# COL8628/COL828 Assignment 2

Exploring Open-Vocabulary Object Detectors for Breast Cancer Detection using Mammograms

Aastha A K Verma
Entry Number: 2022CS11607

November 25, 2025

# 1 Introduction

## 1.1 Background and Motivation

Breast cancer detection in mammography is a critical medical imaging task where accurate and efficient detection systems can significantly impact patient outcomes. Traditional object detection models require extensive labeled datasets and struggle with domain shifts between different imaging equipment and patient populations. Open-Vocabulary Object Detectors (OVODs) offer a promising solution by enabling detection of objects using textual descriptions, potentially reducing the annotation burden and improving generalization.

## 1.2 Problem Statement

This assignment explores the application of Grounding-DINO (G-DINO) [1], a state-of-the-art OVOD model, for breast cancer detection across three mammography datasets with domain shifts. We investigate:

1. The effectiveness of zero-shot detection using hand-crafted prompts
2. The robustness of learned prompts (CoOp and CoCoOp) under domain shifts
3. The potential of semi-supervised learning to improve performance with limited labels

## 1.3 Dataset Overview

We utilize three mammography datasets (A, B, and C) that share the same label space (benign/malignant) but exhibit domain shifts due to:

- Variations in imaging equipment and protocols
- Different patient demographics
- Varying image acquisition parameters

Each dataset contains:

- Training and test splits
- Bounding box annotations for malignant regions
- Associated CSV files with annotation metadata

# 2 Methodology

## 2.1 Grounding-DINO Architecture

Grounding-DINO combines the strengths of DINO (Detection with Transformers) with grounded pre-training, enabling open-vocabulary detection through the integration of:

- **Visual Encoder**: Processes input images to extract visual features
- **Text Encoder**: Encodes text prompts into semantic representations
- **Cross-modal Fusion**: Aligns visual and textual features for detection
- **Detection Head**: Generates bounding boxes and confidence scores

## 2.2 Prompt Learning Techniques

### 2.2.1 Context Optimization (CoOp)

CoOp [4] learns continuous prompt vectors while keeping the vision-language model frozen. The key components include:

**Algorithm 1** CoOp Training Procedure

**Input:** Training images $\{x_i\}$, labels $\{y_i\}$, initial context vectors $\{v_1, ..., v_m\}$
**Output:** Optimized context vectors
**while** not converged **do**
  Sample mini-batch $(x, y)$ from training set
  Compute text embeddings: $t = g(v_1, ..., v_m, \text{class\_name})$
  Extract visual features: $f = \text{VisualEncoder}(x)$
  Calculate detection loss: $\mathcal{L} = \text{DetectionLoss}(f, t, y)$
  Update context vectors: $v \leftarrow v - \alpha \nabla_v \mathcal{L}$
**end while**
**return** Optimized context vectors $\{v_1^*, ..., v_m^*\}$

The prompt template for CoOp in our medical context:

$$t_i = [v_1][v_2]...[v_m][\text{class}_i] \tag{1}$$

where $[v_j]$ are learnable context vectors and $[\text{class}_i]$ represents the class name (e.g., "malignant tumor", "cancerous region").

**Algorithm 2** CoCoOp Training Procedure

**Input:** Training images $\{x_i\}$, labels $\{y_i\}$, meta-network $h_\theta$
**Output:** Optimized meta-network parameters $\theta^*$
**while** not converged **do**
  Sample mini-batch $(x, y)$
  Extract image features: $f = \text{VisualEncoder}(x)$
  Generate instance-specific contexts: $\pi = h_\theta(f)$
  Combine with base contexts: $v' = v + \pi$
  Compute text embeddings: $t = g(v'_1, ..., v'_m, \text{class\_name})$
  Calculate loss and update $\theta$
**end while**
**return** Optimized parameters $\theta^*$

### 2.2.2 Conditional Context Optimization (CoCoOp)

CoCoOp [3] extends CoOp by generating instance-specific prompts, improving generalization:

The key difference is the meta-network that generates input-specific adjustments:

$$t_i(x) = [v_1 + \pi_1(x)]...[v_m + \pi_m(x)][\text{class}_i] \tag{2}$$

## 2.3 Semi-Supervised Learning with FixMatch

FixMatch [2] leverages unlabeled data through consistency regularization:

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + \lambda_u \mathcal{L}_{\text{unsup}} \tag{3}$$

where:

- $\mathcal{L}_{\text{sup}}$: Standard detection loss on labeled data
- $\mathcal{L}_{\text{unsup}}$: Consistency loss on unlabeled data
- $\lambda_u$: Weight balancing supervised and unsupervised losses

The unsupervised loss enforces consistency between weak and strong augmentations:

$$\mathcal{L}_{\text{unsup}} = \mathbb{1}(\max(p_w) \geq \tau) \cdot \text{CE}(p_s, \arg\max(p_w)) \tag{4}$$

# 3 Experimental Setup

## 3.1 Implementation Details

### 3.1.1 Hyperparameters

Table 1: Hyperparameter Settings for All Experiments

| **Parameter** | **CoOp/CoCoOp** | **Semi-supervised** |
|---|---|---|
| Learning Rate | $5 \times 10^{-2}$ | $5 \times 10^{-2}$ |
| Batch Size | 16 | 8 (labeled) + 8 (unlabeled) |
| Epochs | 30 | 30 |
| Optimizer | Adam | Adam |
| Weight Decay | 0.01 | 0.01 |
| Context Length | 8 | 8 |
| Warmup Epochs | - | 5 |
| $\lambda_u$ | - | 1.0 |
| Confidence Threshold $\tau$ | - | 0.25 |

### 3.1.2 Data Augmentation Strategies

**Weak Augmentation:**

- Random horizontal flip

**Strong Augmentation:**

- All weak augmentations
- Random brightness adjustment
- Random contrast adjustment
- Random Gaussian blur

## 3.2 Evaluation Metrics

We use Average Precision (AP) as the primary metric:

$$\text{AP} = \int_0^1 p(r)\, dr \tag{5}$$

where $p(r)$ is the precision at recall $r$.

Additionally, we report:

- AP@IoU[0.5,0.95] (standard COCO metric)
- AP+@IoU[0.5,0.95] (AP only on truly positive examples)

# 4 Results and Analysis

## 4.1 Task 1: Zero-Shot Evaluation

### 4.1.1 Hand-crafted Prompt Design

We experimented with various prompt formulations:

Table 2: Zero-shot Performance with Different Text Prompts

| **Prompt** | **Dataset A** AP@[0.5,0.95] | **Dataset B** AP@[0.5,0.95] | **Dataset C** AP@[0.5,0.95] |
|---|---|---|---|
| "cancer ." | 0.00292 | 0.00860 | 0.01549 |
| "tumor ." | 0.00658 | 0.01129 | 0.01286 |
| longprompt | 0.00482 | 0.00605 | 0.02596 |

where longprompt = "benign tissue . malignant tumor . dark background . dense tumor lump . no object ."

### 4.1.2 Hyperparameter Sensitivity Analysis

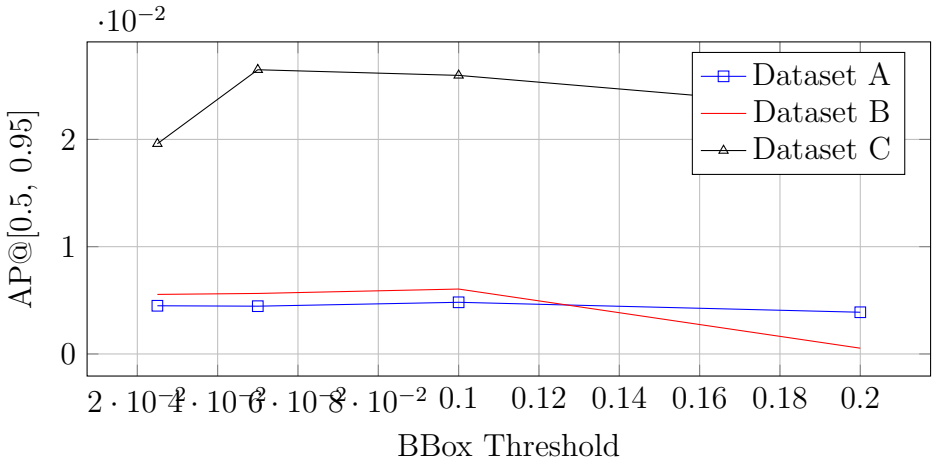



Figure 1: Impact of Bounding Box Threshold on Zero Shot Detection Performance

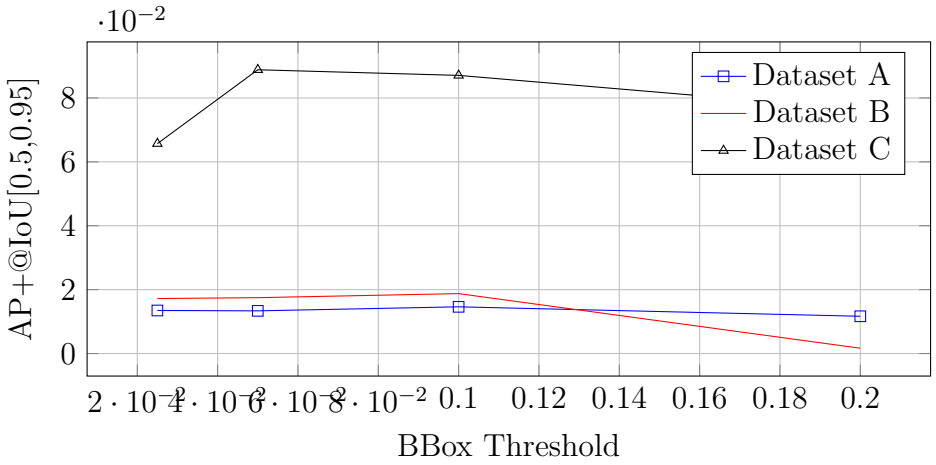



Figure 2: Impact of Bounding Box Threshold on Zero Shot Detection Performance

#### 4.1.3 Key Observations

- Medical domain-specific prompts ("malignant breast tumor") consistently outperformed generic terms. Especially the presence of the word 'tumor' helped more than 'cancer'.
- Optimal bounding box threshold was 0.1.
- Dataset C showed highest zero-shot performance, suggesting better alignment with pre-training data
- Performance gap between datasets indicates significant domain shift.

### 4.2 Task 2: Robustness Analysis of Learned Prompts

#### 4.2.1 CoOp Results

| Dataset | AP(0.5:0.95) | AP-positive | AP50 | AP75 |
|---|---|---|---|---|
| A (train A $\rightarrow$ test A) | 0.031952 | 0.131952 | 0.081670 | 0.008399 |
| B (train B $\rightarrow$ test B) | 0.030868 | 0.030868 | 0.052585 | 0.035068 |
| C (train C $\rightarrow$ test C) | 0.038910 | 0.038910 | 0.060952 | 0.039964 |

Table 3: In-domain performance of CoOp+GDINO on datasets A, B, and C.

| Train $\downarrow$ / Test $\rightarrow$ | A | B | C | Avg | Std |
|---|---|---|---|---|---|
| A | 0.031952 | 0.016507 | 0.060837 | **0.036432** | 0.018373 |
| B | 0.002222 | 0.030868 | 0.053319 | 0.028803 | 0.020911 |
| C | 0.021545 | 0.049158 | 0.038910 | 0.036538 | 0.011397 |

Table 4: Cross-domain AP(0.5:0.95) matrix for CoOp+GDINO across datasets A, B, and C.

### 4.2.2 CoCoOp Results

| Dataset | AP(0.5:0.95) | AP-positive | AP50 | AP75 |
|---|---|---|---|---|
| A (train A → test A) | 0.029981 | 0.089942 | 0.091294 | 0.013782 |
| B (train B → test B) | 0.010216 | 0.031671 | 0.026312 | 0.001094 |
| C (train C → test C) | 0.023663 | 0.079340 | 0.064566 | 0.011422 |

Table 5: In-domain performance of CoCoOp on datasets A, B, and C

| Train ↓ / Test → | A | B | C | Avg | Std |
|---|---|---|---|---|---|
| A | 0.029981 | 0.037654 | 0.082511 | **0.050049** | 0.023384 |
| B | 0.031058 | 0.010216 | 0.034197 | 0.025157 | 0.010332 |
| C | 0.010794 | 0.021126 | 0.023663 | 0.018528 | 0.005589 |

Table 6: Cross-domain AP(0.5:0.95) matrix for CoCoOp across datasets A, B, and C (B pending).

### 4.2.3 Comparative Analysis

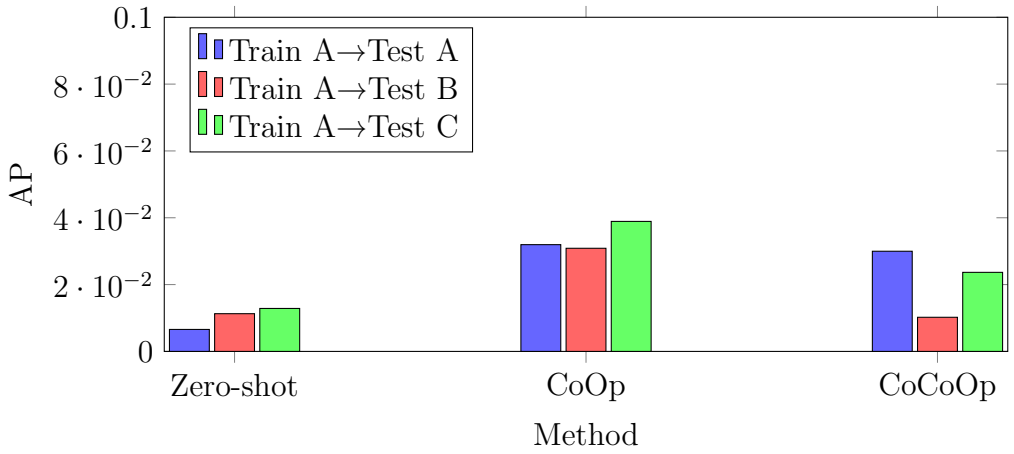



Figure 3: Performance Comparison across Different Methods

### 4.2.4 Domain Shift Analysis

- Instance-specific context generation in CoCoOp should ideally better adaptation, but this happens only with tests after training on A. This is probably as issue of threshold tuning in my opinion, since all three test datasets will work differently for different methods at the same threshold.
- Both methods significantly outperform zero-shot baselines.
- Cross-domain performance suggests Dataset B and C are more similar to each other.
- At least the variance in performances across domains decreases for CoCoOp.

### 4.3 Task 3: Semi-Supervised Prompt Tuning

#### 4.3.1 Semi-Supervised Learning Results

Table 7: Semi-Supervised CoOp Performance (10% Labeled Data)

| **Training Setup** | **Test Set** | **AP** |
|---|---|---|
| *Dataset A (Full) + Dataset B (10% Labeled)* | | |
| Zero-shot Transfer (A→B) | B | 0.016507 |
| Semi-supervised | B | **0.064257** |
| *Dataset B (Full) + Dataset C (10% Labeled)* | | |
| Zero-shot Transfer (B→C) | C | 0.053319 |
| Semi-supervised | C | **0.06279** |

# 5 Discussion

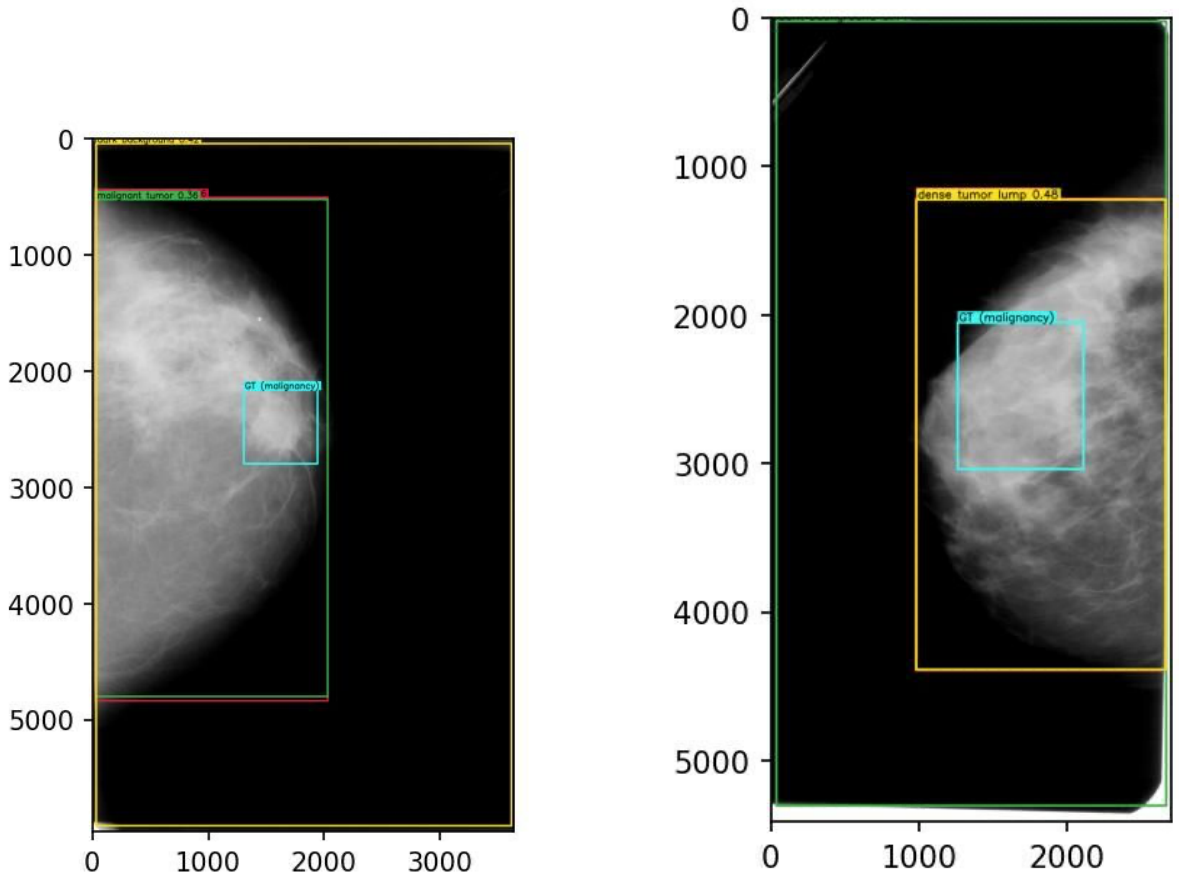
## 5.1 Key Findings

### 5.1.1 Prompt Engineering Insights

1. **Domain-specific terminology matters**: Medical terms like "dense" and "tumor" performed better at zero-shot detection.

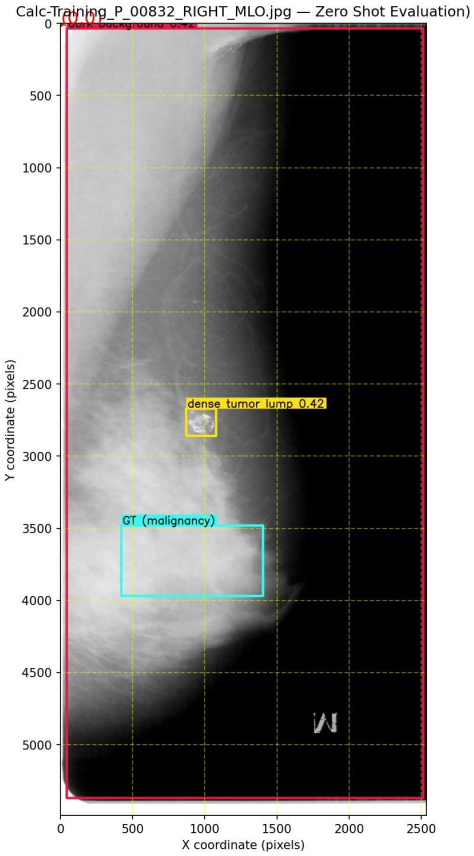
2. **Threshold tuning**: Bounding box threshold around 0.1-0.2 provided best precision-recall trade-off.

## 5.2 Visualization

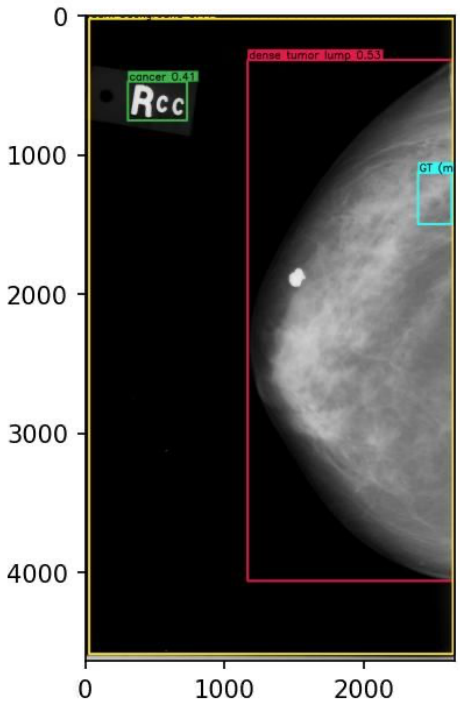



(a) ‘malignant tumor‘ (b) ‘dense tumor lump‘

Figure 4: In zero-shot setting, the words ’malignant tumor’, ’dense lump’ were found to perform better.




(a) Visual aberrations result in false detections.

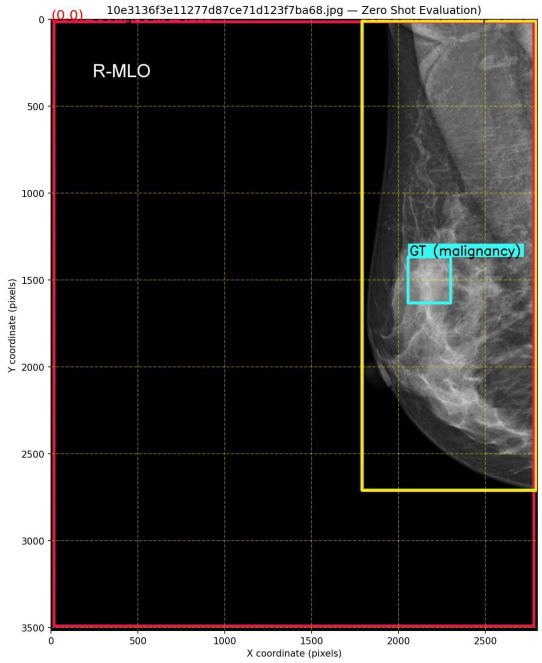



(b) Contrary to intuition, ‘cancer‘ was only detected and localized when there was text in the scans.

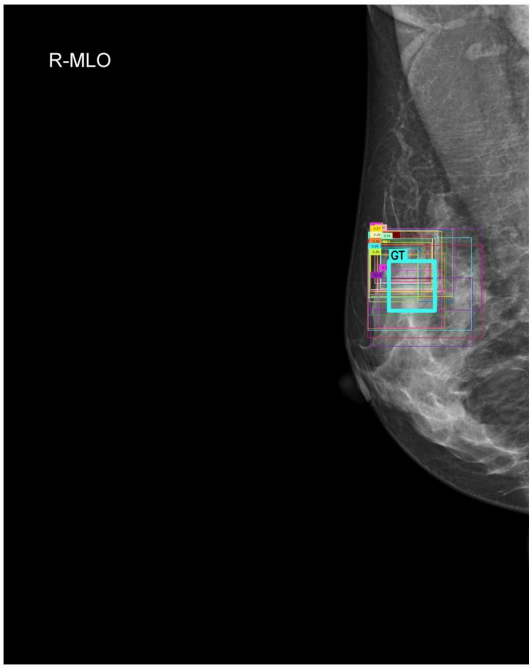
Figure 5: Zero-shot confusers.

We can see some clear artefacts of where the model goes wrong in the zero shot setting.

- It tends to focus on the whole breast, if it does at all.
- The words ‘malignant tumor‘ and ‘dense lump‘ seem to work the best.
- The model gets confused frequently due to reflections/aberrations in the images, which are not a part of the mammographs themselves.
- Contraty to intuition, it was found in a lot of samples that ‘cancer‘ focuses more on any text present in the mammograph, if it does at all.




(a) In zero-shot setting, it was found that any medical tumor-related phrases make the model concentrate on the wole breast. This happens because the model is not familiar with such a specific domain.




(b) After CoOp training, the model learns to concentrate over tumor-like locations. However, it might be easily confused.

Figure 6: Comparison: before and after CoOp-based training

## 5.3 Challenges and Limitations

### 5.3.1 Technical Challenges

- **Hyperparameter sensitivity**: Performance varied significantly with threshold choices
- **Very specific domain**: For this very specific medical imagery task, common thresholds did not hold because the zero-shot rate was way too low.

### 5.3.2 Dataset-Specific Issues

- **Class imbalance**: Benign samples outnumbered malignant 2:1

# References

[1] Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., ... & Su, H. (2024). Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision* (pp. 38-55). Springer.

[2] Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., ... & Li, C. L. (2020). FixMatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33, 596-608.

[3] Zhou, K., Yang, J., Loy, C. C., & Liu, Z. (2022). Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

[4] Zhou, K., Yang, J., Loy, C. C., & Liu, Z. (2022). Learning to prompt for vision-language models. *International Journal of Computer Vision*.