Comparison of CNN,Logistic Regression and SVM Models for Brain Tumor Detection

Convolutional Neural Networks(CNN)

CNNs are sub categories of deep learning which are based on layered architecture. They extract information about features regarding the images. They are used for image classification.

Support Vector Machines(SVM)

This is supervised learning algorithm. It is particularly used for binary classification algorithms and manages non-linear decision boundaries.

Logistic Regression

It helps in prediction of binary outcomes which is based on statistics.

Methodology

Data Collection and Processing

For the comparision, two different datasets were used. A dataset of brain MRI images labeled as either “yes” or “no” was used. Another dataset consists of training and testing modules.

Model Training

The CNN model is trained by splitting the data into 70% for training data, 15% for validation data and 15% for test data. The architecture of CNN was designed by implementing layers like batch normalization.
CNN used best model to evaluate performance of the model, so the best accuarcy is displayed in below figure.

```
    X_train,
    y_train,
    epochs=30,
    validation_data=(X_val,y_val),
    callbacks=cb)
from tensorflow import keras

best_model = keras.models.load_model("./bestmodel.keras")
best_model.save("./bestmodel.keras")
```

```
Epoch 1/30
3/4 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step - accuracy: 0.5573 - loss: 0.6897
Epoch 1: val_loss improved from inf to 0.69632, saving model to ./bestmodel.keras
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 269ms/step - accuracy: 0.5344 - loss: 0.6909 - val_accuracy: 0.4500 - val_loss: 0.6963
Epoch 2/30
3/4 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step - accuracy: 0.5503 - loss: 0.6905
Epoch 2: val_loss did not improve from 0.69632
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 90ms/step - accuracy: 0.5302 - loss: 0.6919 - val_accuracy: 0.4500 - val_loss: 0.6964
Epoch 3/30
3/4 ━━━━━━━━━━━━━━━━━━━━ 0s 81ms/step - accuracy: 0.5000 - loss: 0.6932
Epoch 3: val_loss did not improve from 0.69632
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.5000 - loss: 0.6932 - val_accuracy: 0.4500 - val_loss: 0.6964
Epoch 4/30
3/4 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.4948 - loss: 0.6930
Epoch 4: val_loss did not improve from 0.69632
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 70ms/step - accuracy: 0.4969 - loss: 0.6930 - val_accuracy: 0.4500 - val_loss: 0.6964
Epoch 5/30
3/4 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - accuracy: 0.4844 - loss: 0.6942
Epoch 5: val_loss did not improve from 0.69632
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 87ms/step - accuracy: 0.4906 - loss: 0.6938 - val_accuracy: 0.4500 - val_loss: 0.6964
Epoch 6/30
3/4 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.5417 - loss: 0.6919
Epoch 6: val_loss did not improve from 0.69632
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step - accuracy: 0.5250 - loss: 0.6929 - val_accuracy: 0.4500 - val_loss: 0.6963
```

SVM was used with feature scaling and model was trained using SVM and Logistic Regression.

Accuracy Comparision

Accuracy was considered as a prime factor for model performance evaluation.
The CNN model achieved 33.333% accuracy.(as depicted below)

```
acc=model.evaluate(test_data)[1]
print(f"Our Model Accuracy is {acc*100} %")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 3s 3s/step - accuracy: 0.3333 - loss: 0.7241
Our Model Accuracy is 33.33333432674408 %
```

After training the model using SVM and Logistic Regression , the performance was evaluated with score evaluation. This demonstrated that SVM was better for the dataset. While training the testing score SVM has higher accuracy as compared to Logistice Regression.

```
#Evaluation
print("Training Score:",lg.score(xtrain,ytrain))
print("Testing Score:",lg.score(xtest,ytest))
```

```
Training Score: 1.0
Testing Score: 0.9551020408163265
```

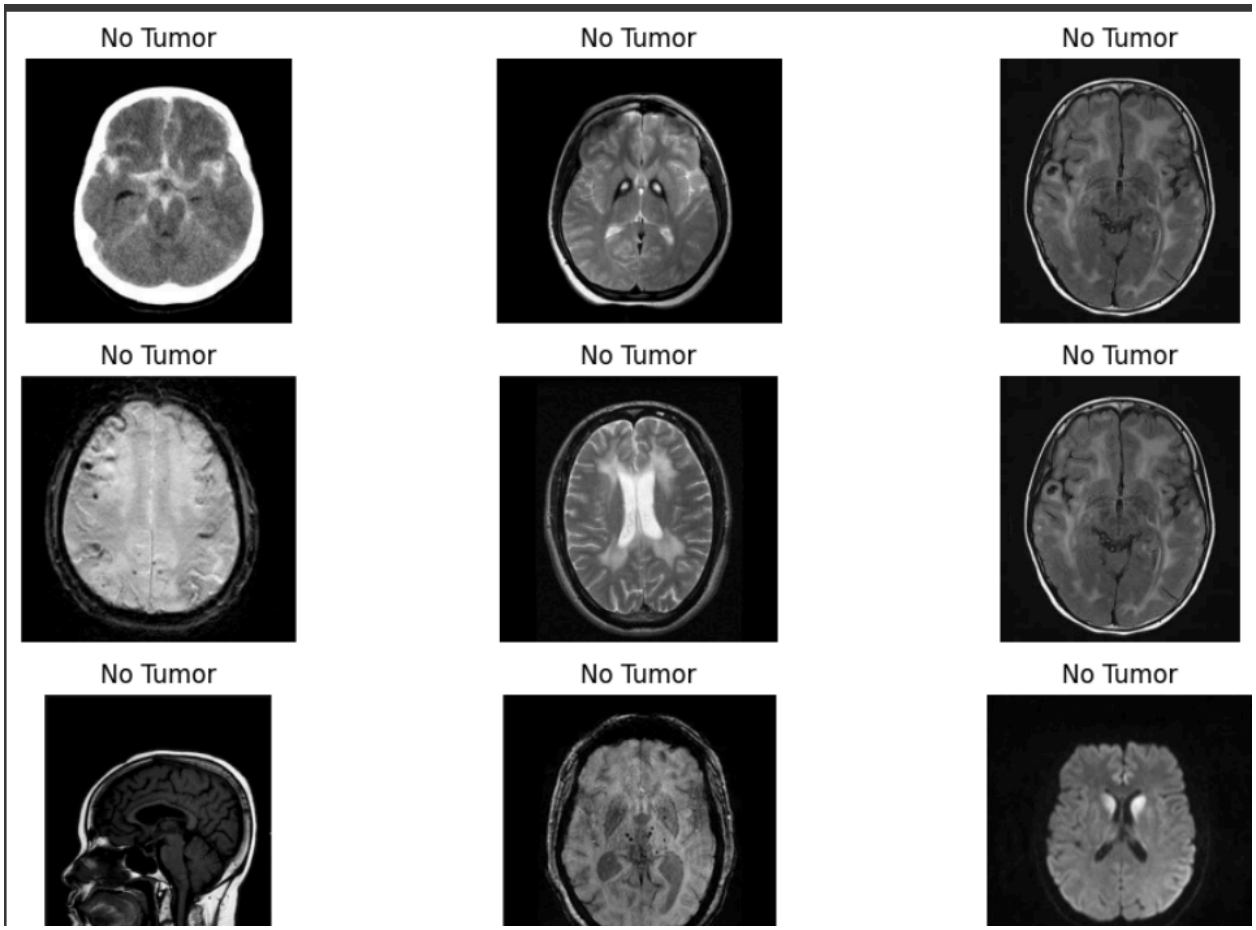
```
print("Training Score:",sv.score(xtrain,ytrain))
print("Testing Score:",sv.score(xtest,ytest))
```

```
Training Score: 0.9918116683725691
Testing Score: 0.9591836734693877
```

Conclusion Of Results

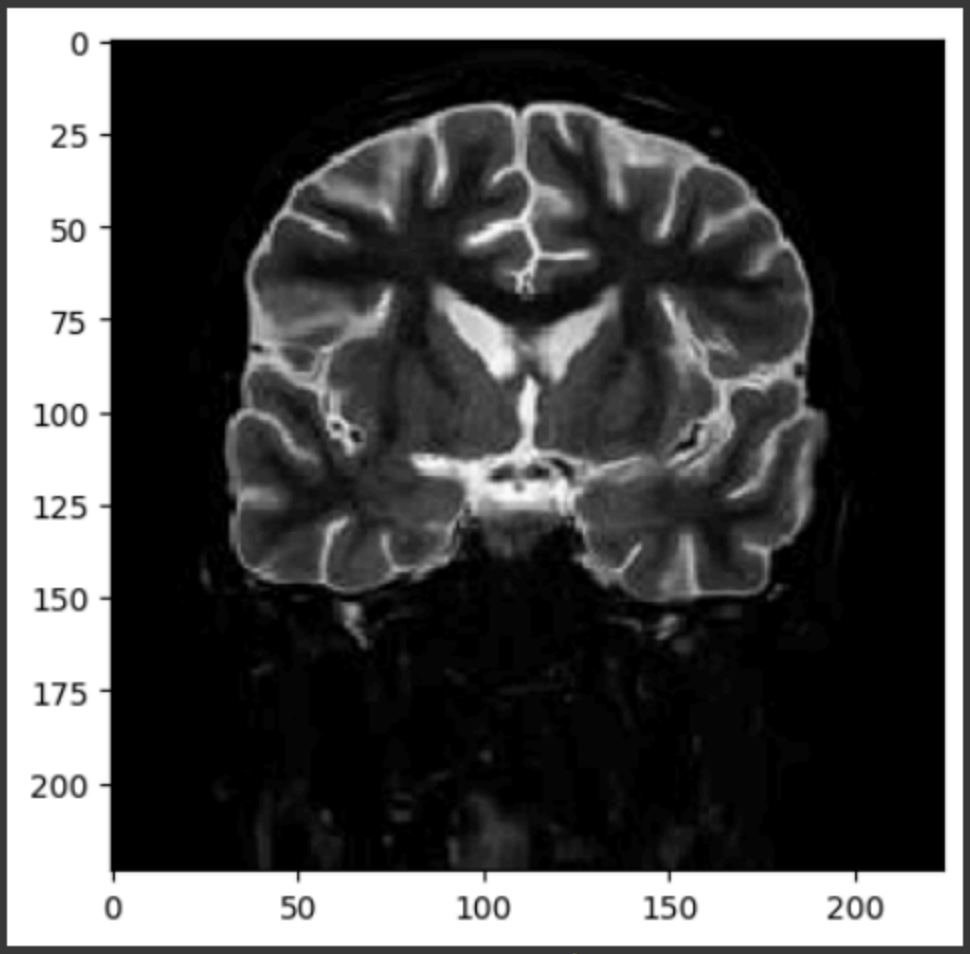
SVM and Logistic Regression can be effective for smaller datasets and for handling the accuracy of image classification , these models are not helpful.

The final rsults of SVM model classification for brain tumor detection are as follows:



The CNN model classifies the images as follows:

```
[ ]  if pred==0:
       print("the MRI is having a Tumor")
     else:
       print("the MRI is not having a Tumor")
```




```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step
the MRI is having a Tumor
```

CNN has better performance with high accuracy as this model learns by extracting fetaures from images and analyze the small details.