are shown with the following equation:

$$weight = \frac{DataSetSize}{ClassCount * ClassSize_c}$$

Where c indicates the class that the value ClassSize pertains to.

To train the data, we utilized stratified k-fold for validation folds, the purpose of utilizing stratified instead of regular k-fold is in order to preserve class distribution while doing the training/validation split, there have been instances where when using regular k-fold, the validation set did not contain any samples from the minority class.

Through viewing the k-fold means plots, we were able to determine if there was overfitting or underfitting present in the cnn. The method taken for overfitting was the inclusion of dropout layers, where the method taken for underfitting was applying more layers to the architecture, or adjusting the dropout amount if dropout layers were present. Adding/removing layers was also presented in an attempt to improve the validation accuracy. When iterating through the epochs of a specific architecture being fitted, the model who performed the highest was the model to be saved for that specific fold. An example of a plot is shown below:



This image is in respect to architecture version 4 of the architecture for its third fold, overfitting can be seen.

B. SVM Model

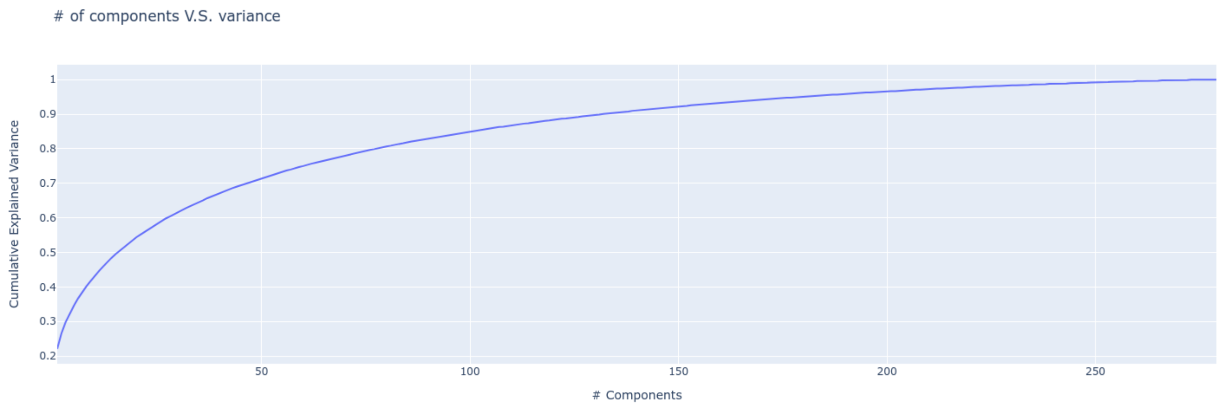
For the SVM model, the procedure is divided into the following steps:

1. Get in sight of the pixel dimension size, which is 256 * 256 * 3 = 196608. This value is just for one image. Therefore, in order to avoid the curse of dimensionality, we need to perform dimensionality reduction. In our project, we selected principal component analysis (PCA) as our technique.

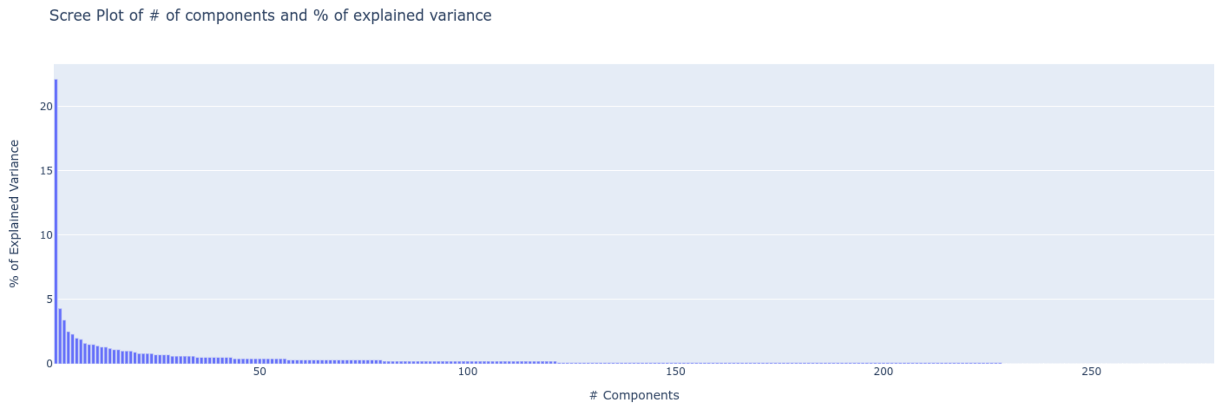
2. Standardization the $train_x$ and $test_x$ so that the value of pixel is rescaled with 0 mean and standard deviation of 1.

3. Dimensionality Reduction (PCA) to reduce the dimension of the data. Despite the fact that PCA is not a technique for selecting features, we used it to select our features. The choice of number of components to keep is based on the percentage of variance. The higher the variance is, the less data is losing, and

vice versa. After the experiment, we discovered the following plots:



(Fig 1: number of component vs cumulative explained variance)



(Fig 2: number of components and % of explained variance)

As shown in Fig 2, PC1 accounts for 22.1% of the variance, but the components after it drop significantly. This entails PC1 is the most significant component since it represents more data than other components. However, we conduct an experiment by carrying out how the different percentage of variance would impact the model's accuracy (both weighted and unweighted). In our experiment, we selected 70%, 80%, 90%, 95%, 97%, 99%, and 100% variance for PCA, which corresponds to 46, 77, 131, 179, 206, 244, and 275 numbers of components. After applying the PCA on the dataset (train and test set), the dimension reduced to 279.

4. Model building with SVM classifiers where each of the models is trained with respect to different numbers of a percentage of variance on PCA. Then the model is evaluated with k-fold validation (5 folds) where the weighted and unweighted accuracies are computed. The model's hyperparameters are configured as follows: rbf as kernel and random state of 1. This configuration is to have an insight of how the model performs.

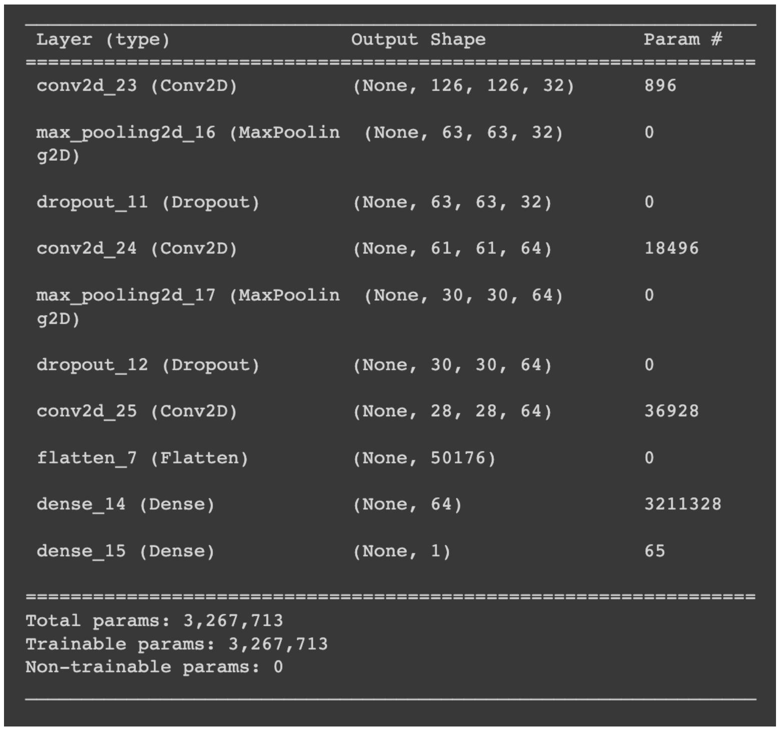
We further experiment with the SVM by tuning different hyperparameters such as *gamma* and *c* while maintaining rbf as our choice of kernel. In this procedure, we utilized grid search cross validation to search the optimal hyperparameters for the SVM model. As a result, the optimal hyperparameters are 0.1 for *c* and *scale* for *gamma*, and these configurations do not impact the accuracy of the model regardless of tuned or non-tune the hyperparameters.

II. Classification Model Results

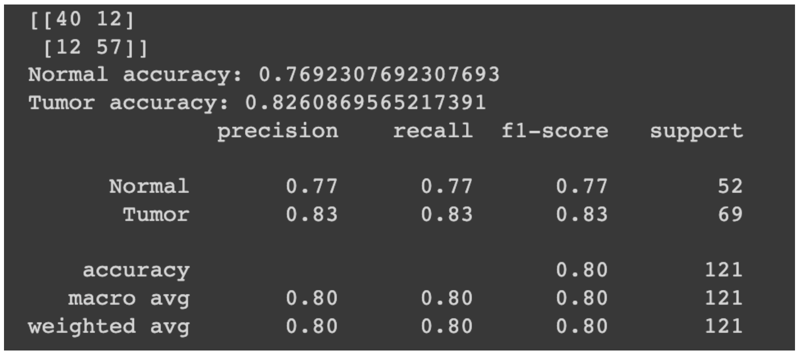
A. CNN Model

There were a total of 7 cnn architectures that were created, below will be the respective architecture and performance for each architecture utilizing the highest validation accuracy fold:

Architecture Version 1:

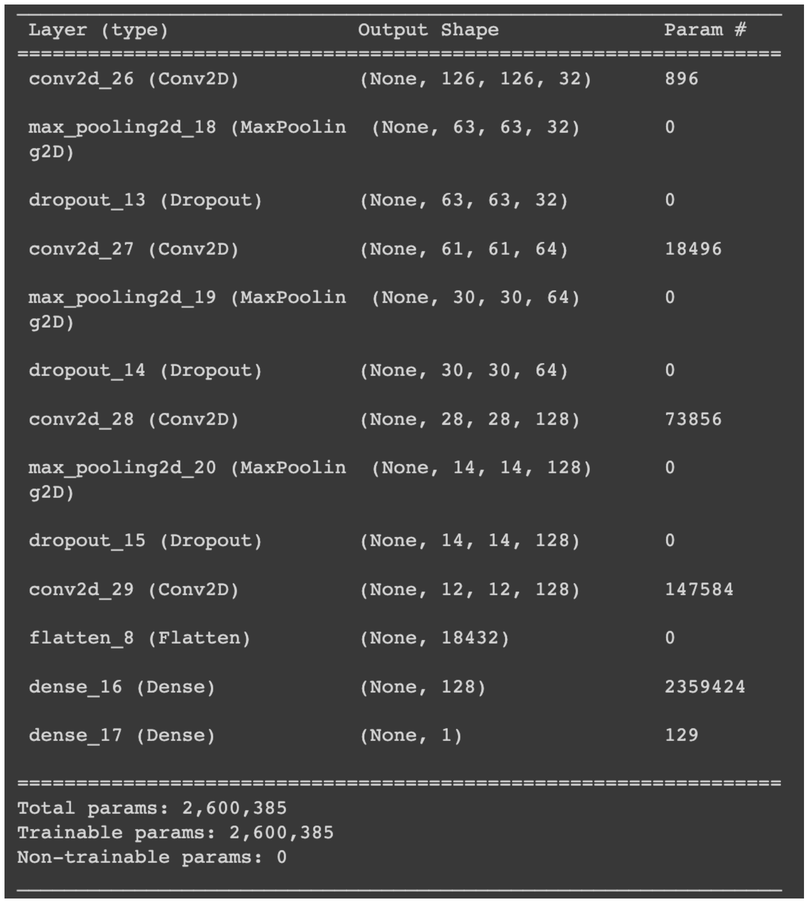


```
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_23 (Conv2D)          (None, 126, 126, 32)      896       
                                                                 
 max_pooling2d_16 (MaxPoolin  (None, 63, 63, 32)       0         
 g2D)                                                            
                                                                 
 dropout_11 (Dropout)        (None, 63, 63, 32)        0         
                                                                 
 conv2d_24 (Conv2D)          (None, 61, 61, 64)        18496     
                                                                 
 max_pooling2d_17 (MaxPoolin  (None, 30, 30, 64)       0         
 g2D)                                                            
                                                                 
 dropout_12 (Dropout)        (None, 30, 30, 64)        0         
                                                                 
 conv2d_25 (Conv2D)          (None, 28, 28, 64)        36928     
                                                                 
 flatten_7 (Flatten)         (None, 50176)             0         
                                                                 
 dense_14 (Dense)            (None, 64)                3211328   
                                                                 
 dense_15 (Dense)            (None, 1)                 65        
                                                                 
=================================================================
Total params: 3,267,713
Trainable params: 3,267,713
Non-trainable params: 0
_________________________________________________________________
```

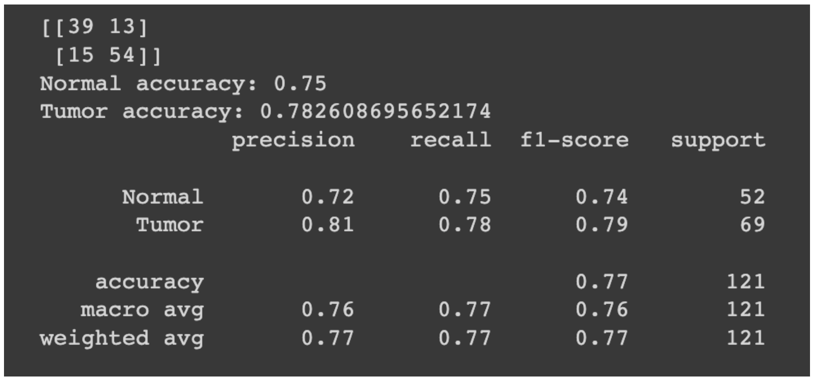


```
[[40 12]
 [12 57]]
Normal accuracy: 0.7692307692307693
Tumor accuracy: 0.8260869565217391
              precision    recall  f1-score   support

      Normal       0.77      0.77      0.77        52
       Tumor       0.83      0.83      0.83        69

    accuracy                           0.80       121
   macro avg       0.80      0.80      0.80       121
weighted avg       0.80      0.80      0.80       121
```

Architecture Version 2:

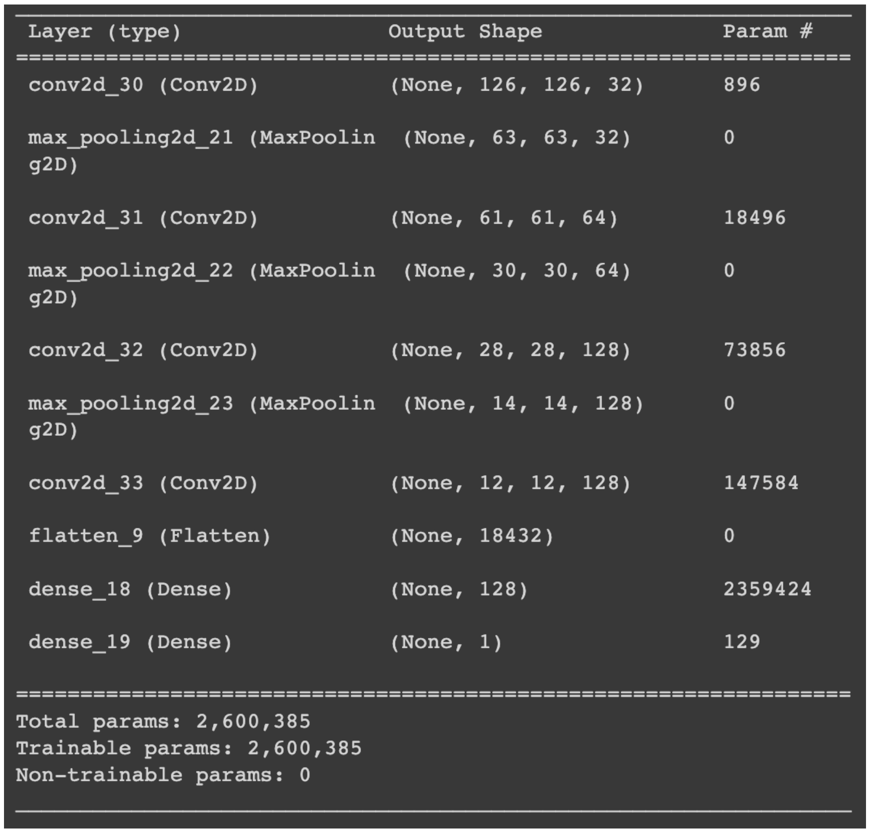


```
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_26 (Conv2D)          (None, 126, 126, 32)      896

 max_pooling2d_18 (MaxPoolin  (None, 63, 63, 32)       0
 g2D)

 dropout_13 (Dropout)        (None, 63, 63, 32)        0

 conv2d_27 (Conv2D)          (None, 61, 61, 64)        18496

 max_pooling2d_19 (MaxPoolin  (None, 30, 30, 64)       0
 g2D)

 dropout_14 (Dropout)        (None, 30, 30, 64)        0

 conv2d_28 (Conv2D)          (None, 28, 28, 128)       73856

 max_pooling2d_20 (MaxPoolin  (None, 14, 14, 128)      0
 g2D)

 dropout_15 (Dropout)        (None, 14, 14, 128)       0

 conv2d_29 (Conv2D)          (None, 12, 12, 128)       147584

 flatten_8 (Flatten)         (None, 18432)             0

 dense_16 (Dense)            (None, 128)               2359424

 dense_17 (Dense)            (None, 1)                 129

=================================================================
Total params: 2,600,385
Trainable params: 2,600,385
Non-trainable params: 0
_________________________________________________________________
```

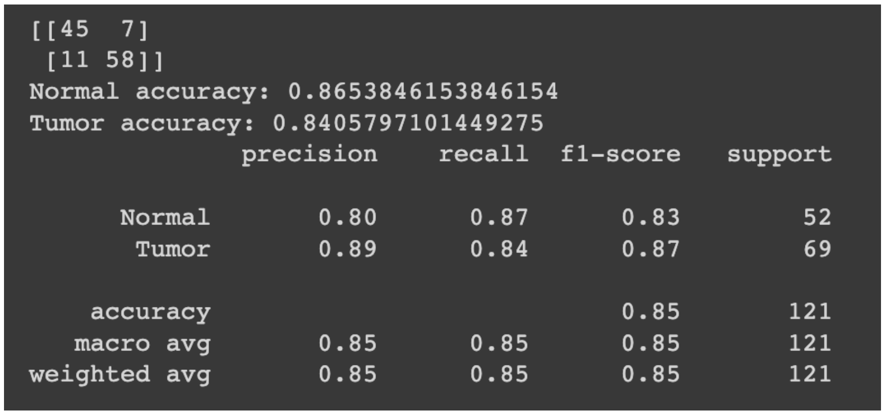


```
[[39 13]
 [15 54]]
Normal accuracy: 0.75
Tumor accuracy: 0.782608695652174
              precision    recall  f1-score   support

      Normal       0.72      0.75      0.74        52
       Tumor       0.81      0.78      0.79        69

    accuracy                           0.77       121
   macro avg       0.76      0.77      0.76       121
weighted avg       0.77      0.77      0.77       121
```

Architecture Version 3:

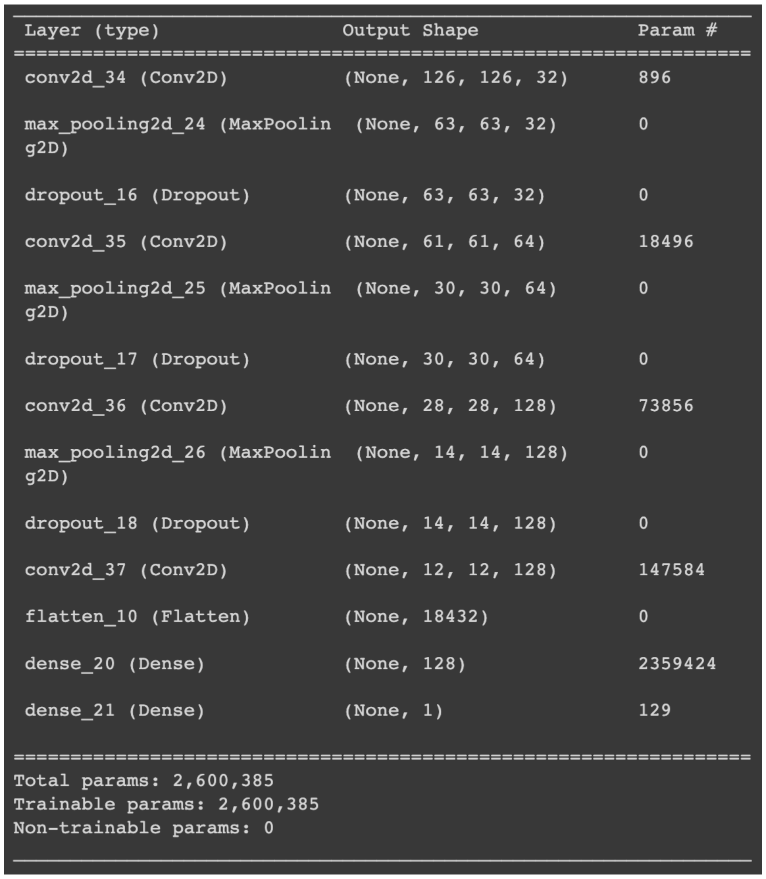


```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_30 (Conv2D)          (None, 126, 126, 32)      896

 max_pooling2d_21 (MaxPoolin  (None, 63, 63, 32)       0
 g2D)

 conv2d_31 (Conv2D)          (None, 61, 61, 64)        18496

 max_pooling2d_22 (MaxPoolin  (None, 30, 30, 64)       0
 g2D)

 conv2d_32 (Conv2D)          (None, 28, 28, 128)       73856

 max_pooling2d_23 (MaxPoolin  (None, 14, 14, 128)      0
 g2D)

 conv2d_33 (Conv2D)          (None, 12, 12, 128)       147584

 flatten_9 (Flatten)         (None, 18432)             0

 dense_18 (Dense)            (None, 128)               2359424

 dense_19 (Dense)            (None, 1)                 129

=================================================================
Total params: 2,600,385
Trainable params: 2,600,385
Non-trainable params: 0
_________________________________________________________________
```

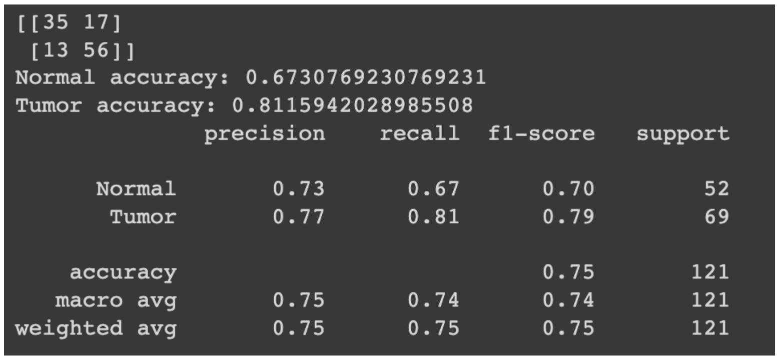


```
[[45  7]
 [11 58]]
Normal accuracy: 0.8653846153846154
Tumor accuracy: 0.8405797101449275
              precision    recall  f1-score   support

      Normal       0.80      0.87      0.83        52
       Tumor       0.89      0.84      0.87        69

    accuracy                           0.85       121
   macro avg       0.85      0.85      0.85       121
weighted avg       0.85      0.85      0.85       121
```

Architecture Version 4:

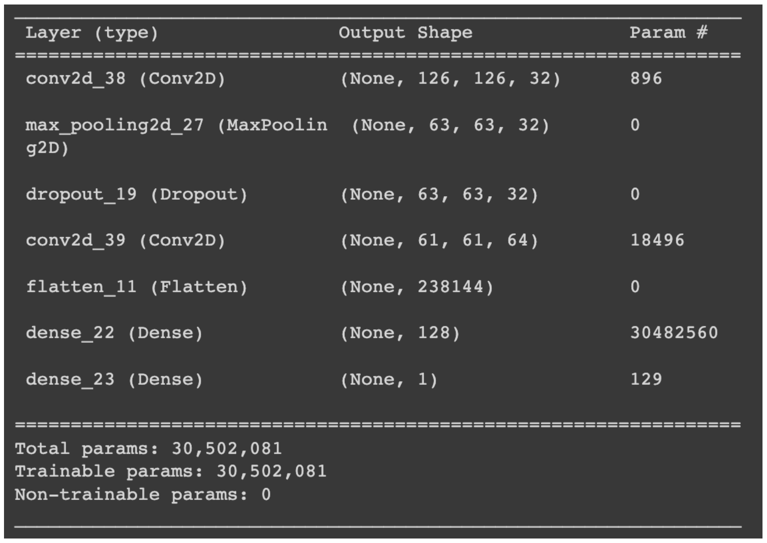


```
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_34 (Conv2D)          (None, 126, 126, 32)      896

 max_pooling2d_24 (MaxPoolin  (None, 63, 63, 32)       0
 g2D)

 dropout_16 (Dropout)        (None, 63, 63, 32)        0

 conv2d_35 (Conv2D)          (None, 61, 61, 64)        18496

 max_pooling2d_25 (MaxPoolin  (None, 30, 30, 64)       0
 g2D)

 dropout_17 (Dropout)        (None, 30, 30, 64)        0

 conv2d_36 (Conv2D)          (None, 28, 28, 128)       73856

 max_pooling2d_26 (MaxPoolin  (None, 14, 14, 128)      0
 g2D)

 dropout_18 (Dropout)        (None, 14, 14, 128)       0

 conv2d_37 (Conv2D)          (None, 12, 12, 128)       147584

 flatten_10 (Flatten)        (None, 18432)             0

 dense_20 (Dense)            (None, 128)               2359424

 dense_21 (Dense)            (None, 1)                 129

=================================================================
Total params: 2,600,385
Trainable params: 2,600,385
Non-trainable params: 0
_________________________________________________________________
```

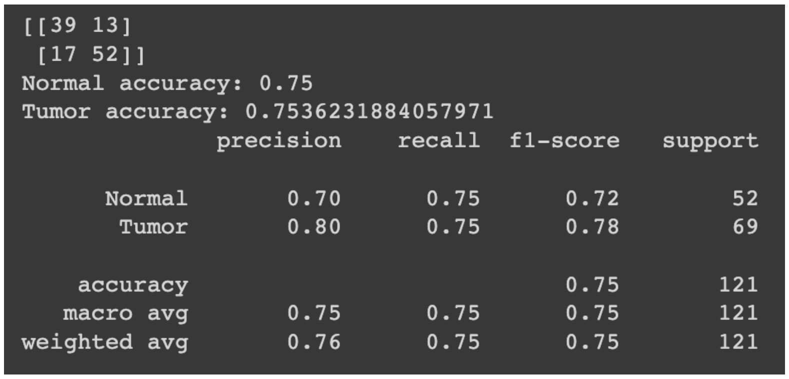


```
[[35 17]
 [13 56]]
Normal accuracy: 0.6730769230769231
Tumor accuracy: 0.8115942028985508
              precision    recall  f1-score   support

      Normal       0.73      0.67      0.70        52
       Tumor       0.77      0.81      0.79        69

    accuracy                           0.75       121
   macro avg       0.75      0.74      0.74       121
weighted avg       0.75      0.75      0.75       121
```

Architecture Version 5:

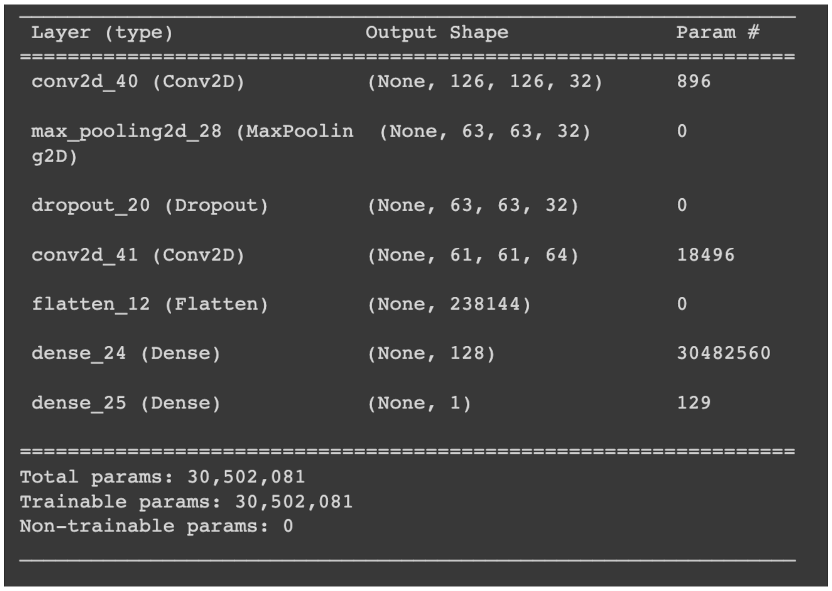


```
Layer (type)                Output Shape              Param #
=================================================================
 conv2d_38 (Conv2D)          (None, 126, 126, 32)      896

 max_pooling2d_27 (MaxPoolin  (None, 63, 63, 32)       0
 g2D)

 dropout_19 (Dropout)        (None, 63, 63, 32)        0

 conv2d_39 (Conv2D)          (None, 61, 61, 64)        18496

 flatten_11 (Flatten)        (None, 238144)            0

 dense_22 (Dense)            (None, 128)               30482560

 dense_23 (Dense)            (None, 1)                 129

=================================================================
Total params: 30,502,081
Trainable params: 30,502,081
Non-trainable params: 0
_________________________________________________________________
```

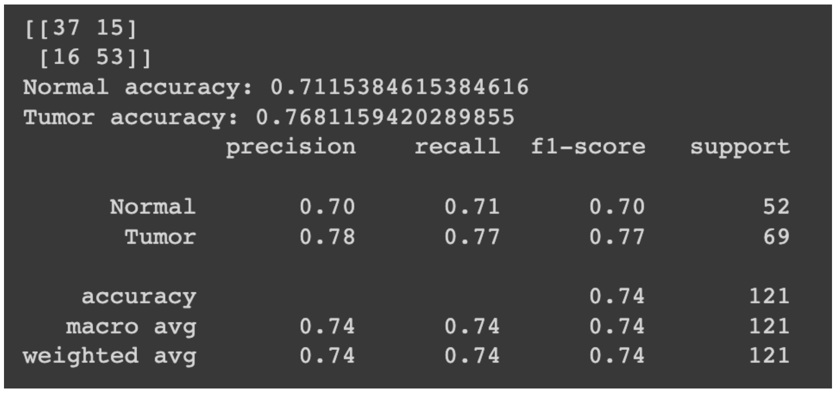


```
[[39 13]
 [17 52]]
Normal accuracy: 0.75
Tumor accuracy: 0.7536231884057971
              precision    recall  f1-score   support

      Normal       0.70      0.75      0.72        52
       Tumor       0.80      0.75      0.78        69

    accuracy                           0.75       121
   macro avg       0.75      0.75      0.75       121
weighted avg       0.76      0.75      0.75       121
```

Architecture Version 6:

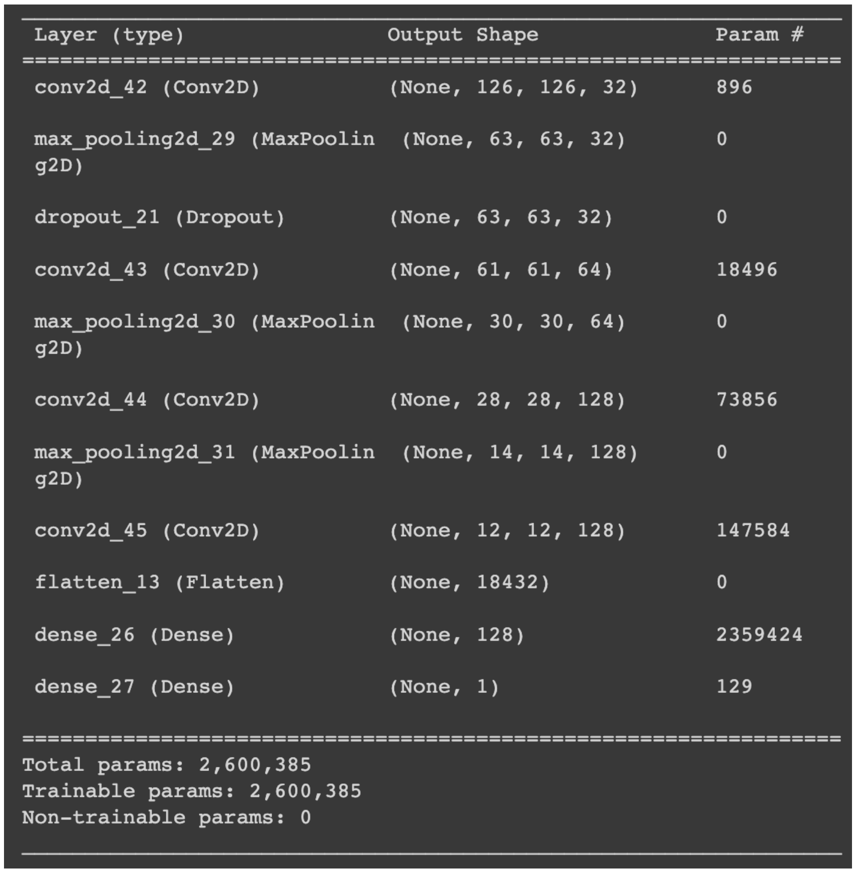


```
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_40 (Conv2D)          (None, 126, 126, 32)      896

 max_pooling2d_28 (MaxPoolin  (None, 63, 63, 32)       0
 g2D)

 dropout_20 (Dropout)        (None, 63, 63, 32)        0

 conv2d_41 (Conv2D)          (None, 61, 61, 64)        18496

 flatten_12 (Flatten)        (None, 238144)            0

 dense_24 (Dense)            (None, 128)               30482560

 dense_25 (Dense)            (None, 1)                 129

=================================================================
Total params: 30,502,081
Trainable params: 30,502,081
Non-trainable params: 0
_________________________________________________________________
```

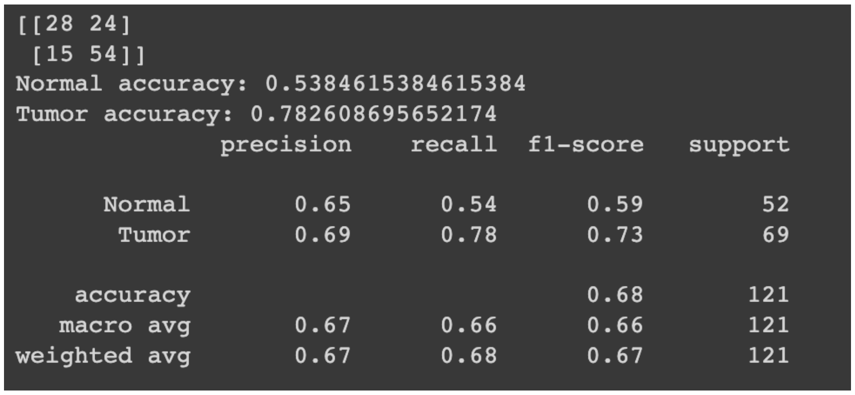


```
[[37 15]
 [16 53]]
Normal accuracy: 0.7115384615384616
Tumor accuracy: 0.7681159420289855
              precision    recall  f1-score   support

      Normal       0.70      0.71      0.70        52
       Tumor       0.78      0.77      0.77        69

    accuracy                           0.74       121
   macro avg       0.74      0.74      0.74       121
weighted avg       0.74      0.74      0.74       121
```

Architecture Version 7:

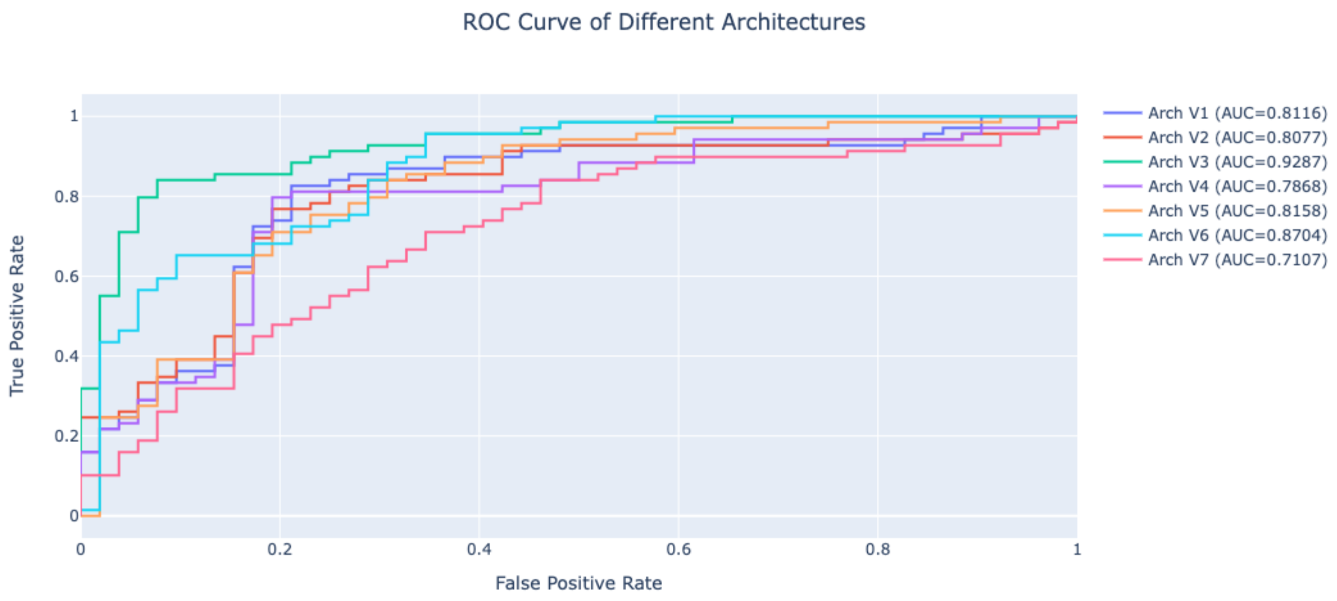


```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_42 (Conv2D)          (None, 126, 126, 32)      896

 max_pooling2d_29 (MaxPoolin  (None, 63, 63, 32)       0
 g2D)

 dropout_21 (Dropout)        (None, 63, 63, 32)        0

 conv2d_43 (Conv2D)          (None, 61, 61, 64)        18496

 max_pooling2d_30 (MaxPoolin  (None, 30, 30, 64)       0
 g2D)

 conv2d_44 (Conv2D)          (None, 28, 28, 128)       73856

 max_pooling2d_31 (MaxPoolin  (None, 14, 14, 128)      0
 g2D)

 conv2d_45 (Conv2D)          (None, 12, 12, 128)       147584

 flatten_13 (Flatten)        (None, 18432)             0

 dense_26 (Dense)            (None, 128)               2359424

 dense_27 (Dense)            (None, 1)                 129

=================================================================
Total params: 2,600,385
Trainable params: 2,600,385
Non-trainable params: 0
_________________________________________________________________
```



```
[[28 24]
 [15 54]]
Normal accuracy: 0.5384615384615384
Tumor accuracy: 0.782608695652174
              precision    recall  f1-score   support

      Normal       0.65      0.54      0.59        52
       Tumor       0.69      0.78      0.73        69

    accuracy                           0.68       121
   macro avg       0.67      0.66      0.66       121
weighted avg       0.67      0.68      0.67       121
```

ROC Curve:



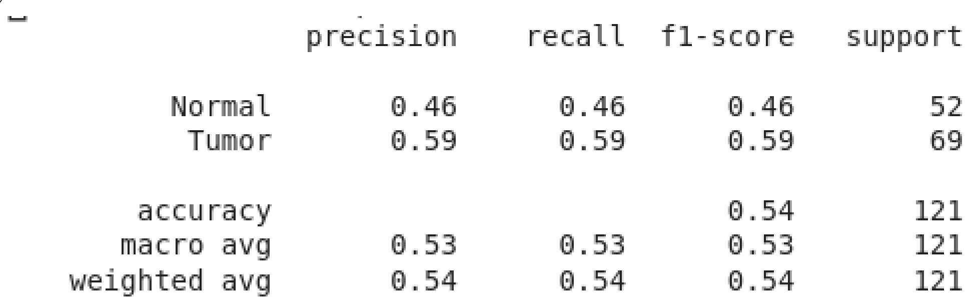
Stratified K-Fold Validation (Unweighted) Accuracy						
	Fold 1	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Avg**
Arch V1	**76.786**	**87.500**	**73.214**	**80.357**	**72.727**	**78.117**
Arch V2	**78.571**	**85.714**	**78.571**	**80.357**	**78.182**	**80.279**
Arch V3	**91.071**	**91.071**	**85.714**	**78.571**	**790.909**	**87.467**
Arch V4	**78.571**	**85.714**	**71.429**	**76.786**	**80.000**	**78.500**
Arch V5	**82.143**	**87.500**	**75.000**	**83.929**	**89.091**	**83.533**
Arch V6	**76.786**	**83.929**	**75.000**	**78.571**	**83.636**	**79.584**
Arch V7	**75.000**	**89.286**	**75.000**	**80.357**	**81.818**	**80.2922**

Concluding Findings:

The concluding findings from this experiment is that architecture version 3 had yielded the best performance in terms of average validation accuracy from the stratified k-fold with a value of 87.467, the highest AUC with a value of 0.9287, and individual class accuracy with values of 86.54% and 84.06% for Normal and Tumor class respectively. As a result, the most effective architecture that will most likely be utilized is the third version of the cnn architecture.

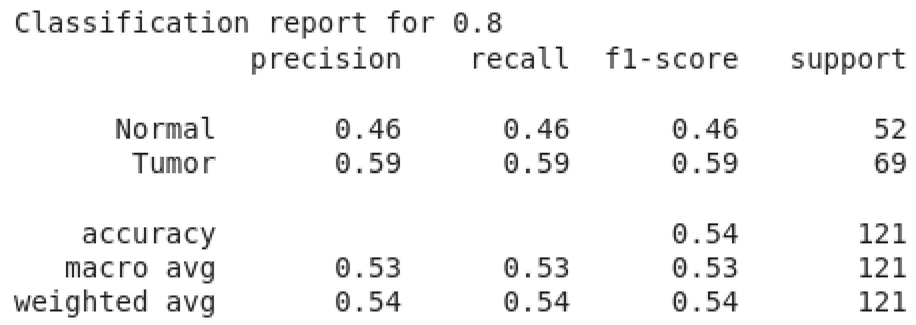
B. SVM Model

70% variance



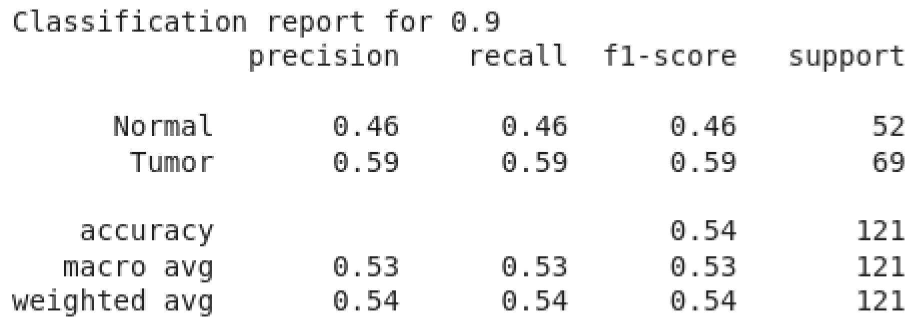
```
              precision    recall  f1-score   support

      Normal       0.46      0.46      0.46        52
       Tumor       0.59      0.59      0.59        69

    accuracy                           0.54       121
   macro avg       0.53      0.53      0.53       121
weighted avg       0.54      0.54      0.54       121
```

80% variance



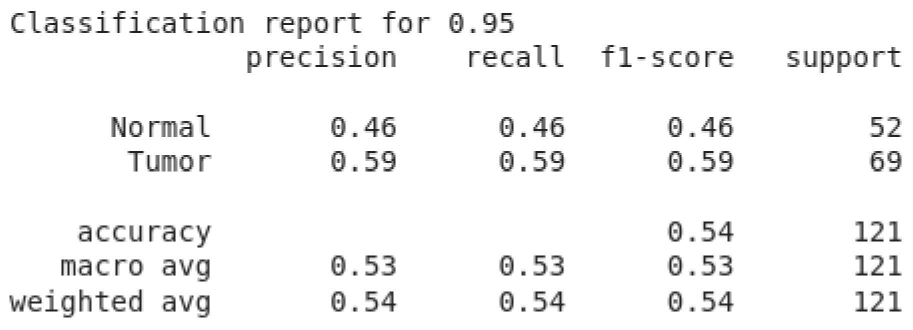
```
Classification report for 0.8
              precision    recall  f1-score   support

      Normal       0.46      0.46      0.46        52
       Tumor       0.59      0.59      0.59        69

    accuracy                           0.54       121
   macro avg       0.53      0.53      0.53       121
weighted avg       0.54      0.54      0.54       121
```

90% variance



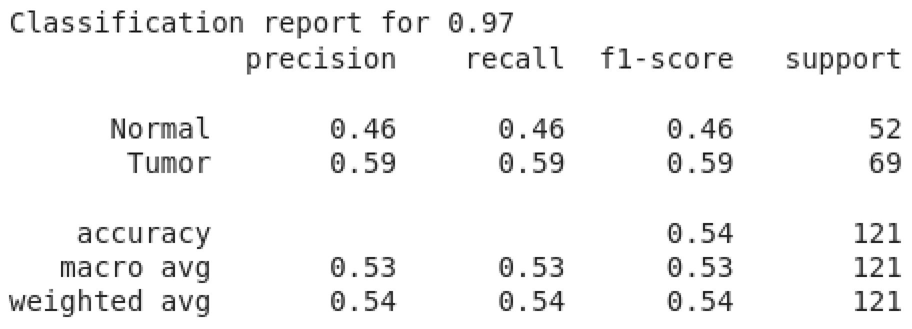
```
Classification report for 0.9
              precision    recall  f1-score   support

      Normal       0.46      0.46      0.46        52
       Tumor       0.59      0.59      0.59        69

    accuracy                           0.54       121
   macro avg       0.53      0.53      0.53       121
weighted avg       0.54      0.54      0.54       121
```

95% variance



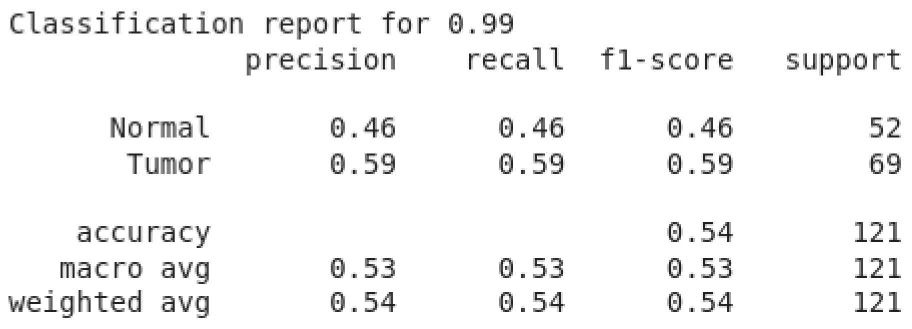
```
Classification report for 0.95
              precision    recall  f1-score   support

      Normal       0.46      0.46      0.46        52
       Tumor       0.59      0.59      0.59        69

    accuracy                           0.54       121
   macro avg       0.53      0.53      0.53       121
weighted avg       0.54      0.54      0.54       121
```

97% variance



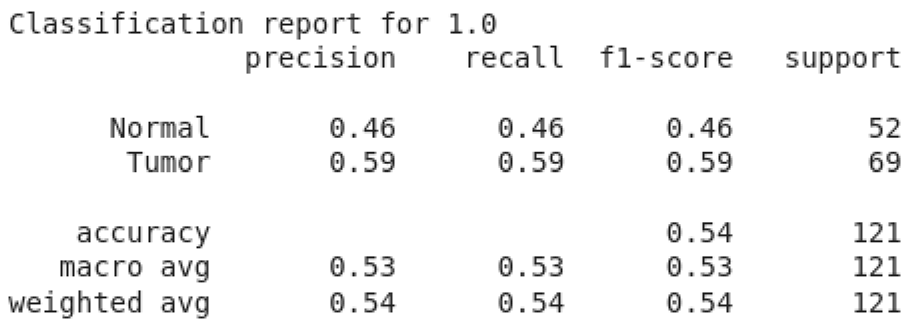
```
Classification report for 0.97
              precision    recall  f1-score   support

      Normal       0.46      0.46      0.46        52
       Tumor       0.59      0.59      0.59        69

    accuracy                           0.54       121
   macro avg       0.53      0.53      0.53       121
weighted avg       0.54      0.54      0.54       121
```

99% variance



```
Classification report for 0.99
              precision    recall  f1-score   support

      Normal       0.46      0.46      0.46        52
       Tumor       0.59      0.59      0.59        69

    accuracy                           0.54       121
   macro avg       0.53      0.53      0.53       121
weighted avg       0.54      0.54      0.54       121
```

100% variance

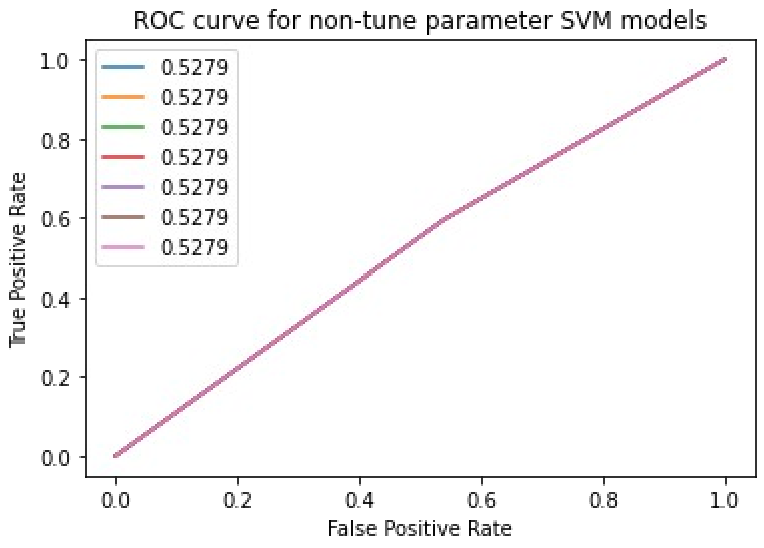


```
Classification report for 1.0
              precision    recall  f1-score   support

      Normal       0.46      0.46      0.46        52
       Tumor       0.59      0.59      0.59        69

    accuracy                           0.54       121
   macro avg       0.53      0.53      0.53       121
weighted avg       0.54      0.54      0.54       121
```

Per class accuracy

```
Accuracy per class (no parameter tunning):
The variance: 0.7
Normal accuracy: 46.15384615384615, Tumor accuracy: 59.42028985507246

The variance: 0.8
Normal accuracy: 46.15384615384615, Tumor accuracy: 59.42028985507246

The variance: 0.9
Normal accuracy: 46.15384615384615, Tumor accuracy: 59.42028985507246

The variance: 0.95
Normal accuracy: 46.15384615384615, Tumor accuracy: 59.42028985507246

The variance: 0.97
Normal accuracy: 46.15384615384615, Tumor accuracy: 59.42028985507246

The variance: 0.99
Normal accuracy: 46.15384615384615, Tumor accuracy: 59.42028985507246

The variance: 1.0
Normal accuracy: 46.15384615384615, Tumor accuracy: 59.42028985507246
```

ROC Curve



Stratified K-Fold Validation (Unweighted) Accuracy

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg

70% variance	53.5256	53.5256	52.0624	53.2469	52.7871	53.0295
80% variance	53.5256	53.5256	52.0624	53.2469	52.7871	53.0295
90% variance	53.5256	53.5256	52.0624	53.2469	52.7871	53.0295
95% variance	53.5256	53.5256	52.0624	53.2469	52.7871	53.0295
97% variance	53.5256	53.5256	52.0624	53.2469	52.7871	53.0295
99% variance	53.5256	53.5256	52.0624	53.2469	52.7871	53.0295
100% variance	53.5256	53.5256	52.0624	53.2469	52.7871	53.0295

Concluding Findings:

As we can see from the results, the SVM model does not get impacted by the percentage of the variance of the PCA as the average folds, accuracy per class, and ROC curve are the same. In other words, you can probably choose any percentage of variance for the PCA. When compared to the CNN, the SVM achieves very low accuracies, and further research is needed to determine why the accuracy is so low.