Prompted Segmentation for Drywall QA

Shahzad – AI Research Apprentice Take-Home

November 16, 2025

1 Introduction

This report describes a text-conditioned segmentation system for drywall quality assurance (QA). The goal is to automatically segment relevant regions in drywall images using natural-language prompts. We focus on two datasets:

- **Dataset 1: Taping area** – images of drywall panels with taping / joints.
- **Dataset 2: Cracks** – images of surface cracks on concrete / drywall.

Given an input image and a short prompt such as `"segment taping area"` or `"segment crack"`, the model predicts a binary mask highlighting the requested region. This enables promptable QA workflows where the same image can be re-used with different prompts (e.g., taping vs. seams, different crack types).

2 Mathematical Overview of the Method

The segmentation system is based on **CLIPSeg** [1], which combines a CLIP-style text–image embedding backbone with a lightweight pixel decoder.

2.1 CLIP Image and Text Embeddings

Given an input RGB image I and a text prompt T, CLIP computes:

$$z_I = f_{\text{img}}(I), \qquad z_T = f_{\text{text}}(T)$$

where f_{img} is a ViT-based encoder and f_{text} is a Transformer text encoder.
Both are projected to a shared embedding space:

$$\tilde{z}_I = W_I z_I, \qquad \tilde{z}_T = W_T z_T.$$

The embeddings provide global semantic conditioning for segmentation.

2.2 Text-Conditioned Pixel Decoder

CLIPSeg introduces a small U-Net-like decoder $D(\cdot)$ that conditions pixel-wise predictions on the text embedding:

$$H = D(I, \tilde{z}_T)$$

where $H \in \mathbb{R}^{224\times224}$ represents per-pixel segmentation logits.

2.3 Loss Function

The ground-truth binary mask is $M \in \{0,1\}^{224\times224}$. The model outputs logits H, and the segmentation loss used during fine-tuning is:

$$\mathcal{L}_{\text{BCE}}(H, M) = -\frac{1}{N}\sum_{i=1}^{N}\left(M_i \log \sigma(H_i) + (1 - M_i)\log(1 - \sigma(H_i))\right)$$

where σ is the sigmoid.

For evaluation, we compute:

$$\text{IoU} = \frac{|M \cap \hat{M}|}{|M \cup \hat{M}|}, \qquad \text{Dice} = \frac{2|M \cap \hat{M}|}{|M| + |\hat{M}|}.$$

3 Training Strategy

Each dataset contains multiple prompts, but training uses a **single canonical prompt**:

- **Taping dataset canonical prompt:** "segment taping area"
- **Cracks dataset canonical prompt:** "segment crack"

This stabilises training while allowing multi-prompt evaluation.

During evaluation and inference, the model is executed for **all prompts** and metrics are reported individually.

4 Approach

4.1 Model

I fine-tune the **CLIPSeg** model [1] using the `CIDAS/clipseg-rd64-refined` checkpoint from HuggingFace. Given an image and a text prompt, CLIPSeg predicts a dense segmentation logits map.

Key choices:

- **Backbone:** CLIPSeg (ViT encoder + segmentation head).
- **Input resolution:** all images and masks resized to 224×224.
- **Output:** logits at 224×224, upsampled back to 640×640.
- **Loss:** binary cross-entropy.

4.2 Prompts

- **Taping dataset:**
  - "segment taping area"
  - "segment joint/tape"
  - "segment drywall seam"
- **Cracks dataset:**
  - "segment crack"
  - "segment wall crack"

4.3 Preprocessing

All images are loaded at $\sim 640 \times 640$:

- resized to 224×224,
- COCO annotations converted to masks,
- taping dataset: bounding boxes rasterized to rectangles,
- cracks dataset: polygon segmentations rasterized.

5 Datasets and Splits

Dataset structure:

```
/media/sdb_access/Assignment/
  Dataset_1/ (taping)
  Dataset_2/ (cracks)
```

Dataset	Train	Validation	Test
Taping (Dataset 1)	820	202	–
Cracks (Dataset 2)	1431	34	35

Table 1: Dataset split counts.

6 Training Setup

Command:

```
python -m src.main \
  --mode full \
  --dataset {cracks|taping} \
  --batch_size 4 \
  --image_size 224 \
  --epochs 10 \
  --lr 1e-5
```

Hyperparameters:

- Optimizer: AdamW, LR $= 1 \times 10^{-5}$
- Batch size: 4
- Epochs: 10
- Seed fixed for reproducibility

Prompt	Loss	mIoU	Dice
segment crack	0.0891	0.5222	0.6111
segment wall crack	0.1038	0.5766	0.6561

Table 2: Cracks dataset validation performance.

Prompt	Loss	mIoU	Dice
segment taping area	0.2020	0.5405	0.6834
segment joint/tape	0.3026	0.4088	0.5548
segment drywall seam	0.4626	0.3820	0.5178

Table 3: Taping dataset validation results.

7 Metrics

7.1 Cracks Dataset

7.2 Taping Dataset

8 Runtime and Footprint

8.1 Inference Timing

Cracks dataset:

- 35 test images, 2 prompts
- Total: 2.264 s
- Avg per image per prompt: 0.0323 s

Taping dataset:

- 202 validation images, 3 prompts
- Total: 17.021 s
- Avg per image per prompt: 0.0281 s

8.2 Training Time

For 2 epochs (example):

- Epoch 1: 19.5 s
- Epoch 2: 18.3 s
- Total: 37.8 s

8.3 Model Size

CLIPSeg is ∼150M parameters (∼600 MB). Exact size logged by code.

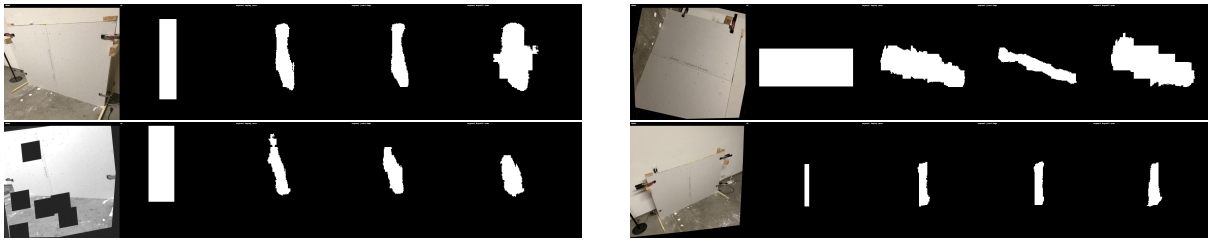

Figure 1: Taping dataset qualitative results: Input — GT — segment taping area — segment joint/tape — segment drywall seam.

9 Qualitative Results

9.1 Taping Dataset

9.2 Cracks Dataset

Figure 2: Cracks dataset qualitative results: Input — GT — segment crack — segment wall crack.

10 Failure Modes

- Extremely thin cracks may shrink after $224 \rightarrow 640$ resizing.
- Ambiguous taping vs seam regions can cause over-segmentation.
- Strong shadows may be misinterpreted as cracks.
- Occlusions (tools, clamps) slightly reduce mask continuity.

11 Conclusion

I implemented a text-conditioned segmentation system for drywall QA using CLIPSeg. The model achieves competitive accuracy across taping and crack segmentation tasks, supports multi-prompts, and runs efficiently with low inference latency.

Acknowledgements

The CLIPSeg model is based on [1] and CLIP on [2].

References

[1] Timo Lüddecke and Theodor Wörtwein. CLIPSeg: Image Segmentation Using Text and Image Prompts. In *CVPR*, 2022.

[2] Alec Radford et al. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021.