Brain Tumor detection using Deep learning

A MINI PROJECT REPORT

Submitted by

NARESH B(1920106054)

NAVIN J(1920106055)

PRATHEESH T(1920106068)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

SONA COLLEGE OF TECHNOLOGY,
SALEM-5 (Autonomous)

ANNA UNIVERSITY : CHENNAI 600 025

NOVEMBER 2023

BONAFIDE CERTIFICATE

Certified that this project report "**Brain Tumor detection using Deep learning**" is the bonafied work of "**NARESH B(1920106054), NAVIN J(1920106055), PRATHEESH T (1920106068)** "who carried out the project work under my supervision.

SIGNATURE
Dr.J.Akilandeswari
Professor
HEAD OF THE DEPARTMENT
Department of Information Technology
Sona College of Technology,
Salem-636 005.

SIGNATURE
Mr.A.NaveenKumar
Assistant Professor
SUPERVISOR
Department of Information Technology
Sona College of Technology,
Salem-636 005.

Submitted for Mini Project -I viva voce examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First and foremost, we thank to power of almighty for showing us inner peace and for all blessings Special gratitude to our parents, for showing their support and love always.

We express our sincere thanks to chairman Sri.C.VALLIAPPA and principal Dr.S.R.R.SENTHILKUMAR for providing adequate facilities to complete the project.

We are immensely grateful to our Head of Information Technology Department, Dr.J.AKILANDESWARI for the continue encouragement to complete the project.

We heartfully thank our project coordinator Dr.P.SHANMUGARAJA for all of his support and guidance throughout this project. Your leadership has been instrumental in our success.

We express our heartfelt thanks to our project supervisor Mr.A.NAVEENKUMAR for his valuable guidance and fruitful discussions throughout the course of the project work.

We feel proud in sharing this success with all our department faculty, staff members and friends who helped directly and indirectly to completing this project successfully.

ABSTRACT

Brain tumors are a major cause of cancer death, and early diagnosis is essential for improving patient outcomes. Magnetic resonance imaging (MRI) is a common imaging modality used to detect brain tumors, but manual tumor segmentation is a time-consuming and challenging task. The objective of this research project is to investigate the use of the Segment Anything foundation model for automatic brain tumor detection and segmentation. The Segment Anything foundation model is a large language model that can be prompted to perform a variety of tasks, including image segmentation. The proposed research project has the potential to develop a new and effective tool for brain tumor detection and segmentation. The model can be used to improve the accuracy and efficiency of tumor diagnosis, and it can help to save lives.

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT

In recent days, brain tumors are a major cause of cancer death the main reason for this is manual brain tumor segmentation is a time-consuming and a challenging task. The objective of this research project is to investigate the use of Segment Anything Model (SAM) for brain tumor segmentation. The use of Auto Encoders also resulted in segmentation of brain tumor in blurred images. The proposed research project has the potential to develop a new and effective tool for brain tumor segmentation.

Steps involved in segmenting:

✓ MRI scan image collection

Images obtained from roboflow dataset, BRATS dataset.

✓ Box annotation

Determine tumor using utility functions.

✓ Segmentation

Segment tumor from image.

✓ Auto Encoders

Use auto encoders for noisy image.

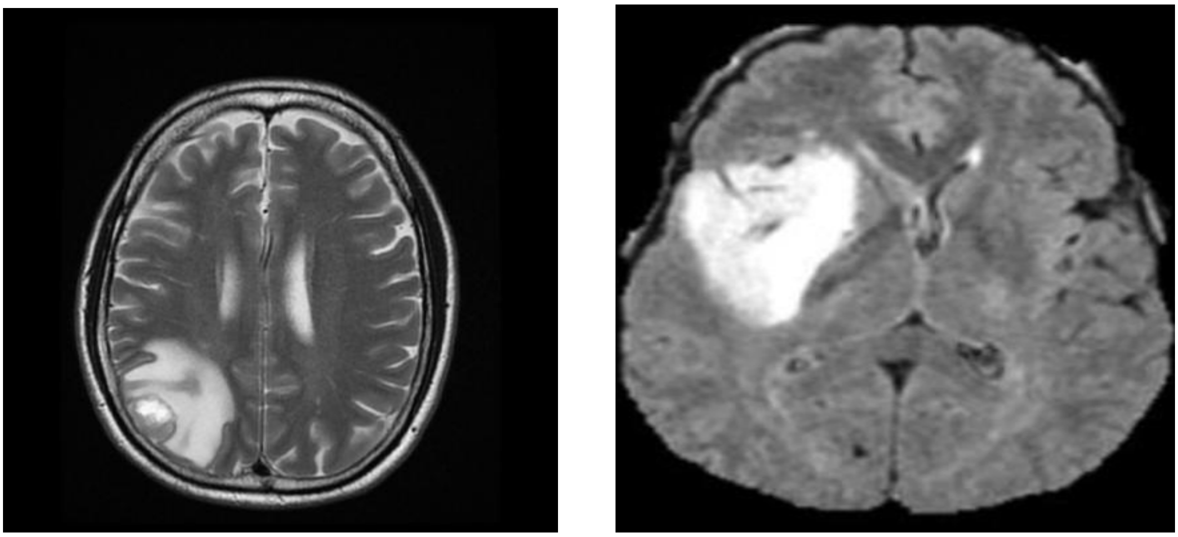


Figure 1.1 Sample MRI image with tumor

1.2 PROBLEM IDENTIFICATION

The task is to segment the tumor precisely from the available Magnetic Resonance Image (MRI) scan. The uniqueness of the project lies in the use of the newly developed Segment Anything Model (SAM) for tumor segmentation, as the manual segmentation might be time consuming. Also there lies a major difficulty in the tumor segmentation of blurred and noisy images which can be achieved by the help of auto encoders.

CHAPTER 2

LITERATURE SURVEY

This chapter addresses about the prior designed models and its limitations.

2.1 LIMITATIONS OF PRIOR ART

In previous work, researchers segmented the tumor for the Magnetic Resonance Image (MRI) with the various segmentation methods. Various segmentation methods include Convolutional Neural Network (CNN) pictured in MR images. They have also implemented various architectures based on Convolutional Neural Network[1]. The best was achieved by the implementation of Deep Neural Network (DNN) with many hidden layers specifically which are adapted to the image data. Researchers included both local features as well as more global contextual features simultaneously.

In surveys conducted in the recent years, it is evident that we can detect and segment the brain tumor from MRI scan images. The survey concluded that there are majorly two techniques on which the segmentation is performed, namely Fuzzy Transform and Morphological operation, these two techniques are used to design the image in MRI. With this developed design the researchers developed the boundaries of the brain tumor and calculate the actual area of tumor. An efficient braintumor detection by integrating modified texture based region growing and cellular automata edge detection. The above proposed method proved to be efficient in treatment of brain tumor and also in removal of tumor.

The working on image to perform different operation is known as image processing. Digital Image is a collection of a defined number of pixels. Each feature has a particular pixel value at a specific position. Image restoration can be considered as an image enrichment where the final image is printed with more important features that create more appealing for the spectator but it is not essential to create genuine data from the point of a scientific vision. Autoencoder is one of the techniques in artificial neural network to analyze and predict a huge volume of data with different many features. This technique has three stages namely the first one is encoding second one is compression and third one is decoding. Denoising autoencoder is an enhanced technique for image processing to reconstruct the damaged image to original image. Denoising autoencoder techniques are applying on the medical image to get the high quality image in digital image processing. In this paper, stochastic gradient descent algorithm is applied to predict the damaged pixel and replaced by new pixel values. The result is better than previous result in medical images.

The literature survey reveals that Magnetic Resonance Imaging (MRI) is a valuable technique for identifying brain tumors in the human mind. To classify tumors into distinct categories such as meningioma, glioma, and pituitary, the analysis of MRI images plays a crucial role. Recent advancements in deep learning have proven to be powerful tools for image classification tasks. This study focuses on the application of the Faster Region-based Convolutional Neural Networks (Faster R-CNN) technique, implemented through the TensorFlow library. The research utilizes a publicly available dataset comprising 3,064 MRI brain images, covering glioma, pituitary, and meningiomas across 233 patients. Results demonstrate that the Faster R-CNN approach achieves an accuracy of 91.66%, surpassing previous work on the same dataset.

2.2 SEGMENTATION MODELS

Segmentation has been a fundamental task in computer vision, with a rich history of model development and evolution. Early segmentation techniques primarily focused on simple thresholding and edge detection methods, providing foundational concepts for image analysis. In the late 20th century, the introduction of region-based methods like the watershed algorithm marked a significant advancement. However, it was in the 2010s that the field witnessed a revolutionary transformation with the emergence of deep learning-based segmentation models.

The U-Net architecture in 2015 played a pivotal role by enabling pixel-wise predictions and quickly became a cornerstone in medical image analysis. Subsequently, models like Mask R-CNN and Fully Convolutional Networks (FCN) further expanded the horizons of segmentation tasks. These innovations set the stage for the development of sophisticated models like Deeplab, PSPNet, and the advent of real-time semantic segmentation. As we delve deeper into this research paper,we will explore the historical lineage and significant milestones that have shaped the landscape of segmentation models, providing a comprehensive understanding of their evolution and contemporary capabilities.

2.3 SEGMENT ANYTHING MODEL (SAM)

The "Segment Anything" model represents a groundbreaking advancement in the field of computer vision and semantic segmentation. Introduced in recent years, this innovative approach transcends conventional segmentation models, offering the ability to segment a wide array of objects and elements in images. Unlike traditional models that primarily target well-defined object classes, "Segment Anything" operates under the premise that any distinguishable entity in an image can be precisely delineated. This model leverages the power of deep learning, utilizing convolutional neural networks (CNNs) and advanced architectures to achieve high precision and accuracy. By training on diverse datasets that encompass a myriad of object categories, the "Segment Anything" model showcases remarkable adaptability, capable of segmenting objects, scenes, text, animals, and virtually any visual element that can be delineated. The versatility and broad scope of this model have significant implications for applications ranging from autonomous vehicles and robotics to content moderation and image analysis.

At the core of the "Segment Anything" model lies a sophisticated neural network architecture designed to recognize and delineate objects with unparalleled precision. This architecture utilizes multi-scale feature extraction, contextual information integration, and post-processing techniques to refine segmentation results. The model's training process involves massive datasets and extensive annotations, enabling it to discern fine-grained details and complex scenes. An intriguing aspect of the "Segment Anything" model is its ability to handle instances where traditional segmentation models may falter. It can differentiate overlapping objects, handle objects with intricate shapes, and adapt to various environmental conditions. Furthermore, this model's adaptability to novel or uncommon object categories makes it a robust tool for researchers and developers seeking a comprehensive approach to image segmentation. Its deployment is poised to revolutionize a wide range of industries, including healthcare, e-commerce, surveillance, and beyond, by unlocking new possibilities for object delineation and image understanding.

2.4 PROPOSED APPROACH

The main focus of the proposed approach is to segment the tumor present in the brain using the available Magnetic Resonance Image (MRI) scans with the implementation of Segment Anything Model (SAM). The images of the tumors and the MRI scans were collected from the roboflow and BRATS dataset. The objective of this research project is to investigate the use of the Segment Anything foundation model for automatic brain tumor detection and segmentation.

The model also perform well when the noisy images are smoothened by the Auto encoders which also results in better segmentation of tumor. Along with the segmentation of the tumor with the produced mask from the SAM, we also identify the position of the tumor that is present in the brain based on the identified box annotation. As an additional feature the live segmentation is also introduced by the model which allows to visualize the real time segmenting based on the input boundary boxes

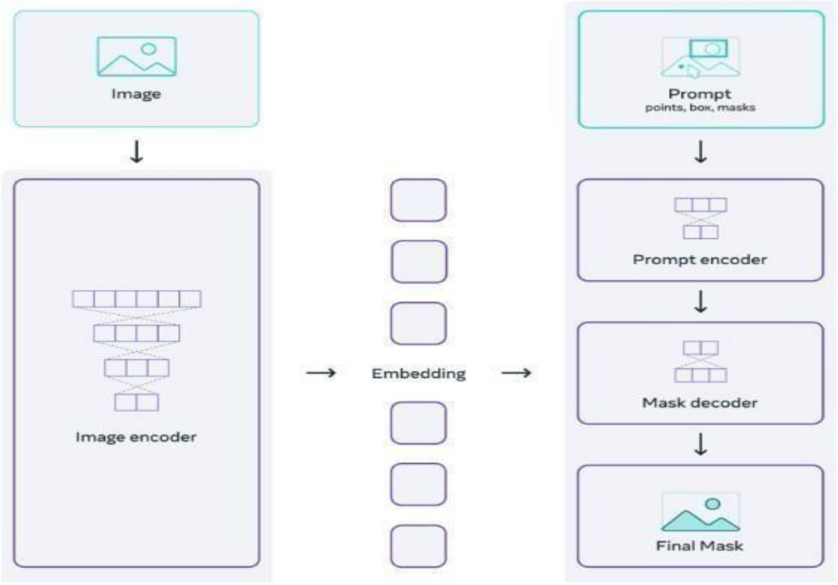


Figure.2.1 Architecture of the proposed system for segment anything model

CHAPTER 3
PROJECT DESCRIPTION

This chapter addresses about the requirements, modules and packages essential for the project.

3.1 SOFTWARE REQUIREMENTS

- ✓ Linux Operating System/Windows
- ✓ Python Platform (Anaconda2, Spyder, Jupyter, Colab)
- ✓ GPU (For better processing speed)
- ✓ Modern Web Browser

3.2 BLOCK DIAGRAM

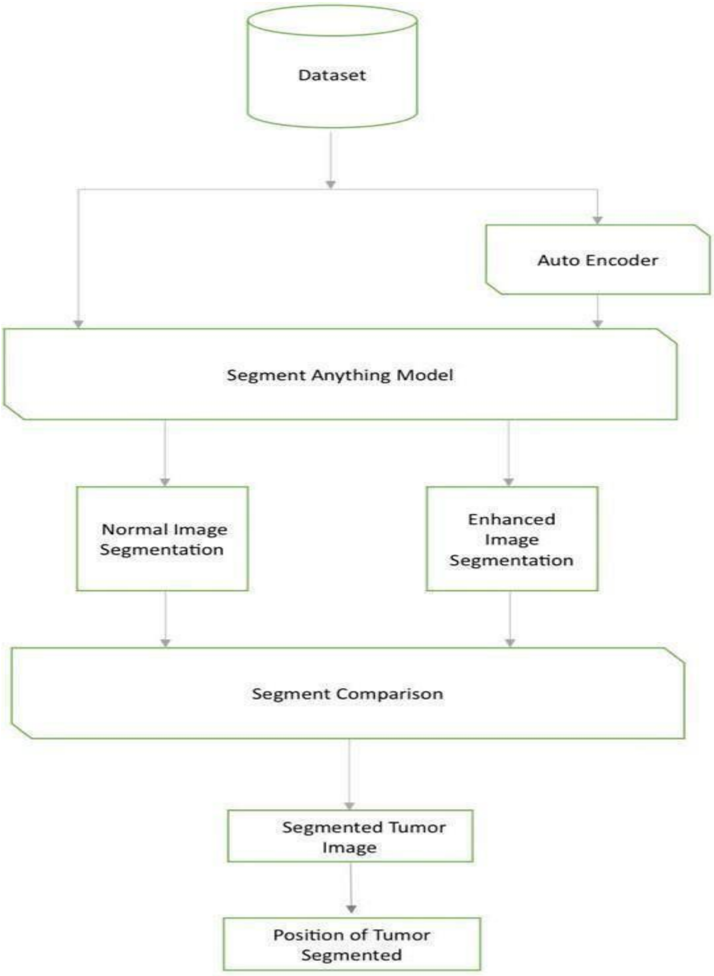


Figure 3.1 : Block diagram

3.3 INPUT(DATA)

Data in the form of images along with the box annotation is provided from the roboflow dataset which contains different types of brain tumors that are diagnosed by the doctors. The tumors are also present in various locations and also in various numbers which tends to appear more realistic in nature. The box annotation serves as the input ground truth for the Segment Anything Model which segments the tumor from the input image. The BRATS dataset(7GB) is used for the purpose of live segmentation of the tumor using the Segment Anything Model, the input boundaries for these images will be provided in realtime, which acts as the base layer for segmentation.

3.4 AUTO ENCODERS

The auto encoder has facilitated the way of segmenting the tumor from the blurred image which contains a lot of noisy in it . This way proves to be effective in segmenting the tumor for the noisy images with proper segmentation results. Autoencoders play a pivotal role in image segmentation by virtue of their inherent capacity to capture and compress salient features of images while subsequently restoring them to their original form. These neural networks are structured as two main components: the encoder and the decoder. The encoder analyzes the input image and generates a condensed representation, often referred to as the latent space or bottleneck layer, where crucial image features are encoded. This compressed representation retains vital information about the image's content and context. Subsequently, the decoder's role is to reconstruct the segmented image from this compact representation. This process of encoding and decoding allows autoencoders to segregate and highlight distinctive image features, resulting in a form of unsupervised segmentation.

The power of autoencoders in image segmentation lies in their ability to discern underlying patterns, shapes, and structures in images without relying on predefined object categories or manual annotations. They can adapt to various segmentation tasks by learning the most discriminative features within the given dataset. Moreover, autoencoders can facilitate both image denoising and feature enhancement, which are invaluable in refining the quality and accuracy of segmentation.

3.5 POSITION IDENTIFICATON

The tumor segmentation lies as the major objective of the project, along with that after the tumor segmentation, the working model also provides the position of the tumor that is present in the brain MRI scan.

This is actually done with the help of two data which is the ground truth that is detected by the Segment Anything Model which is used to segment the tumor and the other is the live segmenting boundary boxes. These various data provide the coordinates for the above lying tumor mask, the x-axis and y-axis coordinates from the detection masks, i.e. the mask with a higher are of interference can used for the position identification of the tumor.

3.6 MODULES AND PACKAGES

3.6.1 SEGMENTATION

The Magnetic Resonance Imaging (MRI) scans are the inputs of the Segment Anything Model. With the help of utility functions the Ground truth of the segmented image and predict the tumor for Segmentation is rederived. The newly proposed large language model which is a part of Deep Learning. The Segment Anything Model is used to accurately segment tumors from Magnetic Resonance Imaging(MRI) scans. The ground truth category is obtained by several utility function which are intended for its specific purpose.

- Annotation2Detection :

The core of the segmentation happens based on the conversion that is done by this utility function, which takes the input of a list of coco annotations and convert it into to model readable and processable annotation which is termed as detection object. It returns an NumPy array that consists of several detected detections.

- Path Identification :

It is the functionality of the combination of three utility functions that are used to identify the path of the image along with its annotation file and read and retrieve its specific images related information like class id, box area and the coordinates of its axis.

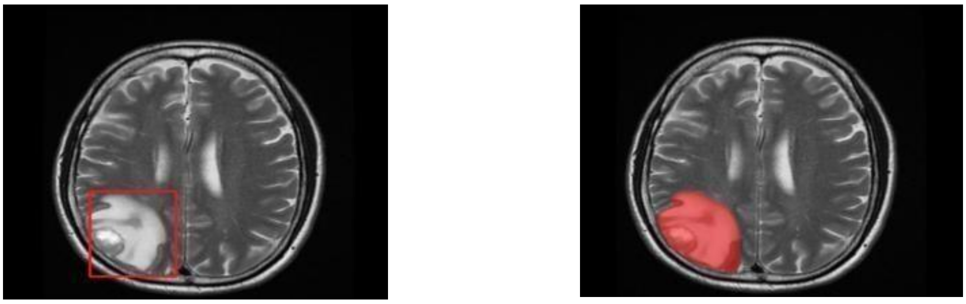


Figure 3.2 : Image Segmentation

3.6.2 IMAGE ENHANCEMENT

The Auto encoders are applied to various kind of images namely , blurred and noisy images. The partially segmented images can be auto encoded and can be again input to original Segment Anything Model for better tumor segmentation. Various types of methods for smoothening and enhancement are applied and tested for the MRI image scan like the Block Matching and 3 Dimensional filtering which proves to be effective and accurate compared to the Gaussian and the Bi – Lateral encoders. In this model, the auto encoders are mainly applied in order to smoothen the Magnetic Resonance Imaging(MRI) scans by clearing the unwanted noisy in the image and soften the edges of the segmenting image for better segmentation output.

The auto encoders comes under the Deep learning domain namely the computer vision, The model uses the python library bm3d and the skimage library for enhancement purpose. The binary encoded image is initially converted to an array of continuous floating point values using the skimage library and after processing converted back to its original state.

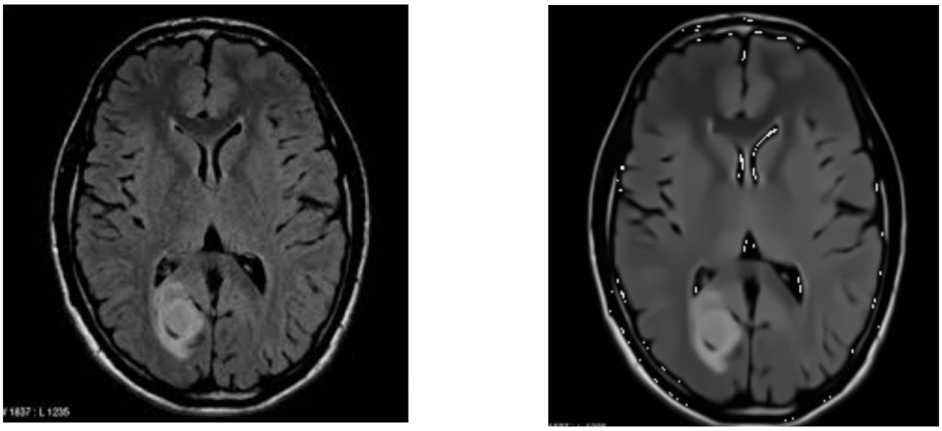


Figure 3.3 : Image Enhancement

3.6.3 ACCURACY COMPARISON

The model itself does not provide any libraries for accuracy evaluation and does not support any kind of metrics such as training and testing accuracy. However, with the help of fine tuning of the parameters of the Segment Anything Model it is visually evident that the segmentation accuracy is comparatively high than the previous model and original image. This fine tuning of the model is based on the various number of masks that are generated during the detection of tumor by the Segment Anything Model. With the detection are the model can accurately find the best possible tumor segmented images from the available detected mask images.

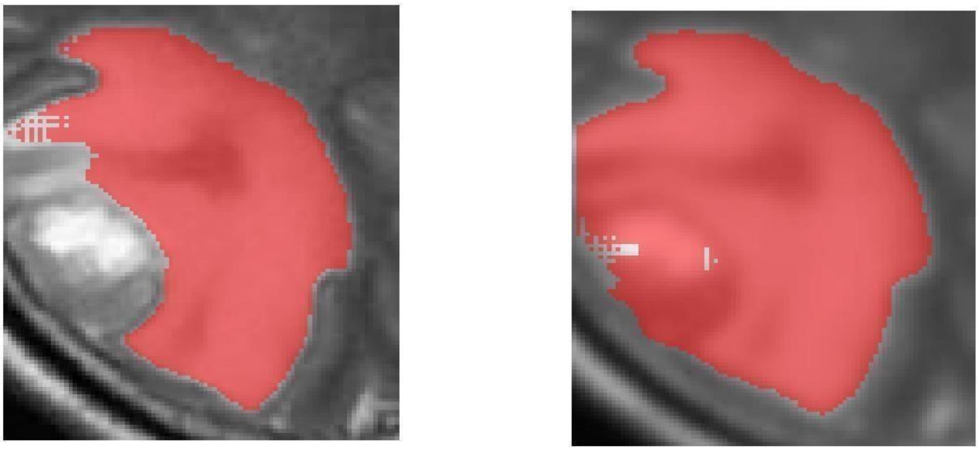


Figure 3.4 Image Comparison

3.6.4 LIVE SEGMENTING

The ability to perform live segmentation of brain tumors by drawing instant boundary boxes using the "Segment Anything" model signifies a significant advancement in medical imaging and healthcare. This innovative approach leverages the capabilities of deep learning and computer vision to provide real-time, user-guided segmentation of brain tumors directly from medical scans. With this system, medical professionals can interactively delineate tumor regions within a patient's brain images by simply drawing boundary boxes around the affected areas. This interactive process not only enhances the speed of diagnosis but also empowers medical practitioners with a more intuitive and precise tool for identifying and tracking brain tumors.

The significance of live brain tumor segmentation through instant boundary boxes lies in its potential to expedite clinical decision-making, treatment planning, and monitoring of patient progress. By streamlining the segmentation process and making it accessible in real-time, this technology has the potential to revolutionize how healthcare professionals approach brain tumor diagnosis and treatment, ultimately leading to improved patient outcomes.

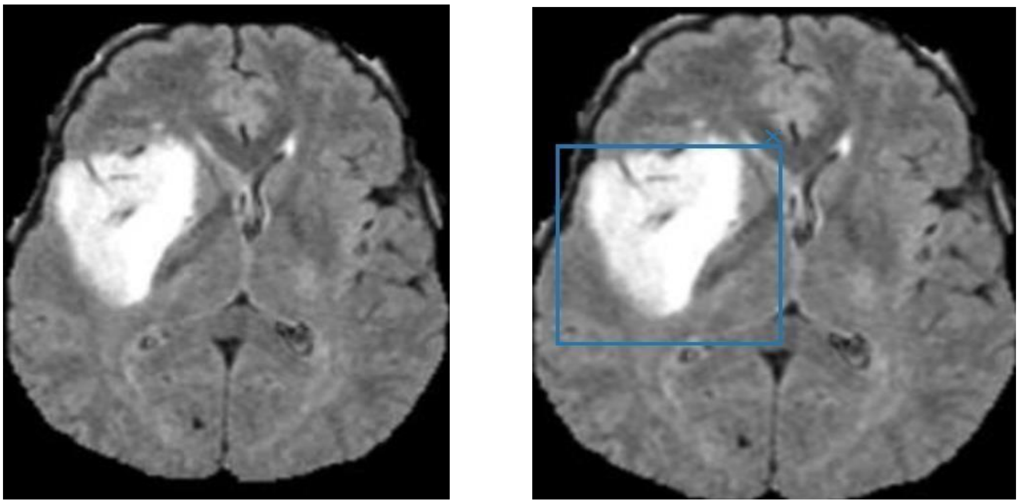


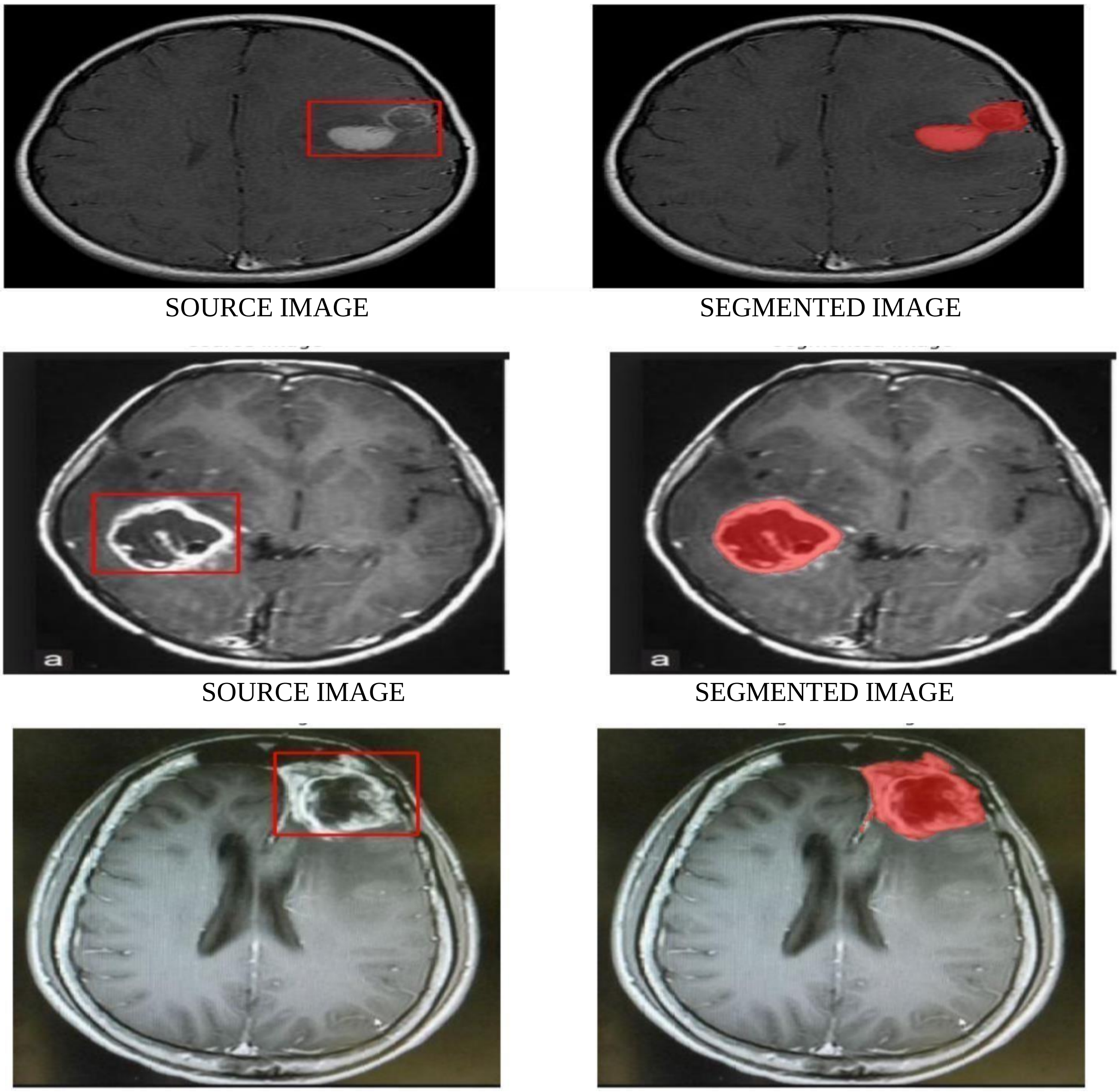
Figure 3.5: Live Segmenting

CHAPTER 4

RESULTS

This chapter addresses the results of the project with provided interface.

4.1 TUMOR SEGMENTED IMAGES



SOURCE IMAGE

SEGMENTED IMAGE

SOURCE IMAGE

SEGMENTED IMAGE

SOURCE IMAGE

SEGMENTED IMAGE

Figure 4.1 : Result of Tumor Segmented Images

4.2 AUTO ENCODED IMAGE

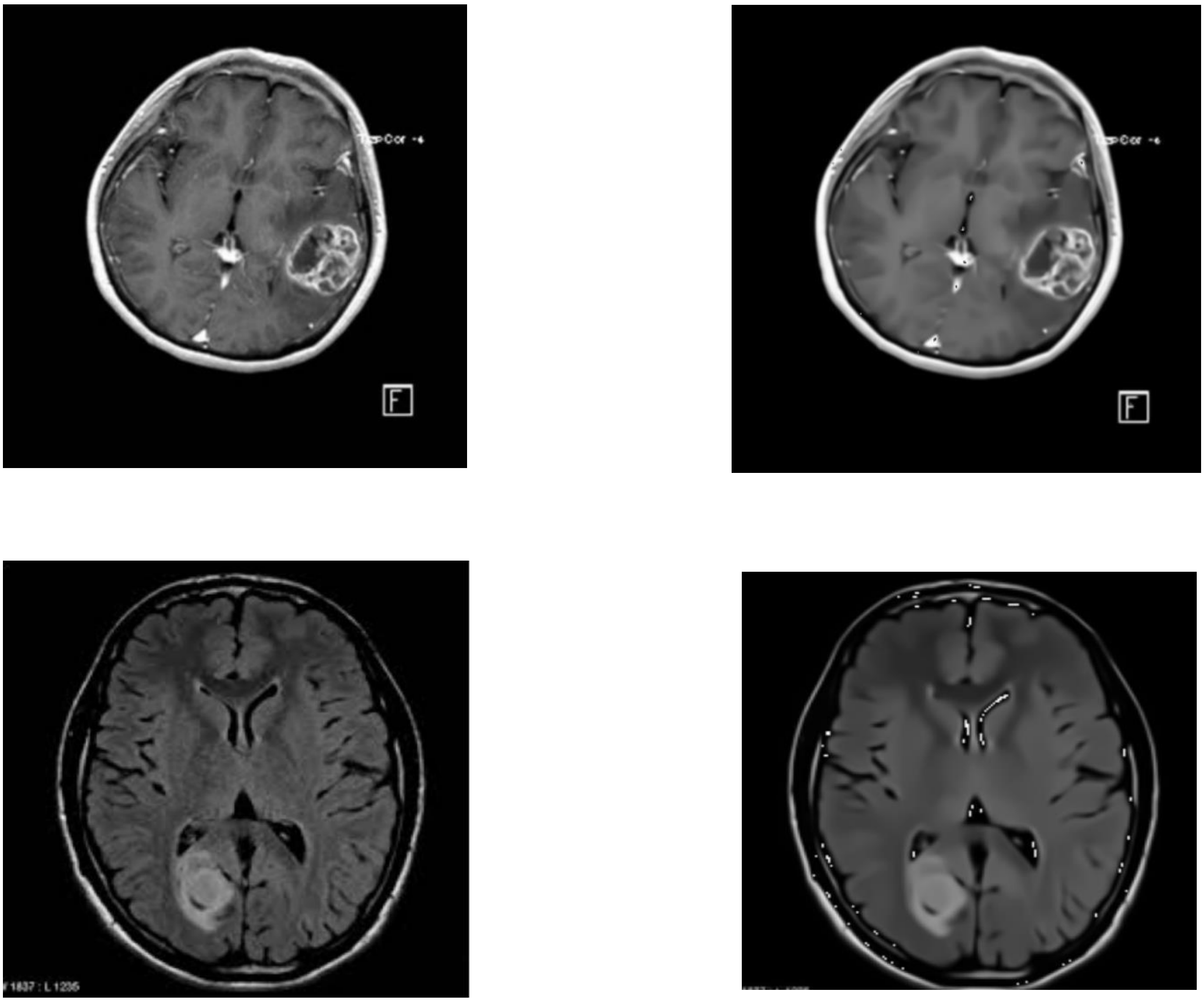


Figure 4.2: Result of Auto Encoded Image

4.3 IMPROVED SEGMENTATION

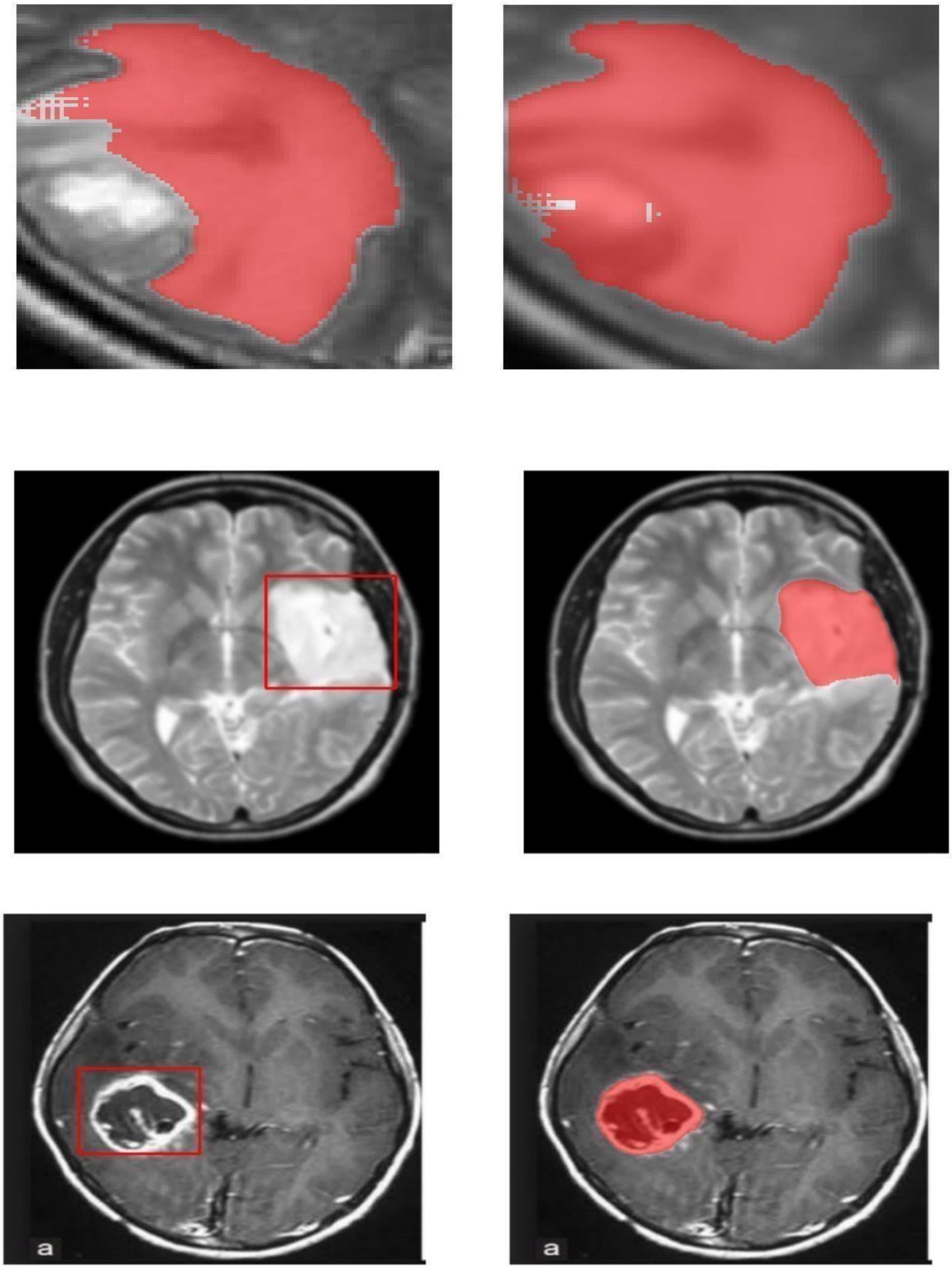
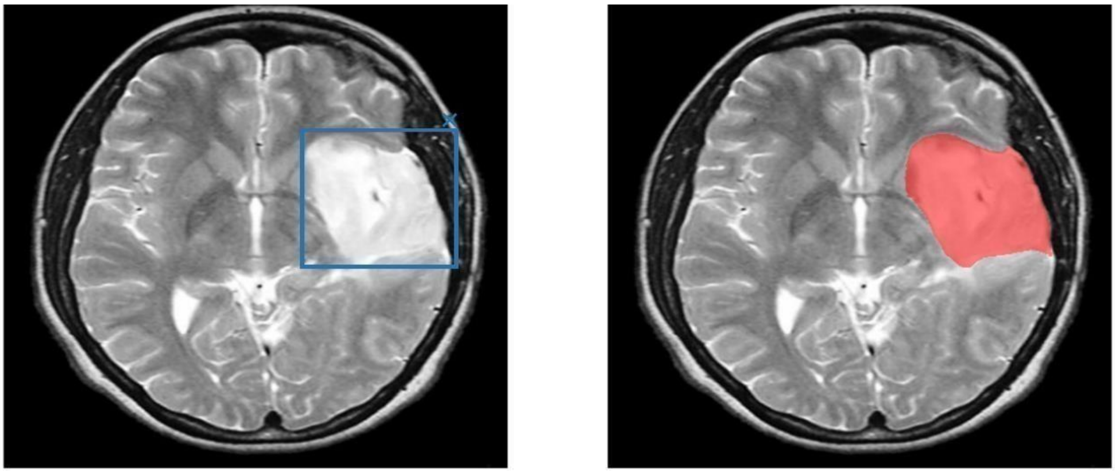
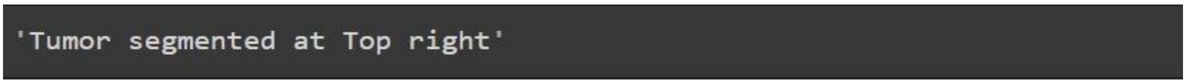


Figure 4.3: Result of Improved Segmentation

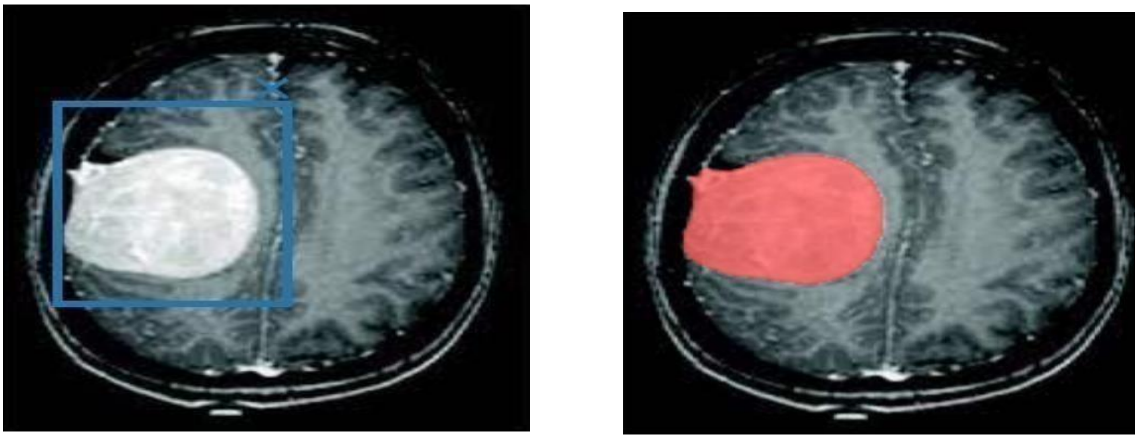
4.4 LIVE SEGMENTING



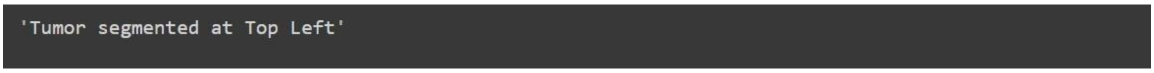
SOURCE IMAGE SEGMENTED IMAGE



```
'Tumor segmented at Top right'
```



SOURCE IMAGE SEGMENTED IMAGE



```
'Tumor segmented at Top Left'
```

Figure 4.4 : Result of Live Segmenting

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONCLUSION

Segment Anything Model (SAM) is an interesting way to think about the applicability of Image segmenting using Deep Learning in making automated segmentation of brain tumors from Magnetic Resonance Imaging (MRI) scans. It is being utilized in health care and medical diagnosis and sometimes, for practical purposes. Segment Anything Model provides many exciting opportunities to learn about different ways of segmenting masks from input images. Being able to analyze tweets in real-time, and determine the segments that underlies the potential tumor, adds a new dimension to the existing segmentation models.

Segmenting based on the live boundary box drawing can help the medical persons to visualize the presence of tumor in the brain and the provided information about the positioning of the tumor will be an added advantage.

5.2 FUTURE ENHANCEMENT:

The task of segmentation, especially in the domain of medical and health care, is still in the developing stage and far from complete. So a couple of ideas are proposed which are worth exploring in the future and may result in further improved performance. Right now we have worked with only the very simplest model; we can improve those models by adding extra information like developing a library or a method for accuracy evaluation of the Segment Anything Model. This can take the existing ground truth values and the detected detection values, which are basically an array of RGB values and perform the pixels level comparison to measure the accuracy of the Segment Anything Model. Additionaly, the model under performs when there exists multiple tumor in the same MRI scan and also the size of the different tumors lies in a different range, the Segment Anything Model is capable of segmenting either of the images especially the one with the large are of inference.

APPENDICES
APPENDIX 1
SAMPLE SCRIPT

Tumor Detection

Code:

Tumor segmentation

```
import os
HOME = os.getcwd()
print("HOME", HOME)
import os
HOME = os.getcwd()
print("HOME", HOME)
!pip install -q jupyter_bbox_widget roboflow dataclasses-json supervision
%cd {HOME}

!mkdir {HOME}/weights
%cd {HOME}/weights

!wget -q https://dl.fbaipublicfiles.com/segment_anything/sam_vit_h_4b8939.pth
import os

CHECKPOINT_PATH = os.path.join(HOME, "weights", "sam_vit_h_4b8939.pth")
print(CHECKPOINT_PATH, "; EXISTS : ", os.path.isfile(CHECKPOINT_PATH))

%cd {HOME}
!mkdir {HOME}/data
%cd {HOME}/data
import torch

DEVICE = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
MODEL_TYPE = "vit_h"
!pip install segment_anything
from segment_anything import sam_model_registry, SamAutomaticMaskGenerator,
SamPredictor
sam =
sam_model_registry[MODEL_TYPE](checkpoint=CHECKPOINT_PATH).to(device=D
EVICE)
```

```
mask_predictor = SamPredictor(sam)
import numpy as np
from dataclasses import dataclass
from typing import List, Tuple, Union, Optional
from dataclasses_json import dataclass_json
from supervision import Detections


@dataclass_json
@dataclass
class COCOCategory:
    id: int
    name: str
    supercategory: str


@dataclass_json
@dataclass
class COCOImage:
    id: int
    width: int
    height: int
    file_name: str
    license: int
    date_captured: str
    coco_url: Optional[str] = None
    flickr_url: Optional[str] = None


@dataclass_json
@dataclass
class COCOAnnotation:
    id: int
    image_id: int
    category_id: int
    segmentation: List[List[float]]
    area: float
    bbox: Tuple[float, float, float, float]
    iscrowd: int
```

```
@dataclass_json
@dataclass
class COCOLicense:
    id: int
    name: str
    url: str


@dataclass_json
@dataclass
class COCOJson:
    images: List[COCOImage]
    annotations: List[COCOAnnotation]
    categories: List[COCOCategory]
    licenses: List[COCOLicense]


def load_coco_json(json_file: str) -> COCOJson:
    import json

    with open(json_file, "r") as f:
        json_data = json.load(f)

    return COCOJson.from_dict(json_data)


class COCOJsonUtility:
    @staticmethod
    def get_annotations_by_image_id(coco_data: COCOJson, image_id: int) ->
List[COCOAnnotation]:
        return [annotation for annotation in coco_data.annotations if annotation.image_id
== image_id]

    @staticmethod
    def get_annotations_by_image_path(coco_data: COCOJson, image_path: str) ->
Optional[List[COCOAnnotation]]:
        image = COCOJsonUtility.get_image_by_path(coco_data, image_path)if
        image:
            return COCOJsonUtility.get_annotations_by_image_id(coco_data, image.id)
        else:
            return None
    @staticmethod
    def get_image_by_path(coco_data: COCOJson, image_path: str) ->
Optional[COCOImage]:
        for image in coco_data.images:
            if image.file_name == image_path:
                return image
        return None

    @staticmethod
    def annotations2detections(annotations: List[COCOAnnotation]) -> Detections:
        class_id, xyxy = [], []

        for annotation in annotations:
            x_min, y_min, width, height = annotation.bbox
            print(annotation.category_id)
            class_id.append(annotation.category_id)
            xyxy.append([
                x_min,
                y_min,
                x_min + width,
                y_min + height
            ])

        return Detections(
            xyxy=np.array(xyxy, dtype=int),
            class_id=np.array(class_id, dtype=int)
        )
%cd {HOME}

import roboflow
from roboflow import Roboflow

roboflow.login()

rf = Roboflow()

project = rf.workspace("hashira-fhxpj").project("mri-brain-tumor")
dataset = project.version(1).download("coco")
import os

DATA_SET_SUBDIRECTORY = "test"
ANNOTATIONS_FILE_NAME = "_annotations.coco.json"
IMAGES_DIRECTORY_PATH = os.path.join(dataset.location,
DATA_SET_SUBDIRECTORY)
```

```
ANNOTATIONS_FILE_PATH = os.path.join(dataset.location,
DATA_SET_SUBDIRECTORY, ANNOTATIONS_FILE_NAME)
coco_data = load_coco_json(json_file=ANNOTATIONS_FILE_PATH)

CLASSES = [
    category.name
    for category
    in coco_data.categories
    if category.supercategory != 'none'
]

IMAGES = [
    image.file_name
    for image
    in coco_data.images
]
import cv2
import numpy as np
import supervision as sv

EXAMPLE_IMAGE_NAME = 'Y33_jpg.rf.f3e7bf77085ecd9dd6c9cfbc268b8fc0.jpg'
EXAMPLE_IMAGE_PATH = '/content/MRI-BRAIN-TUMOR-
1/test/Y33_jpg.rf.f3e7bf77085ecd9dd6c9cfbc268b8fc0.jpg'

# load dataset annotations
annotations = COCOJsonUtility.get_annotations_by_image_path(coco_data=coco_data,
image_path=EXAMPLE_IMAGE_NAME)
ground_truth = COCOJsonUtility.annotations2detections(annotations=annotations)

# small hack - coco numerate classes from 1, model from 0 + we drop first redundant
class from coco json
ground_truth.class_id = ground_truth.class_id - 1

# load image
image_bgr = cv2.imread(EXAMPLE_IMAGE_PATH)
image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)

# initiate annotator
box_annotator = sv.BoxAnnotator(color=sv.Color.red())
mask_annotator = sv.MaskAnnotator(color=sv.Color.red())

# annotate ground truth
annotated_frame_ground_truth = box_annotator.annotate(scene=image_bgr.copy(),
detections=ground_truth, skip_label=True)
```

```
# run SAM inference
mask_predictor.set_image(image_rgb)

masks, scores, logits = mask_predictor.predict(
    box=ground_truth.xyxy[0],
    multimask_output=True
)

detections = sv.Detections(
    xyxy=sv.mask_to_xyxy(masks=masks),
    mask=masks
)
detections  =  detections[detections.area ==  np.max(detections.area)]
#print(detections)
#print(len(detections))
detections.class_id = [1]
#print(detections)
#print(len(detections))
annotated_image = mask_annotator.annotate(scene=image_bgr.copy(),
detections=detections)


sv.plot_images_grid(
    images=[annotated_frame_ground_truth, annotated_image],
    grid_size=(1, 2),
    titles=[", "]
)
```

AUTO ENCODERS

```
from skimage import img_as_float
from skimage.metrics import peak_signal_noise_ratio
from matplotlib import pyplot as plt
from skimage import io
from scipy import ndimage as nd
from google.colab.patches import cv2_imshow
noisy_img  = img_as_float(io.imread("/content/im.jpg"))
plt.imshow(noisy_img, cmap='gray')
from skimage.restoration import (denoise_tv_chambolle, denoise_bilateral,
                                 denoise_wavelet, estimate_sigma)
from skimage import img_as_float
noisy_img = img_as_float(io.imread("/content/im.jpeg"))
```

```
plt.imshow(noisy_img, cmap='gray')
!pip install bm3d
import matplotlib.pyplot as plt
from skimage import io, img_as_float
from skimage.metrics import peak_signal_noise_ratio
import bm3d
import numpy as np
BM3D_denoised_image = bm3d.bm3d(noisy_img, sigma_psd=0.1,
stage_arg=bm3d.BM3DStages.ALL_STAGES)
plt.imshow(BM3D_denoised_image, cmap='gray')
```

LIVE SEGMENTING

```
%cd {HOME}

import sys
!{sys.executable} -m pip install 'git+https://github.com/facebookresearch/segment-anything.git'

!pip install -q jupyter_bbox_widget roboflow dataclasses-json supervision

%cd {HOME}
!mkdir {HOME}/weights
%cd {HOME}/weights

!wget -q https://dl.fbaipublicfiles.com/segment_anything/sam_vit_h_4b8939.pth

import os

CHECKPOINT_PATH = os.path.join(HOME, "weights", "sam_vit_h_4b8939.pth")
print(CHECKPOINT_PATH, "; exist:", os.path.isfile(CHECKPOINT_PATH))

import torch

DEVICE = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
MODEL_TYPE = "vit_h"

from segment_anything import sam_model_registry, SamAutomaticMaskGenerator,
SamPredictor
```

```
sam =
sam_model_registry[MODEL_TYPE](checkpoint=CHECKPOINT_PATH).to(device=D
EVICE)
mask_generator = SamAutomaticMaskGenerator(sam)
import base64

def encode_image(filepath):
    with open(filepath, 'rb') as f:
        image_bytes = f.read()
    encoded = str(base64.b64encode(image_bytes), 'utf-8')
    return "data:image/jpg;base64,"+encoded

IMAGE_PATH = '/content/in.png'

IS_COLAB = True

if IS_COLAB:
    fromgoogle.colab import output
    output.enable_custom_widget_manager()

from jupyter_bbox_widget import BBoxWidget

widget = BBoxWidget()
widget.image = encode_image(IMAGE_PATH)
widget
print(widget.bboxes)
b = widget.bboxes[0]
x = b['x']
y = b['y']
w = b['width']
h = b['height']
position = "center"
if (x < 64 and y < 76):
  position = "Top Left"
elif (x < 64 and y > 76):
  position = "Bottom Left"
elif (x > 64 and y > 76):
  position = "Bottom right"
else:
  position = "Top right"
# x > 80 and y > 127)
Print(position)
box_annotator = sv.BoxAnnotator(color=sv.Color.red())
```

```
mask_annotator = sv.MaskAnnotator(color=sv.Color.red(), color_map='index')

detections = sv.Detections(
    xyxy=sv.mask_to_xyxy(masks=masks),
    mask=masks
)
detections = detections[detections.area == np.max(detections.area)]

source_image = box_annotator.annotate(scene=image_bgr.copy(), detections=detections,
skip_label=True)
segmented_image = mask_annotator.annotate(scene=image_bgr.copy(),
detections=detections)

sv.plot_images_grid(
    images=[source_image, segmented_image],
    grid_size=(1, 2),
    titles=['source image', 'segmented image']
)
```

REFERENCES

[1] K. Venu, P. Natesan, N. Sasipriyaa and S. Poorani, "Review on Brain Tumor Segmentation Methods using Convolution Neural Network for MRI Images," 2018 International Conference on Intelligent Computing and Communication for Smart World (I2C2SW), Erode, India, 2018, pp. 291-295, doi: 10.1109/I2C2SW45816.2018.8997387.

[2] Alexander Kirillov, undefined., et al, "Segment Anything," 2023.

[3] K. Psychogyios, H. C. Leligou, F. Melissari, S. Bourou, Z. Anastasakis and T. Zahariadis, "SAMStyler: Enhancing Visual Creativity With Neural Style Transfer and Segment Anything Model (SAM)," in IEEE Access, vol. 11, pp. 100256-100267, 2023, doi: 10.1109/ACCESS.2023.3315235.

[4] Irmak, E. Multi-Classification of Brain Tumor MRI Images Using Deep Convolutional Neural Network with Fully Optimized Framework. Iran J Sci Technol Trans Electr Eng 45, 1015–1036 (2021)

[5] S. M. Kulkarni and G. Sundari, "A Review on Image Segmentation for Brain Tumor Detection," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2018, pp. 552-555, doi: 10.1109/ICECA.2018.8474893.

[6] Y. Zhong, S. Jia and Y. Hu, "Denoising Auto-Encoder Network Combined Classfication Module for Brain Tumors Detection," 2022 3rd International Conference on Electronic Communication and Artificial Intelligence (IWECAI), Zhuhai, China, 2022, pp. 540-543, doi: 10.1109/IWECAI55315.2022.00110.

[7] R. S. Prabha and M. Vadivel, "R-CNN Based Brain Tumor Segmentation Using MRI Images," 2022 International Conference on Futuristic Technologies (INCOFT), Belgaum, India, 2022, pp. 1-4, doi: 10.1109/INCOFT55651.2022.10094426.

[8] V. S. Kumar and V. Jayalakshmi, "Reconstructing the Medical Image by Autoencoder with Stochastic Processing in Neural Network," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp. 1521-1526, doi: 10.1109/ICIRCA51532.2021.9545041.

[9] A. Wulandari, R. Sigit and M. M. Bachtiar, "Brain Tumor Segmentation to Calculate Percentage Tumor Using MRI," 2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC), Bali, Indonesia, 2018, pp. 292-296, doi: 10.1109/KCIC.2018.8628591.

[10] S. Alkassar, M. A. M. Abdullah and B. A. Jebur, "Automatic Brain Tumour Segmentation using fully Convolution Network and Transfer Learning," 2019 2nd International Conference on Electrical, Communication, Computer, Power and Control Engineering (ICECCPCE), Mosul, Iraq, 2019, pp. 188-192, doi: 10.1109/ICECCPCE46549.2019.203771.