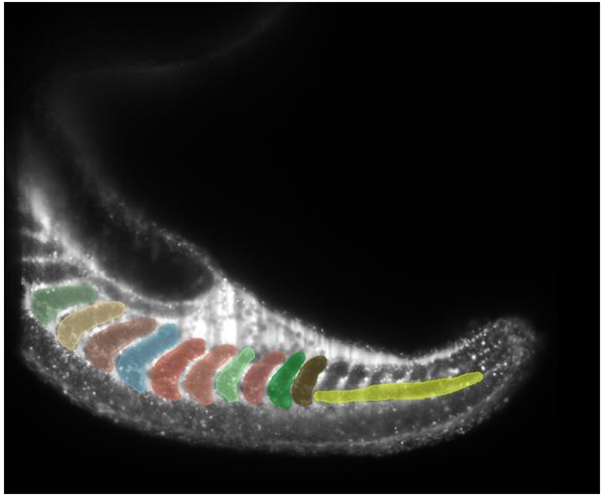


Automatic Image Segmentation of Light-Sheet Zebrafish Scans

MICROBS Lab

Supervising Professor :
Selman Sakar

Supervising TA :
Artur Krzysztof Banach

Aly Elbindary

School
of Engineering
MICROBS

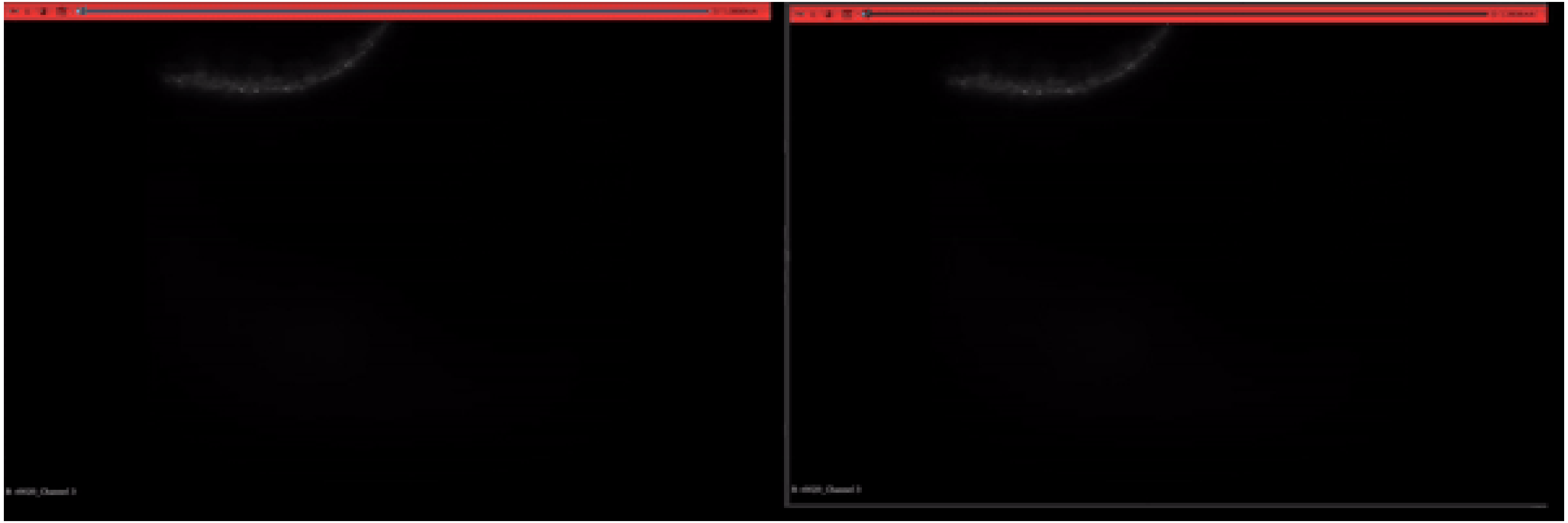
Outline

- Background & Objectives
- Methodology
- Results
- Discussion

Background & Objectives

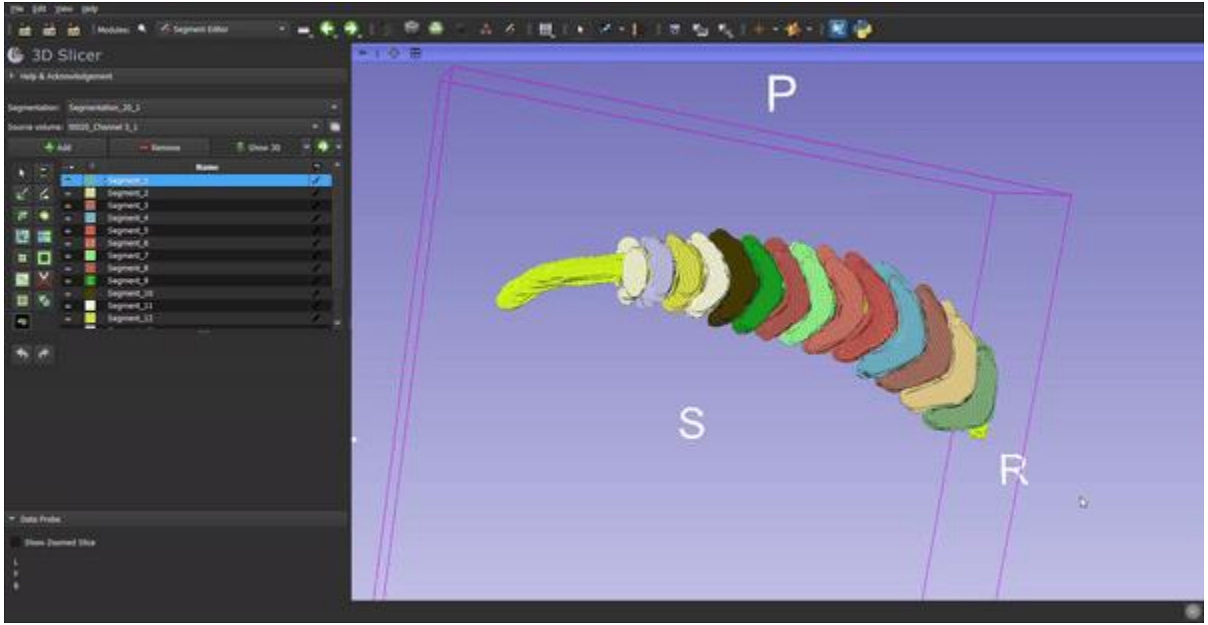
Goals & Motivation

- What do we want to achieve? Why?



Goals & Motivation

- What do we want to achieve? Why?



Segmentation: Segmentation_3D_1

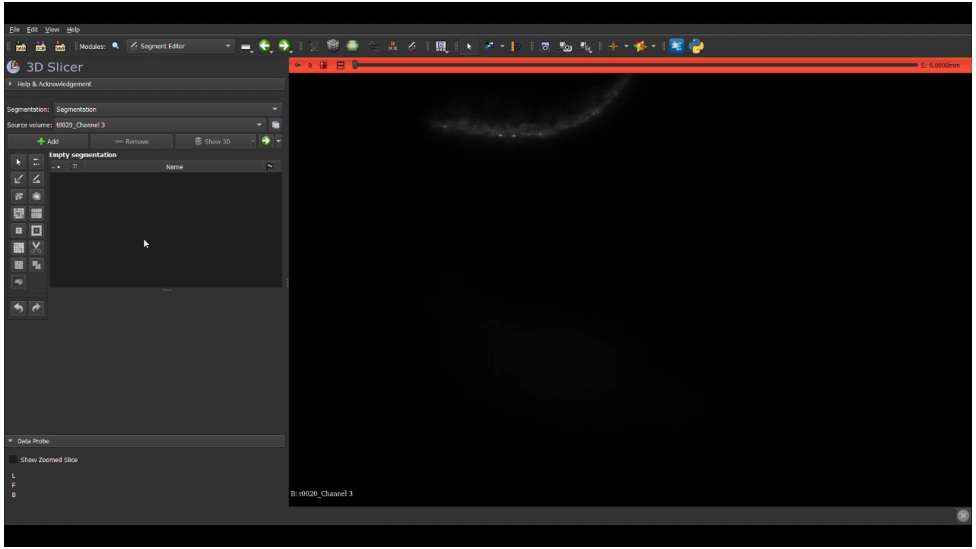
Source volume: 00020_Channel 3_1

Segment_1
Segment_2
Segment_3
Segment_4
Segment_5
Segment_6
Segment_7
Segment_8
Segment_9
Segment_10
Segment_11
Segment_12

Methodology

Methodology

- Manual Segmentation

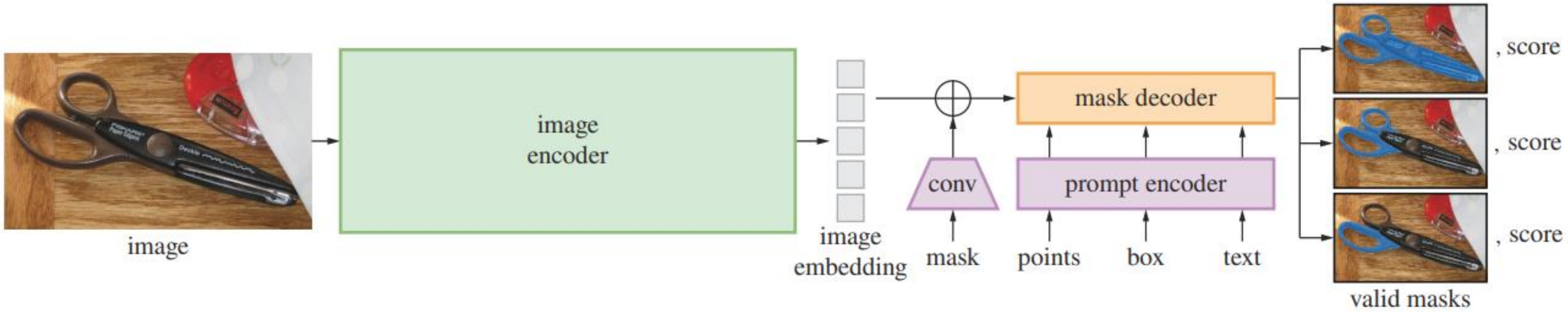


Segmentation: Segmentation

Source volume: t0020_Channel 3

Methodology

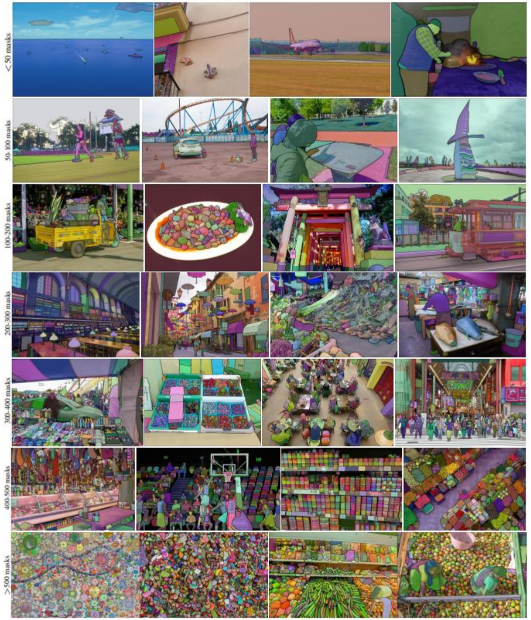
- Automatic Segmentation – Introduction to SAM



Segment Anything Model (SAM) – Architecture Overview [5]

Methodology

- Automatic Segmentation – Introduction to SAM



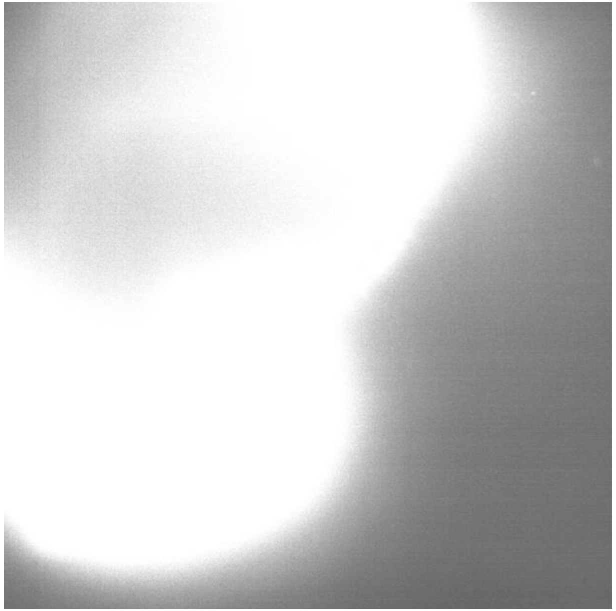
Example images with overlaid masks generated by SAM [5]

Speaker

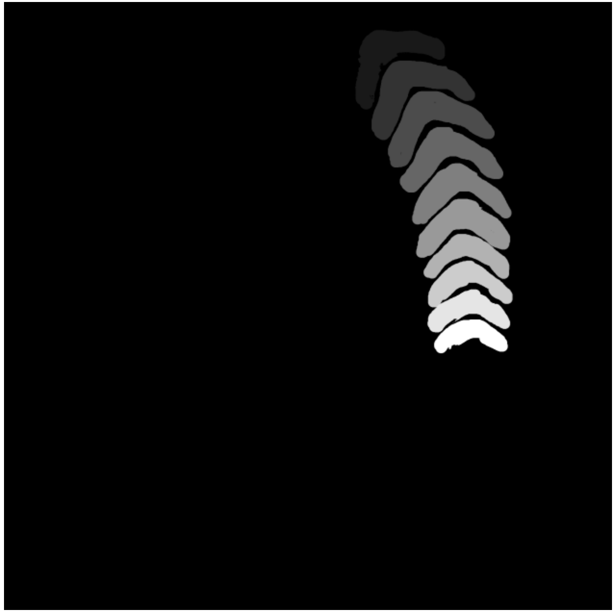
Methodology

- Raw Data Adaptation

Raw Zebrafish Frame Loaded Without Adaptation



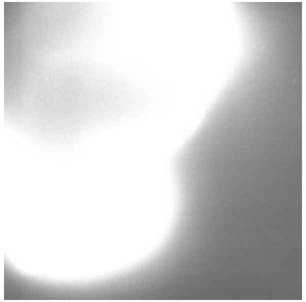
Raw Mask Frame Loaded Without Adaptation



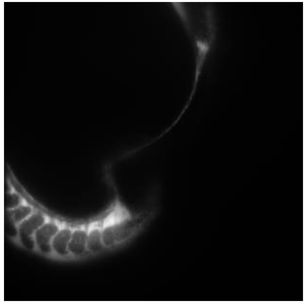
Methodology

- Raw Data Adaptation

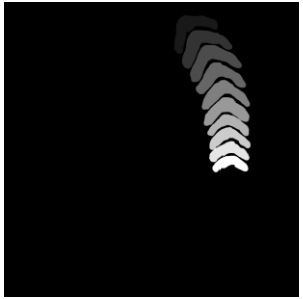
Before Pixel Conversion



After Pixel Conversion

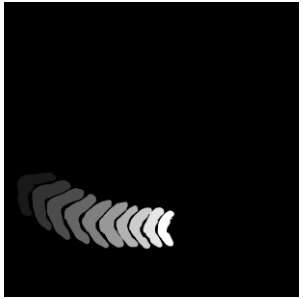


Raw Mask



90° Rotation Around Center

180° Rotation Around Middle Horizontal Axis



Methodology

- Loss Function & Evaluation Metrics

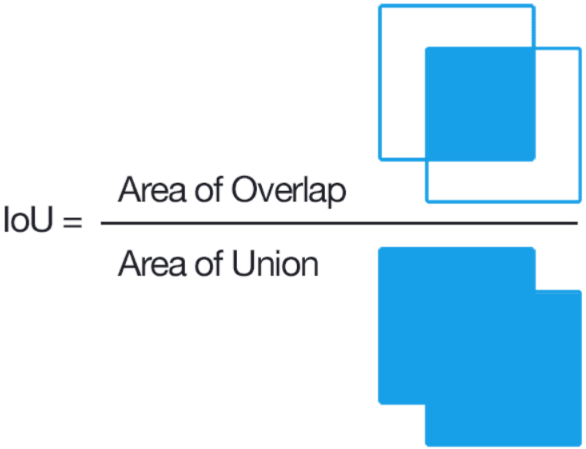
Loss Function : Focal Loss [3]

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise} \end{cases}$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise} \end{cases}$$

$$\mathrm{FL}(p_t) = (1-p_t)^{\gamma} CE(p_t)$$

Evaluation Metric : Intersection over Union (IoU) [7]



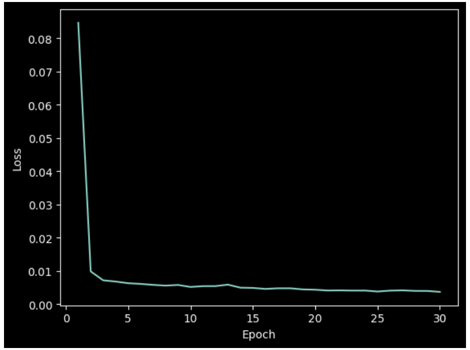
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Results

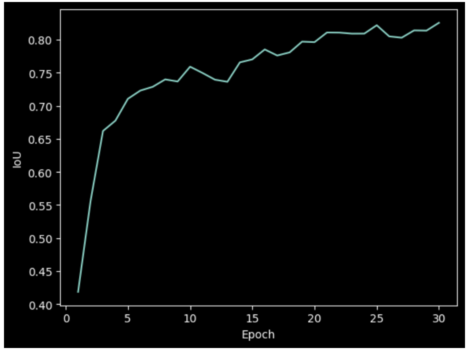
Results

- Finetuning SAM – Version : facebook/sam-vit-base (93.7M)

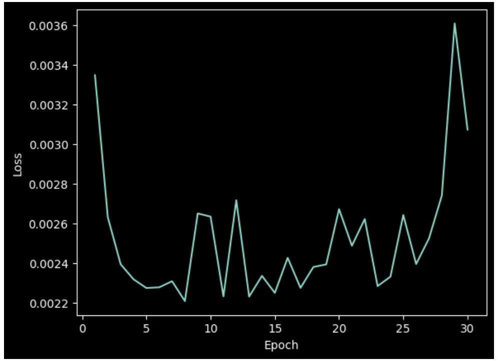
Train Loss



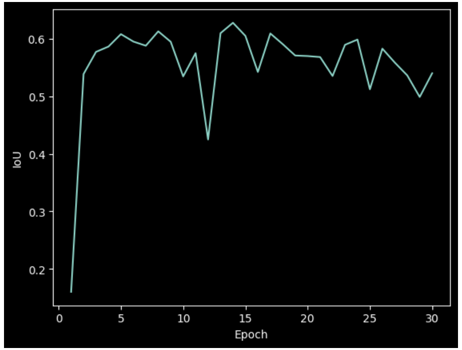
Train IoU



Test Loss

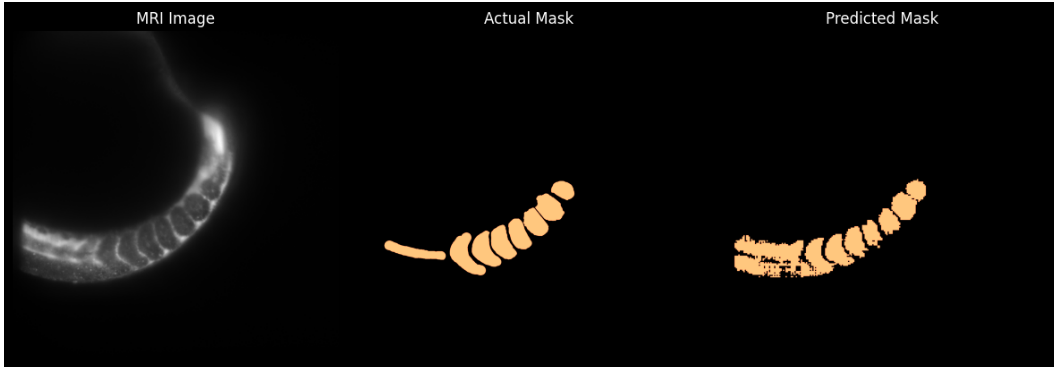


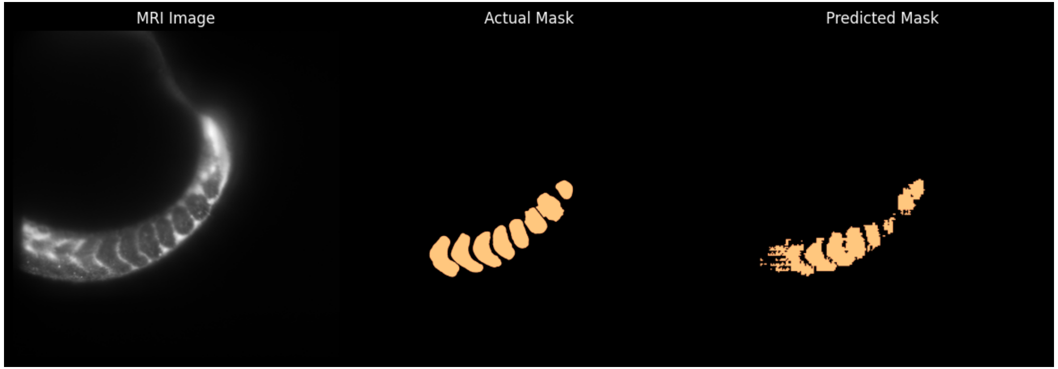
Test IoU



Results

- Finetuning SAM – Version : facebook/sam-vit-base (93.7M)

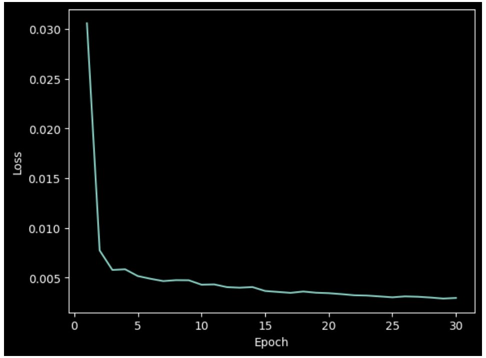




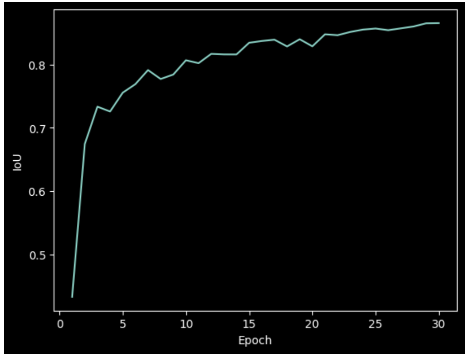
Results

- Finetuning SAM – Version : facebook/sam-vit-huge (641M)

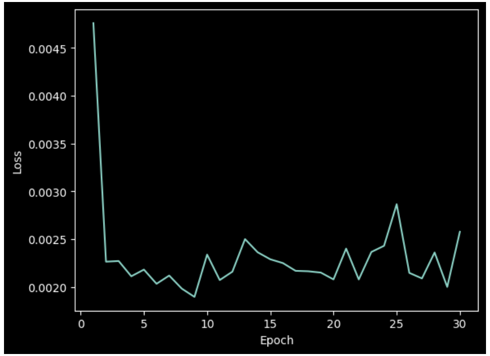
Train Loss



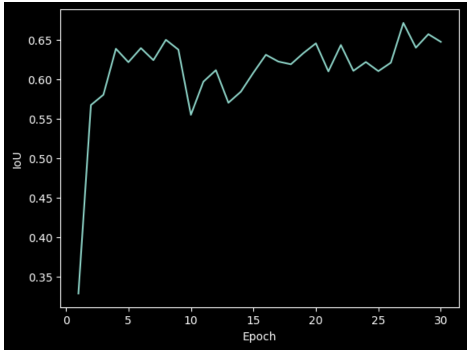
Train IoU



Test Loss



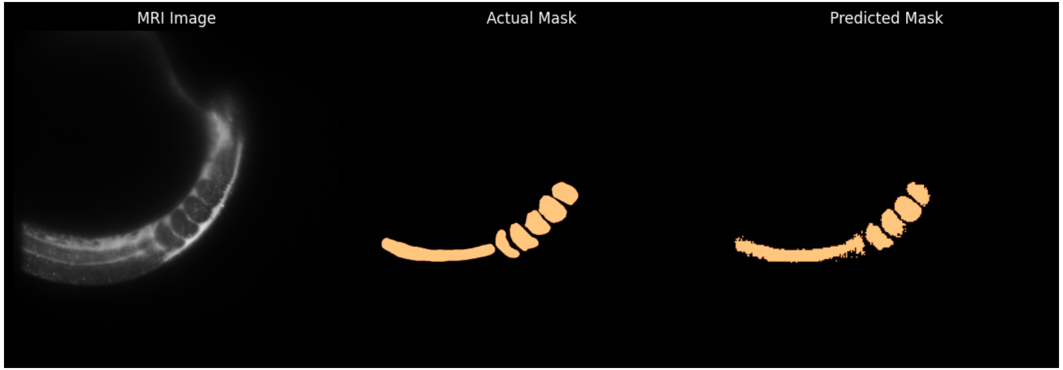
Test IoU

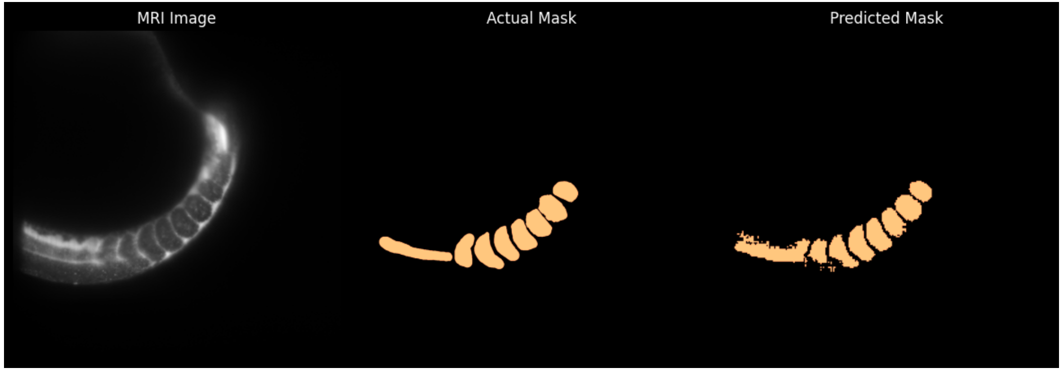


Robotics Semester Project

Results

- Finetuning SAM – Version : facebook/sam-vit-huge (641M)

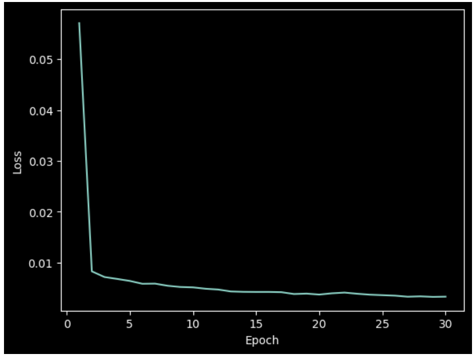




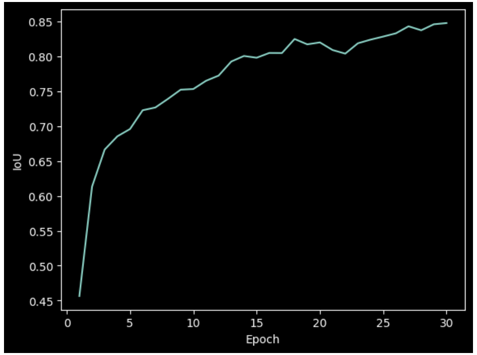
Results

- Finetuning SAM – Version : wanglab/medsam-vit-base

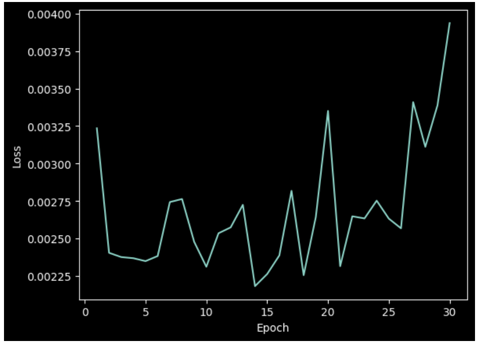
Train Loss



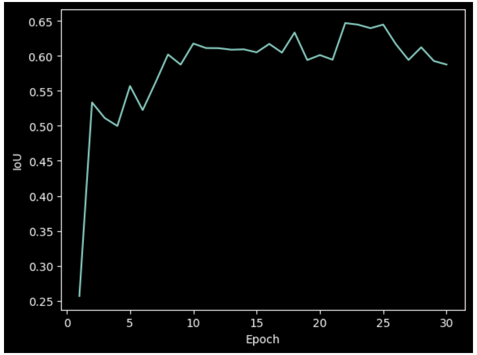
Train IoU



Test Loss

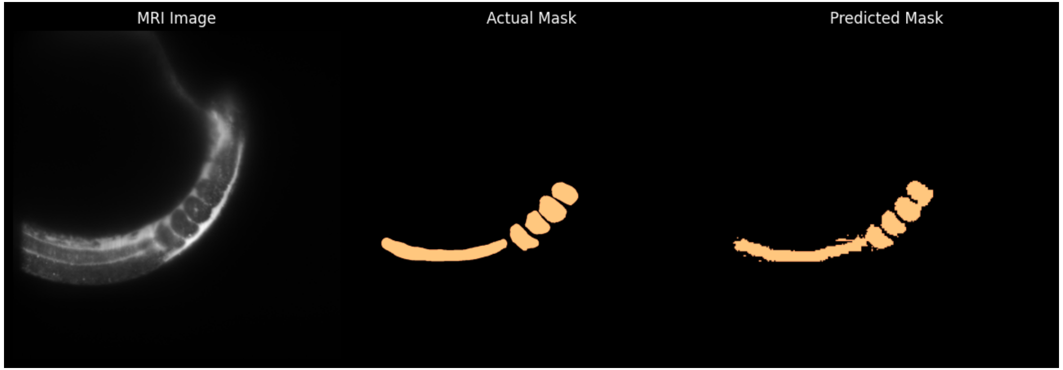


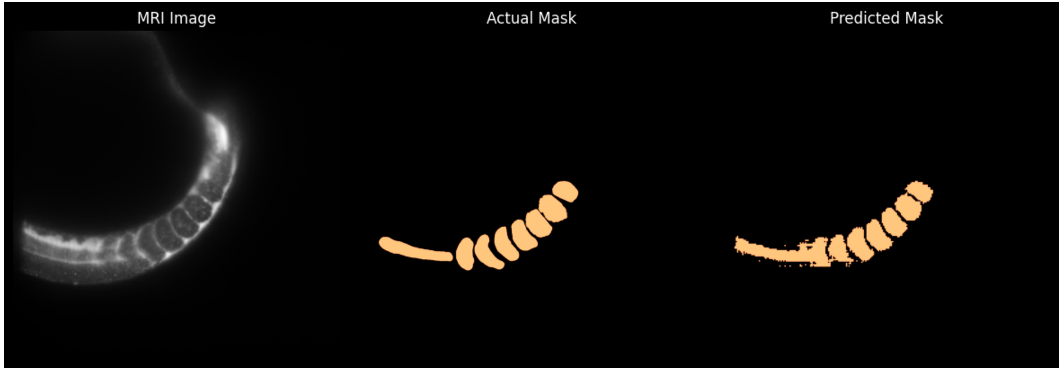
Test IoU



Results

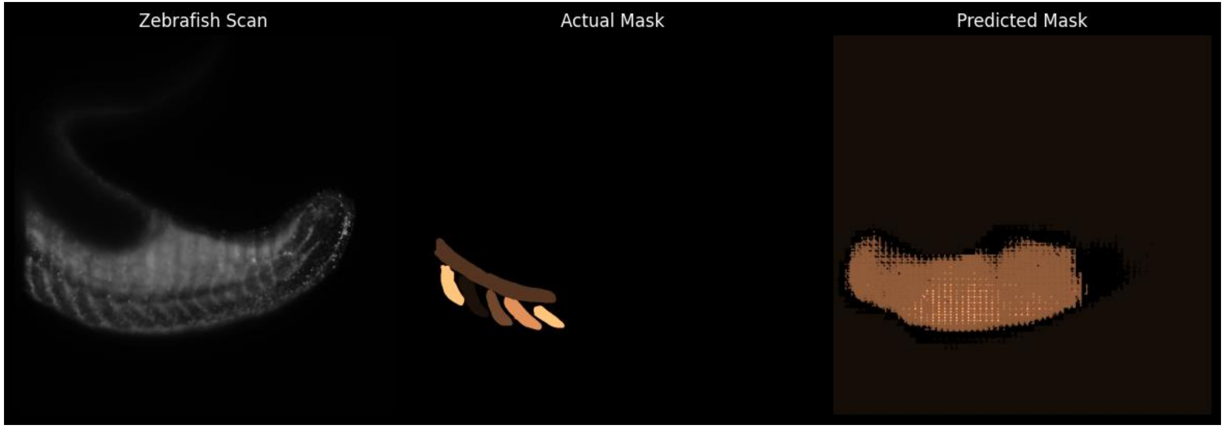
- Finetuning SAM – Version : wanglab/medsam-vit-base





Results

- Finetuning SAM – Multi-Class Image Segmentation

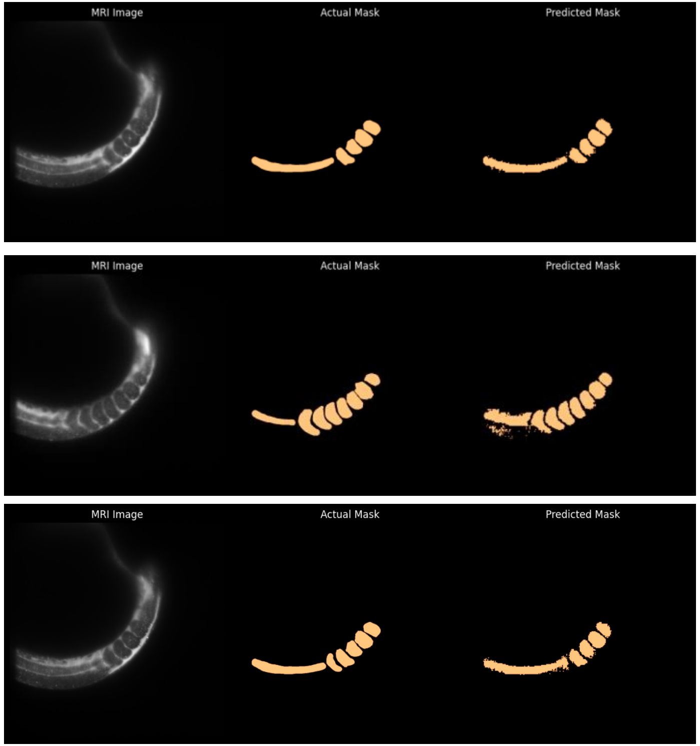


Discussion

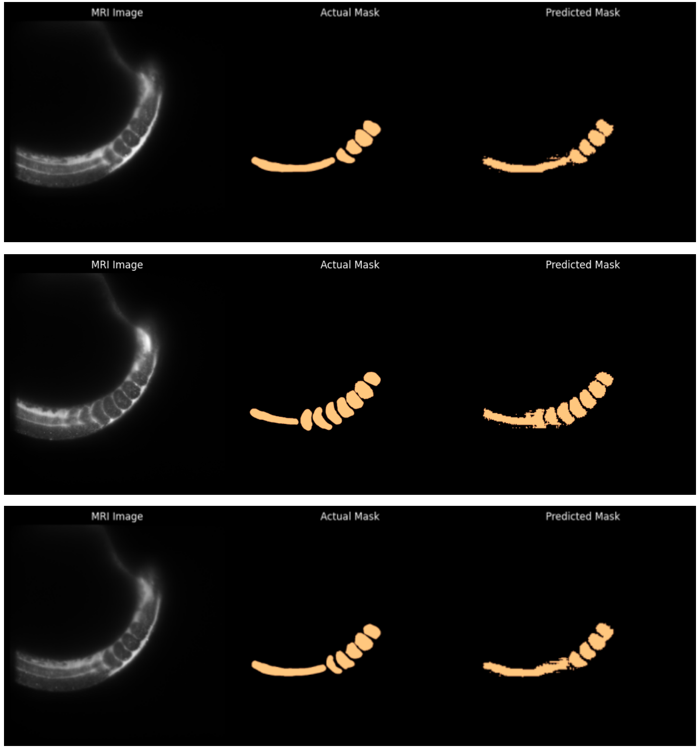
Discussion

- Results Analysis – Which Model Yields The Better Results?

facebook/sam-vit-huge Model inferences



wanglab/medsam-vit-base Model inferences



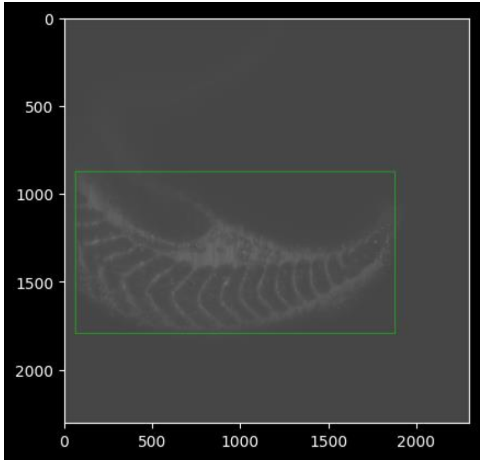
Discussion

- Drawbacks of SAM & Potential Improvements

Improvements List :

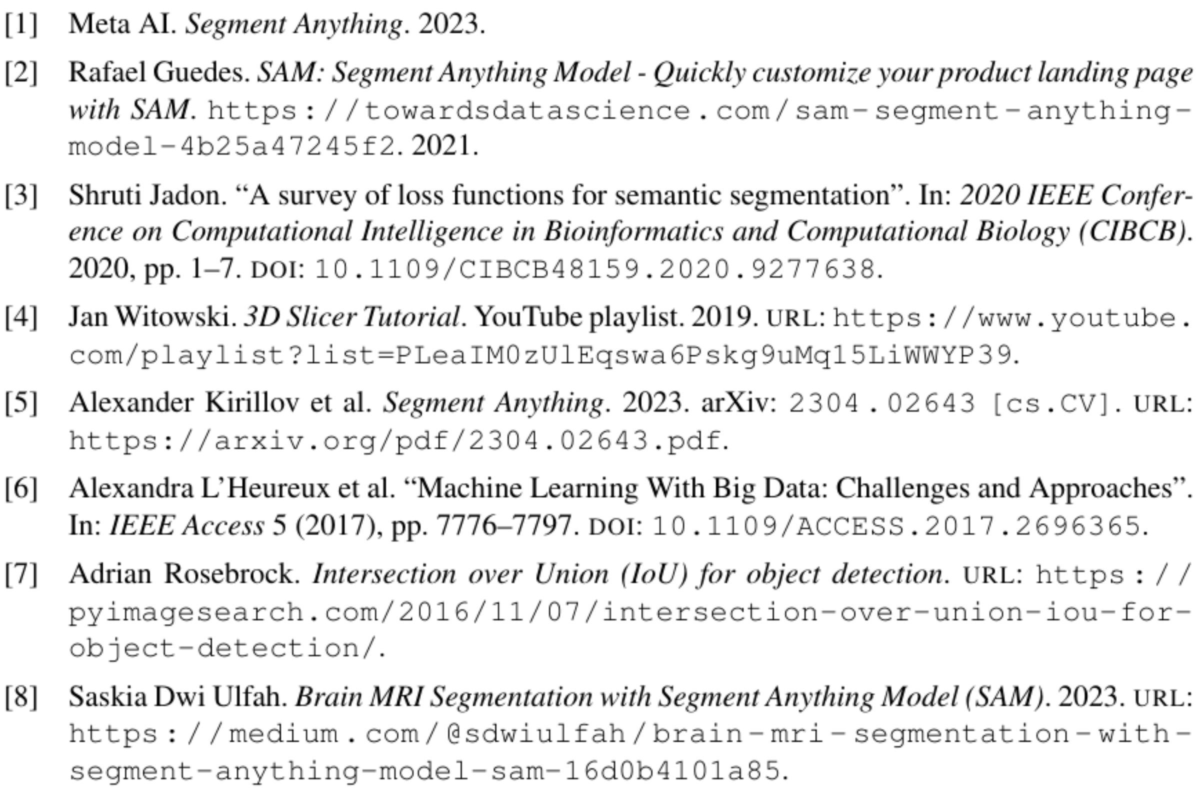
- Hyperparameter Search
- Multi-Class Segmentation Fix
- Data Augmentation & Improvements

Prompt Bounding Box Example



Summary & Conclusion

References



[1] Meta AI. *Segment Anything*. 2023.

[2] Rafael Guedes. *SAM: Segment Anything Model - Quickly customize your product landing page with SAM*. https://towardsdatascience.com/sam-segment-anything-model-4b25a47245f2. 2021.

[3] Shruti Jadon. "A survey of loss functions for semantic segmentation". In: *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. 2020, pp. 1–7. DOI: 10.1109/CIBCB48159.2020.9277638.

[4] Jan Witowski. *3D Slicer Tutorial*. YouTube playlist. 2019. URL: https://www.youtube.com/playlist?list=PLeaIM0zUlEqswa6Pskg9uMq15LiWWYP39.

[5] Alexander Kirillov et al. *Segment Anything*. 2023. arXiv: 2304.02643 [cs.CV]. URL: https://arxiv.org/pdf/2304.02643.pdf.

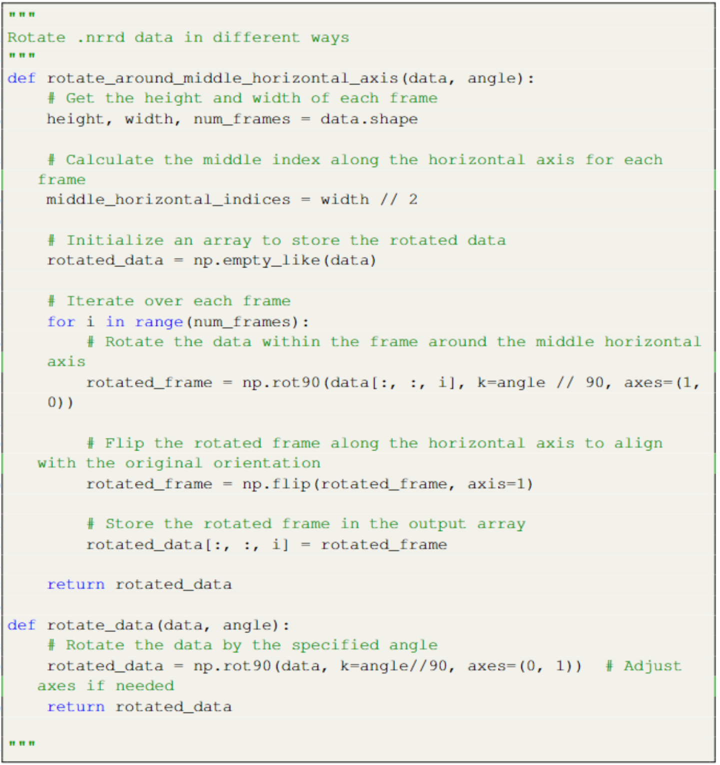
[6] Alexandra L'Heureux et al. "Machine Learning With Big Data: Challenges and Approaches". In: *IEEE Access* 5 (2017), pp. 7776–7797. DOI: 10.1109/ACCESS.2017.2696365.

[7] Adrian Rosebrock. *Intersection over Union (IoU) for object detection*. URL: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.

[8] Saskia Dwi Ulfah. *Brain MRI Segmentation with Segment Anything Model (SAM)*. 2023. URL: https://medium.com/@sdwiulfah/brain-mri-segmentation-with-segment-anything-model-sam-16d0b4101a85.

Appendix

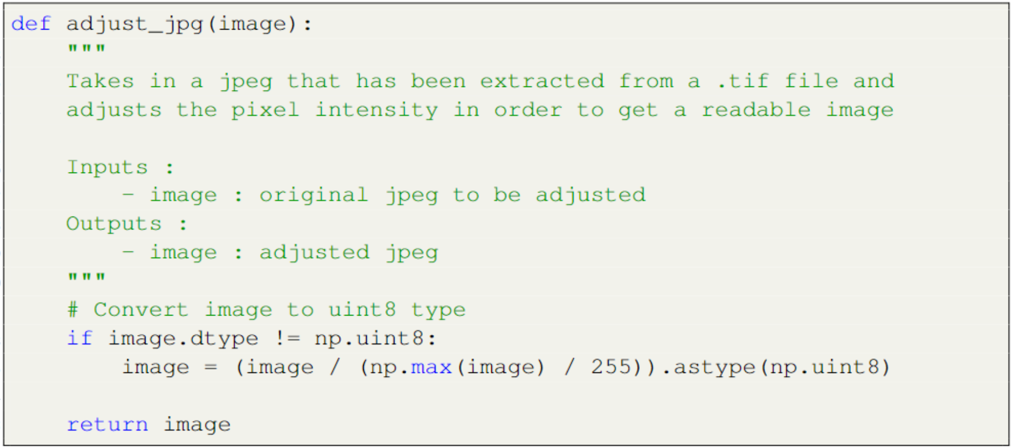
Python Functions that Allow Proper Mask Rotation



```python
"""
Rotate .nrrd data in different ways
"""
def rotate_around_middle_horizontal_axis(data, angle):
    # Get the height and width of each frame
    height, width, num_frames = data.shape

    # Calculate the middle index along the horizontal axis for each frame
    middle_horizontal_indices = width // 2

    # Initialize an array to store the rotated data
    rotated_data = np.empty_like(data)

    # Iterate over each frame
    for i in range(num_frames):
        # Rotate the data within the frame around the middle horizontal axis
        rotated_frame = np.rot90(data[:, :, i], k=angle // 90, axes=(1, 0))

        # Flip the rotated frame along the horizontal axis to align with the original orientation
        rotated_frame = np.flip(rotated_frame, axis=1)

        # Store the rotated frame in the output array
        rotated_data[:, :, i] = rotated_frame

    return rotated_data

def rotate_data(data, angle):
    # Rotate the data by the specified angle
    rotated_data = np.rot90(data, k=angle//90, axes=(0, 1))  # Adjust axes if needed
    return rotated_data

"""
```

Appendix

Python Function that Allows .tif Pixel Value Conversion

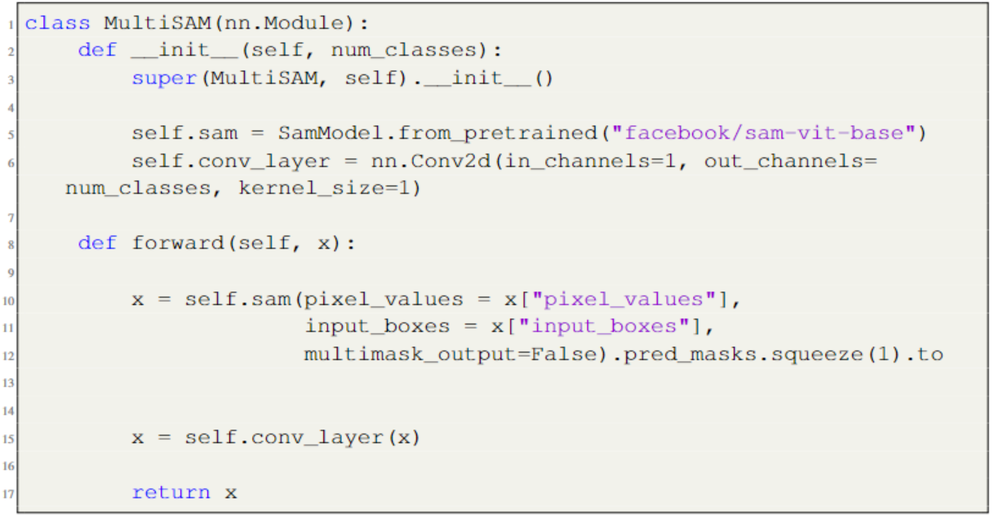


```
def adjust_jpg(image):
    """
    Takes in a jpeg that has been extracted from a .tif file and
    adjusts the pixel intensity in order to get a readable image

    Inputs :
        - image : original jpeg to be adjusted
    Outputs :
        - image : adjusted jpeg
    """
    # Convert image to uint8 type
    if image.dtype != np.uint8:
        image = (image / (np.max(image) / 255)).astype(np.uint8)

    return image
```

Speaker

Appendix

Python Code for Multi-Class SAM Model



```
class MultiSAM(nn.Module):
    def __init__(self, num_classes):
        super(MultiSAM, self).__init__()

        self.sam = SamModel.from_pretrained("facebook/sam-vit-base")
        self.conv_layer = nn.Conv2d(in_channels=1, out_channels=
   num_classes, kernel_size=1)

    def forward(self, x):

        x = self.sam(pixel_values = x["pixel_values"],
                     input_boxes = x["input_boxes"],
                     multimask_output=False).pred_masks.squeeze(1).to


        x = self.conv_layer(x)

        return x
```