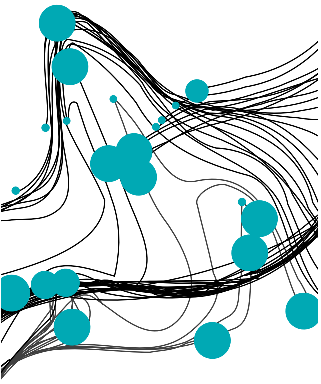
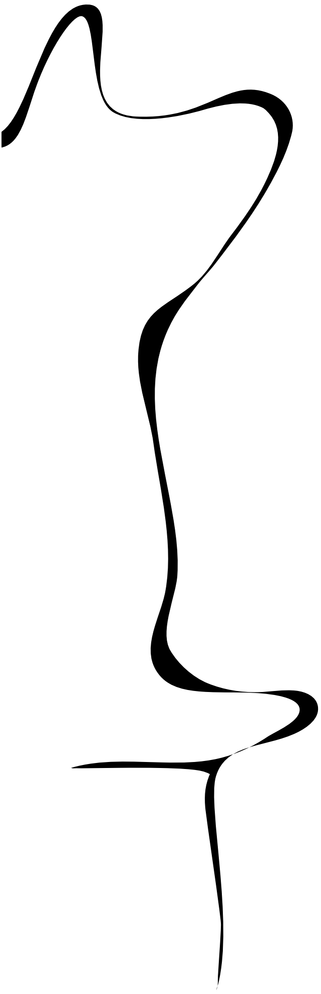


.05171

DMB

DATA MANAGEMENT
AND
BIOMETRICS



MAMMOGRAPHY SEGMENTATION USING VISUAL PROMPTS AND FEW SAMPLES

Matteo Bronkhorst

MASTER'S ASSIGNMENT

Committee:
dr. Nicola Strisciuglio
ing. Florian W. Hahn
dr. Jeroen Veltman
Sheryasi Pathak, MSc

June, 2024

2024DMB0004
Data Management and Biometrics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Mammography segmentation using visual prompts and few samples

Matteo Bronkhorst
Data Management and Biometrics
University of Twente
Enschede, the Netherlands
m.bronkhorst-1@student.utwente.nl

***Abstract*—To improve the effectiveness of mammography screening, developing better diagnostic tools is paramount. For large populations, automated annotation of patient exams can significantly alleviate the workload of radiologists. Segmentation of tumorous regions in an image provides more localization context than classification, but the lack of fine-grained segmentation labels often complicates training segmentation models directly on the data available at local hospitals. We optimize three state-of-the-art segmentation models for mammography using the CBIS-DDSM, and evaluate their performance on a manually annotated subset of our private dataset. Out of a U-Net, Segmentation Transformer, and Segment Anything Model, the latter performed best by a large margin on the CBIS-DDSM, achieving an IoU of 27.42% as opposed to the U-Net achieving 9.11% and the Segmentation Transformer with 8.26%. However, performance on this public mammography dataset was unrepresentative of the zero-shot transfer performance on our private dataset. Future research should focus on similarly assessing the usability of other public mammography datasets for training diagnostic tools that are effective in clinical settings.**

***Index Terms*—Mammography segmentation, deep neural network, mammography images, pixel-wise label, region of interest, zero-shot transfer**

I. Introduction

In 2020, breast cancer was the world's most prevalent form of cancer with 2.26 million new cases and 685 thousand deaths[1][2]. Earlier treatment of breast cancer improves outcomes[3], and so does mammography screening in asymptomatic populations[4]. Manual screening relies on expert clinicians, resulting in a workload that increases with population size. Thus the aim of developing automated methods of breast cancer diagnosis is to improve the diagnostic tools for clinicians and alleviate their workload by acting as an additional reader.

Placing a focus on aiding the diagnostic process means that a binary classifier of malignancy is not enough; it will not provide a radiologist with information about where a tumor might be, or what the shape of the tumor is. Instead, we focus on a task where the output provides localization context: semantic segmentation of mammographic images. Semantic segmentation entails assigning a class label to each pixel of an image. In our work, we apply segmentation models to a private mammography dataset recently collected by local hospital Ziekenhuis Groep Twente (Hospital Group Twente). This dataset is annotated with image-level labels, but mostly lacks pixel-wise class annotations. To allow the evaluation of segmentation models when applied to this private dataset, 206 images were recently annotated by a radiologist at ZGT.

A. Challenges and Contributions

In this work, we address **two challenges**.

1) Local hospitals lack pixel-wise annotations for their datasets. Due to privacy concerns, medical data collections are generally kept private. Consequently, the responsibility for fine-grained annotation of these private datasets falls on the hospitals. This prevents hospitals from using their in-house mammography datasets for training segmentation models with pixel-wise supervision.
2) Public mammography datasets are small when compared to commonly used natural image datasets. Both natural image datasets with object annotations (CIFAR-10[5], SA-1B[6], ADE20K[7]) and without object annotations (ImageNet[8]) surpass the sizes of public mammography datasets[9, 10]. State-of-the-art computer vision models thrive especially in high-data regimes[11][6]. This hinders using large segmentation models for mammography segmentation.

In combination, these challenges make it difficult for local hospitals to understand the effectiveness of state-of-the-art segmentation models for use within their organization. We address these challenges with a **threefold of contributions**.

1) We establish a framework for training on public mammography datasets and evaluating on unseen, out-of-distribution data from a different clinical source. This allows us to assess the zero-shot performance of models on our private dataset. The highest overlap we achieve on the private dataset is 31.91% (IoU) with a U-Net. Importantly, we conclude that model performance varies greatly between the CBIS-DDSM and the private dataset.
2) With this framework, we compare three state-of-the-art segmentation models trained with pixel-wise supervision on the CBIS-DDSM. We find that the U-Net[12] outperforms the Segmentation Transformer[13], and that the Segment Anything Model[6] greatly improves over both.
3) For the Segmentation Transformer, we compare the effectiveness of full fine-tuning to Visual Prompt Tun-

ing[14]. Our results show that VPT produces equal if not better performance than full fine-tuning. This indicates that parameter-efficient fine-tuning methods can be suitable for the mammography domain.

B. Research Question

In order to keep our research relevant to local hospitals, we pose the following research question: **”How effectively does segmentation performance on public mammography datasets transfer to the private dataset collected by Ziekenhuis Groep Twente?”** This phrasing is general enough to remain useful for other local hospitals, while still measurable and specific.

C. Background

To familiarize the reader with the Vision Transformer (ViT) based models we have used in our research, we shortly describe the developments that led to their emergence. The field of Natural Language Processing has seen a lot of recent progress, largely induced by the debut of the Transformer and the emergence of large pre-trained models. Works like BERT[15] and GPT[16] result in pre-trained models often referred to as ‘foundation models’, as they provide a solid base for fine-tuning to various downstream tasks. Subsequent to this progress in NLP, the field of computer vision has seen innovation in the form of the Vision Transformer. This Transformer variant specially designed for vision tasks has since competed with the more traditional CNNs. With the Vision Transformer as a basis, models like the Segmentation Transformer[13] and the Segment Anything Model[6] were developed to extend the Vision Transformer’s capabilities beyond classification to segmentation. The Segmentation Transformer and the Segment Anything Model are considered foundation models for segmentation tasks.

The different ways in which a foundation model can be fine-tuned to a downstream task are legion. One method of particular interest to our work is Visual Prompt Tuning[14], which has been developed specifically for Vision Transformer architectures. Visual Prompt Tuning works well when few training annotations are available for the downstream task, which means it should be well-suited for mammography. We detail the differences between the Vision Transformer, Segmentation Transformer, and Segment Anything Model in Section II-B.

II. Related Works

We discuss a number of works that are similar to ours, on which we build, or are otherwise relevant to our research.

A. Mammography

1) Datasets: The New York University Breast Cancer Screening Dataset v1.0 (NYU BCSD v1.0)[17] was introduced in 2019. It is a mammography screening dataset of considerable size, counting 229,426 patient exams, totalling 1,001,093 digital images. While the dataset itself has been kept unpublished, the details of how it was collected and curated are available in a public report by Wu *et al.* [17]. Such a report can contain useful lessons for the collection of similar datasets at local hospitals. They mention that a total of 5,832 exams led to biopsies within 120 days after the exam. These exams have been annotated with pixel-level cancer labels. These annotations were collected by providing a group of radiologists with the corresponding pathology reports and asking them to retrospectively annotate lesions that had been selected for biopsies. Out of the total 5,832 exams presented for annotation to the radiologists, 3,917 exams were marked with at least one region. While this may seem like a large dataset, it has not been made available publicly, and it should be emphasized that out of 229,426 total exams, 3,917 exams (1.7%) have been annotated with pixel-level labels. In comparison to other mammography datasets[9][10], this makes it a large mammography dataset in the context of unsupervised segmentation, but not when we consider the pixel-wise supervised learning task.

The Curated Breast Imaging Subset of Digital Database for Screening Mammography (CBIS-DDSM)[9] is a curated, updated and standardized subset of DDSM. While CBIS-DDSM was published in 2017, the underlying DDSM, a set of scanned film mammography studies, was collected in the United States and released in 1997. This makes the DDSM a rather old dataset, and since the exams were initially stored on film, the scanning of film mammograms has likely introduced an extra layer of noise and other visual artefacts. The publication of CBIS-DDSM in 2017 has allowed plenty of methods to be trained and evaluated on this public dataset. This abundance of research with this dataset makes it a suitable choice for use in our work to enable comparison to other research. The CBIS-DDSM is a dataset with 1,624 patient exams with Region Of Interest (ROI) annotations, for 1,566 patients. The exams contain 1644 annotated cases of which 753 are calcifications and 891 are masses. These exams contain 3,032 images, with 3,103 annotated image regions. Each image contains at least one annotated region, and a few images contain more than one region.

The VinDr-Mammo dataset collected in Vietnam, released in 2022, is a mammography dataset that contains 20,000 images from 5,000 studies. Its annotations include rectangular regions of interest for exams that needed follow-up examination. The work by Nguyen *et al.* [10] accompanying its publication presents a summary of mammography datasets, including the NYU BCSD v1.0 and the CBIS-DDSM. It shows that datasets have differing levels of annotations for regions of interest and that the number of studies per dataset varies wildly. The summary mentions four datasets that contain annotations of the type “contour enclosing the finding” (i.e. pixel-level annotations). These four are DDSM, INBreast, NYU Dataset, and CSAW-CC. As mentioned previously, the NYU Breast Cancer Screening dataset is not publicly available and cannot be used in our work. INBreast is small compared to the other three: it consists of 115 studies. The two datasets most suitable for our purposes are (CBIS-)DDSM and CSAW-CC. CSAW-CC concerns a carefully compiled set of 8723 patients in total, of which 7850 were selected from a healthy control and 873 were diagnosed with breast cancer. For each image, CSAW-CC includes the final diagnosis (cancer or no-cancer.) This is different from the malignant/benign labels for CBIS-DDSM, but makes it no less qualified for developing tools that assist in malignancy detection and localization. We have not used the CSAW-CC dataset in our experiments.

The private dataset we use in our work has been collected at Ziekenhuis Groep Twente (ZGT), and provides image-level class labels. It consists of 84,299 images, 15,991 patients, and 21,013 exams. It concerns exams that were conducted anywhere from 2013 through 2020, all of them digital. The total number of images is 4 times as many as VinDr-Mammo, and about 27 times as many as CBIS-DDSM. To facilitate the evaluation of segmentation models, 206 images have recently been annotated by a radiologist at ZGT. Similar to the collection of annotations for the NYU BCSD[17], the radiologist was presented with pathology reports corresponding to the patient exams from which the images were taken. The radiologist was asked to retrospectively annotate regions of interest.

2) Segmentation: Michael *et al.* [18] present a survey of mammography segmentation. They arrange the methods into three categories: classical, machine learning, and deep learning. For each work, they mention reported metrics but do not clearly distinguish per work whether metrics are reported over full mammograms or cropped regions of interest. This makes it difficult to objectively compare the performance of the various works listed in the survey. They conclude that the deep learning methods seem to be the most promising and that the U-Net is a popular choice because it requires few annotated images.

The **U-Net** is a convolutional model developed specifically for biomedical image segmentation[12]. It involves an encoder and decoder with residual connections. The encoder performs a step-wise compression of the feature space, and the decoder has a shape inverse to that of the encoder, allowing it to expand the feature space and produce a segmentation map. Each level of the encoder connects through a residual connection to the corresponding level of the decoder, allowing the more narrow parts of the network to focus on high-level patterns. The U-Net has been applied numerous times to CBIS-DDSM: Connected-UNets[19], Connected-SegNets[20], ConnectedUNets++[21], and Mammo-SAM[22] use the U-Net as a baseline to compare other methods to. This positions the U-Net as a suitable baseline candidate in our work, facilitating direct comparison to these other works.

An example of research that reports metrics over cropped regions, mass regions in particular, is Connected-UNets by Baccouche *et al.* [19]. Connected-UNets combines two U-Nets sequentially, adding extra residual connections between the two networks. It also integrates Atrous Spatial Pyramid Pooling in its proposed architecture. In advocacy for its proposed architecture, a comparison is made to the Dice and IoU score of various other models. On the CBIS-DDSM, their standard U-Net achieves a test score of 64.87% IoU and 78.62% Dice score. They conclude that their proposed Connected-UNet architecture outperforms the other experiments, including the standard U-Net.

ConnectedUNets++ by Sarker *et al.* [21] bases on Connected-UNets and introduces an improved iteration over their model. Additionally, ConnectedUNets++ performs mass segmentation on full mammographic images instead of cropped regions. This results in drastically lower reported metrics: a standard U-Net achieves 27% IoU and 41% Dice score on the CBIS-DDSM test set. Similarly to Baccouche *et al.*, Sarker *et al.* conclude that their proposed ConnectedUNets++ architecture outperforms the other architectures, including the standard U-Net.

In contrast to the supervised U-Net and models with similar task definitions, there also exist unsupervised approaches to mammography segmentation. Examples are GMIC[23] and GLAM[24]. The Globally-aware Multiple Instance Classifier (GMIC) by Shen *et al.* [23] presents a classification method designed to tackle two challenges that arise from properties that differentiate medical images from natural images: higher resolutions and (usually) smaller regions of interest. GMIC does this by combining a memory-efficient but coarse global network to identify regions of interest, and a more high-capacity network to collect details from the identified regions. Importantly, it combines the information computed over different patches to make the final class prediction. Its global module can be used for segmentation purposes.

Global Local Activation Mapping (GLAM) by Shen *et al.* [23] goes further than GMIC but uses a similar approach. Just like GMIC, it makes use of a global, local, and fusion module, and it is capable of discerning between malignant and benign lesions. In GLAM, both the global and the local modules output a segmentation map, while in GMIC only the local module produces a saliency map. Both GMIC and GLAM are trained for region selection in a weakly supervised manner, only relying on image-level labels and the multiple-instance nature of region selection. GMIC and GLAM were trained and evaluated on the NYU dataset, resulting in per-class Dice scores of (32.5%, 24.0%) and (39.0%, 33.5%) for GMIC and GLAM respectively, for classes (*malignant*, *benign*). Compared to a supervised U-Net trained on the pixel-level annotated subset of the NYU dataset which evaluated at Dice scores of (50.4%, 41.2%), the GMIC and GLAM approaches are less precise in segmenting regions of interest. Nevertheless, in the absence of pixel-level annotation, weakly supervised methods like these are worth considering for use by local hospitals.

B. Foundation models

The term 'foundation models' refers to large-scale models pre-trained on large datasets to support a wide range of downstream tasks. Since the introduction of Transformers by Vaswani *et al.* [25] in 2017, foundation models have revolutionized the field of Natural Language Processing. Examples of well-known Transformer-based foundation models are BERT[15], Generative Pre-trained Transformers[16], and successors of these works. A common factor in many recent

Large Language Models (LLMs) is their use of large text corpora for pre-training for a task that induces the model to learn a general form of language representation/understanding, and their remarkable performance on subsequent more specific downstream tasks.

The Transformer's rise to prominence in NLP has prompted research into how to effectively utilize similar architectures for Computer Vision, resulting in the Vision Transformer by Dosovitskiy *et al.* [11] and the Swin Transformer by Liu *et al.* Both tackle an important difference in information density between text and vision: while text is made up of information-dense units (words), it is less straightforward to directly represent images in units of high information density. The Vision Transformer (ViT)[11] does this by subdividing an image into a set of equal-sized patches and applying a linear projection to each patch to produce a visual token for each patch. Together with added position embeddings, these tokens are then fed to the Transformer. The ViT architecture exhibits a quadratic relationship between input size and computational cost of the attention mechanism[11]. Shifted Windows Transformers (Swin Transformers)[26] take a different approach: the Swin Transformer divides an image into separate local windows within which self-attention is performed. Keeping the number of patches within a local window constant means that while the amount of local windows in an image increases linearly with image resolution, the computational cost of the attention mechanism within the local window stays constant, leading to a linear relationship between input size and computational cost.

In Figures 1a and 1b we illustrate the architecture of the ViT. Figure 1a highlights the main contributions of Dosovitskiy *et al.*: dividing the image into patches, and linearly projecting these patches. Following the Vision Transformer, plenty of research has produced models that incorporate ViTs as a cornerstone for their architecture. Among these are also the Segmentation Transformer and the Segment Anything Model which we list in Section II-B1.

1) Segmentation Models: There exist many models pre-trained for segmentation tasks. In this section we detail the two most relevant to our experiments. The **Segmentation Transformer** (SETR) by Zheng *et al.* [13] is a ViT-based model designed for semantic segmentation. The SETR has an encoder-decoder structure, where the encoder is a ViT, and the decoder can have various designs. The three decoders explored by Zheng *et al.* [13] are *Naive upsampling (Naive)*, *Progressive upsampling (PUP)*, and *Multi-level feature aggregation (MLA)*. In Figures 1a and 1c we present diagrams to visually demonstrate the correspondence between the ViT (Figure 1a) and SETR. The implementation of SETR by Zheng *et al.* was based on the MMSegmentation toolbox[27], which provides a framework with building blocks for implementing semantic segmentation experiments. The SETR has since been incorporated into the MMSegmentation toolbox, paired with reproduction experiments. While Zheng *et al.* [13] found the MLA decoder to be most effective, the subsequent reproduction by the authors of MMSegmentation championed the PUP decoder. In our experiments, we will be using the SETR-PUP architecture.

The **Segment Anything Model** by Kirillov *et al.* [6] combines a number of previously developed models including CLIP[28] and ViTDet[29]. The contribution of Kirillov *et al.* [6] is threefold:

- Dataset: The SA-1B is the largest segmentation dataset to date, consisting of 1.1 billion masks and 11 million images.
- Task: a redefinition of the segmentation task as promptable segmentation. The goal of the task is to produce a valid segmentation mask given any segmentation prompt.
- Model: trained on the SA-1B dataset for the redefined task, a modular Segment Anything Model that incorporates a ViT-based image encoder, a prompt encoder, and a decoder.

The prompts used by the Segment Anything Model capture quite some use cases: points, bounding boxes, masks, and text.

SETR and SAM are not the only pre-trained segmentation models available. Since we wish to focus on Vision Transformers, and both SETR and SAM incorporate a Vision Transformer into their respective architectures, these models are well-suited for our research. Other ViT-based segmentation models definitely exist, like Segmenter[30], HRViT[31], and many more mentioned in a survey on semantic segmentation with ViTs by Thisanke *et al.* [32]. We do not delve into their details, because we will not be using them in our experiments. This is not to say that they are not viable candidates for mammography segmentation, or that we've determined SETR and SAM to be in some way superior to these other models.

C. Fine-tuning methods

When considering how to optimize a pre-trained Vision Transformer to a different downstream task, there are plenty of fine-tuning strategies to choose from. The options include but are certainly not limited to:

- Full fine-tuning: fine-tuning all parameters in the network.
- Partial fine-tuning: fine-tuning parts of the network, while freezing other parts of the network. For example, freezing the first *n* layers, and fine-tuning the later layers like done by Yosinski *et al.* [33].
- Adapter-based methods: freezing the model paired with inserting adapter layers into the model, helping to augment the latent representation of each layer with new information that is beneficial to the end task. Examples of this are Explicit Visual Prompting[34], and Low-Rank Adaptation[35] (LoRA). In the latter, trainable low-rank matrices are injected into the model. LoRA reduces the amount of trainable parameters, is considered more parameter efficient than full fine-tuning, and according to Hu *et al.* [35] it lowers GPU memory requirements.
- Prompt-based methods: introduce learnable inputs to learn task-specific prompts. An example called **Visual Prompt Tuning** (VPT)[14] concatenates sets of learnable parameters to the input of each layer of the Vision Transformer.

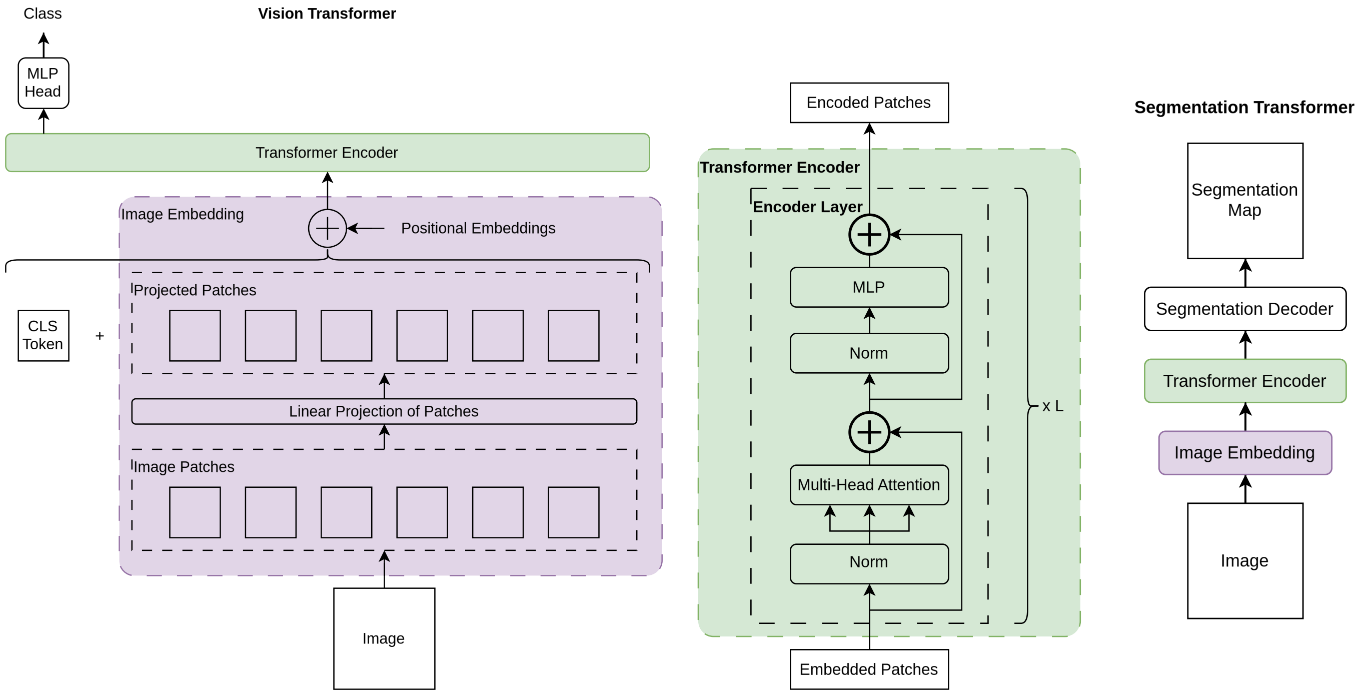


Fig. 1: Schematic overviews: The Transformer Encoder indicated with green, the Image Embedding indicated with purple. (a) Vision Transformer architecture introduced by Dosovitskiy *et al.* [11]. *CLS Token* indicates the learned special token prepended to the projected patch embeddings. The MLP classification head takes as input the final encoded embedding at the same index as this classification token was inserted (i.e., index 0). (b) Transformer Encoder used as part of the Vision Transformer, as presented by Dosovitskiy *et al.* [11]. *L* indicates the number of encoder layers. Similar types of encoders are used in works like BERT[15] and GPT[16]. (c) Schematic overview of the SETR architecture introduced by Zheng *et al.* [13]. In contrast to the ViT, no CLS token is inserted in the image embeddings for the Segmentation Transformer. All final embeddings are used as input to the decoder head. The resulting segmentation map assigns each pixel one of the possible classes.

D. Foundation models applied to Mammography Segmentation

At the moment of writing, there are few works that apply either the Segmentation Transformer or the Segment Anything Model to the mammography domain. This indicates a gap in research. We have found only three works that apply the Segment Anything Model to mammography. These are discussed below. Looking a bit broader, we observe that other Transformer variants have been applied to mammography, but these either use Swin Transformers or propose modifications that seem rather complex and specific. We do not discuss their details, because they are difficult to relate directly to our research into fine-tuning pre-trained models.

Both Ahmadi *et al.* [36] and Hu *et al.* [37] have applied SAM without fine-tuning to breast tumor detection. Hu *et al.* [37] focused on ultrasound imaging and exploring the behaviour of SAM at various ViT backbone sizes. They did not use mammography datasets. Ahmadi *et al.* [36] applied SAM to both ultrasound and mammography, and compared its performance to a U-Net trained on a breast ultrasound dataset. They report that the U-Net was trained on mammography images and that it outperforms the pre-trained SAM. While they claim that their findings highlight the importance of selecting deep learning architectures tailored for medical image segmentation, it seems that their comparison is skewed: they apply the Segment Anything Model in a zero-shot fashion, without adapting it to the target dataset. Their conclusion that the Segment Anything Model is ”less adaptable to various tasks and datasets” than the U-Net architecture seems unsubstantiated when they have evaluated the performance of SAM without optimizing for mammography.

A recent work by Xiong *et al.* [22], titled Mammo-SAM, mentions a crucial domain gap between the medical domain and the SA-1B dataset on which SAM has been trained. To bridge this gap, they designed an adapter-based fine-tuning method for SAM. Their adapter-based fine-tuning method is different from what we explore in this paper, since we have not used adapters to fine-tune the Segment Anything Model. Rather, we have applied full fine-tuning to SAM. We also investigated the possibility of applying a different parameter-efficient method to the SAM: Visual Prompt Tuning (see Section VI.) There is another major point in which their work differs from ours: they remove the prompt encoder. In doing so, they divert from the promptable segmentation task that

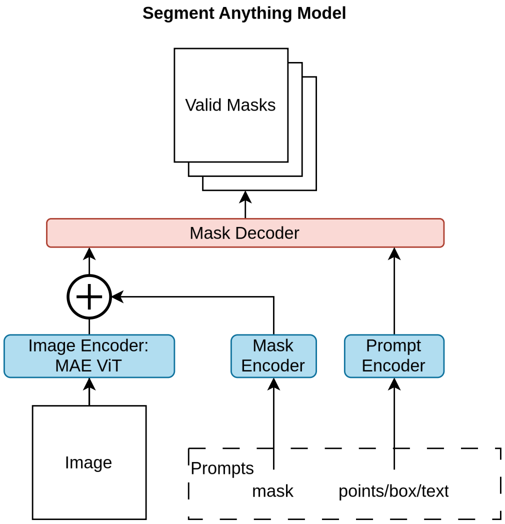


Fig. 2: Schematic overview of the SAM architecture introduced by Kirillov *et al.* [6]. Encoding elements indicated with blue, decoder with red. Compared to the SETR, we see that SAM is markedly different in its inclusion of segmentation prompts and a prompt encoder, and that it produces three candidates for a valid mask instead of one. It uses a ViT-based image encoder pre-trained for masked auto encoding (MAE). The Mask Encoder is made up of convolutional layers. The Prompt Encoder represents points and boxes using positional encodings and learned embeddings. The Prompt Encoder uses CLIP[28] for encoding text prompts.

was presented by Kirillov *et al.* [6]. The motivation supplied for removing the prompt encoder is that their work focuses on automatic segmentation without manual prompts. In our work, we keep the prompt encoder because we think segmentation prompts will be useful for good segmentation performance on small regions of interest.

III. Methodology

In this work, we aim for an approach that is deployable in several hospitals by adapting it to specific characteristics of the datasets and problems at hand. The research question posed in Section I-B can be asked by other local hospitals than just Ziekenhuis Groep Twente. At the same time, ZGT is not the only hospital with a collection of private mammography images lacking segmentation labels. Manually collecting segmentation labels for the entire private dataset and using them to train supervised segmentation models would allow for the best match between the trained model and the task it is applied to in a clinical environment. Applying this approach at another hospital would again involve the arduous process of manually labelling a large mammography dataset. To achieve similar results across hospitals, each hospital would need to produce annotations of similarly high quality. Instead, we present an approach that can be applied with comparatively little manual annotation involved. This means using public datasets where possible and only collecting manual annotations for testing.

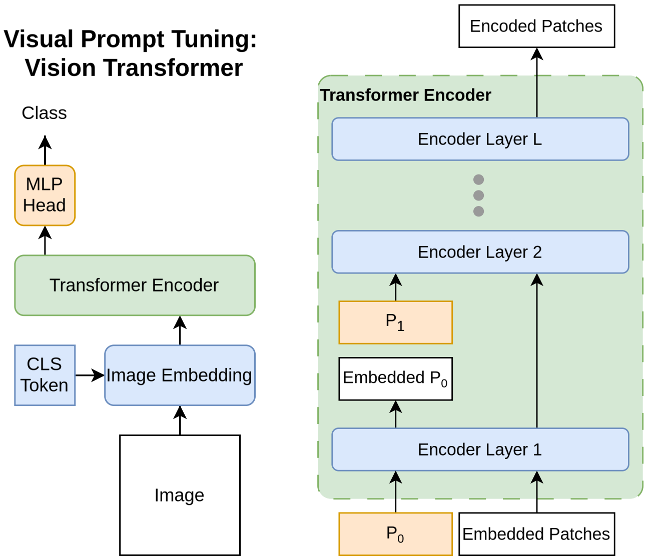


Fig. 3: Schematic overview of VPT-Deep, as introduced by Jia *et al.* [14]. Blue and orange indicate frozen and tuned parameters, respectively. **(a)** High-level overview of Visual Prompt Tuning applied to the Vision Transformer. The Transformer Encoder is shown in green, because it is partly tuned and partly frozen. **(b)** A more detailed visualization of the Transformer Encoder during Visual Prompt Tuning of the Vision Transformer, according to [14]. Before each encoder layer, learnable prompts are prepended to the embedded patches. Before each encoder layer, learnable prompts are prepended to the embedded patches. Before feeding to each next layer, the embedded prompts are discarded and replaced with another layer of learnable prompts.

A. Approach

In Figure 4 we schematically illustrate our approach to optimizing (pre-trained) models for mammography segmentation and evaluating them on an unseen dataset. In the sections below, we discuss the models we have selected to compare through this framework, and the methods for optimizing these models for the public mammography dataset. We give a detailed definition of the task that the models are meant to solve in Section III-A3. In Section III-B we describe the loss functions used for each model, and the metrics we use to analyze the performance of each experiment.

1) Selected Models: The models that we have selected for our experiments are:

- U-Net[12]
- SETR[13]
- SAM[6]

The models that have performed exceptionally well on computer vision benchmarks throughout the last decade, and seem most promising for our purpose, are often either based on convolutions or attention mechanisms. Our reason for experimenting mainly with Vision Transformer-based models is that ViTs have been shown to excel compared to similarly

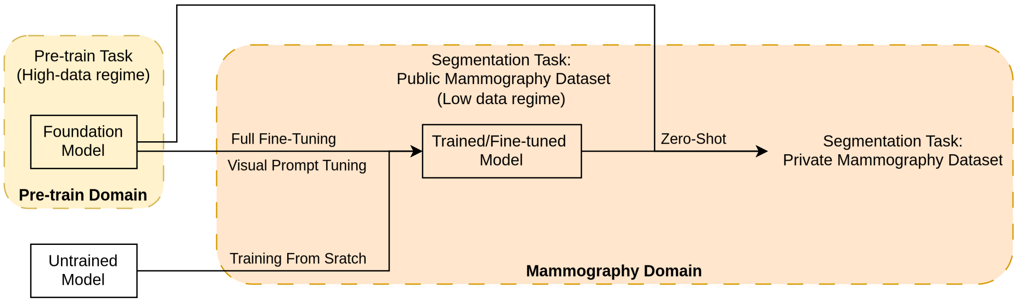


Fig. 4: Main overview of our framework for optimizing foundation models for mammography. We take pre-trained foundation models and optimize them for a public mammography dataset, after which we apply them to a private dataset that the models have not seen before. As an alternative to pre-trained models, we also include untrained models that we directly optimize for mammography without pre-training.

sized ResNets when (pre-)trained on larger datasets[11]. This suggests that it is easier for ViTs to absorb more information from larger datasets than convolutional models do. Given that the ViT is pre-trained on very large datasets, the right training or fine-tuning method might result in better performance on downstream tasks like mammography.

2) Model Training and Adaptation: To optimize models for the segmentation task on a mammography dataset of choice, we consider three options:

- Training the model from scratch.
- Zero-shot application of a pre-trained model.
- Fine-tuning a pre-trained model using training samples from the downstream task.

It is important to note that these three options are not applicable to every model. Since the Segment Anything Model has been trained on the largest segmentation dataset to date, we refrain from training the same architecture from scratch. It seems likely that a model of this size would overfit on most mammography datasets due to their limited size.

The modus operandi for the U-Net is to train from scratch[19–21]. We have not identified suitable pre-trained U-Net models. Since we are focusing on the use of ViT-based models as foundation models, we will not be treating the U-Net as a foundation model; we do not apply any pre-trained U-Net in zero-shot fashion, and we do not fine-tune any pre-trained U-Net. We only optimize the U-Net by training from scratch. Our experiments include the U-Net as a CNN-based complement to the ViT-based Segmentation Transformer.

In Section IV we detail the exact combinations in which models, optimization methods, and datasets have been applied. For the Segmentation Transformer experiments, we use the SETR-PUP architecture, as this is the architecture found most effective in the reproduction experiments conducted by the authors of the MMSegmentation toolbox. For the Segment Anything Model, we limit our focus to point prompts.

To analyze the effectiveness of parameter-efficient tuning methods in the mammography domain, we include Visual Prompt Tuning in our framework as an alternative to full fine-tuning. This method of fine-tuning is designed for Vision Transformers and is not directly applicable to the U-Net.

3) Task Definition: Generally, mammography segmentation centres around identifying regions of interest from an input image. For the U-Net and the SETR, the only input given is the full image, and we task the model with identifying and marking any regions of interest in the image. In some cases, there are multiple separately annotated regions of different abnormality classes. We have merged these regions into one binary ground truth map per mammogram. The U-Net and SETR can both be configured to output a segmentation map that maps each pixel to one of N classes, so it is possible to approach multi-class segmentation for mammography. Still, for simplicity, we have lumped all abnormality classes together under the label ”Region of Interest”. This results in a binary output map (i.e. $N = 2$) by both the SETR and U-Net.

From the three models we have selected, the SAM is special in the inputs it accepts, and the outputs it produces. It takes extra inputs in the form of prompts, and it does not produce a segmentation map assigning one of variable N classes to each pixel. The SAM is not constructed to give semantic information about an object, but only to produce a valid binary mask from an input image and any accompanying prompts. More precisely, the SAM produces three separate

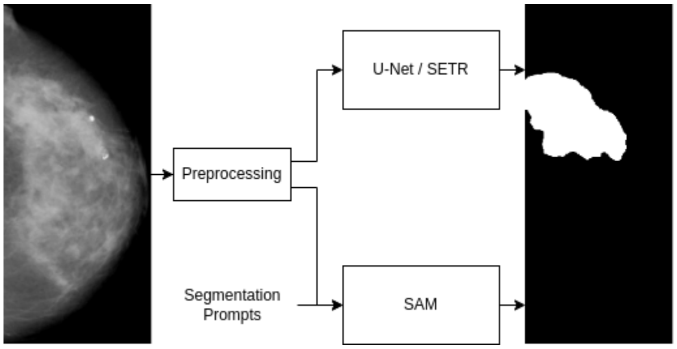


Fig. 5: Overview of the inputs and outputs of the three models we use in our experiments. On the left is a sample from the CBIS-DDSM dataset, on the right the corresponding ground truth for a region of interest. The Segment Anything Model requires segmentation prompts as extra inputs.

masks that could all be valid, and also estimates IoU scores corresponding to these masks. These estimated scores are then used to rank the masks, and during inference the mask with the best estimated score is chosen as the final output. This results in a binary output map from the Segment Anything Model. (Accurate IoU estimations are crucial for making a correct choice between candidate masks.)

We provide the SAM with point prompts during both inference and training. For the automatic selection of these points, we follow the approach by Kirillov *et al.* [6]. During training, these points are coordinates of randomly selected pixels in the ground truth mask. For inference, we use the Euclidian Distance Transform to find the centre of the mask.

B. Loss, Metrics, and Evaluation

In our experiments, we use four loss functions: Binary Cross-Entropy, Dice loss, Focal loss, and Mean Squared Error. In this section, we describe their theoretical background, the specific definitions we used for these losses, and why we used these losses. We measure the performance of each model with the following set of metrics: Dice, IoU, trainable parameter count, and maximum GPU memory usage. We also visually compare ground truth to predictions from various models by exporting a small set of random samples (see Figures 6a and 6b.) The formulas we used for IoU, Dice, BCE, Focal, and MSE are as follows:

$$Dice = \frac{2I + \beta}{I + U + \beta}$$

Where I is the intersection between prediction and ground truth, U is the union between prediction and ground truth, and β is a very small constant (1e-7). Including β in the denominator ensures that the formula is defined for the case of an empty ground truth mask (all zeros). Including it in the numerator ensures that when the prediction and ground truth are both empty, this is counted as perfect overlap, giving a Dice score of 1.

$$IoU = \frac{I + \beta}{U + \beta}$$

$$BCE = -[y \log(x) + (1 - y) \log(1 - x)]$$

Where y is the target pixel value, taking values of either 0 or 1. x is the predicted logit, taking values anywhere from 0 to 1. We take the mean BCE over all pixels of a mask.

$$Focal = -[\alpha(1-x)^{\gamma} y \log(x) + (1-\alpha)x^{\gamma}(1-y)\log(1-x)]$$

Or, the more compact definition from Lin *et al.* [38]:

$$Focal = -\alpha_t(1 - x_t)^{\gamma} \log(x_t)$$

And for the Mean Squared Error we have of course used

$$MSE = (y - x)^2$$

Zheng *et al.* [13] make use of a pixel-wise cross-entropy loss, and Sarker *et al.* [21] use the *binary* cross-entropy loss. We follow their example by using the binary cross-entropy loss for both the SETR and U-Net. Kirillov *et al.* [6] report using a mixture of focal loss and dice loss for the segmentation masks, and we do the same for our experiments with the SAM. Specifically, $L_{mask} = 20Focal + Dice$. The SAM produces three possible segmentation masks, and three corresponding predictions for the overlap score (IoU) of each of these masks. A loss function is also calculated for each of the IoU predictions: $L_{iou} = MSE$. Kirillov *et al.* [6] mentions that they calculate the loss ($L_{mask} + L_{iou}$) for all three mask candidates. Then, they *backpropagate the loss only for the best of these masks*. We do the same. However, there is ambiguity to this instruction from Kirillov *et al.* [6]. We discuss this further in paragraph V-A2a.

IV. Experiments

A. Datasets

We use two datasets: the Curated Breast Imaging Subset of Digital Database for Screening Mammography (CBIS-DDSM), and the private dataset provided by Ziekenhuis Groep Twente. We describe their details in Section II-A1.

B. Shared Implementation Details

We run all of our experiments on a high-performance cluster, managed using SLURM. We've used an NVIDIA A40 GPU with 48GB memory. For all training experiments, we use a shared set of implementation details: Because the learning rate is typically task-sensitive, we use the same learning rate of 0.0001 that was also applied for experiments on CBIS-DDSM by Sarker *et al.* [21]. Following Sarker *et al.* [21], we apply learning rate reduction upon a plateau of the validation loss, with a patience of 25 epochs and a reduction factor of 10, and we apply early stopping when the training loss has not improved for 40 epochs. Following Jia *et al.* [14] and Kirillov *et al.* [6] we use the AdamW optimizer, with β_1 = 0.9, β_2 = 0.999.

C. U-Net

We have trained a U-Net for CBIS-DDSM segmentation from scratch. The precise details for its architecture have been copied from Sarker *et al.* [21]. The U-Net has been trained at an input resolution of 224x224, resizing each image to fit using bilinear interpolation. We apply Contrastive Limited Adaptive Histogram Equalization (CLAHE), as is also used by Sarker *et al.* [21]. We do not perform any other data preprocessing/augmentation steps. A batch size of 16 was used.

D. Segmentation Transformer

We have trained and fine-tuned a Segmentation Transformer for CBIS-DDSM segmentation in three ways: training a randomly initialized SETR From Scratch (FS), Full fine-Tuning (FT) a pre-trained SETR, and Visual Prompt Tuning (VPT) a pre-trained SETR. Each time we have used the ViT-L backbone architecture. For pre-trained backbone weights, we use those provided by Dosovitskiy *et al.* [11]. Similar to the

U-Net, an input resolution of 224x224 has been used, with bilinear interpolation, and finally CLAHE was applied. Again, a batch size of 16 was used.

The SETR can be trained with auxiliary heads for better performance. To fairly compare baseline performances between models, we choose not to use auxiliary heads for the SETR since we do not use auxiliary heads for the U-Net and the SAM. We are not interested in producing a state-of-the-art solution, and while the auxiliary heads may improve absolute performance, they do not seem essential to understanding the behaviour of the model during fine-tuning in circumstances of low sample counts.

During full fine-tuning of the SETR, the learning rate applied to the decoder is 10.0 times that of the overall learning rate (following Zheng *et al.* [13]). This is done because the decoder is randomly initialized and its parameters likely need more updating than the pre-trained backbone. However, when applying Visual Prompt Tuning to the SETR, or when training the SETR from scratch (i.e. without loading a pre-trained backbone), we set this decoder learning rate multiplier to 1.0 as was also done by Jia *et al.* [14].

Jia *et al.* [14] originally implemented VPT for various ViT-like architectures, and these implementations are available in their GitHub repository[1]. The included architectures are ViT[11], Masked Auto Encoder ViT[39], and a MoCo-v3 ViT[40]. Jia *et al.* [14] also conducted experiments that applied VPT to the SETR, but the combination of SETR and VPT is not found in their repository. Nevertheless, the implementation of VPT for the ViT has proven sufficient basis for re-implementing VPT for the SETR. After publication by Zheng *et al.* [13], the SETR has been incorporated into the MMSegmentation toolbox[27]. In accordance with Jia *et al.* [14], we used the MMSegmentation repository as a basis for our implementation of VPT for the SETR. See Section VII for code.

Jia *et al.* [14] have performed ablations of VPT with different lengths, depths, and prompt locations. They show that VPT-Deep is close to optimal depth if not optimal (i.e. inserting prompts into all transformer layers) and find prepending to be most effective. We follow their example by prepending prompts at each layer of the SETR. However, where Jia *et al.* [14] recommend searching for an appropriate prompt length, we simply fix the prompt length at 50 tokens.

E. Segment Anything Model

We have used the repository[2] published by Kirillov *et al.* [6] as the basis for implementing testing and fine-tuning SAM. Unlike the framework we used for the SETR, the SAM repository did not include any training code. The focus of the repository seems to be on demonstrating the behaviour of SAM, rather than revealing more information about how their experiments were conducted.

We used the PyTorch models and pre-trained weights that the Segment Anything repository provides and wrote the code necessary for our experiments around that. Like the U-Net and SETR experiments, we've applied early stopping, learning rate reduction, and the AdamW optimizer. We bilinearly interpolate the images to fit the desired input size, but in contrast to the other two models, the input dimensions for SAM are 1024x1024. We have not applied CLAHE for SAM. Instead, we normalize pixel values the same way as Kirillov *et al.* [6]: z-score normalization for the RGB channel distributions from the SA-1B dataset. The batch size is 1 instead of 16, due to unexpectedly high memory usage at higher batch sizes. This seems to be an issue with our implementation that we have been unable to resolve.

Model	IoU (%)	Dice (%)	Trainable (M)	GPU mem (GB)
U-Net	9.11	12.39	28.99	6.735
SETR-FS	4.50	6.69	307.43	12.211
SETR-FT	7.95	11.36	307.43	12.334
SETR-VPT	8.26	11.30	5.36	14.301
SAM-ZS	2.05	3.60	0.00	4.439
SAM-FT	27.42	39.84	312.34	8.029

TABLE I: All models trained on the CBIS-DDSM train set, evaluated on the CBIS-DDSM test set. The postfixes ZS/FS/FT/VPT indicate ”Zero-Shot”/”From Scratch”/”Full Tuning”/”Visual Prompt Tuning” respectively. The SAM experiments are separated from the rest to emphasize that SAM is given segmentation prompts which the others lack.

F. Evaluation on ZGT Dataset

We have had to make a few changes to our implementation before being able to evaluate on the unseen private dataset. The first is that we incorporated the small constant β into the numerators of the IoU and Dice formulas in Section III-B. The other is that we needed to define the behaviour of point prompt selection when a ground truth mask is empty. In such cases, during inference we select the point that is in the middle of the background (i.e. the middle of the image). During training we select random pixels in the image.

Model	IoU (%)	Dice (%)
U-Net	31.91	33.18
SETR-FS	16.57	17.13
SETR-FT	9.64	11.14
SETR-VPT	15.44	16.97
SAM-ZS	0.52	0.96
SAM-FT	13.70	19.20

TABLE II: All models trained on the CBIS-DDSM train set, evaluated on the manually annotated samples from the private dataset.

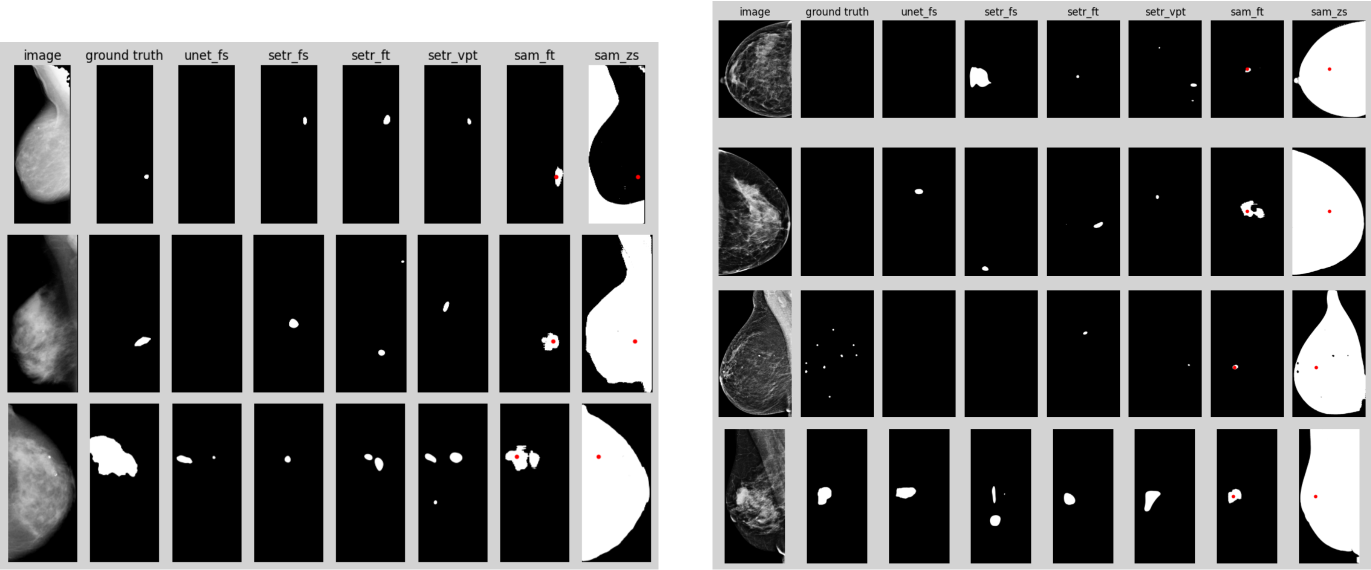
V. Results: Analysis and Discussion

A. CBIS-DDSM

Clearly, Table I shows that the best-performing model for the CBIS-DDSM dataset is SAM-FT. Before discussing the performance of the Segment Anything Model, we reflect on the relative differences between the U-Net and the Segmentation Transformer.

[1] https://github.com/KMnP/vpt

[2] https://github.com/facebookresearch/segment-anything



a) CBIS-DDSM. These samples were taken from the validation set.

b) Ziekenhuis Groep Twente private dataset. Two of the ground truths do not contain annotated regions.

Fig. 6: Example segmentation outputs for both datasets. Image and ground truth are shown in the left two columns of each subfigure. The red dots are the locations indicated by the point prompts added as input during the SAM experiments.

1) U-Net and Segmentation Transformer: Among the models that do not use segmentation prompts, Table I shows the U-Net as the best-performing model by achieving the highest overlap metrics (IoU and Dice) of its group. It has the smallest GPU memory footprint, using just over half of the GPU memory needed for the smallest competing SETR. Only in terms of parameter-efficiency can we argue in favor of the SETR-VPT over the U-Net. The SETR-VPT shows overlap metrics that come close to those of the U-Net, using 5 times fewer trainable parameters to achieve it.

We can see that for the models that were trained from scratch, i.e. U-Net and SETR-FS, the model with fewer trainable parameters is more effective. SETR-FS has the lowest IoU and Dice out of the U-Net and SETR experiments, while the U-Net has the highest overlap metrics. This is unsurprising, because Dosovitskiy *et al.* [11] found that only with “larger datasets (14M - 300M images)” the Vision Transformer starts to outperform CNNs on similar tasks. Since the ViT is such an important component of the SETR, it was to be expected that training the SETR from scratch on a dataset with 3,103 annotated image regions would not yield the best performance.

As expected, SETR-VPT uses far fewer trainable parameters than SETR-FS and SETR-FT. Nevertheless, SETR-VPT shows comparable if not better overlap than SETR-FT. This confirms the notion from Jia *et al.* [14] that Visual Prompt Tuning is a parameter-efficient approach that can achieve similar performance with fewer trainable parameters. It demonstrates that Visual Prompt Tuning can achieve similar results to full fine-tuning the Segmentation Transformer when applied to mammography. SETR-VPT is the most parameter-efficient of all models, barring SAM-ZS. It does come at the price of a larger GPU memory footprint, needing roughly 2 GB more than the other two SETR experiments.

2) Segment Anything Model: SAM-FT is the best-performing approach for CBIS-DDSM by a large margin. The overlap metrics for SAM-FT in Table I are far beyond those obtained for the U-Net and SETR. SAM-FT improves over the best competing model with a factor of 3, both for IoU and Dice. This shows that there are crucial differences between the Segment Anything Model and the Segmentation Transformer that allow the Segment Anything Model to model the mammography segmentation task a lot better. Architecturally, the SETR is closest to the SAM, making the SETR-FT experiment the one with the fewest methodological differences to SAM-FT. There are four key areas we can identify in which the SETR-FT and the SAM-FT differ:

- Pre-train dataset: The ViT backbone of the SETR has been pre-trained on the ImageNet-21k[8], and the decoder of the SETR has not been pre-trained. The SAM has been pre-trained on the SA-1B dataset
- Pre-train task: The ViT backbone of the SETR has been pre-trained for classification, while SAM has been pre-trained for generic object segmentation. The SETR is constructed to output a single segmentation mask, while the SAM has been constructed with a mechanism for choosing between three possibly valid segmentation masks. The objective function that is used to optimize the SAM is also different from those used for the U-Net and the SETR, as we describe in Section III-B.
- Model architecture: The SETR consists of a single encoder and decoder. There are three encoding elements in the SAM: the Image Encoder, the Mask Encoder,

and the Prompt Encoder. The Mask Decoder of the SAM is a variant of a Transformer[25] block, while the PUP Decoder of the SETR is based on convolutional operations.

- Input: Our SETR experiments operate at an input resolution of 224x224. In contrast, the SAM has been designed for 1024x1024 inputs. This is a large difference in image quality. The SAM also is given the point prompts, equipping the model with a localization hint that the SETR does not receive. These two factors theoretically provide the SAM with an information advantage over the SETR.

Further experiments are necessary to conclude which of these areas are the determining factors in the performance difference between SAM-FT and SETR-FT.

a) Predicted IoU Scores: For analyzing the training and validation outputs of SAM-FT, we calculated metrics not only for the final masks but also per individual mask prediction head. We noticed that the average IoU for two of three prediction heads was higher than the average IoU of the final selected mask. This means that the mask selection mechanism is often selecting a suboptimal mask; there is performance to be gained by selecting a fixed segmentation head, instead of using the predicted IoU scores to select the mask.

We suspect this is because the predicted IoU for incorrect masks are overconfident. In Section III-B, we mention that we only backpropagate L_{sam} for the mask with the lowest calculated $L_{mask} + L_{iou}$. This means that when an estimation of an IoU score is very wrong as measured by L_{iou}, this makes it less likely that the model will learn to predict that score better. Instead, we think a better approach is to backpropagate L_{iou} for all mask heads, and to only backpropagate L_{mask} for the mask with the lowest calculated $L_{mask} + L_{iou}$.

B. ZGT Dataset

At first glance, the results in Table II seem utterly discordant with those found in Table I. The model that shows best performance on the unseen ZGT dataset is the U-Net, followed by SETR-FS, and SETR-VPT. The performance of SETR-FS relative to other models is especially surprising, since it showed the weakest evaluation on CBIS-DDSM. On the ZGT dataset, the U-Net unexpectedly outshines SAM-FT. A likely culprit for this is that 97 out of the total 206 images lack any region annotation. More so than the U-Net, SAM-FT falls prey to false positives: SAM-FT seems to be looking for tumours that do not exist. The evaluation results on our private dataset show that the CBIS-DDSM does not accurately match the distribution of our dataset. Without evaluation on our private dataset, this would have led to an incomplete impression of the performance of segmentation models.

There are interesting observations to be made about the example outputs in Figures 6a and 6b. SAM-ZS behaves according to what one might expect from a model pretrained for segmentation outside of the medical domain. It can accurately segment the breast from each image, although we see an inverted map for one sample. We see there are cases where all models produce outputs in the neighbourhood of the ground truth (i.e. the bottom row of Figure 6b). SAM-FT has learnt to produce much smaller masks, but still always annotates around the point prompt. The most important thing to note is that in the ZGT dataset, there are samples with empty ground truths. Two examples of this are shown in Figure 6b, where we see that most models incorrectly predict a non-empty mask. This indicates that the models have not learnt well enough to identify the absence of tumors.

VI. Future Work

Time and computational power are never in unlimited supply. We compared three state-of-the-art models and applied them in various ways to the mammography domain. While there are additional models and fine-tuning methods we have not explored, and certain mammography datasets we did not use, these are typical constraints faced in research. Some directions for future research are particularly noteworthy, and we list them below.

We have not conducted an exhaustive search of hyperparameters for each model, instead relying on parameters used in other works. For instance, we are uncertain whether the batch sizes and learning rates used in our experiments were optimal choices. Specifically for the Segment Anything Model, as mentioned in Section IV-E, we were unable to experiment with higher batch sizes. A basic exploration of what hyperparameters suit each specific model on the task of mammography segmentation might allow for a fairer performance comparison between models.

1) Visual Prompt Tuning for the Segment Anything Model: As we discussed in Section IV-D, we have succesfully implemented Visual Prompt Tuning for the Segmentation Transformer. We also intended to implement VPT for SAM, which had not been done before. This would result in a type of contribution similar to the adapter-based fine-tuning technique from [22]. Unfortunately, implementing VPT for the SAM is significantly different than for the SETR. We have taken a good look at the details of the SAM, but were unable to comprehensively implement VPT for the image encoder used in the SAM. Where the SETR is a direct adaptation of the ViT into a model designed for segmentation, the SAM is not. Rather, there is an extra step between the ViT from [11], and the ViT variant used as the image encoder in [6]: the modifications introduced as part of the ViTDet[29]; For most of the encoder layers of the pre-trained Vision Transformer, global attention has been replaced with window attention. The window attention is computed over non-overlapping windows of the encoder layer's input. The ViTDet also introduced relative positional biases at each encoder layer. Because of these modifications, it is not entirely straightforward to apply Visual Prompt Tuning as it was originally designed. Concretely, we ran into two issues:

1) VPT-Deep prepends prompts to the inputs of each encoder layer. For global attention, this makes sense, because new information can be gained by computing attention between the prompts and the regular inputs. In the context of window attention, this would run the risk

of putting all prompts in one of several windows, and only providing added value for that specific window. It seems likely that this will impact the effectiveness of the prompts, because the model will not be allowed to learn new task-related information for large parts of the image, or only in a rather indirect fashion.

2) Secondly, the learned relative positional encodings have a size that is dependent on the input length of each encoder layer. This means that if we increase the length of the layer's input, the layer's positional encoding and the input are no longer compatible sizes. This requires some way of resizing the positional encodings, while still preserving their learned function.

One way of tackling the first obstacle is by choosing to employ a reduced version of VPT; only prepending at the layers that use global attention. An option that offers more freedom is prepending prompts per window. For the second obstacle, we notice that [29] mentions inspiration for the relative positional embedding being taken from [26]. There it is mentioned that such positional encodings can be adapted to different window sizes through bi-cubic interpolation. These are however *potential* remedies, and we have not taken the time to put them into practice. Further time and experiments would be needed to come to a conclusion about the effectiveness of combining VPT with SAM.

A. Limitations

There is a small number of inherent limitations to our work. The first limitation is that while the principle of training on a public dataset and subsequently evaluating transfer performance on a private dataset can be applied to other tasks than just binary segmentation, there needs to be a certain level of correspondence between the annotations that exist for the public dataset and those that can be collected for the private dataset. Concretely, the task definition should be the same for both datasets. So far, we have only trained models for tumor instance segmentation. This requires manual binary annotation of regions of interest on the private samples. In the case of semantic segmentation, e.g. recognizing the abnormality class of the tumor, the process of manual annotation will also need to take encoding this information into account. When the task becomes more fine-grained, so do the required manual annotations, requiring more effort by the experts who provide these annotations.

Secondly, not every option for optimizing and evaluating listed in Section III-A2 applies to every model. For instance, the Zero-Shot application of a SETR is only well-defined when its pre-train task has the same number of classes as the zero-shot task. An example that would not work, would be pre-training a SETR for ADE20K segmentation with 150 segmentation classes, and subsequently trying to apply this model to a binary segmentation task like the mammography segmentation task we have considered in this work. The best way to work around this seems to be to find a pre-train task with the same number of segmentation classes as the zero-shot task.

Finally, we remark that the manually collected set of annotations covers less than 1% of the images present in the private dataset from Ziekenhuis Groep Twente (206 out of 84,299 images have been annotated.) This raises concerns about the representativeness of the annotated subset, and the robustness of conclusions drawn based on evaluation with this subset.

VII. Conclusions

We establish a framework for training state-of-the-art models on public mammography datasets, and assessing their subsequent performance on unseen out-of-distribution data. We've used this framework to compare performance of three models: the U-Net, the Segmentation Transformer, and the Segment Anything Model. The model that showed by far the best performance on the public CBIS-DDSM is the Segment Anything Model. Importantly however, we've seen that this result does not mean that high performance on other mammography datasets is guaranteed. When applying the SAM to a private dataset from Ziekenhuis Groep Twente, overlap metrics drop. The likely cause is samples with deliberately empty ground truth masks. We publish all code used in our experiments.[3]

References

[1] World Health Organization, "Cancer," *www.who.int*, Feb. 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/cancer.

[2] World Health Organization, "Breast cancer," *www.who.int*, Jul. 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/breast-cancer.

[3] M. Richards, A. Westcombe, S. Love, P. Littlejohns, and A. Ramirez, "Influence of delay on survival in patients with breast cancer: A systematic review," *The Lancet*, vol. 353, no. 9159, pp. 1119–1126, Apr. 1999. DOI: 10.1016/s0140-6736(99)02143-1.

[4] L. Nyström, I. Andersson, N. Bjurstam, J. Frisell, B. Nordenskjöld, and L. E. Rutqvist, "Long-term effects of mammography screening: Updated overview of the swedish randomised trials," *The Lancet*, vol. 359, no. 9310, pp. 909–919, Mar. 2002. DOI: 10.1016/s0140-6736(02)08020-0.

[5] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:18268744.

[6] A. Kirillov *et al.*, *Segment anything*, 2023. arXiv: 2304.02643 [cs.CV].

[7] B. Zhou *et al.*, "Semantic understanding of scenes through the ade20k dataset," *International Journal of Computer Vision*, vol. 127, no. 3, pp. 302–321, Dec. 2018. DOI: 10.1007/s11263-018-1140-0.

[3] https://github.com/DrumsnChocolate/final-project

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[9] R. E. Lee, F. Gimenez, A. Hoogi, K. Miyake, M. Gorovoy, and D. L. Rubin, "A curated mammography data set for use in computer-aided detection and diagnosis research," *Scientific Data*, vol. 4, no. 1, Dec. 2017. DOI: 10.1038/sdata.2017.177. [Online]. Available: https://doi.org/10.1038/sdata.2017.177.

[10] H. T. Nguyen *et al.*, "Vindr-mammo: A large-scale benchmark dataset for computer-aided diagnosis in full-field digital mammography," *medRxiv*, 2022. DOI: 10.1101/2022.03.07.22272009. [Online]. Available: https://www.medrxiv.org/content/early/2022/03/10/2022.03.07.22272009.

[11] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020. arXiv: 2010.11929. [Online]. Available: https://arxiv.org/abs/2010.11929.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. arXiv: 1505.04597. [Online]. Available: http://arxiv.org/abs/1505.04597.

[13] S. Zheng *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," *CoRR*, vol. abs/2012.15840, 2020. arXiv: 2012.15840. [Online]. Available: https://arxiv.org/abs/2012.15840.

[14] M. Jia *et al.*, *Visual prompt tuning*, 2022. arXiv: 2203.12119 [cs.CV].

[15] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. arXiv: 1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805.

[16] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:49313245.

[17] N. Wu *et al.*, "The nyu breast cancer screening dataset v1.0," 2019.

[18] E. Michael, H. Ma, H. Li, F. Kulwa, and J. Li, *Breast cancer segmentation methods: Current status and future potentials*, 2023. arXiv: 2305.09880 [cs.CV].

[19] A. Baccouche, B. Garcia-Zapirain, C. C. Olea, and A. S. Elmaghraby, "Connected-unets: A deep learning architecture for breast mass segmentation," *npj Breast Cancer*, vol. 7, Dec. 2021. DOI: 10.1038/s41523-021-00358-x.

[20] M. Alkhaleefah *et al.*, "Connected-segnets: A deep learning model for breast tumor segmentation from x-ray images," *Cancers*, vol. 14, Aug. 2022. DOI: 10.3390/cancers14164030.

[21] P. Sarker, S. Sarker, G. Bebis, and A. Tavakkoli, *Connectedunets++: Mass segmentation from whole mammographic images*, 2022. arXiv: 2210.13668 [eess.IV].

[22] X. Xiong, C. Wang, W. Li, and G. Li, "Mammo-sam: Adapting foundation segment anything model for automatic breast mass segmentation in whole mammograms," in *Machine Learning in Medical Imaging*, Cham: Springer Nature Switzerland, 2024, pp. 176–185, ISBN: 978-3-031-45673-2.

[23] Y. Shen *et al.*, "An interpretable classifier for high-resolution breast cancer screening images utilizing weakly supervised localization," *CoRR*, vol. abs/2002.07613, 2020. arXiv: 2002.07613. [Online]. Available: https://arxiv.org/abs/2002.07613.

[24] K. Liu, Y. Shen, N. Wu, J. Chledowski, C. Fernandez-Granda, and K. J. Geras, "Weakly-supervised high-resolution segmentation of mammography images for breast cancer diagnosis," *CoRR*, vol. abs/2106.07049, 2021. arXiv: 2106.07049. [Online]. Available: https://arxiv.org/abs/2106.07049.

[25] A. Vaswani *et al.*, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. arXiv: 1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762.

[26] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," *CoRR*, vol. abs/2103.14030, 2021. arXiv: 2103.14030. [Online]. Available: https://arxiv.org/abs/2103.14030.

[27] M. Contributors, *MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark*, https://github.com/open-mmlab/mmsegmentation, 2020.

[28] A. Radford *et al.*, *Learning transferable visual models from natural language supervision*, 2021. arXiv: 2103.00020 [cs.CV].

[29] Y. Li, H. Mao, R. Girshick, and K. He, "Exploring plain vision transformer backbones for object detection," in *Computer Vision – ECCV 2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Cham: Springer Nature Switzerland, 2022, pp. 280–296, ISBN: 978-3-031-20077-9.

[30] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, *Segmenter: Transformer for semantic segmentation*, 2021. arXiv: 2105.05633 [cs.CV].

[31] J. Gu *et al.*, *Multi-scale high-resolution vision transformer for semantic segmentation*, 2021. arXiv: 2111.01236 [cs.CV].

[32] H. Thisanke, C. Deshan, K. Chamith, S. Seneviratne, R. Vidanaarachchi, and D. Herath, *Semantic segmentation using vision transformers: A survey*, 2023. arXiv: 2305.03273 [cs.CV].

[33] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, *How transferable are features in deep neural networks?* 2014. arXiv: 1411.1792 [cs.LG].

[34] W. Liu, X. Shen, C.-M. Pun, and X. Cun, *Explicit visual prompting for low-level structure segmentations*, 2023. arXiv: 2303.10883 [cs.CV].

[35] E. J. Hu *et al.*, *Lora: Low-rank adaptation of large language models*, 2021. arXiv: 2106.09685 [cs.CL].

[36] M. Ahmadi *et al.*, *Comparative analysis of segment anything model and u-net for breast tumor detection in ultrasound and mammography images*, 2024. arXiv: 2306.12510 [eess.IV].

[37] M. Hu, Y. Li, and X. Yang, *Breastsam: A study of segment anything model for breast tumor detection in ultrasound images*, 2023. arXiv: 2305.12447 [eess.IV].

[38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, *Focal loss for dense object detection*, 2018. arXiv: 1708.02002 [cs.CV].

[39] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. B. Girshick, "Masked autoencoders are scalable vision learners," *CoRR*, vol. abs/2111.06377, 2021. arXiv: 2111.06377. [Online]. Available: https://arxiv.org/abs/2111.06377.

[40] X. Chen, S. Xie, and K. He, *An empirical study of training self-supervised vision transformers*, 2021. arXiv: 2104.02057 [cs.CV].

Appendix A
Reproduction: U-Net experiments in relation to results reported in ConnectedUNets++[21]

In order to anchor our results with other reports, in this appendix we attempt to reproduce the experiments conducted by Sarker *et al.* [21]. We prefer comparing to Sarker *et al.* [21] over Baccouche *et al.* [19] because the former use entire images (just like our work does), while the latter evaluate on regions. We did not find a published implementation of the models used by Sarker *et al.* [21]. We did notice a publicly available repository for some of the experiments by Baccouche *et al.* [19][4]. Since Sarker *et al.* [21] derived from Baccouche *et al.* [19] it seems unlikely that Sarker *et al.* [21] and Baccouche *et al.* [19] are using significantly different architectures for what they call a "basic U-Net", so we have incorporated some of the implementation by Baccouche *et al.* [19] into our own experiments.

Our reproduction experiments have similar implementation details to those laid out in Section IV-C. These reproduction experiments can be thought of as a bridge between the U-Net used by Sarker *et al.* [21] and our U-Net result in Table I. We are aware of two important differences between our U-Net and the U-Net used by Sarker *et al.* [21]:

1) We use the AdamW optimizer in our experiments, while Sarker *et al.* [21] used the Adam optimizer.
2) We choose to tackle segmentation of not only mass abnormalities: we also include calcification abnormalities in the task. The inclusion of calcification cases likely makes the model's task more difficult, since it needs to recognize a more diverse set of abnormalities.

To account for these differences, in Table III all models were trained with the Adam optimizer, and we've trained one U-Net on both masses+calcifications and one U-Net on only masses (Mass-only).

model	IoU (%)	Dice (%)
U-Net Mass-only (Sarker *et al.* [21])	27	41
U-Net Mass-only (ours)	15.13	19.95
U-Net (ours)	10.30	13.70

TABLE III: Reproduction results for the U-Net architecture, using the Adam optimizer.

In Table III it is clear that our results differ from Sarker *et al.* [21]. We also see that training the U-Net for the task of segmenting both calcifications and masses results in lower overall performance, as expected.

There could be an implementation detail that either we missed, or was not mentioned in their paper. There are many other possible reasons why there might be a performance difference. However, the performance difference seems unlikely to be random. Access to the original code of the experiments conducted by Sarker *et al.* [21] would be very helpful in finding out the key differentiating factor in our training approaches.

From our perspective, a likely culprit could have been the use of a different moment to checkpoint the model before testing. Since the training loss is monitored and early stopping is applied after 40 epochs of not improving, it might make quite a big difference at what point the model's state is saved for evaluation. In all of our U-Net experiments, we have saved the model *after* these 40 last epochs. It could well be that Sarker *et al.* [21] checkpoint the model at its 'best' i.e. at the point of lowest training loss. This could explain a large part of the difference in evaluation metrics. To investigate the likelihood of this being the cause of any performance deficit, for *U-Net Mass-only (ours)* we have plotted the progression of the IoU and Dice metrics, and the validation and training loss for all epochs. See Figure 7.

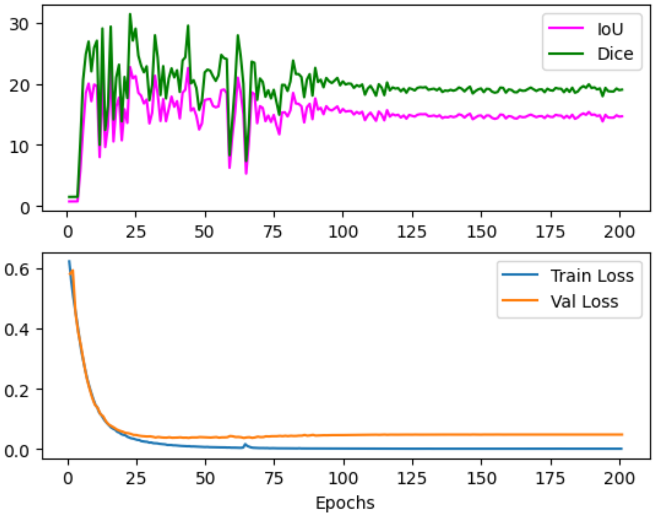


Fig. 7: U-Net Mass-only (ours): Validation metrics progression over all epochs, together with validation loss and training loss. There is no obvious deterioration in validation performance over the last 40 epochs.

None of the metrics in Figure 7 have deteriorated drastically over the last 40 epochs. Since the validation metrics are close to the test result in Table III, our idea that the performance difference could be explained by a different checkpointing moment falls flat.

[4] https://github.com/AsmaBaccouche/Connected-Unets-and-more