Brain Tumor Project

- **Implemented By Using DeepLearning**
  - **Keras**
  - **Tensorflow**

In [2]:
```
## Required Libraries For the model
import numpy as np
import matplotlib.pyplot as plt
import keras
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Flatten,MaxPool2D,Dropout, BatchNormaliza
from tensorflow.keras.preprocessing.image import load_img, img_to_array, array_to_i
import os
from pathlib import Path
import warnings
warnings.filterwarnings('ignore')
```

In [3]:
```
## load the data
data_dir = ('/content/drive/MyDrive/tumor')
number_of_img=['jpeg','jpg','bmp','png']
```

In [4]:
```
## total images in tumor and no tumor

tumor_len = len(os.listdir(os.path.join(data_dir,"meningioma")))
notumor_len = len(os.listdir(os.path.join(data_dir,"notumor")))
print("Total Tumor Images: ", tumor_len)
print("Total No_Tumor Images: ", notumor_len)
```

```
Total Tumor Images:  1339
Total No_Tumor Images:  1595
```

In [5]:
```
for image_class in os.listdir(data_dir):   ## it reads two files that is meningioma
    for image in os.listdir(os.path.join(data_dir, image_class)):  ## reads the all
        image_path = os.path.join(data_dir, image_class, image)
        try :
            img = cv2.imread(image_path)  ## cv2 reads and loads the image specifie
            tip = imghdr.what(image_path) ## determine type image inthe byte stream
            if tip not in image_exts:
                print("Image not in exists list {}".format(image_path))
                os.remove(image_path)
        except Exception as e:
            print("Issue with image {}".format(image_path))
```

```
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0421.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0839.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0549.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0745.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0746.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0615.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0890.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0758.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0551.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0359.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0666.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0420.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0687.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_1049.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_1014.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0904.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0524.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0532.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0450.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0871.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0678.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0413.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0691.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0571.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0489.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0426.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0641.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0742.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0773.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0546.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0574.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0432.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0792.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0525.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0672.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0673.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0619.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0575.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0387.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0514.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0964.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0739.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_1015.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0941.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0608.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0808.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0750.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0771.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0961.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0376.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0522.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0412.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0561.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0475.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0373.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_1023.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0604.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0399.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0444.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0989.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0892.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0898.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_1041.jpg
Issue with image /content/drive/MyDrive/tumor/notumor/Tr-no_0732.jpg
```

```
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0079.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0523.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0074.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0430.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0187.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0314.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0096.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0470.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0260.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0142.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0310.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0305.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0439.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0274.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0341.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0238.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0290.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0177.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0205.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0366.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0249.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0426.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0407.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0283.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0472.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0376.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0434.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0143.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0478.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0174.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0071.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0409.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0363.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0183.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0095.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0110.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0240.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0242.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0119.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0521.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0386.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0092.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0316.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0137.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0418.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0329.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0300.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0206.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0351.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0121.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0464.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0417.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0517.jpg
Issue with image /content/drive/MyDrive/tumor/meningioma/Tr-me_0236.jpg
```

In [6]:

```
data = tf.keras.utils.image_dataset_from_directory(directory='/content/drive/MyDriv
                                                   image_size=(224,224),
                                                   batch_size=32,
                                                   label_mode='binary')

## scale data in rgb of 255
data_scale = data.map(lambda x,y: (x/255, y))
data_iterator = data_scale.as_numpy_iterator()
batch  = data_iterator.next()
```

```
print("Image shape: ",batch[0].shape)
print("Labels shape: ",batch[1].shape)
```

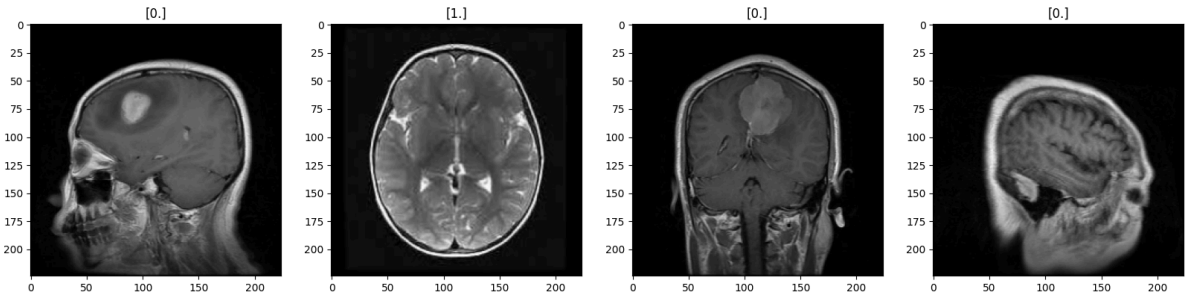
```
Found 2934 files belonging to 2 classes.
Image shape:  (32, 224, 224, 3)
Labels shape:  (32, 1)
```

Split Data Train, Validation, and Test

In [7]:
```
train = data = tf.keras.utils.image_dataset_from_directory(directory='/content/driv
                                                           image_size=(224,224),
                                                           batch_size=32,
                                                           label_mode='binary',
                                                           seed=123,
                                                           validation_split=0.2,
                                                           subset='training',
                                                           shuffle=True)
train_scale = train.map(lambda x,y: (x/255, y))
train_iterator = train_scale.as_numpy_iterator()
batch  = train_iterator.next()
print("Image shape: ",batch[0].shape)
print("Labels shape: ",batch[1].shape)
```

```
Found 2934 files belonging to 2 classes.
Using 2348 files for training.
Image shape:  (32, 224, 224, 3)
Labels shape:  (32, 1)
```

In [8]:
```
fig, ax = plt.subplots(ncols=4, figsize=(20,20))
for idx, img in enumerate(batch[0][:4]):
  ax[idx].imshow(img)
  ax[idx].title.set_text(batch[1][idx])
```



- 0 = Brain Tumor
- 1 = No Tumor

In [9]:
```
val = tf.keras.utils.image_dataset_from_directory(directory="/content/drive/MyDrive
                                                  validation_split=0.15,
                                                  batch_size=32,
                                                  image_size=(224,224),
                                                  seed=123,
                                                  subset='validation',
                                                  label_mode='binary')
val_scale = val.map(lambda x,y: (x/255,y))
val_iter = val_scale.as_numpy_iterator()
batch = val_iter.next()
print("Image shape", batch[0].shape)
print("Labels shape", batch[1].shape)
```

```
Found 2934 files belonging to 2 classes.
Using 440 files for validation.
Image shape (32, 224, 224, 3)
Labels shape (32, 1)
```

In [10]:
```
test = int(len(data)*0.15)
test = data.take(test)
```

Out[10]:
```
<_TakeDataset element_spec=(TensorSpec(shape=(None, 224, 224, 3), dtype=tf.float3
2, name=None), TensorSpec(shape=(None, 1), dtype=tf.float32, name=None))>
```

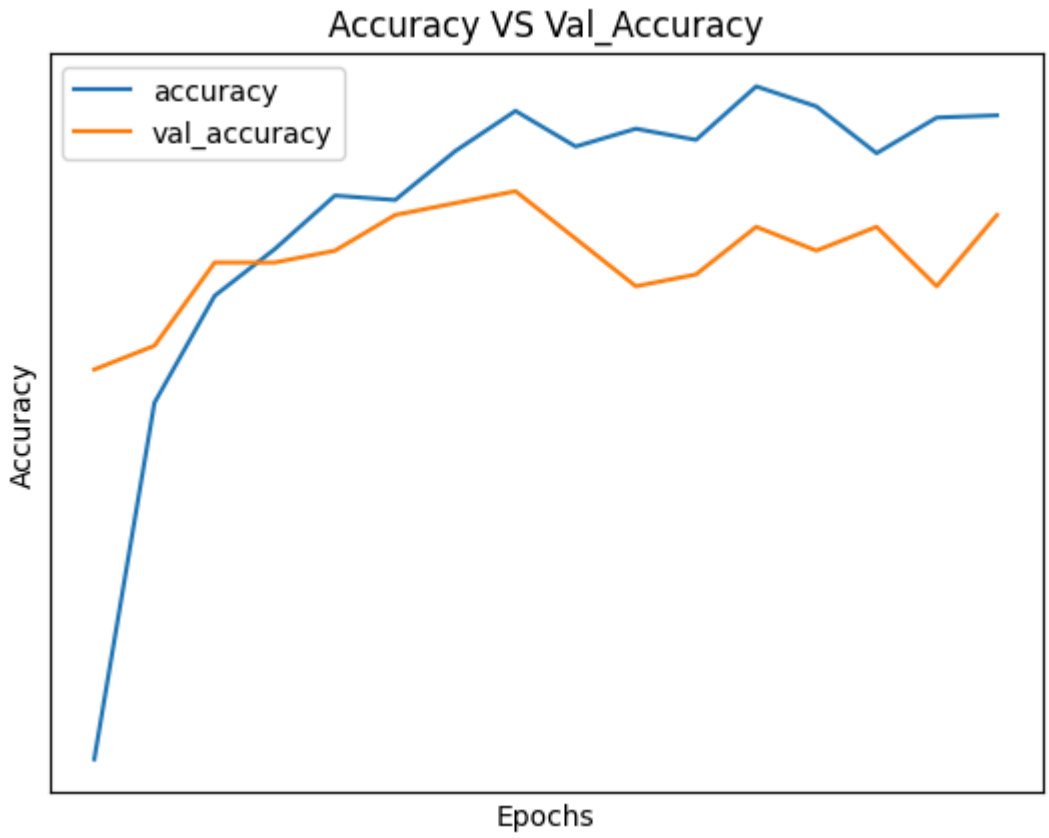
Model Building

In [12]:
```
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
model_earlystop = EarlyStopping(monitor='accuracy', min_delta=0.01, patience=8, ver
## Feature Extraction In CNN
## froming model into Sequential
model = Sequential()

## Now adding model required filters, kernel_size, activation, input_shape
## input layers
model.add(Conv2D(filters=30, kernel_size=(3,3), activation='relu', input_shape=(224

## Hidden layer
model.add(Conv2D(filters=50, kernel_size=(3,3), activation='relu')) ## hidden layer
model.add(MaxPool2D(pool_size=(2,2))) ## MaxPool2D which down samples the input alo

model.add(Conv2D(filters=80, kernel_size=(3,3), activation='relu')) ## hidden layer
model.add(MaxPool2D(pool_size=(2,2)))

model.add(Conv2D(filters=120, kernel_size=(3,3), activation='relu')) ## hidden laye
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.3))

model.add(Flatten()) ##convert the 2D array into 1D array

## classification and Probabilistic Distribution or output layers
model.add(Dense(units=40, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(units=1, activation='sigmoid'))

## model.compile
model.compile(optimizer='adam', loss=tf.losses.BinaryCrossentropy(), metrics=['accu
hist = model.fit(train, epochs=30, validation_data = val, validation_split=0.2, ini
```

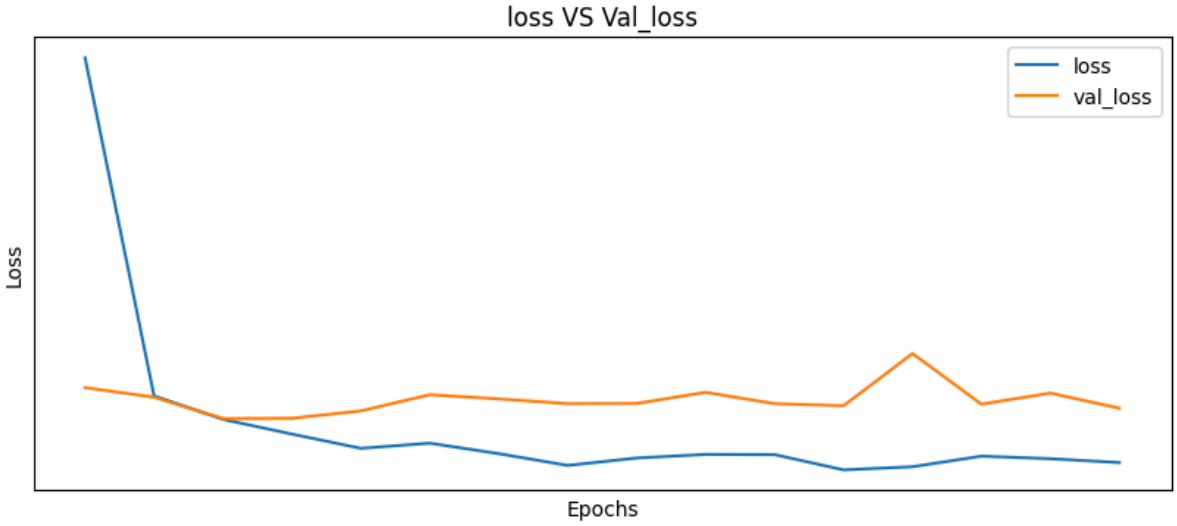
```
Epoch 4/30
74/74 [==============================] - 32s 255ms/step - loss: 0.9532 - accuracy:
0.8641 - val_loss: 0.2072 - val_accuracy: 0.9386
Epoch 5/30
74/74 [==============================] - 17s 214ms/step - loss: 0.1886 - accuracy:
0.9323 - val_loss: 0.1857 - val_accuracy: 0.9432
Epoch 6/30
74/74 [==============================] - 16s 200ms/step - loss: 0.1355 - accuracy:
0.9527 - val_loss: 0.1372 - val_accuracy: 0.9591
Epoch 7/30
74/74 [==============================] - 16s 201ms/step - loss: 0.1022 - accuracy:
0.9617 - val_loss: 0.1379 - val_accuracy: 0.9591
Epoch 8/30
74/74 [==============================] - 16s 203ms/step - loss: 0.0699 - accuracy:
0.9719 - val_loss: 0.1546 - val_accuracy: 0.9614
Epoch 9/30
74/74 [==============================] - 16s 204ms/step - loss: 0.0815 - accuracy:
0.9710 - val_loss: 0.1913 - val_accuracy: 0.9682
Epoch 10/30
74/74 [==============================] - 16s 207ms/step - loss: 0.0578 - accuracy:
0.9804 - val_loss: 0.1818 - val_accuracy: 0.9705
Epoch 11/30
74/74 [==============================] - 16s 204ms/step - loss: 0.0313 - accuracy:
0.9881 - val_loss: 0.1710 - val_accuracy: 0.9727
Epoch 12/30
74/74 [==============================] - 16s 198ms/step - loss: 0.0480 - accuracy:
0.9813 - val_loss: 0.1714 - val_accuracy: 0.9636
Epoch 13/30
74/74 [==============================] - 16s 199ms/step - loss: 0.0562 - accuracy:
0.9847 - val_loss: 0.1967 - val_accuracy: 0.9545
Epoch 14/30
74/74 [==============================] - 16s 199ms/step - loss: 0.0557 - accuracy:
0.9825 - val_loss: 0.1709 - val_accuracy: 0.9568
Epoch 15/30
74/74 [==============================] - 15s 195ms/step - loss: 0.0211 - accuracy:
0.9928 - val_loss: 0.1664 - val_accuracy: 0.9659
Epoch 16/30
74/74 [==============================] - 15s 196ms/step - loss: 0.0283 - accuracy:
0.9889 - val_loss: 0.2844 - val_accuracy: 0.9614
Epoch 17/30
74/74 [==============================] - 15s 197ms/step - loss: 0.0522 - accuracy:
0.9800 - val_loss: 0.1699 - val_accuracy: 0.9659
Epoch 18/30
74/74 [==============================] - 15s 195ms/step - loss: 0.0464 - accuracy:
0.9868 - val_loss: 0.1949 - val_accuracy: 0.9545
Epoch 19/30
74/74 [==============================] - 15s 196ms/step - loss: 0.0377 - accuracy:
0.9872 - val_loss: 0.1607 - val_accuracy: 0.9682
Epoch 19: early stopping
```

In [16]:

```
## plot accuracy and val_accuracy
fig = plt.figure()
plt.plot(hist.history['accuracy'], label='accuracy')
plt.plot(hist.history['val_accuracy'], label='val_accuracy')
plt.title("Accuracy VS Val_Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.xticks([])
plt.yticks([])
#plt.ylim([0.5,1.0])
plt.legend()
plt.show()
```



In [20]:
```
## plot loss and val_loss
plt.figure(figsize=(10,4))
plt.plot(hist.history['loss'], label='loss')
plt.plot(hist.history['val_loss'], label='val_loss')
plt.title("loss VS Val_loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.xticks([])
plt.yticks([])
#plt.ylim()
plt.legend()
plt.show()
```



Model Tuner

In [22]:
```
pip install keras-tuner
```

```
Collecting keras-tuner
  Downloading keras_tuner-1.4.6-py3-none-any.whl (128 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 128.9/128.9 kB 4.4 MB/s eta 0:00:00
Requirement already satisfied: keras in /usr/local/lib/python3.10/dist-packages (f
rom keras-tuner) (2.15.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-package
s (from keras-tuner) (23.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages
(from keras-tuner) (2.31.0)
Collecting kt-legacy (from keras-tuner)
  Downloading kt_legacy-1.0.5-py3-none-any.whl (9.6 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.
10/dist-packages (from requests->keras-tuner) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-pack
ages (from requests->keras-tuner) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dis
t-packages (from requests->keras-tuner) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dis
t-packages (from requests->keras-tuner) (2024.2.2)
Installing collected packages: kt-legacy, keras-tuner
Successfully installed keras-tuner-1.4.6 kt-legacy-1.0.5
```

In [23]:

```
## tuner library
import keras_tuner
import keras
from kerastuner import RandomSearch
```

In [24]:

```
## function to create hyperparameter to find best model
def build_model2(hp):
  model = keras.Sequential(
      [keras.layers.Conv2D(
           filters = hp.Int('conv_1_filter', min_value=15, max_value=80, step=16),
           kernel_size = hp.Choice('conv_1_kernel', values=[2,5]),
           activation='relu',
           input_shape = (224,224,3)
       ),
       keras.layers.Conv2D(
           filters = hp.Int('conv_2_filter', min_value=30, max_value=100, step=16),
           kernel_size = hp.Choice('conv_2_kernel', values=[2,5]),
           activation='relu'
       ),
       keras.layers.MaxPool2D(pool_size = (2,2)),
       keras.layers.Dropout(rate = hp.Float('dropout', 0.3,0.5)),
       keras.layers.Conv2D(
           filters=hp.Int('conv_3_filter', min_value=80, max_value=200, step=32),
           kernel_size=hp.Choice('conv_3_kernel', values=[2,5]),
           activation='relu'
       ),
       keras.layers.MaxPool2D(pool_size = (2,2)),
       keras.layers.Dropout(rate = hp.Float('dropout', 0.3,0.5)),
       keras.layers.Flatten(),
       keras.layers.Dense(
           units = hp.Int('Dense_units', min_value=80, max_value=200, step=32),
           activation='relu'
       ),
       keras.layers.Dropout(rate = hp.Float('dropout', 0.3,0.5)),
       keras.layers.Dense(units=1, activation='sigmoid')])
  model.compile(optimizer=keras.optimizers.Adam(hp.Choice('learning_rate',values=[1
                loss = keras.losses.BinaryCrossentropy(),
                metrics=['accuracy'])
  return model
```

In [25]:
```
## using RandomSearch to select the best model using hypertuned model
tuner = RandomSearch(hypermodel=build_model2, objective='val_accuracy', max_trials=
## search the best model
tuner.search(train, epochs=5, validation_data=val, validation_split=0.2)
```

```
Trial 3 Complete [00h 02m 49s]
val_accuracy: 0.949999988079071

Best val_accuracy So Far: 0.949999988079071
Total elapsed time: 00h 07m 23s
```

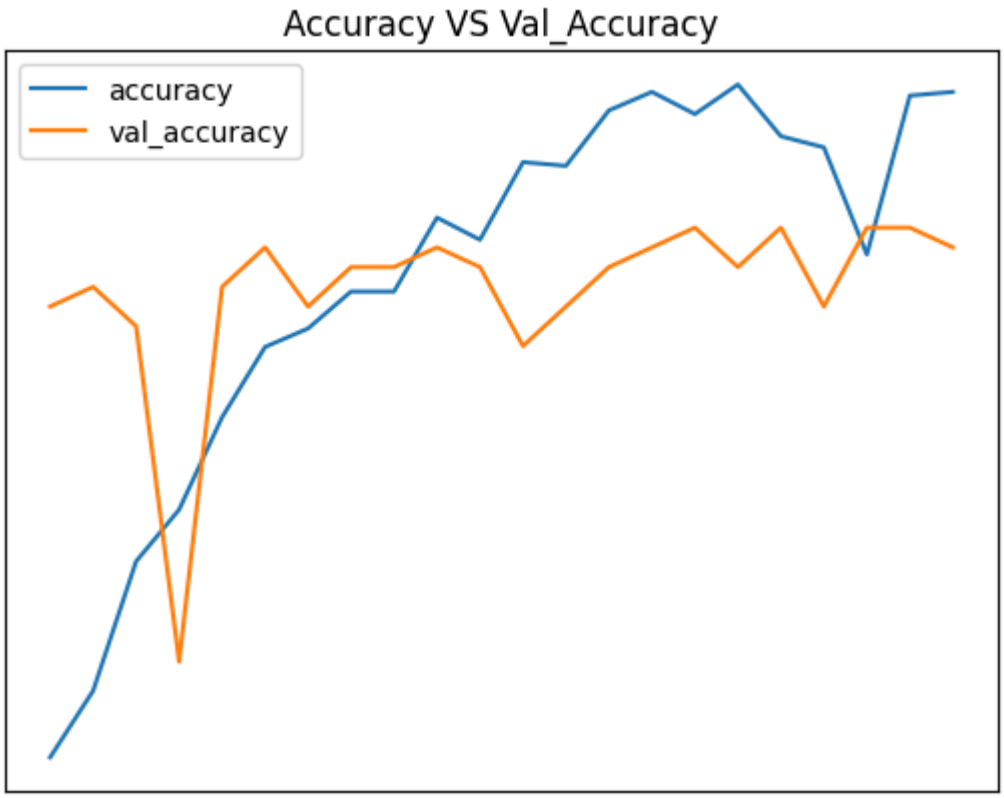
In [26]:
```
## select the best model in which we have tuned before
best_model = tuner.get_best_models()[0]
```

In [27]:
```
## Early Stopping if there is no improvement model accuracy model get stopped
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
model_earlystop = EarlyStopping(monitor='accuracy', min_delta=0.01, patience=8, ver

## model fit
model_train2 = best_model.fit(train, epochs=30, validation_data = val, validation_s
```

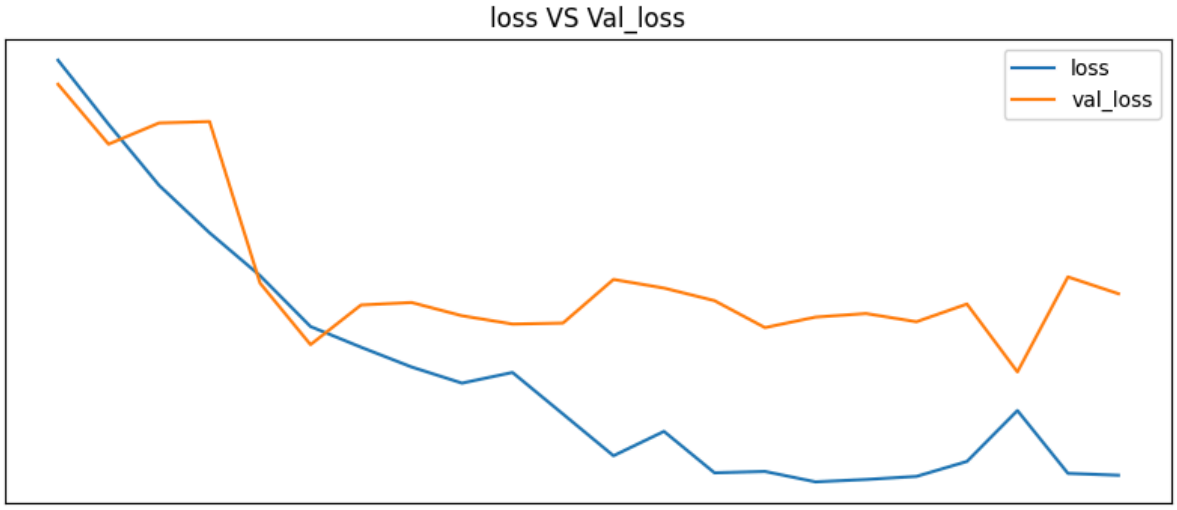
```
Epoch 4/30
74/74 [==============================] - 24s 274ms/step - loss: 0.2491 - accuracy:
0.9003 - val_loss: 0.2386 - val_accuracy: 0.9523
Epoch 5/30
74/74 [==============================] - 22s 280ms/step - loss: 0.2213 - accuracy:
0.9080 - val_loss: 0.2127 - val_accuracy: 0.9545
Epoch 6/30
74/74 [==============================] - 21s 276ms/step - loss: 0.1949 - accuracy:
0.9229 - val_loss: 0.2219 - val_accuracy: 0.9500
Epoch 7/30
74/74 [==============================] - 22s 293ms/step - loss: 0.1743 - accuracy:
0.9289 - val_loss: 0.2225 - val_accuracy: 0.9114
Epoch 8/30
74/74 [==============================] - 23s 296ms/step - loss: 0.1557 - accuracy:
0.9395 - val_loss: 0.1525 - val_accuracy: 0.9545
Epoch 9/30
74/74 [==============================] - 21s 274ms/step - loss: 0.1337 - accuracy:
0.9476 - val_loss: 0.1258 - val_accuracy: 0.9591
Epoch 10/30
74/74 [==============================] - 36s 479ms/step - loss: 0.1247 - accuracy:
0.9497 - val_loss: 0.1430 - val_accuracy: 0.9523
Epoch 11/30
74/74 [==============================] - 21s 277ms/step - loss: 0.1161 - accuracy:
0.9540 - val_loss: 0.1441 - val_accuracy: 0.9568
Epoch 12/30
74/74 [==============================] - 21s 270ms/step - loss: 0.1091 - accuracy:
0.9540 - val_loss: 0.1383 - val_accuracy: 0.9568
Epoch 13/30
74/74 [==============================] - 22s 282ms/step - loss: 0.1138 - accuracy:
0.9625 - val_loss: 0.1347 - val_accuracy: 0.9591
Epoch 14/30
74/74 [==============================] - 21s 274ms/step - loss: 0.0957 - accuracy:
0.9600 - val_loss: 0.1352 - val_accuracy: 0.9568
Epoch 15/30
74/74 [==============================] - 21s 273ms/step - loss: 0.0777 - accuracy:
0.9689 - val_loss: 0.1541 - val_accuracy: 0.9477
Epoch 16/30
74/74 [==============================] - 21s 268ms/step - loss: 0.0882 - accuracy:
0.9685 - val_loss: 0.1503 - val_accuracy: 0.9523
Epoch 17/30
74/74 [==============================] - 22s 281ms/step - loss: 0.0703 - accuracy:
0.9749 - val_loss: 0.1449 - val_accuracy: 0.9568
Epoch 18/30
74/74 [==============================] - 21s 270ms/step - loss: 0.0708 - accuracy:
0.9770 - val_loss: 0.1332 - val_accuracy: 0.9591
Epoch 19/30
74/74 [==============================] - 21s 279ms/step - loss: 0.0663 - accuracy:
0.9744 - val_loss: 0.1377 - val_accuracy: 0.9614
Epoch 20/30
74/74 [==============================] - 21s 272ms/step - loss: 0.0674 - accuracy:
0.9779 - val_loss: 0.1393 - val_accuracy: 0.9568
Epoch 21/30
74/74 [==============================] - 21s 277ms/step - loss: 0.0687 - accuracy:
0.9719 - val_loss: 0.1357 - val_accuracy: 0.9614
Epoch 22/30
74/74 [==============================] - 21s 270ms/step - loss: 0.0752 - accuracy:
0.9706 - val_loss: 0.1434 - val_accuracy: 0.9523
Epoch 23/30
74/74 [==============================] - 22s 282ms/step - loss: 0.0972 - accuracy:
0.9583 - val_loss: 0.1140 - val_accuracy: 0.9614
Epoch 24/30
74/74 [==============================] - 21s 272ms/step - loss: 0.0700 - accuracy:
0.9766 - val_loss: 0.1552 - val_accuracy: 0.9614
Epoch 25/30
```

```
74/74 [==============================] - 21s 272ms/step - loss: 0.0692 - accuracy:
0.9770 - val_loss: 0.1479 - val_accuracy: 0.9591
Epoch 25: early stopping
```

In [28]:
```
## check the accuracy and val_accuracy using hypertuned model
fig = plt.figure()
plt.plot(model_train2.history['accuracy'], label='accuracy')
plt.plot(model_train2.history['val_accuracy'], label='val_accuracy')
plt.title("Accuracy VS Val_Accuracy")
plt.xticks([])
plt.yticks([])
plt.legend()
plt.show()
```



In [29]:
```
## check the loss and val_loss using hypertuned model
plt.figure(figsize=(10,4))
plt.plot(model_train2.history['loss'], label='loss')
plt.plot(model_train2.history['val_loss'], label='val_loss')
plt.title("loss VS Val_loss")
plt.xticks([])
plt.yticks([])
plt.legend()
plt.show()
```



- **Comparing To Previous Plot Where We have took Manual Parameters Looking Into the plot we are getting Overfit**
- **After Selecting HyperTuned Model Parameters we are getting Good Plot**

In [30]:
```
## model accuracy using test data
from keras.models import load_model
from tensorflow.keras.metrics import Precision, Recall, BinaryAccuracy
precision = Precision()
recall = Recall()
BinaryAcc = BinaryAccuracy()
```

In [55]:
```
##check accuracy
for batch in test.as_numpy_iterator():
  X,y = batch
  yhat = best_model.predict(X)
  precision.update_state(y, yhat)
  recall.update_state(y, yhat)
  BinaryAcc.update_state(y,yhat)
print(precision.result(),"\n",recall.result(),"\n",BinaryAcc.result())
```

```
1/1 [==============================] - 0s 90ms/step
1/1 [==============================] - 0s 59ms/step
1/1 [==============================] - 0s 56ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 35ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 29ms/step
1/1 [==============================] - 0s 55ms/step
1/1 [==============================] - 0s 36ms/step
tf.Tensor(1.0, shape=(), dtype=float32)
 tf.Tensor(1.0, shape=(), dtype=float32)
 tf.Tensor(1.0, shape=(), dtype=float32)
```

In [51]:
```
import cv2
path = "/content/drive/MyDrive/tumor/meningioma/Tr-me_0022.jpg"
img = cv2.imread(path)
resize = tf.image.resize(img, (224,224))
plt.imshow(img)
input_expand = np.expand_dims(resize/255, axis=0)
pred = best_model.predict(input_expand)[0][0]
if pred>=0.5:
  print("No Tumor")
```

```
else:
  print("Brain Tumor")
```

```
1/1 [==============================] - 0s 111ms/step
Brain Tumor
```