

PHUSE US Connect 2021

Application of Machine Learning (ML) in Tumor Detection in Oncology Study

ANIMESH PATEL

Data Analytics Consultant

animesh9719@gmail.com



Contents



Introduction /
Motivation



Dataset



Image
Processing



Approach



Result



Conclusion

Introduction / Motivation

- According to WHO Cancer is the second leading cause of death globally.
- Early detection significantly improves the chances of survival and prognosis.
- Brain tumor is defined as collection of cells that grow in an abnormal way.
- Brain tumor is affecting the central nervous system.
- Most common and popular method to identify tumor type - magnetic resonance imaging(MRI)
- Radiologist is needed to confirm progression/presence of tumor or not
- Sometime independent assessor and adjudication committee is present

Introduction / Motivation

- Brain tumor detection is complex task.
- Large amount of noise data are difficult for human to handle.
- That is the motivation for us to develop CNN model that help and save times of radiologist.
- Goal: Along with one radiologist, this model can serve the purpose of confirming results i.e. can full fill the need of 2nd radiologist
- Can serve the purpose of adjudication

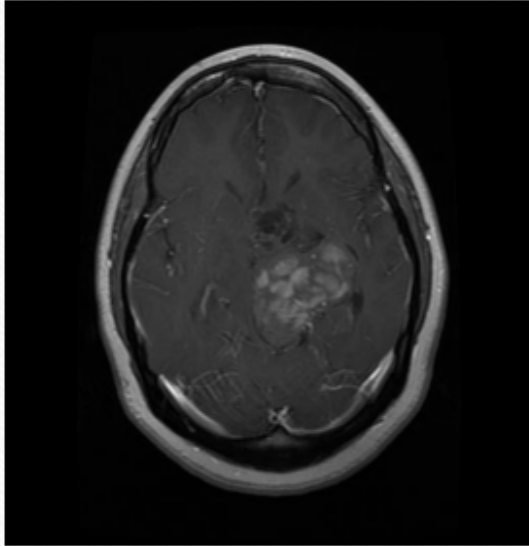
Setting up

- Types of tumors patients MRI data.
 - Glioma
 - Meningioma
 - Pituitary
- The work is divided into 3 parts:
 1. Preprocessing
 2. Create CNN model
 3. Train and Test developed model

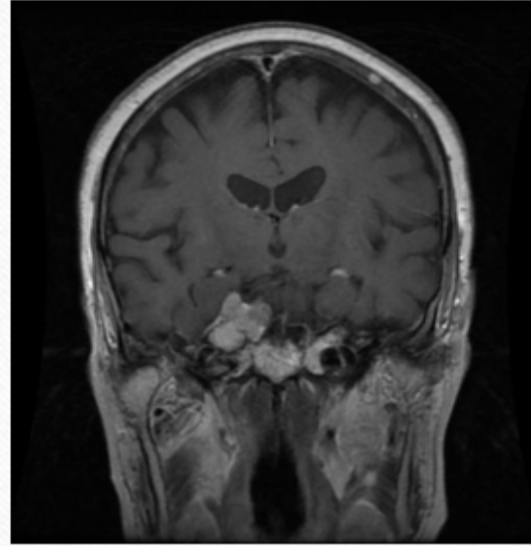
Data Description

- Our tumor dataset contain
 - 1. Three types of brain tumor patients' MRI data.
 - 2. Normal brain MRI data
- Around 3000 MRI images are used to develop our model.
- Dataset is publicly available on an internet.
https://figshare.com/articles/dataset/brain_tumor_dataset/1512427
<https://github.com/SartajBhuvaji/Brain-Tumor-Classification-DataSet>
- All the images have different size so here data preprocessing comes.

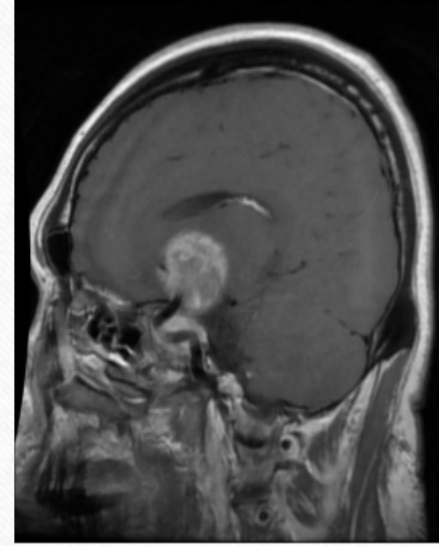
Dataset



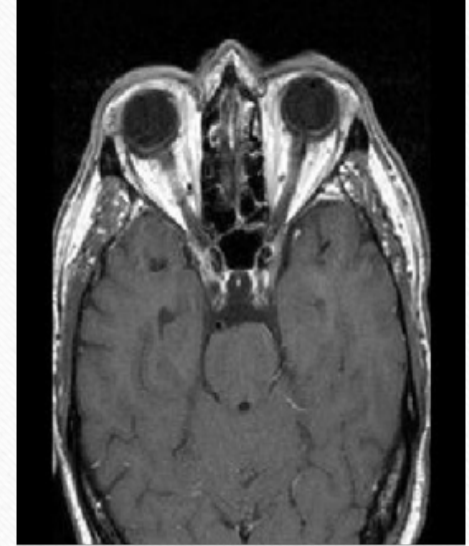
Glioma Tumor



Meningioma Tumor

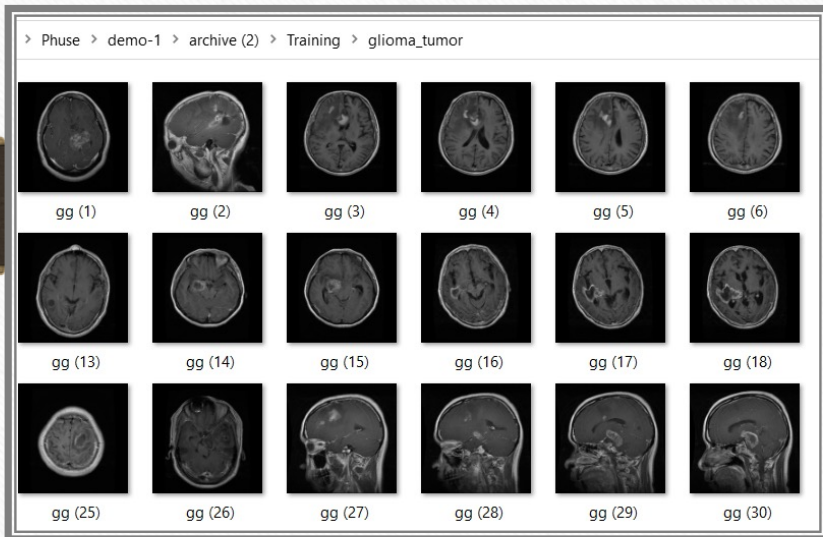


Pituitary Tumor



No Tumor

Data Preprocessing



- We use images data.
- Many preprocessing steps are performed before applying ML model.
- Pre-processing step:
 1. Resize each image to same size for fast processing.
 2. Convert each image to gray scale and store in array.
 3. Normalized Pixel Values.
 4. Split data into Training and Testing dataset.

Resize each image to same size

- To resize an image, OpenCV provides `cv2.resize()` function.
- CNN architecture has many layers which depends on input size therefor resizing is important part.

```
#Resize each Image
```

```
img_size = 128
```

```
image_array = cv2.resize(img_array, (img_size,img_size))
```

Convert each image to gray scale

```
: #Convert Image to Grayscale
```

```
train_data = []
for i in categories:
    train_path = os.path.join(data_dir,i)
    tag = categories.index(i)
    for img in os.listdir(train_path):
        try:
            image_arr = cv2.imread(os.path.join(train_path , img), cv2.IMREAD_GRAYSCALE)
            new_image_array = cv2.resize(image_arr, (img_size,img_size))
            train_data.append([new_image_array , tag])
        except Exception as e:
            pass
```

```
: print(train_data)
```

```
...,
[1, 2, 1, ..., 3, 2, 2],
[0, 0, 0, ..., 3, 2, 2],
[0, 0, 0, ..., 1, 0, 0]], dtype=uint8), 0], [array([[0, 0, 0, ..., 0, 0, 0],
[0, 0, 0, ..., 2, 2, 2],
[0, 0, 1, ..., 3, 2, 2],
...,
```

Normalized Pixel Values

```
#normalize image pixel value.
```

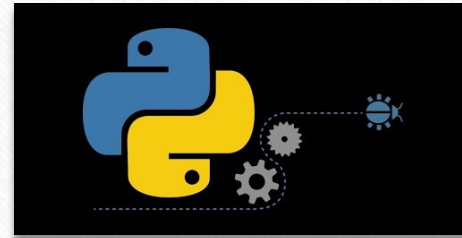
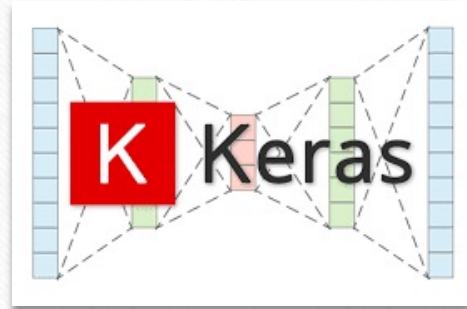
```
X = X/255.0
```

```
X = X.reshape(-1,128,128,1)
```

```
(2870, 128, 128)
```

- Pixel values of image data between 0 and 255 this is not ideal to take large value in neural network.
- So here we normalized pixel value in range of 0 to 1.

Approach



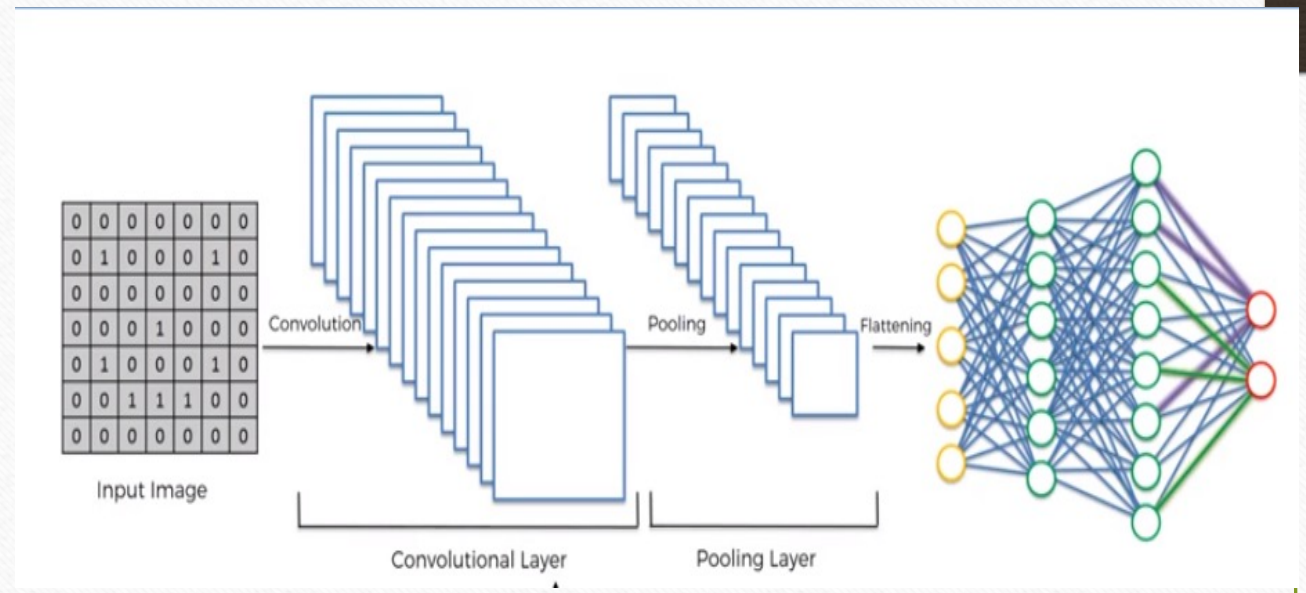
- We used Python programming language with using Keras and TensorFlow packages. Implement our model on Jupyter notebook.
- By using Convolutional Neural Network (CNN) we performed MRI-image detection.

Convolutional Neural Network(CNN)

- Convolutional neural network (CNN) is one of the main categories to do images recognition, images classifications, Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

CNN Building Blocks

1. Convolutional Layer
2. Pooling layer
3. ReLU Layer
4. Fully Connected Layer



Our CNN Model

```
model = Sequential()

model.add(Conv2D(filters = 64, kernel_size = (5,5),padding = 'Same', activation = 'relu', input_shape = (128,128,1)))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.2))
model.add(Conv2D(filters = 128, kernel_size = (3,3),padding = 'Same', activation = 'relu'))
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))
model.add(Dropout(0.2))
model.add(Conv2D(filters = 128, kernel_size = (3,3),padding = 'Same', activation = 'relu'))
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))
model.add(Dropout(0.2))
model.add(Conv2D(filters = 128, kernel_size = (2,2),padding = 'Same', activation = 'relu'))
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))
model.add(Dropout(0.2))
model.add(Conv2D(filters = 256, kernel_size = (2,2),padding = 'Same', activation = 'relu'))
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))
model.add(Dropout(0.2))

model.add(Flatten())
model.add(Dense(1024, activation = "relu"))
model.add(Dropout(0.5))

model.add(Dense(4, activation = "softmax"))
optimizer = Adam(lr=0.001, beta_1=0.9, beta_2=0.999)
model.compile(optimizer = optimizer , loss = "categorical_crossentropy", metrics=["accuracy"])
```

Result

Training Accuracy

```
score = model.evaluate(X_train, y_train, verbose = 0)
score[1]
```

0.9750430583953857

Testing Accuracy

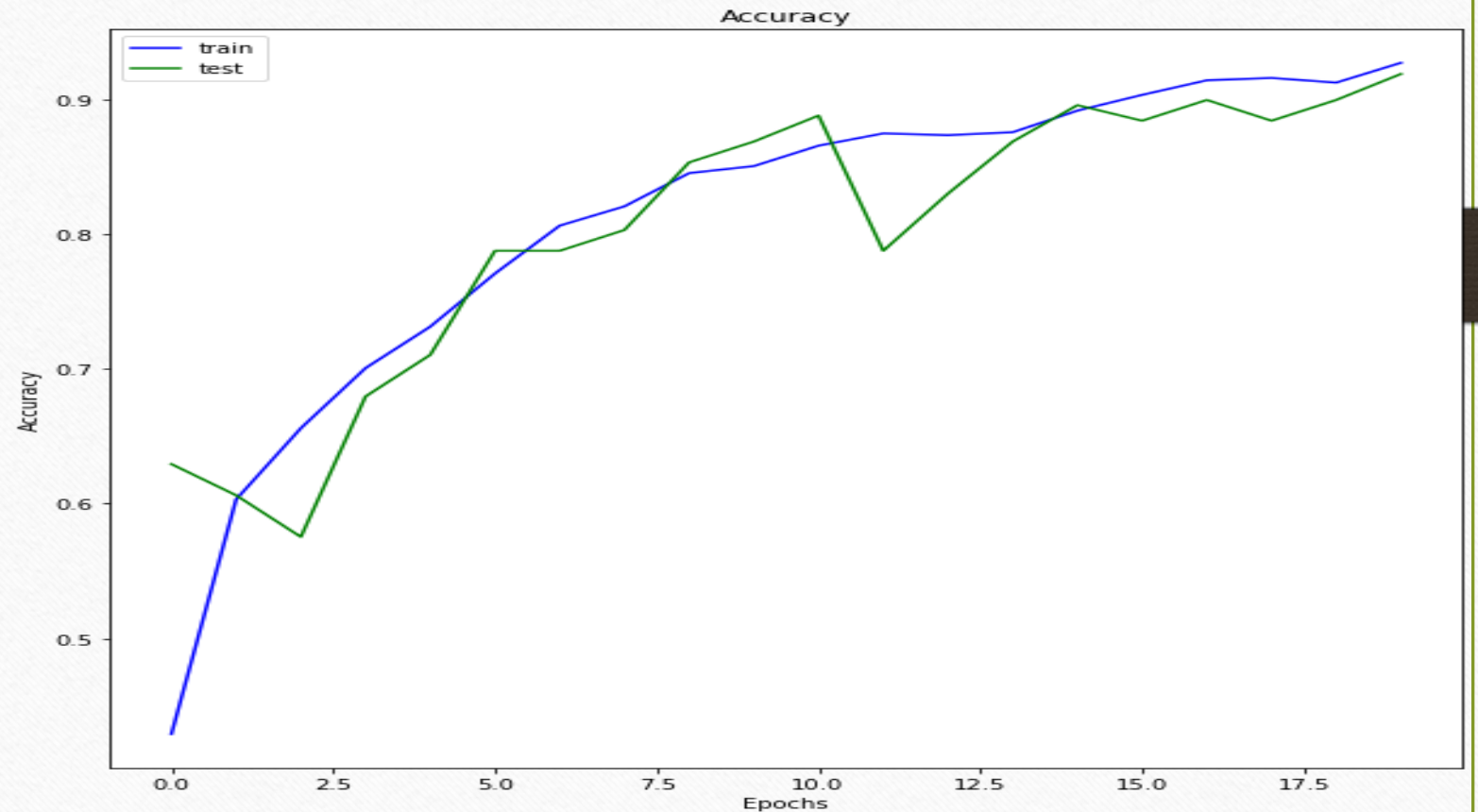
```
score = model.evaluate(X_test, y_test, verbose = 0)
score[1]
```

0.9163762927055359

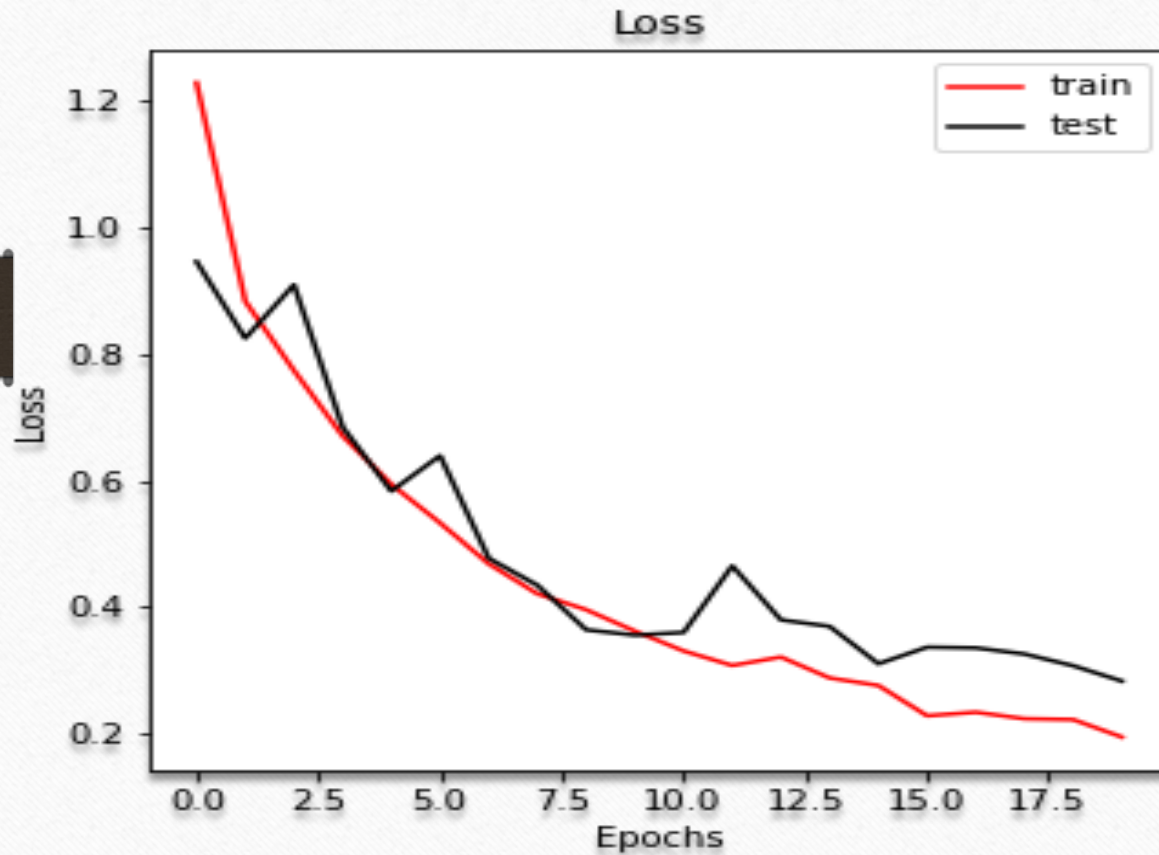
- We run all the MRI images for 20 times, and we got highest training and testing accuracy. We also investigate a training and testing loss to check whether our model goes to underfit or overfit.

Result

- Here we see the number of epoch are increased the Training and testing accuracy also increased.



Result



- On other side training and testing Loss are decreased while number of epoch are increased. That is the sign of our model is give us an optimal result.

Conclusion

- Detection of brain tumors has a very significant role to play in medical interventions.
- We are able to create and train model that gave us very high accuracy. This model can be taken and developed to specific need of company.
- With the using of this automated detection model, it will help to save radiologist time and saving cost in clinical research.
- Can be good option for adjudication (3rd Radiologist) in oncology clinical trial.

Thank You...

Questions.?

Questions

- 1. Why you used CNN.?
- 2. Does this model detect which type of brain tumor.?
- 3. What is Epoch.?
- 4. If we train model more number of times did, we get better result.?
- 5. Can we used another images for testing.?