Interim Project Report- CSE 4020-Fall 2021

December 18, 2021

1 Team members

Divyanu Baheti - 19BCE1045
Apoorv Yadav - 19BCE1163

2 Code of academic integrity

I affirm that:

- This work is my own original work and is not a borrowed work, either from other students or from assignments for other courses.
- I have not given or received any unauthorized help on this assignment.
- This submission is free from:
    1. Plagiarism
    2. Fabrication of facts
    3. Unauthorized assistance
    4. Collusion
- This submission gives proper credit to sources and references, acknowledges the contributions and ideas of others relevant to this academic work.
- This submission gives proper credit to sources and references, acknowledges the contributions and ideas of others relevant to this academic work.

- This submission was prepared by me fully adhering to the rules that govern this assignment regarding resource material, electronic aids, copying, collaborating with others, or engaging in any other behaviour that subverts the purpose of the assignment and the directions of the teacher.

3 Problem statement of the project

Our objective in this project is to create an image classification model that can predict Brain MRI scans that belong to one of the four classes with a reasonably high accuracy. Our dataset has more than 3000 Brain MRI scans which are categorized in four classes - Glioma Tumor, Meningioma Tumor, Pituitary Tumor and No Tumor.

4 Motivation for choosing the problem

Brain Tumor is one of the deadliest disease any human can suffer from. Machine Learning bought a whole new dimension to the field of Disease-Detection.

Identification:
Many a times there has been various errors from doctors while giving reports be it like sometimes.

- The doctor makes a mistake and falsely dragonize a patient.
- The doctor fails to dragonize a patient.

4.1 Literature survey

Research work by 9 different authors has been discussed based on varied deep learning techniques and architectures adopted by them.

Sakshi Ahuja et al.,[1] used transfer learning and superpixel technique for detection of brain tumor and brain segmentation respectively. The dataset used was from BRATS 2019 brain tumor segmentation challenge and this model was trained on the VGG 19 transfer learning model. Using the superpixel technique the tumor was divided between LGG and HGG images. This resulted in an average of dice index of 0.934 in opposition to ground truth data.

Hajar Cherguif et al.,[2] used U-Net for the semantic segmentation of medical images. To develop a good convoluted 2D segmentation network, U-Net architecture was used. BRATS 2017 dataset was used for testing and evaluating the model proposed. The U-Net architecture proposed had 27 convolutional layers, 4 deconvolutional layers, Dice-coef of 0.81.

Chirodip Lodh Choudhury et al.,[3] made the use of deep learning techniques involving deep neural networks and also incorporated it with a Convolutional Neural Network model to get the accurate results of MRI scans. A 3-layer CNN architecture was proposed which was further connected to a fully Connected Neural Network. F-score equal to 97.33 and an accuracy equal to 96.05% was achieved.

Ahmad Habbie et al.,[4] MRI T1 weighted images were taken and using semi-automatic segmentation analyzed the possibility of a brain tumour using an active contour model. The performance of morphological active contour without edge, snake active contour and morphological geodesic active contour was analyzed. MGAC performed the best among all three as suggested by the data.

Neelum et al.,[5] used a concatenation approach for the deep learning model in this paper and the possibility of having a brain tumor was analyzed. Pre trained deep learning models which are Inception - v3 and DenseNet201 were used to detect and classify brain tumors. Inception - v3 model was pre trained to extract the features and these features were concatenated for tumor classification. Then, the classification part was done by a softmax classifier.

Ms. Swati Jayade et al.,[6] used Hybrid Classifiers. The classification of tumors was done into types, malignant and benign. Feature dataset here was prepared by Gray level Co-occurrence Matrix (GLCM) feature extraction method. A hybrid method of classifiers involving KNN and SVM classifiers was proposed to increase efficiency.

Zheshu Jia et al.,[7] the author made a fully automatic heterogeneous segmentation in which SVM (Support Vector Machine) was used. For training and checking the accuracy of tumor detection in MRI images, a classification known as probabilistic neural network classification sytem had been used. Multi spectral brain dataset is used and this model focused on the automated segmentation of meningioma.

DR. Akey Sungheetha, DR. Rajesh Sharma R.[8] used Gabor transform along with the soft and hard clustering for detecting the edges in the CT and MRI images. A total of 4500 and 3000 instances of MRI images and CT were used respectively. K-means clustering was used for the separation of similar features into sub-groups To represent the images in the form of histogram properties, the author used Fuzzy c means.

Parnian Afshar et al.,[9] used a bayesian approach for the classification of brain tumor using capsule networks. To improve the results of tumor detection, capsule network instead of CNN was used as CNN can loose the important spatial information. The team proposed BayesCap framework. To test the proposed model they used a benchmarkbrain tumor dataset.

Further, 19 other research papers regarding the latest research work done in this domain have been reviewed and tabulated in Table 1. Key details such as the dataset and techniques used, year of publication, major observations and accuracy achieved have been presented to aid further research work in this domain.

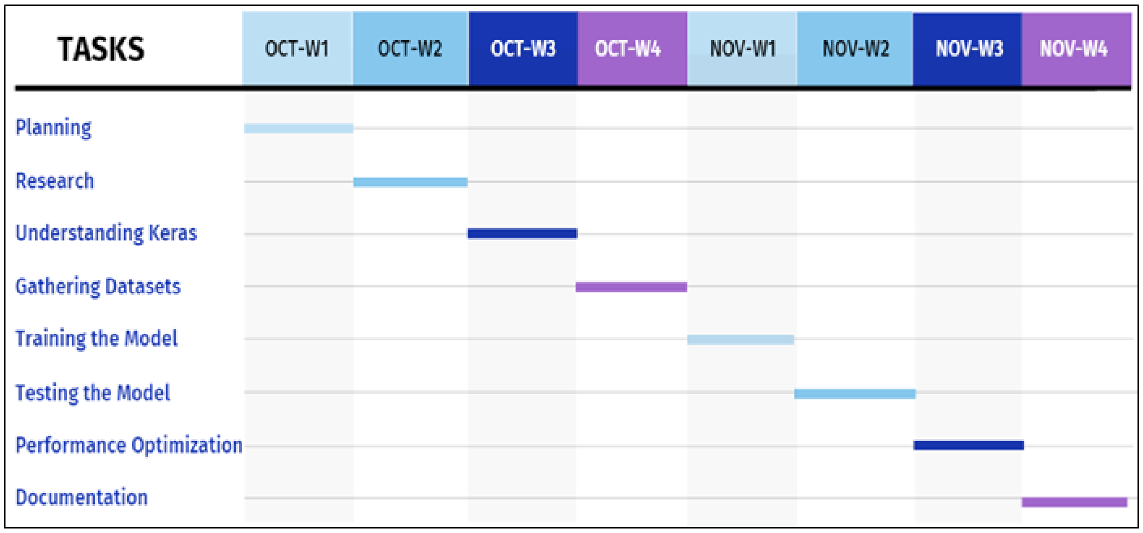
5 Methodology

The hands-on project on Brain Tumor Classification using Keras is divided into following tasks:

- Task 1: Research on Brain Tumor Disease its causes, symptoms and risk factors.

- Task 2: Getting the Dataset and Importing necessary Libraries e.g., Keras.
- Task 3: Creating Directory to separate and store Training and Testing Data
- Task 4: Data Visualization
- Task 5: Create a function to Crop Images
- Task 6: Save Pre-processed Data
- Task 7: Data Augmentation and Data Loader
- Task 8: Creating the Model
- Task 9: Model Training and Model Evaluation
- Task 10: Prediction on Test Data
- Task 11: Performance Optimization
- Task 12: Documentation

Gantt Chart:



5.1 Details of all experiments conducted

- **Image classification Using Keras**

- **Design of the experiment**
  1. Keras is a Machine Learning (ML) library built on top of TensorFlow, making it extremely easy to create and fit Deep Learning model architectures. As they like to say, "Keras API is designed for human beings, not machines."
  2. Keras can be very flexible and powerful; one can gradually peel the layers of complexity and dive deeper to customize almost every little detail.
  3. Moreover, Keras can take advantage of multi-GPU configurations or TPU training because of its TensorFlow-backed backend.
  4. The Keras design pattern to do image classification for brain tumor from scratch, starting from JPEG image files on disk, without leveraging pre-trained weights or a pre-made Keras Application model.
  5. We have used EfficientNetB1 for fine-tuning the classification.

- **Data sets**
  We have used the efficient net model and train it on Brain MRI dataset. This dataset has more than 3000 Brain MRI scans which are categorized in four classes:
  1. Glioma Tumor
  2. Meningioma Tumor
  3. Pituitary Tumor
  4. No Tumor
  The dataset link:
  `https://github.com/Apoorv-17/Brain-Tumor-Classification`

- **Pseudo-code**
  We will use an efficient net model and train it on a Brain MRI dataset. Our objective in this project is to create an image classification model that can predict Brain MRI scans that belong to one of the four classes with a reasonably high accuracy.

  1. START
  2. Importing and setup the necessary environment before hand to start with the task.

3. Loading the Dataset in raw format which has 2 folders namely Training and Testing. Both of which contains images from all the 4 types of Brain Tumors.
4. Filtering out the corrupted images which don't have string 'JFIF' in their header.
4. Next we do the data visualization on the train dataset.
5. We see that the Images have a lot of waste/black space which is not useful and could affect the performance of the model hence create a function which crops out all such spaces.
6. Now we create a new directory and save all the cropped training and testing images.
7. Next we do image data augmentation on the training images.
8. Next we finally build and compile the model 9. After that we do the model training with the epoch value as 7.
10. Next we perform the model and evaluation. 11. We then plot the model accuracy and model loss line graph for better inference on the performance.
12. At last do the model prediction on the test images.
13. END

- **Code**

```
#Importing the packages
import os
import random
from tqdm import tqdm
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import cv2
import tensorflow as tf
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.preprocessing.image import
   ↪ ImageDataGenerator
from tensorflow.keras.preprocessing.image import
```

```
    ↪ array_to_img
from tensorflow.keras.applications import EfficientNetB1

from tensorflow.keras.models import Model
from tensorflow.keras.layers import Flatten,Dense,Conv2D,
    ↪ Dropout,GlobalAveragePooling2D

from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ModelCheckpoint,
    ↪ EarlyStopping
import imutils


# Create Directory for Training Data
os.mkdir("/content/Crop-Brain-MRI")
os.mkdir("/content/Crop-Brain-MRI/glioma_tumor")
os.mkdir("/content/Crop-Brain-MRI/meningioma_tumor")
os.mkdir("/content/Crop-Brain-MRI/no_tumor")
os.mkdir("/content/Crop-Brain-MRI/pituitary_tumor")


# Create Directory for Testing Data
os.mkdir("/content/Test-Data")
os.mkdir("/content/Test-Data/glioma_tumor")
os.mkdir("/content/Test-Data/meningioma_tumor")
os.mkdir("/content/Test-Data/no_tumor")
os.mkdir("/content/Test-Data/pituitary_tumor")

\begin{figure}[htp]
    \centering
    \includegraphics[width=15cm]{gantt.png}
\end{figure}\\



# Data Visualizaion
train_dir = "/content/Brain-Tumor-Classification/Brain-MRI
    ↪ /Training/"
```

```
test_dir = "/content/Brain-Tumor-Classification/Brain-MRI/
    ↪ Testing/"
classes = os.listdir("/content/Brain-Tumor-Classification/
    ↪ Brain-MRI/Training")
files_path_dict = {}

for c in classes:
  files_path_dict[c] = list(map(lambda x : train_dir+c+'/'
      ↪ +x, os.listdir(train_dir+c)))


plt.figure(figsize=(17,17))
index = 0

for c in classes:
  random.shuffle(files_path_dict[c])
  path_list = files_path_dict[c][:5]

  for i in range(1, 5):
    index += 1
    plt.subplot(4, 4, index)
    plt.imshow(load_img(path_list[i]))
    plt.title(c)



# Create a Function to crop images
def crop_image(image, plot=False):

    img_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    img_gray = cv2.GaussianBlur(img_gray, (5, 5), 0)

    img_thresh = cv2.threshold(img_gray, 45, 255, cv2.
        ↪ THRESH_BINARY)[1]
    img_thresh = cv2.erode(img_thresh, None, iterations=2)
    img_thresh = cv2.dilate(img_thresh, None, iterations
        ↪ =2)
    contours = cv2.findContours(img_thresh.copy(), cv2.
        ↪ RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    contours = imutils.grab_contours(contours)
    c = max(contours, key=cv2.contourArea)

    extLeft = tuple(c[c[:, :, 0].argmin()][0])
    extRight = tuple(c[c[:, :, 0].argmax()][0])
    extTop = tuple(c[c[:, :, 1].argmin()][0])
    extBot = tuple(c[c[:, :, 1].argmax()][0])

    new_image = image[extTop[1]:extBot[1], extLeft[0]:
        ↪ extRight[0]]

    if plot:
        plt.figure()
        plt.subplot(1, 2, 1)
        plt.imshow(image)
        plt.tick_params(axis='both', which='both', top=
            ↪ False, bottom=False, left=False, right=False
            ↪ ,labelbottom=False, labeltop=False,
            ↪ labelleft=False, labelright=False)
        plt.title('Original␣Image')
        plt.subplot(1, 2, 2)
        plt.imshow(new_image)
        plt.tick_params(axis='both', which='both',top=False
            ↪ , bottom=False, left=False, right=False,
            ↪ labelbottom=False, labeltop=False, labelleft
            ↪ =False, labelright=False)
        plt.title('Cropped␣Image')
        plt.show()

    return new_image


# Saving the cropped Images
# Crop the Training Images and Save it to the Directory
    ↪ we previously cretaed
```

```
glioma = train_dir + "glioma_tumor"
meningioma = train_dir + "meningioma_tumor"
no_tumor = train_dir + "no_tumor"
pituitary = train_dir + "pituitary_tumor"

j = 0
for i in tqdm(os.listdir(glioma)):
  path = os.path.join(glioma,i)
  img = cv2.imread(path)
  img = crop_image(img, plot=False)
  if img is not None:
    img = cv2.resize(img, (224, 224))
    save_path = "/content/Crop-Brain-MRI/glioma_tumor/" +
        ↪ str(j) + ".jpg"
    cv2.imwrite(save_path, img)
    j = j+1

j = 0
for i in tqdm(os.listdir(meningioma)):
  path = os.path.join(meningioma,i)
  img = cv2.imread(path)
  img = crop_image(img, plot=False)
  if img is not None:
    img = cv2.resize(img, (224, 224))
    save_path = "/content/Crop-Brain-MRI/meningioma_tumor/
        ↪ " + str(j) + ".jpg"
    cv2.imwrite(save_path, img)
    j = j+1

j = 0
for i in tqdm(os.listdir(no_tumor)):
  path = os.path.join(no_tumor,i)
  img = cv2.imread(path)
  img = crop_image(img, plot=False)
  if img is not None:
    img = cv2.resize(img, (224, 224))
    save_path = "/content/Crop-Brain-MRI/no_tumor/" + str(
        ↪ j) + ".jpg"
    cv2.imwrite(save_path, img)
    j = j+1

j = 0
for i in tqdm(os.listdir(pituitary)):
  path = os.path.join(pituitary,i)
  img = cv2.imread(path)
  img = crop_image(img, plot=False)
  if img is not None:
    img = cv2.resize(img, (224, 224))
    save_path = "/content/Crop-Brain-MRI/pituitary_tumor/"
       ↪  + str(j) + ".jpg"
    cv2.imwrite(save_path, img)
    j = j+1



# Crop the Testing Images and Save it to the Directory we
   ↪  previously cretaed
test_glioma = test_dir + "glioma_tumor"
test_meningioma = test_dir + "meningioma_tumor"
test_no_tumor = test_dir + "no_tumor"
test_pituitary = test_dir + "pituitary_tumor"

j = 0
for i in tqdm(os.listdir(test_glioma)):
  path = os.path.join(test_glioma,i)
  img = cv2.imread(path)
  img = crop_image(img, plot=False)
  if img is not None:
    img = cv2.resize(img, (224, 224))
    save_path = "/content/Test-Data/glioma_tumor/" + str(j
       ↪ ) + ".jpg"
    cv2.imwrite(save_path, img)
    j = j+1

j = 0
for i in tqdm(os.listdir(test_meningioma)):
  path = os.path.join(test_meningioma,i)
  img = cv2.imread(path)
  img = crop_image(img, plot=False)
  if img is not None:
    img = cv2.resize(img, (224, 224))
    save_path = "/content/Test-Data/meningioma_tumor/" +
       ↪ str(j) + ".jpg"
    cv2.imwrite(save_path, img)
    j = j+1

j = 0
for i in tqdm(os.listdir(test_no_tumor)):
  path = os.path.join(test_no_tumor,i)
  img = cv2.imread(path)
  img = crop_image(img, plot=False)
  if img is not None:
    img = cv2.resize(img, (224, 224))
    save_path = "/content/Test-Data/no_tumor/" + str(j) +
       ↪ ".jpg"
    cv2.imwrite(save_path, img)
    j = j+1

j = 0
for i in tqdm(os.listdir(test_pituitary)):
  path = os.path.join(test_pituitary,i)
  img = cv2.imread(path)
  img = crop_image(img, plot=False)
  if img is not None:
    img = cv2.resize(img, (224, 224))
    save_path = "/content/Test-Data/pituitary_tumor/" +
       ↪ str(j) + ".jpg"
    cv2.imwrite(save_path, img)
    j = j+1

# Perform Data Augmentation and Prepare the Train,
   ↪ Validation and Test Dataset

# Use Image Data Generator to perform this task.
```

```
datagen = ImageDataGenerator(rotation_range=10,
    ↪ height_shift_range=0.2, horizontal_flip=True,
    ↪ validation_split=0.2)
train_data = datagen.flow_from_directory('/content/Crop-
    ↪ Brain-MRI/',
                                         target_size = (224,
                                             ↪ 224),
                                         batch_size= 32,
                                         class_mode= '
                                             ↪ categorical',
                                         subset= 'training')
valid_data = datagen.flow_from_directory('/content/Crop-
    ↪ Brain-MRI/',
                                         target_size = (224,
                                             ↪ 224),
                                         batch_size= 32,
                                         class_mode= '
                                             ↪ categorical',
                                         subset= 'validation')

test_datagen = ImageDataGenerator()

test_data = datagen.flow_from_directory('/content/Test-
    ↪ Data/',
                                         target_size = (224,
                                             ↪ 224),

                                         class_mode= '
                                             ↪ categorical'
                                        )

# View the class dictionary
print(train_data.class_indices)
print(test_data.class_indices)

# View the augmented data.
sample_x, sample_y = next(train_data)
plt.figure(figsize=(12,9))
```

```
for i in range(6):
  plt.subplot(2,3, i+1)
  sample = array_to_img(sample_x[i])
  plt.axis('off')
  plt.grid(False)
  plt.imshow(sample)
plt.show()

# Build and Compile the Model
# Model building
effnet = EfficientNetB1(weights="imagenet", include_top=
   ↪ False, input_shape=(224, 224,3))
model = effnet.output
model = GlobalAveragePooling2D()(model)
model = Dropout(0.5)(model)
model = Dense(4, activation="softmax")(model)
model = Model(inputs= effnet.input, outputs= model)

model.summary()

# Model Compiling
model.compile(optimizer=Adam(learning_rate=0.0001),loss="
   ↪ categorical_crossentropy",metrics=["accuracy"])
checkpoint = ModelCheckpoint("model.h5",monitor="
   ↪ val_accuracy",save_best_only=True, mode="auto",
   ↪ verbose=1)
earlystopping = EarlyStopping(monitor="val_accuracy",
   ↪ patience=5, mode="auto", verbose=1)

# Model Training and Model Evaluation
# Train the model
history = model.fit(train_data, epochs=7, validation_data=
   ↪ valid_data, verbose=1, callbacks=[checkpoint,
   ↪ earlystopping])

# Plot the training curves
plt.style.use("ggplot")
plt.figure(figsize=(12,6))
```

```
epochs = range(1,8)
plt.subplot(1, 2, 1)
plt.plot(epochs, history.history["accuracy"], "go-")
plt.plot(epochs, history.history["val_accuracy"],"ro-")
plt.title("Model␣Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend(["Train","Val"], loc="upper␣left")

plt.subplot(1, 2, 2)
plt.plot(epochs, history.history["loss"], "go-")
plt.plot(epochs, history.history["val_loss"],"ro-")
plt.title("Model␣loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend(["Train","Val"], loc="upper␣left")
plt.show()

# Evaluate the model on Test Set
model.evaluate(test_data)

# Obtaining Predictions on Test Images
# Obtain Predictions on Test Images
class_dict = {0:"glioma_tumor",1:"meningioma_tumor",2:"
    ↪ no_tumor",3:"pituitary_tumor"}

test_img1 = cv2.imread("/content/Test-Data/pituitary_tumor
    ↪ /3.jpg")
plt.imshow(test_img1)
plt.grid(False)
test_img1 = np.expand_dims(test_img1, axis=0)
pred = model.predict(test_img1)
pred = np.argmax(pred)
pred_class = class_dict[pred]
print(pred_class)
```

- **Inferences from the experiment**
  From the research work which we did on the Brain tumor disease we

got to know many things about it like what are all its types and how and why they are classified into 4 types namely:
1. Glioma Tumor
2. Meningioma Tumor
3. Pituitary Tumor
4. No Tumor

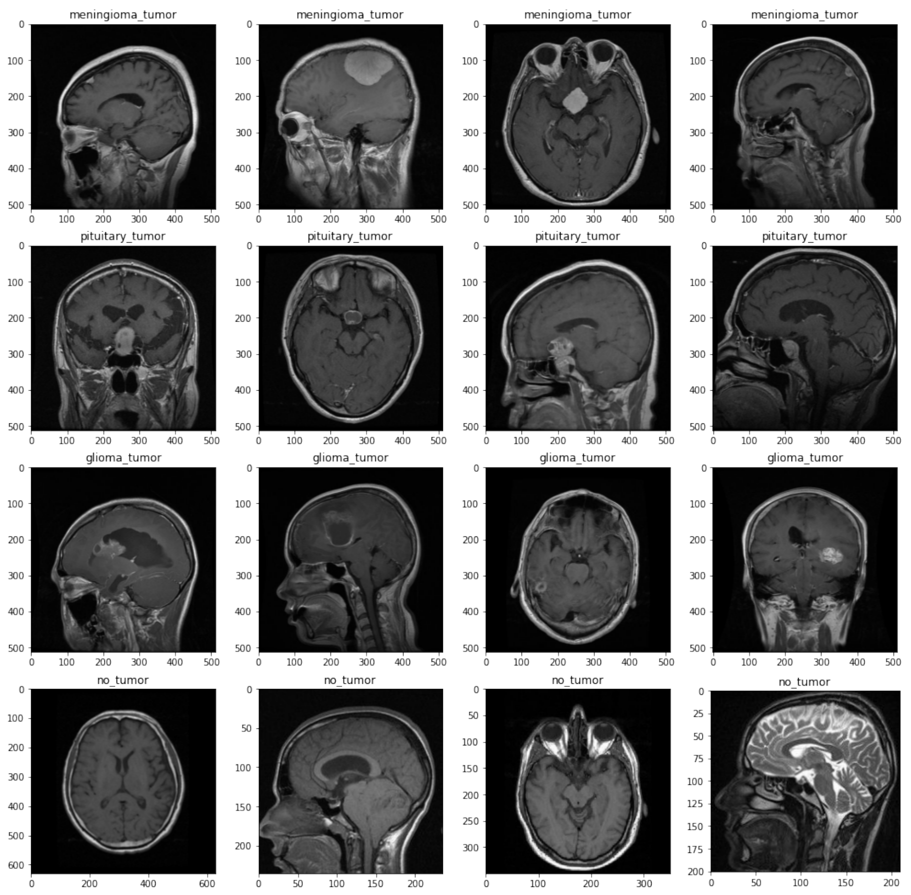


Figure 1: Different types of Brain Tumor

Also with real-time datasets their some time alot of them have unnecessary inputs with them which can hamper the performance of the model hence it is equally important to get rid of such things. Like in the brain MRI samples alot of space was wasted in the black hence we had to remove them before doing any other step.

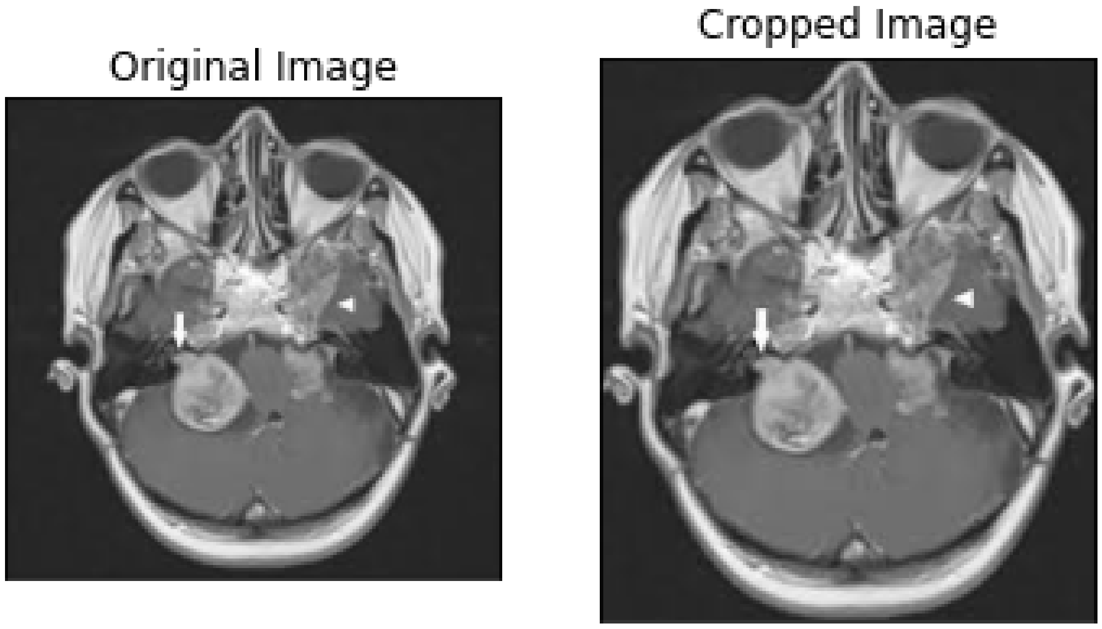


Figure 2: Original v/s Cropped Image

From the work up till we have done we can infer that Keras is extremely easy to create and fit Deep Learning model architectures. It is very flexible and powerful. We can use it to do the brain tumor classification easily and efficiently.
We have finally build a machine learning model which with very high accuracy can predict the type of brain tumor and also if the person is even having brain tumor just from the MRI scan of the brain.

5.2 Solution of the problem

Through this project we have solved the problem of early detection of the Brain Tumor classification with very high accuracy of 0.9778, which is consider quite acceptable in the field of medicine and the model loss of 0.1996. This can help people approach their doctors and get their treatment as early

as possible. Using our site which we have built a person can upload his/her Brain MRI scan and get to know if they have any of the following Brain Tumor:-
1. Glioma Tumor
2. Meningioma Tumor
3. Pituitary Tumor
4. No Tumor

6 Validation/comparison of the results

The results of our experiments are validated in the followings way, like: comparing our methodology with other various Keras methodology adopted by different people, for the same problem. For this comparison we have taken 3 case study of different Keras modules:
1. VGG-16
2. Inception V3
3. EfficientNet B1 (Used by us)

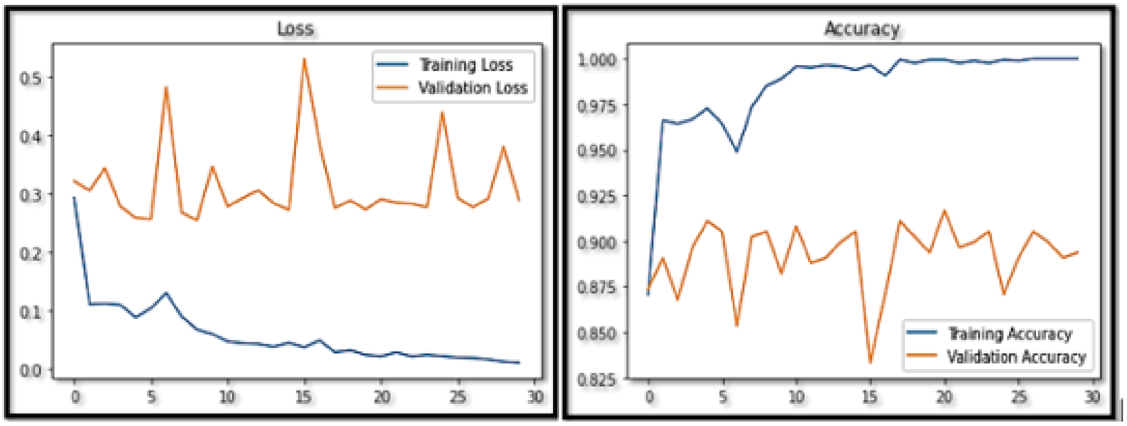


Figure 3: Loss and Accuracy Vs Epoch plots of VGG-16

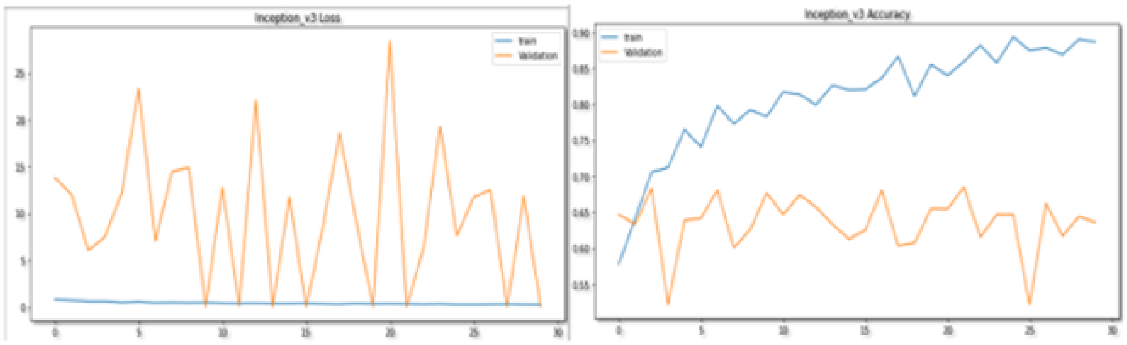


Figure 4: Loss and Accuracy Vs Epoch plots of Inception V3

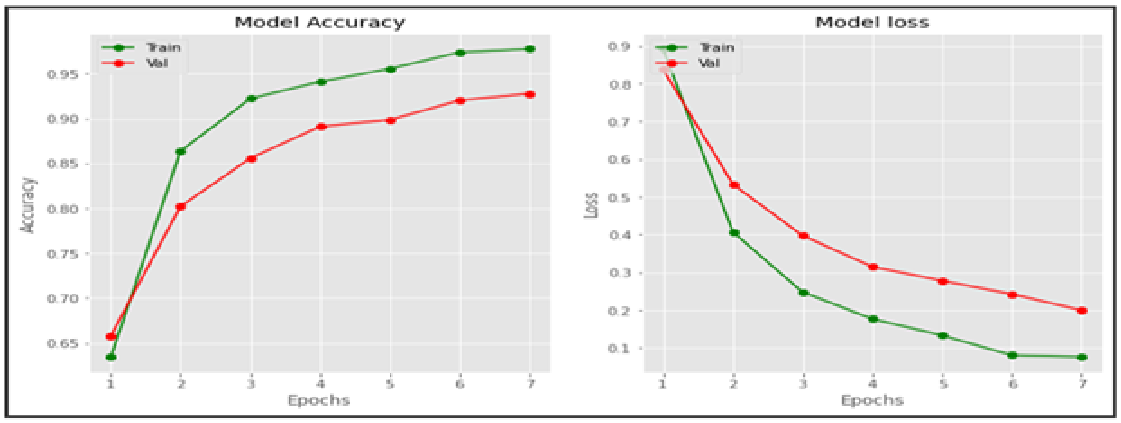
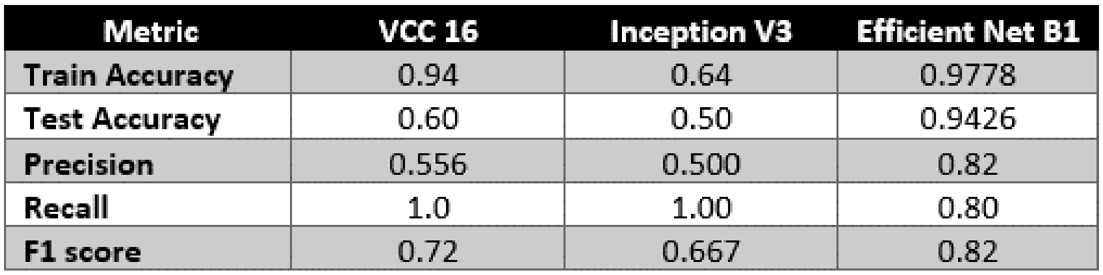


Figure 5: Loss and Accuracy Vs Epoch plots of EfficientNet B1



Metric	VCC 16	Inception V3	Efficient Net B1
Train Accuracy	0.94	0.64	0.9778
Test Accuracy	0.60	0.50	0.9426
Precision	0.556	0.500	0.82
Recall	1.0	1.00	0.80
F1 score	0.72	0.667	0.82

Figure 6: Comparison of Keras models with our model

7 Impact of the project

Since there is such a major impact of technology in recent times. Healthcare is an industry which is facing many developments and this project will have a great impact at making a bridge between the health-Care industry and Machine Learning.

This will reduce the errors in the diagnosis of Brain-Tumor and will bring a revolution in Health-Care industry.

This will greatly influence the involvement of ML in every other field of healthcare which will help in increasing the accuracy of our diagnosis.

8 Contributions of the team members

The weekly-log of all the activities done by the team members:

Team Member	Week1 (01/10/21 to 09/10/21)	Week2 (10/10/21 to 16/10/21)	Week3 (17/10/21 to 23/10/21)	Week4 (24/10/21 to 31/10/21)
Divyanu Baheti 19BCE1045	Planning Workflow	Research on Brain Tumour	Data Gathering	Data Preprocesing
Apoorv Yadav 19BCE1163	Deciding the Deep Learning Framework to be used	Research on Brain Tumour	Understanding Keras	Model Building

Team Member	Week5 (01/11/21 to 06/11/21)	Week6 (07/11/21 to 13/11/21)	Week7 (14/11/21 to 20/11/21)	Week8 (21/11/21 to 30/11/21)
Divyanu Baheti 19BCE1045	Data Agumentation	Model testing	Model Prediction	Documentation
Apoorv Yadav 19BCE1163	Model training	Model Evaluation	Inference Gathering	Documentation

9 Other Information

9.1 A short abstract of the project

Brain Tumor is one of the deadliest disease any human can suffer from. Machine Learning bought a whole new dimension to the field of Disease-Detection.

Identification:
Many a times there has been various errors from doctors while giving reports be it like sometimes.

- The doctor makes a mistake and falsely diagnose a patient.
- The doctor fails to diagnose a patient.

Our objective in this project is to create an image classification model that can predict Brain MRI scans that belong to one of the four classes with a reasonably high accuracy. Our dataset has more than 3000 Brain MRI scans which are categorized in four classes - Glioma Tumor, Meningioma Tumor, Pituitary Tumor and No Tumor.

9.2 Your personal observation at the end of the project

With completing this project and making a Production build we are quite happy that we have contributed to the society as an engineer and as a machine learning expert and with this someones life can be saved with the early detection of the Brain Tumor.

9.3 Challenges faced during the project

Since, CAT1 and CAT2 was being conducted during the project timeline hence most of the work was completed on Sunday after exam. And also to consolidate some work we had done some of the work in the earlier week so as we could give our exams easily and don't deviate from track.

9.4 Scope of converting the project to a Research article/Product

We are planning on going on one step above and build a PRODUCTION BUILD of our model and make a website where people can just upload their MRI scans of the brain and our model will predict if they have a brain tumor and if yes what type of brain tumor which can help them take proper action and contact their Doctor for further procedures.

9.5 Any other information

We have made a Production build for our model using streamlit module and deployed our model to a website whose link is mentioned below where a person can upload his/her Brain MRI scan and get a information regarding

10 References

10.1 Articles in Journals

1. Adebileje, S. A., Ghasemi, K., Aiyelabegan, H. T., and Saligheh Rad, H. (2017). Accurate classification of brain gliomas by discriminate dictionary learning based on projective dictionary pair learning of proton magnetic resonance spectra. Magn. Reson. Chem. 55, 318–322. doi: 10.1002/mrc.4532
2. Afshar, P., Mohammadi, A., and Plataniotis, K. N. (2018). "Brain tumor type classification via capsule networks," in in 2018 25th IEEE International Conference on Image Processing (ICIP), (New Jersey: IEEE), 3129–3133. doi: 10.1109/ICIP.2018.8451379
3. Al-Shaikhli, S. D. S., Yang, M. Y., and Rosenhahn, B. (2014). "Coupled dictionary learning for automatic multi-label brain tumor segmentation in flair MRI images," in In Proceedings of the International Symposium on Visual Computing, (Cham: Springer), 489–500. doi: 10.1007/978-3-319-14249-4-46
4. Al-Shaikhli, S. D. S., Yang, M. Y., and Rosenhahn, B. (2016). Brain tumor classification and segmentation using sparse coding and dictionary learning. Biomed. Tech. 61, 413–429. doi: 10.1515/bmt-2015-0071
5. Anaraki, A. K., Ayati, M., and Kazemi, F. (2019). Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms. Biocybern. Biomed. Eng.

39, 63–74. doi: 10.1016/j.bbe.2018.10.004
6. Cai, X., Ding, C., Nie, F., and Huang, H. (2013). "On the equivalent of low-rank linear regressions and linear discriminant analysis based regressions," in Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (New York: ACM), 1124–1132. doi: 10.1145/2487575.2487701

10.2 Books Other Monographs

1. WHO classification of tumours. IARC Publications Website - WHO Classification of Tumours(n.d.). Retrieved December 16, 2021
2. Louis D, Ohgaki H, Wiestler O et al. The 2007 WHO Classification of Tumours of the Central Nervous System. Acta Neuropathol. 2007;114(2):97-109. doi:10.1007/s00401-007-0243-4 - Pubmed