Deep learning final project
Comparative study of Image Segmentation models such as segment anything and U-Net

Ankur Aggarwal
New York University
aa10336@nyu.edu

Anupam Tiwari
New York University
ast9885@nyu.edu

Abstract

This project presents a comparative study of the Segment Anything Model (SAM) and the U-net model for image segmentation. Despite considerable advancements in natural language processing foundation models, the progress in the field of computer vision for image segmentation remains relatively unchartered. SAM, a recent innovation by Meta AI, attempts to fill this gap as a generalizable, promptable foundation model for image segmentation. To understand the efficacy of SAM, we implement it on medical dataset of breast cancer images on Huggingface that SAM and UNET have not seen, later we compare their performance for image segmentation. The results suggest that SAM demonstrates superior generalization capabilities, offering significant advancements for large-scale, zero-shot image segmentation. Where a UNETS can be very domain focused. Our findings could hold considerable implications for various fields, including medical imaging, autonomous vehicles, and object recognition, where image segmentation plays a crucial role.

Introduction

Image segmentation, a fundamental task in computer vision, finds applications in diverse domains, from medical imaging to autonomous vehicles and object recognition. While natural language processing has seen significant advancements in foundation models, similar progress in computer vision, especially for image segmentation, has been slower. However, a promising development is the foundation model for image segmentation called SAM, developed by Meta AI Research. This model is trained on a broad dataset consisting 1 Billion images called (SA-1B) and is capable of powerful generalization. In this project, we aim to reproduce the SAM model's work and compare it with other encoder-decoder models like U-net for zero-shot images. We implement UNET and SAM, later we train both the segmentation models on (nielsr/breast-cancer) image Medical dataset hosted on huggingface. Later we quatify both the models and compare their performance on medical images. We also implement measure of goodness algorithms such as dice score for each model and capture the segmentation capabilities.

Model Background

SAM: SAM is a recent innovation by Meta AI that attempts to fill the gap in the field of computer vision for image segmentation. SAM is a generalizable, promptable foundation model for image segmentation. This means that SAM can be used to segment images of any kind, even those that it has never seen before. SAM does this by using a technique called prompt-based learning. Prompt-based learning is a method of training a model to perform a task by providing it with a prompt. In the case of SAM, the prompt is a description of the image that needs to be segmented. For example, if you want SAM to segment a picture of a dog, you would provide it with the prompt "segment the dog in this image." SAM would then use this prompt to generate a segmentation mask that identifies the dog in the image.

U-net: U-Net is a convolutional neural network that was developed for biomedical image segmentation at the Computer Science Department of the University of Freiburg, Germany. The architecture of U-Net is symmetric and consists of an encoding (contracting) path and a decoding (expanding) path, which gives it a U-shaped architecture. The encoding path captures the context of the input image, while the decoding path allows for precise localization making it highly effective for tasks like semantic segmentation.

The SAM model architecture is based on the ViT-B/L/H transformer architecture. The transformer architecture is a powerful neural network architecture that has been shown to be effective for a variety of natural language processing

tasks. SAM uses the transformer architecture to learn features from images that are relevant to segmentation.

The SAM model is also equipped with a decoder head that is responsible for generating segmentation masks. The decoder head is trained to generate masks that are accurate and consistent with the features learned by the transformer encoder.

The SAM model has been shown to be effective for a variety of image segmentation tasks. SAM has achieved state-of-the-art results on the COCO and Cityscapes datasets, and has been shown to be effective for segmenting a wide variety of objects, including people, cars, and animals.

The SAM model is a powerful tool for image segmentation. SAM can be used to generate segmentation masks for any object in an image, and has been shown to be effective for a variety of image segmentation tasks.

Here is a more detailed overview of the SAM model architecture:

- Transformer Encoder: The transformer encoder is responsible for learning features from images that are relevant to segmentation. The transformer encoder consists of a stack of self-attention layers. Self-attention is a mechanism that allows the transformer encoder to learn long-range dependencies between pixels in an image.

- Decoder Head: The decoder head is responsible for generating segmentation masks. The decoder head consists of a stack of convolutional layers. Convolutional layers are able to learn spatial relationships between pixels in an image.

- Loss Function: The SAM model is trained using a loss function that measures the discrepancy between the predicted segmentation masks and the ground truth segmentation masks. The loss function is a weighted combination of two terms: a pixel-wise loss term and a boundary loss term. The pixel-wise loss term measures the average pixel-wise difference between the predicted segmentation masks and the ground truth segmentation masks. The boundary loss term measures the average distance between the predicted segmentation masks and the ground truth segmentation masks.

The SAM model is a powerful tool for image segmentation. SAM can be used to generate segmentation masks for any object in an image, and has been shown to be effective for a variety of image segmentation tasks.

UNET network image

U-Net is a fully convolutional network, which means that it only has convolutional layers and no fully connected layers. This makes U-Net well-suited for image segmentation tasks, as convolutional layers are able to learn spatial relationships between pixels in an image. U-Net also has an encoder-decoder architecture, which means that it has two main parts: an encoder and a decoder. The encoder is responsible for extracting features from the input image, while the decoder is responsible for reconstructing the image with the segmented regions.

Here is a more detailed overview of the U-Net model architecture:

- Encoder: The encoder is responsible for extracting features from the input image. The encoder consists of a stack of convolutional layers, each followed by a max pooling layer. The max pooling layers reduce the spatial dimensions of the feature maps, while the convolutional layers learn spatial relationships between pixels.

- Decoder: The decoder is responsible for reconstructing the image with the segmented regions. The decoder consists of a stack of convolutional layers, each followed by a upsampling layer. The upsampling layers increase the spatial dimensions of the feature maps, while the convolutional layers learn to combine the features from the encoder with the features from the decoder to generate the segmented regions.

- Loss Function: The U-Net model is trained using a loss function that measures the discrepancy between the predicted segmentation masks and the ground truth segmentation masks. The loss function is a weighted combination of two terms: a pixel-wise loss term and a boundary loss term. The pixel-wise loss term measures the average pixel-wise difference between the predicted segmentation masks and the ground truth segmentation masks. The boundary loss term measures the average distance between the

predicted segmentation masks and the ground truth segmentation masks.

The U-Net model architecture has been shown to be very effective for image segmentation tasks. U-Net has been shown to be able to achieve state-of-the-art results on a variety of image segmentation datasets. U-Net is a versatile model that can be used for a variety of image segmentation tasks.

Dataset

We are using breast cancer dataset hosted at hugging face under nielsr/breast-cancer

Optimizers

Adam, short for Adaptive Moment Estimation, is an optimization algorithm used in deep learning models to adapt the learning rate for each weight of the neural network. It was proposed by Diederik Kingma and Jimmy Ba in 2015. Adam is a popular choice due to its effectiveness across a wide variety of deep learning tasks and architectures.

In stochastic gradient descent (SGD), all weights are updated with the same learning rate, and it is up to the user to manually adjust this rate over time. However, Adam automatically adjusts the learning rate for each weight based on the computed gradients of the loss with respect to the weight, making it an adaptive learning rate method.

Adam uses the concept of momentum by adding fractions of previous gradients to current ones. This practice helps to dampen oscillations and speed up the search for the optimal weights. Mathematically, it maintains an exponentially decaying average of past gradients (first moment) and an exponentially decaying average of past squared gradients (second moment).

Given a loss function L, and its parameters θ, the gradients of L with respect to θ are computed as g_t (where t is the timestep). The first and second moment running averages (m_t and v_t) are then updated as follows:

$$m_t = \beta1 * m_{(t-1)} + (1 - \beta1) * g_t$$
$$v_t = \beta2 * v_{(t-1)} + (1 - \beta2) * g_t^2$$

Here, β1 and β2 are hyperparameters that control the decay rates of these running averages. Typically, β1 is set to 0.9 and β2 to 0.999.

However, these moving averages are initialized as zero vectors, leading to a bias towards zero at early time steps. Adam introduces bias-corrected estimates to counter this issue:

$$m_t\_hat = m_t / (1 - (\beta1^t))$$
$$v_t\_hat = v_t / (1 - (\beta2^t))$$

Finally, the parameters are updated by:

$$\theta_t = \theta_{(t-1)} - \alpha * m_t\_hat / (sqrt(v_t\_hat) + \varepsilon)$$

Here, α is the learning rate and ε is a small constant added to improve numerical stability (usually set to 10^-8).

Adam is favored for its adaptive learning rate, low memory requirements, and suitability for problems with noisy or sparse gradients. However, it can also be sensitive to the choice of hyperparameters and sometimes may not converge to the optimal solution. Recent variants of Adam, like AdamW and AdamP, have proposed solutions to some of these issues.

Loss function

Dice Loss is a popular loss function used in image segmentation tasks. It is designed to evaluate the similarity of two samples, making it suitable for comparing a predicted segmentation mask and a ground truth mask.

Dice Loss is based on the Sørensen-Dice coefficient, which is a statistic used to gauge the similarity of two sets. For

image segmentation tasks, the two sets are the predicted and actual (ground truth) segmentation masks.

The Sørensen-Dice coefficient (Dice Score) is calculated as follows:

$$\text{Dice Score} = 2 * |X \cap Y| / (|X| + |Y|)$$

Here, $|X \cap Y|$ represents the common elements between the predicted (X) and actual (Y) sets. $|X|$ and $|Y|$ are the total number of elements in the predicted and actual sets, respectively.

For binary segmentation tasks, where X and Y are binary masks, the Dice Score simplifies to:

$$\text{Dice Score} = 2 * \Sigma (X * Y) / (\Sigma X + \Sigma Y)$$

Here, $\Sigma (X * Y)$ is the sum of element-wise multiplication of the predicted and actual masks, and ΣX and ΣY are the total counts of predicted and actual positive pixels, respectively.

Dice Loss is then defined as the complement of the Dice Score:

$$\text{Dice Loss} = 1 - \text{Dice Score}$$

In terms of the binary masks, it becomes:

$$\text{Dice Loss} = 1 - (2 * \Sigma (X * Y) / (\Sigma X + \Sigma Y))$$

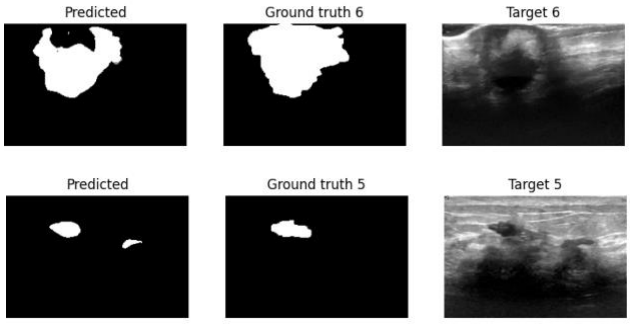
A lower Dice Loss indicates a higher overlap between the predicted and actual segmentation masks and thus signifies better segmentation performance.

It's important to note that the Dice Loss is differentiable, making it suitable for gradient-based optimization methods. It also gives more weight to the classes that hav e fewer samples (less represented), so it works well even on imbalanced data, a common issue in medical imaging tasks.

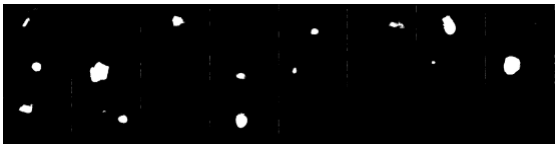
Technical details

Our project utilizes the SA-1B dataset [3] for training and the medical dataset for testing. We have implemented two models for this project - the SAM model and the U-net model. Both models use cross-entropy loss as the loss function, which is typical for segmentation tasks. We used a batch size of 20, and the models were trained for 50 epochs for u-net and 10 epochs for SAM. Since, SAM is already pre-trained on a huge dataset, we did not train it for more than 10 epochs whereas u-net is very domain and hence we used more epochs. We used the Adam optimizer with a learning rate of 0.001 for both models. All the training was carried out on a GPU for faster computations.

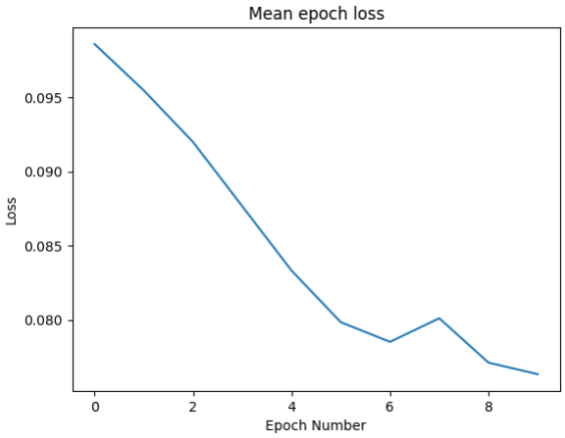
Results



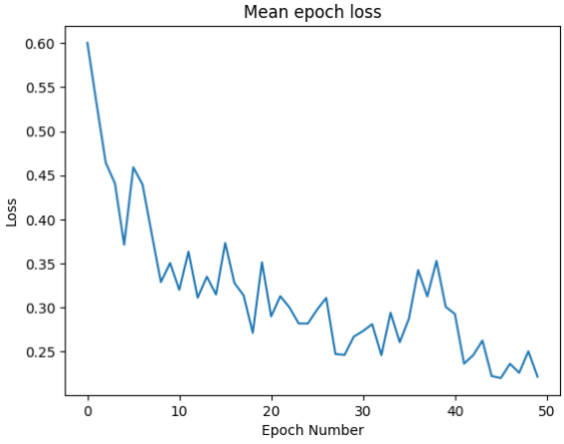
U-net ground truth vs true prediction

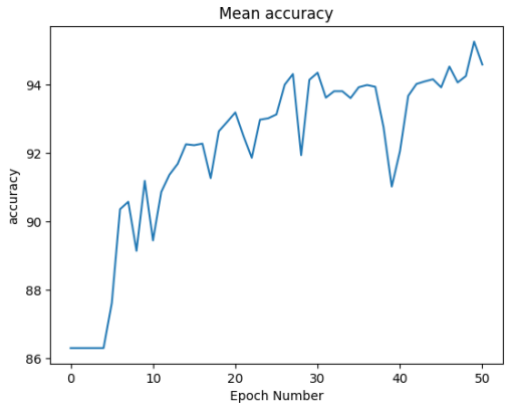


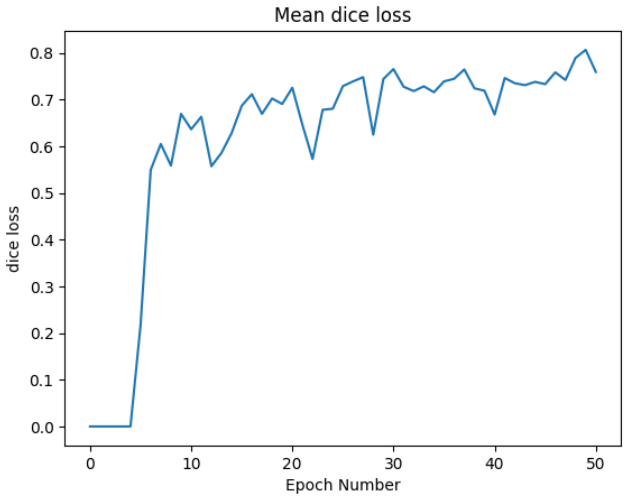
U-net image prediction phases



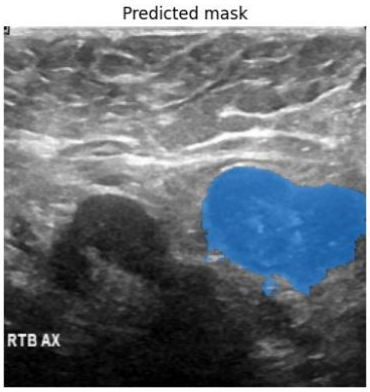
Mean loss of my SAM model

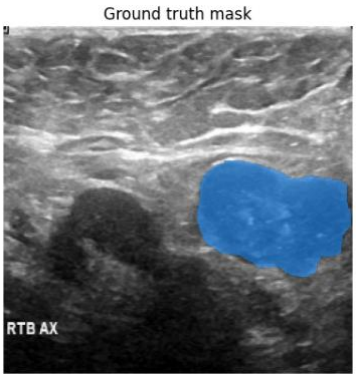






Dice vs mean square vs accuracy





Segment anything model results

Source code

The source code for this project can be accessed at:

https://github.com/anupam-tiwari/deep_learning_final_project

Conclusion

The results suggest that SAM demonstrates superior generalization capabilities, offering significant advancements for large-scale, zero-shot image segmentation. Where a UNETS can be very domain focused. Our findings could hold considerable implications for various fields, including medical imaging, autonomous vehicles, and object recognition, where image segmentation plays a crucial role. Based on the graphs from the results, we can see that u-net's performance increases with more number of epochs and we achieve a 93 percent of accuracy with a dice loss of 0.7, which is good enough for any segmentation model. Thus, u-net can handle domain specific tasks whereas segment anything models is more generalized for a large number of dataset and can be fine tuned for a specific dataset. U-net model is smaller and more efficient whereas segment anything model requires huge amount of training and resources to perform. For the future scope of this work, we plan to test more segmentation models and compare them on wide domains.

References

Aggarwal, A., & Tiwari, A. (2023). Comparative study of Image Segmentation models such as segment anything and U-Net. New York University. Retrieved from https://github.com/anupam-tiwari/deep_learning_final_project

Kirillov, A., He, K., Girshick, R., Rother, C., & Dollar, P. (2023). Segment Anything. arXiv preprint arXiv:2304.02643. Retrieved from https://arxiv.org/abs/2304.02643

Meta AI. (n.d.). SA-1B Dataset. Retrieved from https://ai.facebook.com/datasets/segment-anything/

Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., & Terzopoulos, D. (2022). Image Segmentation Using Deep Learning: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(7), 3523-3542

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional Networks for Biomedical Image Segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer.