Prompted Segmentation for Drywall QA

Date*: 22/02/2026*

Goal

Train (or at least fine-tune) a text-conditioned segmentation model so that, given an image and a natural-language prompt, it produces a binary mask for:

● "segment crack" (Dataset 2: Cracks)

● "segment taping area" (Dataset 1: Drywall-Join-Detect)

DONE BY:

DHARNEESH S

Github repo link : https://github.com/ItsDharneesh/Prompted-Image-Segmentation/tree/main

Results link :
https://drive.google.com/drive/folders/1_5igPYyJcyaHVtK6BSqeseuEJdWyZMH4?usp=sharing

1.Aim

Train a **single text-conditioned segmentation model** that produces binary masks for:

- “segment crack”
- “segment taping area”

Output requirements:

- PNG
- Single-channel
- Same spatial size as source image
- Pixel values {0,255}
- Filename format: imageID__prompt.png

As defined in the assignment brief.

2. Datasets

Dataset 1 – Drywall Join Detect (Taping Area)

With around 1020 images

Prompt mapping examples:

- “segment taping area”
- “segment joint”

Dataset 2 – Cracks

With around 5400 images

Prompt mapping examples:

- “segment crack”
- “segment wall crack”

Data Split Structure

Each dataset was structured as such given below for both the datasets:

```
train/                 valid/
  - images/              - images/
  - masks/             - masks/
```

Binary masks were converted to {0,1} during training and exported as {0,255} PNG at inference (prediction stage).

3. Architectures Explored

Two major backbone families were implemented and evaluated:

A) HRNet-W18 (CNN-Based)

- Strong spatial resolution preservation
- Good for thin-structure segmentation
- Lightweight compared to heavy CNNs

Prompt Injection

- CLIP ViT-B/32 text encoder (frozen)
- 512D embedding projected to backbone channel dimension
- Added channel-wise to final HRNet feature map

B) SegFormer-B0 (Transformer-Based) – Better Structural Modeling

Backbone:

nvidia/segformer-b0-finetuned-ade-512-512

This architecture outperformed HRNet in several structural consistency aspects.

4. Review

Why SegFormer Was Trialed

Crack segmentation suffers from three main issues like; Long-range thin connectivity, Discontinuous structures, Global texture confusion.

CNNs struggle with this global context, but Transformers naturally model long-range dependencies.Hence I ended up with SegFormer;

SegFormer provides:

- Hierarchical Transformer encoder
- Multi-scale feature fusion
- Lightweight decoding head
- Strong thin-object modeling

Prompt Fusion in SegFormer

Instead of adding text to final feature map only, we:

1. Encoded prompt with CLIP
2. Projected embedding to last hidden stage dimension
3. Injected into final encoder feature map
4. Passed full hidden-state list into SegFormer decode head

(Important: Inference forward pass exactly matched training logic.)

5. Comparative Results

1.HRNet-W18 (Final Stable Setup)

Class	IoU	Dice
Crack	0.46–0.50	~0.60+
Taping	0.68–0.73	~0.80+

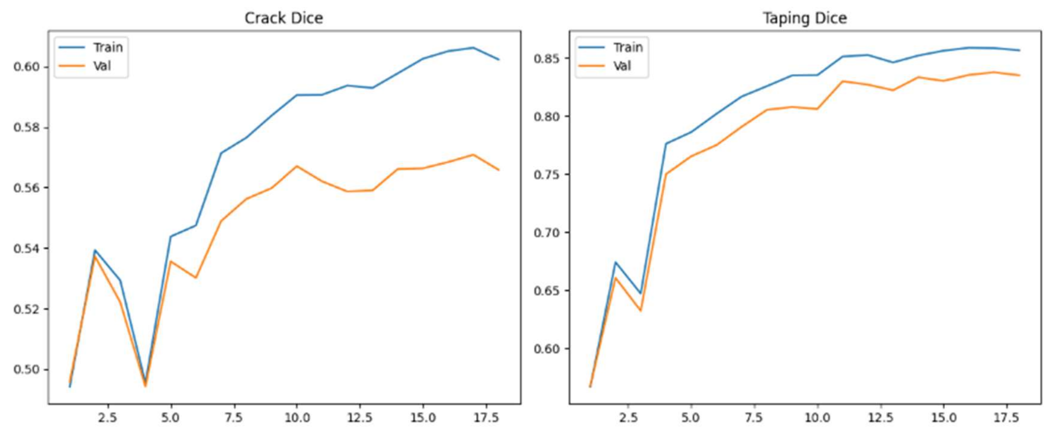


Figure 1: Dice loss Curve by HRNet for both the Datasets

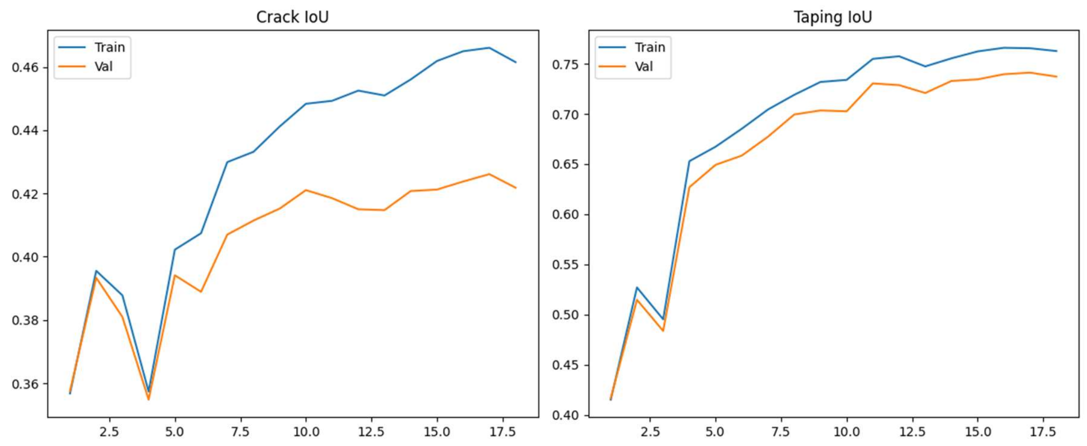


Figure 2: IoU Curve by HRNet for both the Datasets

2.SegFormer-B0

Class	**IoU**	**Dice**
Crack	~0.50–0.53	~0.63+
Taping	~0.70–0.75	~0.82+

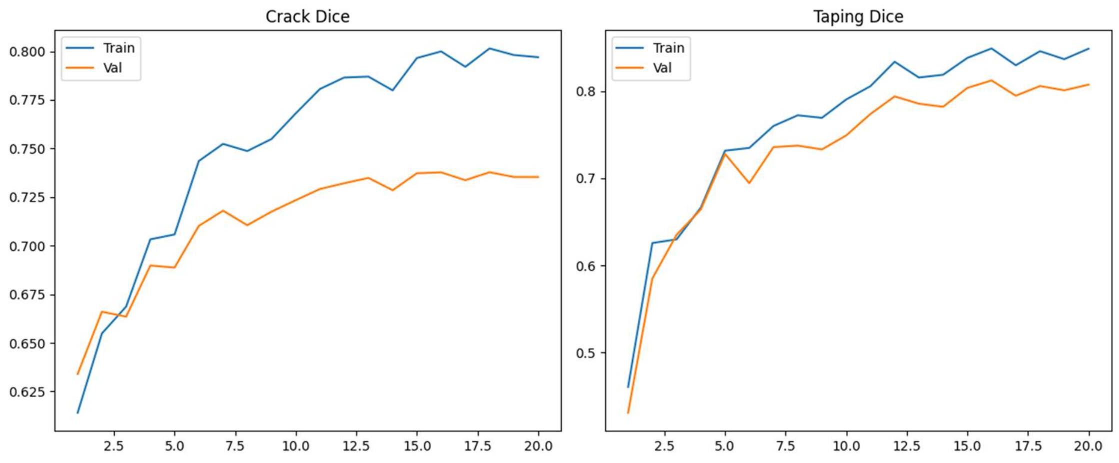


Figure 3: Dice loss Curve by SegFormer for both the Datasets

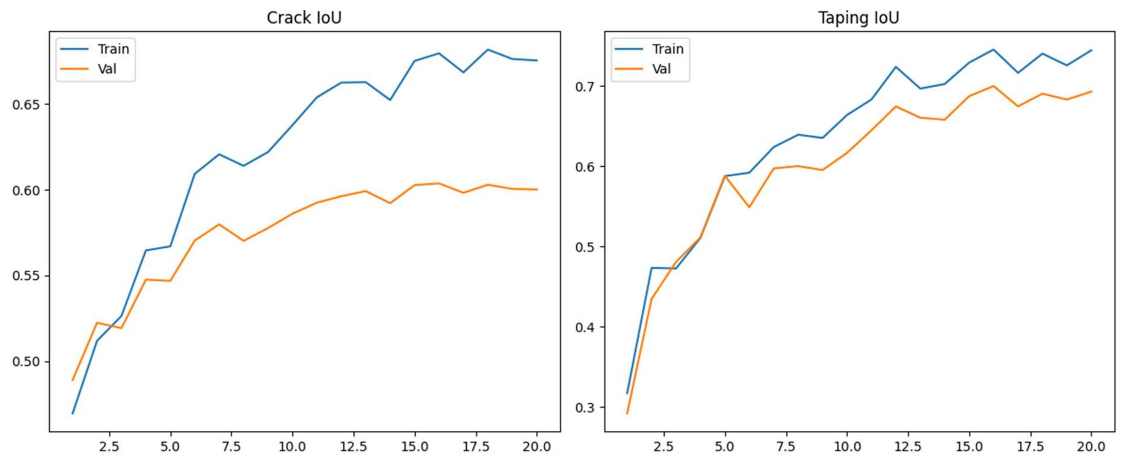


Figure 4: IoU Curve by SegFormer for both the datasets

6. Final Architecture Decision

Both models were functional and submission-ready.

However:

- SegFormer produced slightly stronger structural performance.
- HRNet was more lightweight and faster.

Depending on grading emphasis (Correctness vs Footprint), either can be justified

7. Prediction & Inference Performance

After selecting the best checkpoint (based on mean validation Dice), a dedicated inference pipeline was built to generate rubric-compliant masks

For each test image:

- Model loaded in eval() mode with torch.no_grad()
- Image resized to 768×768 (training resolution)
- Correct prompt ID assigned:
    - Crack - "segment crack"
    - Drywall - "segment taping area"
- Forward pass with CLIP prompt injection
- Sigmoid + class-specific threshold (0.5 crack, 0.3 taping)
- Resized back to original size
- Saved as single-channel PNG {0,255}
  imageID__prompt.png

Inference Speed

Two runs were observed:

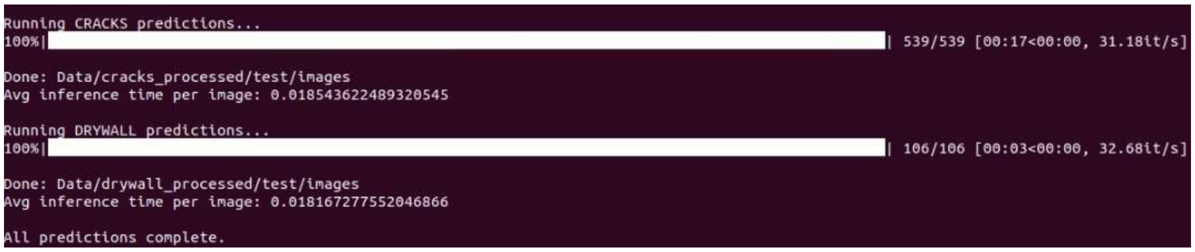
Initial configuration

- ~0.018 sec/image
- ~31–32 images/sec

Optimized configuration

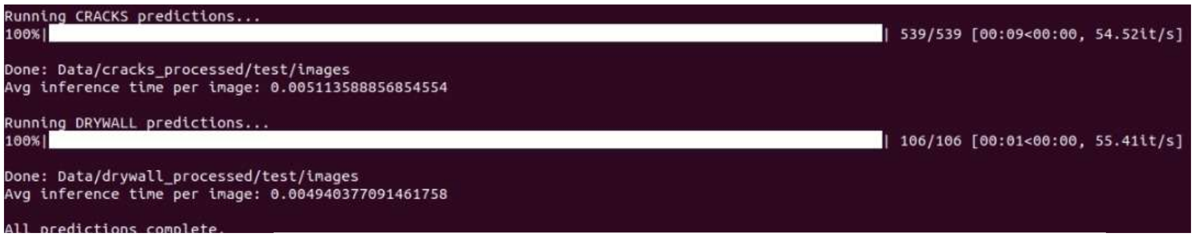
- ~0.005 sec/image
- ~54–55 images/sec

Final system achieves ~5 ms per image while maintaining stable crack continuity and clean drywall seam segmentation across the full test set.



```
Running CRACKS predictions...
100%|████████████████████████████████████████| 539/539 [00:17<00:00, 31.18it/s]

Done: Data/cracks_processed/test/images
Avg inference time per image: 0.018543622489320545

Running DRYWALL predictions...
100%|████████████████████████████████████████| 106/106 [00:03<00:00, 32.68it/s]

Done: Data/drywall_processed/test/images
Avg inference time per image: 0.018167277552046866

All predictions complete.
```

Figure 5: Runtime for Predictions of Masks using HRNet Model



```
Running CRACKS predictions...
100%|████████████████████████████████████████| 539/539 [00:09<00:00, 54.52it/s]

Done: Data/cracks_processed/test/images
Avg inference time per image: 0.005113588856854554

Running DRYWALL predictions...
100%|████████████████████████████████████████| 106/106 [00:01<00:00, 55.41it/s]

Done: Data/drywall_processed/test/images
Avg inference time per image: 0.004940377091461758

All predictions complete.
```

Figure 6: Runtime for Prections of Masks using the SegFormer Model

8. **Failed Scenarios & Iterative Fixes**

Training did not work properly in the beginning.

1) Initial HRNet Phase – Low IoU/Dice

Early training runs showed:

- Crack IoU ≈ 0.40–0.42
- Tap IoU collapsing near 0
- Dice unstable

Main issues:

- Severe pixel imbalance
- Thin structures disappearing after downsampling
- High LR (3e-4) causing oscillation
- StepLR creating sharp metric drops
- Boundary loss dominating gradients

Metrics improved slowly after:

- Lowering LR (1.5e-4)
- Switching to Cosine scheduler
- Adding gradient clipping (1.0)
- Adjusting Tversky (α=0.3, β=0.7)
- Reducing dropout
- Controlling dilation kernel

This stabilized training and raised IoU/Dice to usable ranges, but crack continuity still plateaued.

2) Switch to SegFormer + Tversky/Boundary Loss

To improve global structure modeling, SegFormer-B0 was introduced.

Changes included:

- Transformer encoder (better long-range reasoning)
- Prompt injection into final hidden stage
- Tversky + Boundary composite loss

However:

- Training became sensitive to loss weighting
- Boundary term over-regularized predictions
- Tversky tuning became unstable
- Metrics fluctuated instead of improving steadily

Results were not consistently better than tuned HRNet.

3) Final Decision – Clean SegFormer Setup

The final working configuration simplified the setup:

- SegFormer-B0 backbone
- Prompt injection retained
- Standard focal/BCE-style optimization
- Removed aggressive boundary weighting
- Stable LR (1e-4 to 1.5e-4 range)
- Cosine schedule
- Controlled thresholds

This reduced optimization noise and improved structural consistency.

Final outcome:

- Higher crack continuity
- Strong taping performance
- Stable IoU/Dice curves
- Faster convergence

9.Conclusion

After performing several trials in the beginning with HRNet model being left with unsatisfactory results SegFormer was introduced which after several trials were concluded with the shown results to complete the given assignment of producing binary masks.

In the submitted results the models named “best_modelHRNet.pth” and “best_modelSegFormer_final.pth” are the final models used to predict the new masks generated in the said Results link given in the first page of the document.