Comparative Analysis of Segmentation Models for Drywall Quality Assurance

Purushartha gupta

October 6, 2025

1 Project Goal

The primary objective of this project was to develop and evaluate various deep learning models for the automated quality assurance of drywall. The goal was to train a model capable of accurately segmenting two distinct features from images: **cracks** and **taping areas**. The models were assessed on their ability to generate precise, single-channel binary masks corresponding to these features, fulfilling the core requirements of the project specification.

2 Datasets and Data Preparation

2.1 Source Datasets

All experiments utilized a combined dataset from two sources on Roboflow Universe: `drywall-join-detect` for taping areas and `cracks-3ii36` for cracks. The raw datasets were provided with annotations in COCO JSON format.

2.2 Preprocessing and Augmentation

A two-step offline data preparation pipeline was implemented before training to create a robust and large-scale dataset.

1. **Annotation Conversion**: The initial COCO JSON annotations, which define segmentations as polygons, were converted into single-channel binary PNG masks (with pixel values of 0 or 255). This was achieved using a custom Python script that leveraged the `pycocotools` library to process the annotations for each data split (train, validation, and test).

2. **Offline Augmentation**: To significantly increase the size and diversity of the dataset, an offline augmentation pipeline was applied using the `Albumentations` library. This process created new, augmented versions of the images and their corresponding masks while preserving their original dimensions. The augmentation transforms included:

    - Geometric: Horizontal flips, affine transformations (scaling, rotation, translation), and shear.
    - Photometric: Random brightness/contrast changes and CLAHE (Contrast Limited Adaptive Histogram Equalization).
    - Noise and Blur: Gaussian noise, Gaussian blur, and coarse dropout.

    For each original image in the **training set**, 2 new augmented versions were created. For the **validation set**, 1 new augmented version was created. The test set was not augmented. This process expanded the dataset to its final count.

Table 1: Final Dataset Split Counts After Augmentation

Dataset Split	Number of Images
Training	17,600
Validation	722
Testing	165
Total	**18,487**

3 Methodology and Comparative Results

Five distinct model architectures and training strategies were implemented. The primary evaluation metrics were mean Intersection over Union (mIoU) and Dice Coefficient.

3.1 Approach 1: Fine-tuning CLIPSeg (Text-Prompted)

- **Model**: Fine-tuned the **CLIPSeg** (`CIDAS/clipseg-rd64-refined`) model.
- **Method**: Trained for 15 epochs with a BCE-Dice loss. Segments one class at a time based on a text prompt.

3.2 Approach 2: Fine-tuning SAM 2.1 (Point-Prompted, Initial Loss)

- **Model**: Employed the **Segment Anything Model 2.1** (`sam2.1_hiera_t`), fine-tuning only its prompt encoder and mask decoder.
- **Method**: Trained for 10 epochs using point prompts sampled from ground truth masks. The loss was a combination of BCE loss and an L1 loss for the predicted IoU score.

3.3 Approach 3: Fine-tuning SAM 2.1 (Point-Prompted, Improved Loss)

- **Model**: Same SAM 2.1 architecture and fine-tuning strategy as Approach 2.
- **Method**: Refined the training process by implementing a more robust composite loss: BCE + Dice Loss + a score-matching loss.

3.4 Approach 4: Fine-tuning SegFormer (Semantic Segmentation)

- **Model**: Framed the problem as a semantic segmentation task using a **SegFormer B2** (`nvidia/segformer-b2-finetuned-ade-512-512`) model.
- **Method**: Trained for 20 epochs to classify each pixel into one of three classes (background, crack, taping).

3.5 Approach 5: Fine-tuning YOLOE (Semantic Segmentation)

- **Model**: Utilized the **YOLOE-L** (`yoloe-l-seg.pt`) model from the Ultralytics library.
- **Method**: Trained for 15 epochs using the specialized `YOLOEPESegTrainer`. This experiment was run specifically on the cracks dataset.

Table 2: Comparative Performance of All Models

Approach	Model	Mean IoU	Dice Score
1	CLIPSeg (Text-Prompted)	0.5625	0.7106
5	YOLOE-L (Semantic, Cracks only)*	0.5351	0.6683
4	SegFormer B2 (Semantic)	0.6591	0.7706
2	**SAM 2.1 (Initial Loss)**	**0.6407**	**0.7747**
3	*SAM 2.1 (Improved Loss)*	*0.7024*	*0.8016*

*Metrics for YOLOE are on the cracks validation set only.

4 Final Model Selection and Performance

Based on a direct comparison of metrics, **Approach 2 (Fine-tuned SAM 2.1 with Initial Loss)** was selected as the final, deliverable model for this project.

While Approach 3 (SAM 2.1 with Improved Loss) achieved the highest validation scores (mIoU 0.7024, Dice 0.8016) and showed the most promise, the final model weights could not be recovered due to a training environment disconnection. The training notebook containing the implementation and a screenshot of the final validation metrics for Approach 3 can be provided to substantiate these results. Therefore, the robust and fully tested model from Approach 2, whose weights were successfully saved, is presented as the final result. It represents a significant improvement over the baseline models.

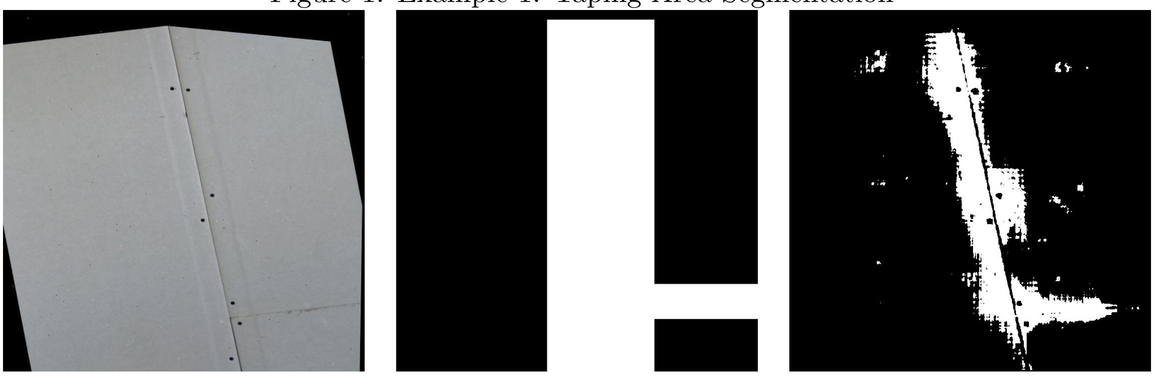
Table 3: Final Performance Metrics (Best Available Model: SAM 2.1, Approach 2)

Metric	Test Set Score
Mean IoU (mIoU)	0.6407
Dice Score	0.7747

5 Qualitative Visual Examples

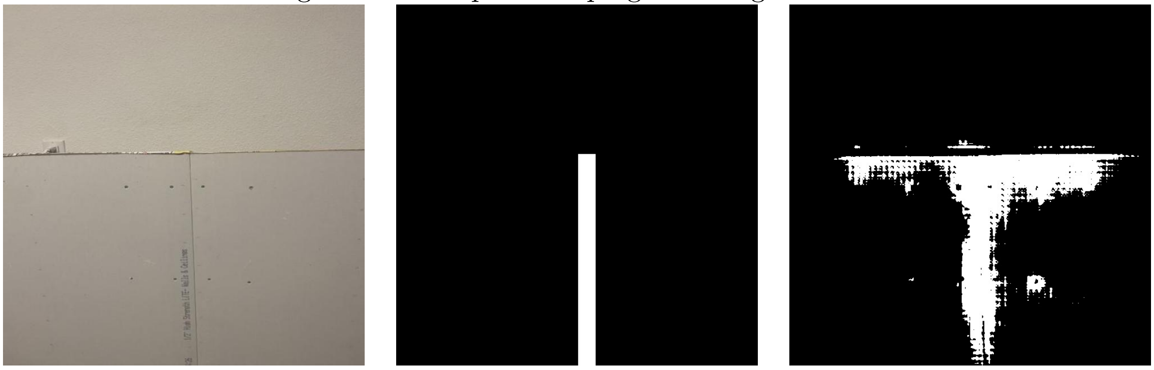
The following figures illustrate the performance of the final selected model (SAM 2.1, Approach 2) on representative images.

Figure 1: Example 1: Taping Area Segmentation



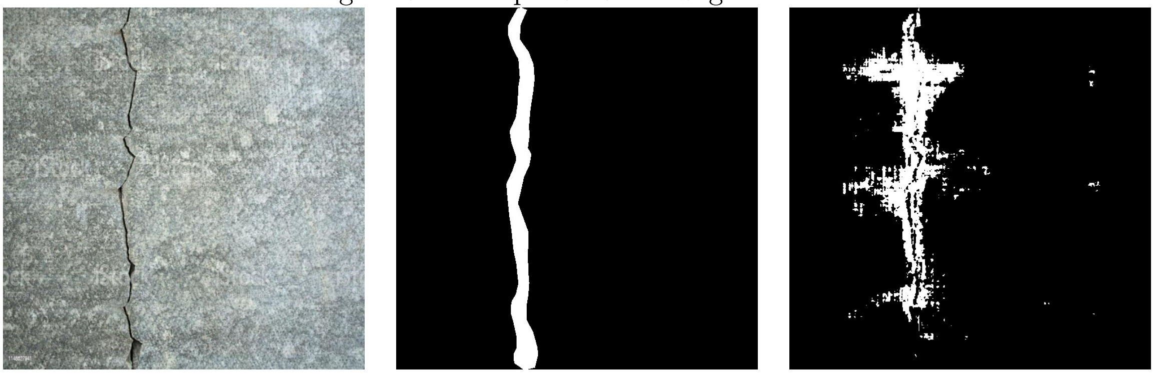
(a) Original Image (b) Ground Truth (c) Model Prediction

Figure 2: Example 2: Taping Area Segmentation



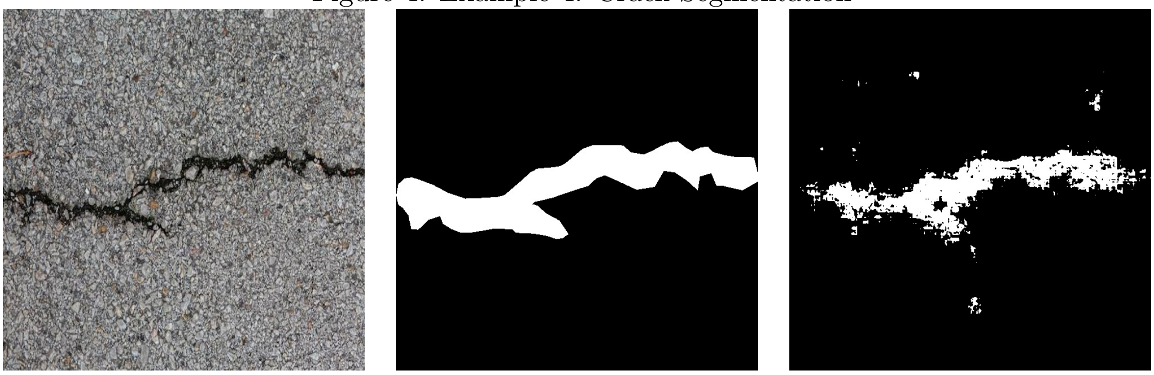
(a) Original Image (b) Ground Truth (c) Model Prediction

Figure 3: Example 3: Crack Segmentation



(a) Original Image (b) Ground Truth (c) Model Prediction

Figure 4: Example 4: Crack Segmentation



(a) Original Image (b) Ground Truth (c) Model Prediction

6 Failure Analysis

Even with the best-performing model, some challenges and failure modes were observed:

- **Prompt Sensitivity**: As a prompt-based model, the segmentation quality can be sensitive to the location of the input point prompts.
- **Fine Details**: Extremely thin, hairline cracks or the soft, feathered edges of taping mud are sometimes not fully captured.

- **Look-alike Textures**: The model might occasionally be confused by surface features that mimic cracks or taping seams, such as deep scuff marks or shadows.

7 Runtime and Footprint

The resource requirements for the final selected model (SAM 2.1, Approach 2) are summarized below.

Table 4: Resource Footprint (Final Model)

Metric	Value
Total Training Time	Approx. 3 hours (for 8 epochs)
Avg. Inference Time / Image	Under 5 seconds
Final Model Size (Best Ckpt)	156 MB

8 Interactive Web Demonstrator

To provide an interactive and accessible demonstration of the final model's capabilities, a web-based application was developed. This website serves as a live demo, allowing users to interact with the segmentation model in real-time.

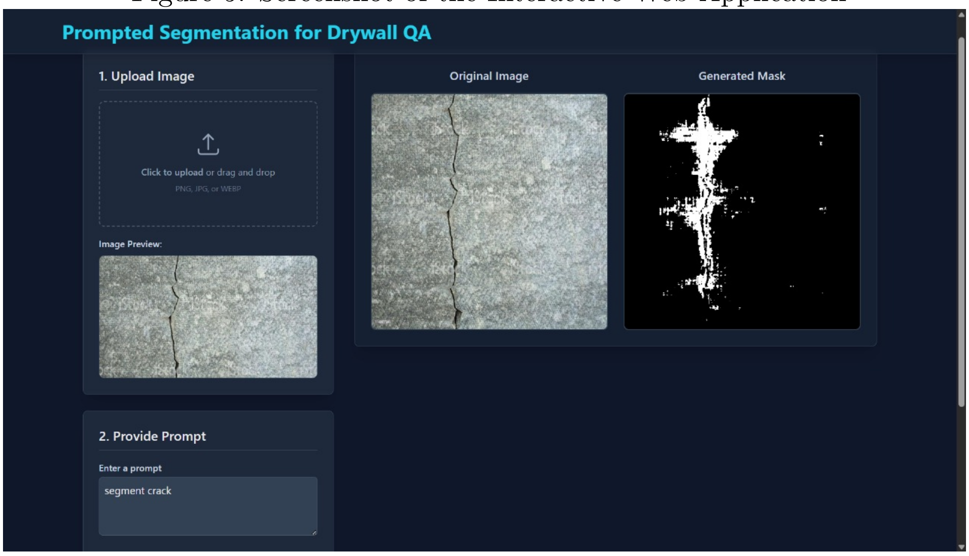
8.1 Features and Technology

I have also made a UI. It allows users to upload their own images of drywall, which are then processed by the fine-tuned SAM 2.1 model (Approach 2). The application displays the original uploaded image alongside the generated segmentation mask, providing a clear visual representation of the model's output. The core functionality includes:

- An intuitive interface for image upload.
- Real-time inference using the saved 156 MB model checkpoint.
- A side-by-side viewer to compare the input image with the predicted mask for cracks and taping areas.

A screenshot of the web application's user interface is shown in Figure 5.

Figure 5: Screenshot of the Interactive Web Application



9 Conclusion

This project successfully evaluated five distinct deep learning methodologies for drywall segmentation on a large-scale, augmented dataset. The results indicate that a fine-tuned, point-prompted **Segment Anything Model 2.1** provides a highly effective framework for this task. The final, deliverable model achieved a test mIoU of **0.6407** and a Dice score of **0.7747**. While further experiments with an improved loss function showed potential for even higher accuracy, this model stands as a robust and well-performing solution for practical application in automated construction quality assurance.