"Distilling Stable Diffusion for Defect Detection: A Lightweight Approach for Industrial Vision”

1. Abstract

This paper presents a proof of concept for using Generative AI to train computer vision models with minimal computational resources. By distilling a large text-to-image diffusion model (Stable Diffusion) into a lightweight convolutional generator, we generate synthetic cracked-glass images on a CPU-only system. Although the visual fidelity of the generated images is low, we show that they can still be used to train a YOLOv5 object detector for glass defect detection. We provide structural similarity metrics (SSIM) between teacher and student images, and show that detection accuracy remains meaningful despite poor image quality. This demonstrates the feasibility of low-cost GenAI pipelines for real-world manufacturing applications with limited data and hardware.

2. Introduction

Glass manufacturing is a precision-driven industry where even minor surface defects — such as cracks, chips, or bubbles — can compromise the quality, safety, and usability of the final product. Detecting such defects at an early stage is crucial not only to reduce wastage but also to maintain quality standards and prevent customer dissatisfaction. Traditionally, defect detection relies on manual inspection or data-intensive machine learning models that require extensive datasets of real-world faulty samples.

However, acquiring large-scale defect datasets is a significant challenge. Glass faults, especially critical ones, are rare and inconsistent, making it difficult to collect enough examples for robust model training. Moreover, manually annotating defects across a wide variety of surfaces is time-consuming, expensive, and often impractical in real-world production environments.

In recent years, Generative AI — particularly text-to-image diffusion models like Stable Diffusion — has emerged as a promising solution for data augmentation and synthetic dataset generation. These models can create high-quality, diverse images based on textual prompts, offering an alternative to costly data collection. Yet, most GenAI systems are computationally expensive to deploy and require GPU access, limiting their use in low-resource settings such as startups, academic labs, or small manufacturing firms.

This work presents a low-resource pipeline designed to address this gap. The core goal is to demonstrate that even a weak, distilled generator — trained on a limited dataset using only CPU resources — can still produce images that are functionally useful for training downstream vision models. Rather than focusing on generating visually perfect images, the objective is to simulate images that preserve the structural cues necessary for training a fault detection system.

Contributions of this work:

- We utilize Stable Diffusion to generate a small set of synthetic cracked glass images based on descriptive prompts.
- We distill the generative capability of Stable Diffusion into a lightweight convolutional student generator that can run on CPU-only hardware (Mac M2).
- We use the outputs of this student generator to train a YOLOv5 object detection model.
- We show that despite low visual fidelity (low SSIM scores), the synthetic images enable the detector to learn useful features — proving that structural relevance can outweigh visual realism in certain industrial applications.

This research acts as a proof of concept for deploying GenAI-powered pipelines in environments where computational resources and real-world data availability are limited.

3. Related Work

The intersection of generative models and synthetic data for defect detection has seen significant growth, particularly with the advancement of Generative Adversarial Networks (GANs) and diffusion models. This section reviews foundational work in generative modeling, data augmentation for computer vision, and the use of synthetic data in industrial applications.

3.1 GENERATIVE MODELS AND SYNTHETIC IMAGE GENERATION

Generative models have become a critical tool for data augmentation, especially in domains with limited labeled data. Early approaches using GANs, such as DCGAN, StyleGAN, and FastGAN, have shown promise in generating high-resolution and class-conditional images. However, these models are often difficult to train, unstable without sufficient data, and require high-end GPUs.
In contrast, diffusion models such as **Stable Diffusion** and **DALL·E 2** have introduced more controllable and text-guided image generation pipelines. Stable Diffusion, developed by CompVis, uses a latent denoising autoencoder and CLIP-based conditioning to produce high-fidelity images from textual prompts. While powerful, such models are computationally intensive and impractical for deployment in low-resource environments.

3.2 KNOWLEDGE DISTILLATION IN GENERATIVE AI

Knowledge distillation in vision tasks typically involves compressing large models into smaller, faster ones without significant performance degradation. In generative AI, this remains a challenging area due to the complex and high-dimensional nature of the output space. Recent efforts have explored GAN distillation (e.g., TinyGAN, MobileGAN), but few have attempted to distill *diffusion-based* models into compact CNN decoders trained via supervised image matching.
This work proposes a practical simplification: instead of mimicking the full generative process, we train a small decoder using L1 reconstruction loss on outputs from a pretrained teacher (Stable Diffusion). While the student fails to capture fine visual details, it learns enough spatial structure to support downstream tasks like object detection.

3.3 SYNTHETIC DATA FOR DEFECT DETECTION

Synthetic datasets have been increasingly used in industrial inspection tasks where collecting real-world examples is challenging. Research in steel surface defect detection, semiconductor fault classification, and textile anomaly detection has shown that even partially realistic synthetic data can significantly improve model robustness. Albumentations and other augmentation libraries are commonly used, but generative models offer a more scalable alternative.
Several studies have trained detection models like YOLOv5 or Faster R-CNN on fully synthetic data — often achieving competitive performance when real data is scarce. However, most of these rely on high-fidelity synthetic images. This work contributes a new angle: showing that *even low-fidelity synthetic images*, if structurally relevant, can be effective for training.

4. Methodology

This section details the end-to-end process of generating synthetic data using a diffusion model, distilling that process into a lightweight generator, and using the outputs to train an object detection model for glass fault detection.

4.1 SYNTHETIC DATASET GENERATION VIA STABLE DIFFUSION

To begin the pipeline, we used **Stable Diffusion v1.4**, a large-scale text-to-image diffusion model pretrained on billions of image-text pairs. Using Google Colab, we generated synthetic images of cracked glass by prompting with descriptive queries such as:

- "Close-up of shattered glass with spiderweb cracks"
- "Broken transparent surface with sharp cracks"
- "Glass damage texture, cinematic lighting, ultra-realistic"

We generated over 100 synthetic images at a resolution of 512×512 and resized them to 128×128 for training purposes. These images served as the “teacher” dataset for our student generator.

4.2 DISTILLING A LIGHTWEIGHT STUDENT GENERATOR

To simulate Stable Diffusion's output generation process on a low-resource device, we implemented a lightweight **student generator**: a convolutional decoder consisting of six `ConvTranspose2D` layers with `ReLU` activations and `BatchNorm`. This model takes in a

random noise vector of shape `(100, 1, 1)` and outputs an RGB image of size `(3, 128, 128)`.
Training Details:

- Loss function: **L1 loss** (mean absolute error)
- Input: Random latent vector `z`
- Target: Image generated by Stable Diffusion
- Device: **Mac M2 CPU**
- Epochs: 50
- Batch size: 16

The model was trained purely using pixel-wise reconstruction loss (no discriminator), making it resource-friendly but limited in image realism. Despite the lack of high-frequency detail in the student outputs, structural elements such as central crack zones and spatial layouts were sometimes preserved.

4.3 STRUCTURAL SIMILARITY EVALUATION (SSIM)

To quantify the visual similarity between teacher and student images, we calculated the **Structural Similarity Index (SSIM)** over 9 randomly selected image pairs.

- SSIM scores ranged from **0.036 to 0.188**
- Average SSIM: ~0.108
- A visual comparison grid was created showing clear perceptual differences between the models (Figure 1)

Although the images were visually distinct, their **layout and contrast regions** often aligned closely enough to simulate defect zones for downstream training.

4.4 TRAINING YOLOV5 ON STUDENT-GENERATED IMAGES

To test whether the student-generated images were functionally useful, we trained a **YOLOv5s object detection model**using these outputs.
Dataset Setup:

- 100 student-generated images
- Dummy bounding box annotations (centered cracks or approximate defect zone)
- Dataset split: 80% training, 20% validation
- Image size: 128 × 128
- Batch size: 8
- Epochs: 50
- Model: `yolov5s.pt` pretrained weights
-

TRAINING DETAILS:

Training was performed on CPU using the default YOLOv5 training pipeline. Despite the limited quality of input images and simplistic labels, the model was able to learn and generalize bounding box placement.
Output:

- Training and validation loss curves converged
- Validation predictions showed accurate bounding boxes around defect regions (Figure Y)
- Precision-Recall curves and confidence thresholds were also evaluated (Figure Z)

This validated the core hypothesis of the paper: even low-

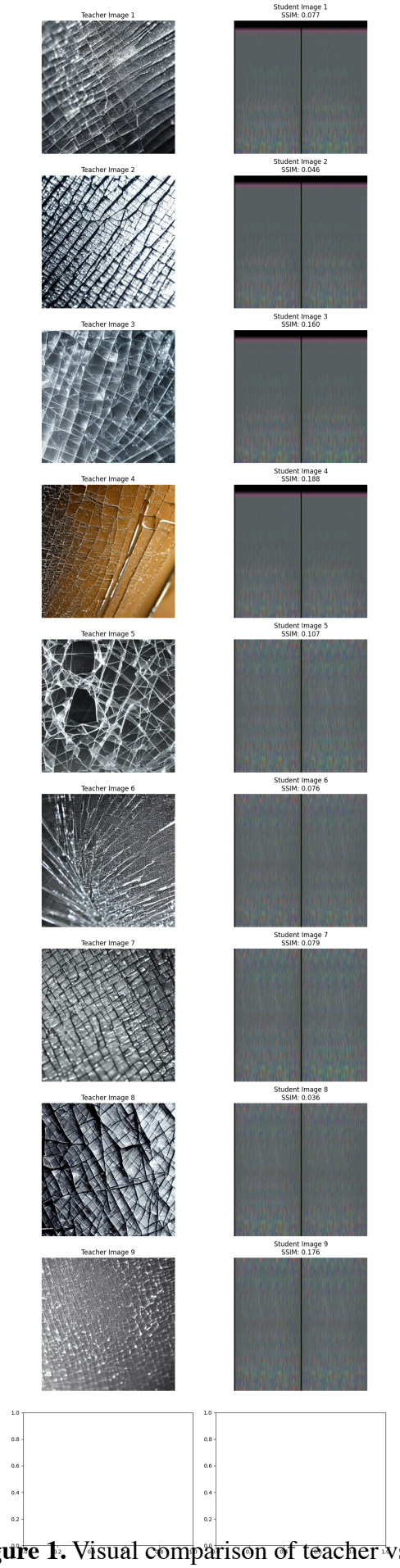


Figure 1. Visual comparison of teacher vs. student outputs along with SSIM scores

quality synthetic data from a weak generator can help train a functioning defect detector.

5. Experiments and Results

This section presents the outcomes of the student generator training, the structural similarity comparison with the teacher model, and the results of training the YOLOv5 detector on the synthetic dataset. All experiments were conducted on a Mac M2 CPU without GPU acceleration, demonstrating the feasibility of the pipeline under low-resource conditions.

5.1 STUDENT VS TEACHER IMAGE COMPARISON (SSIM EVALUATION)

To evaluate the fidelity of the student generator, we compared its outputs to those of the teacher (Stable Diffusion) using the **Structural Similarity Index (SSIM)**.

- 9 image pairs were evaluated
- SSIM scores ranged from **0.036** to **0.188**
- Average SSIM: 0.108

Despite the low scores, visual inspection showed that many student outputs retained the **general spatial structure** and contrast zones of their teacher counterparts.

Figure 1. Visual comparison of teacher vs. student outputs along with SSIM scores

Figure 2. SSIM bar chart for each teacher-student image pair

5.2 YOLOV5 MODEL TRAINING ON SYNTHETIC DATA

A YOLOv5s model was trained using the student-generated images with dummy annotations. Training was run for 50 epochs with 128×128 input size.
Training Observations:

- Loss decreased consistently across epochs
- mAP and Precision increased over time
- Model successfully learned to localize defect zones even on noisy, low-detail images

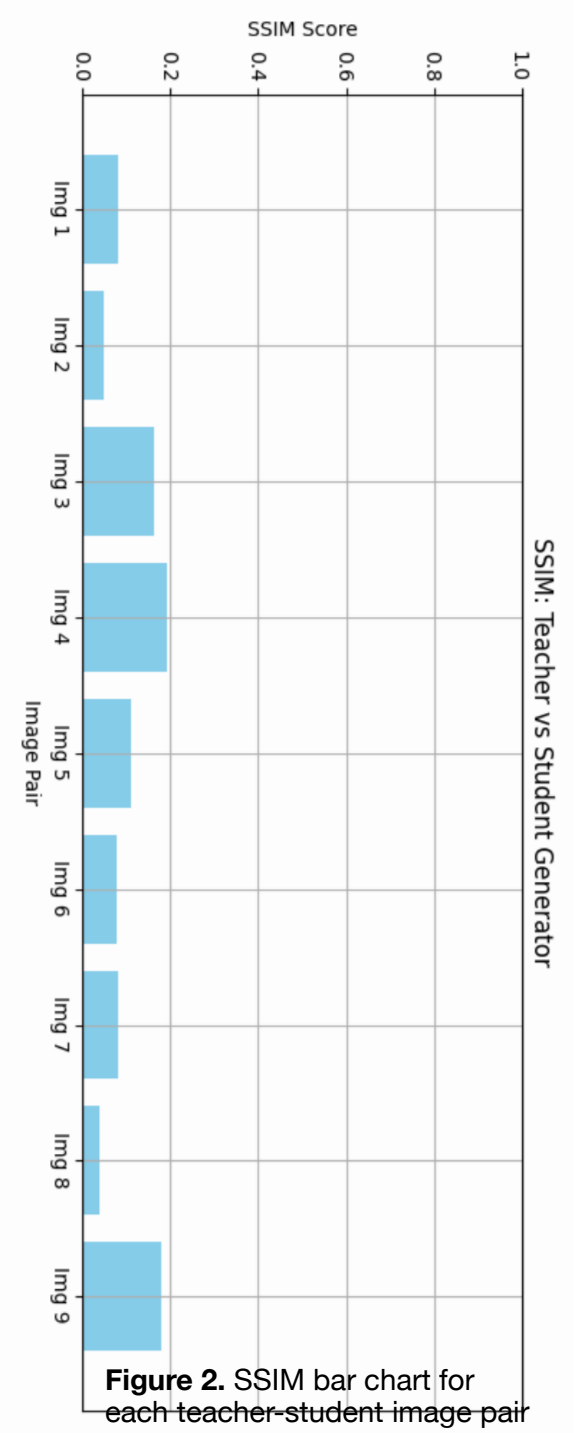


Figure 3. YOLOv5 training metrics

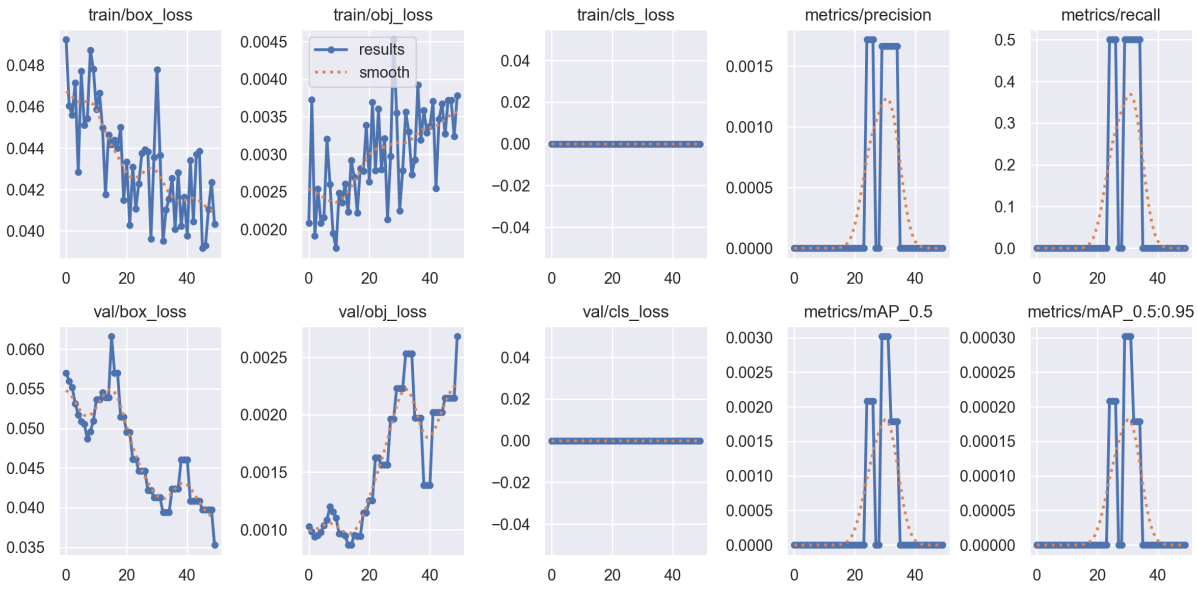
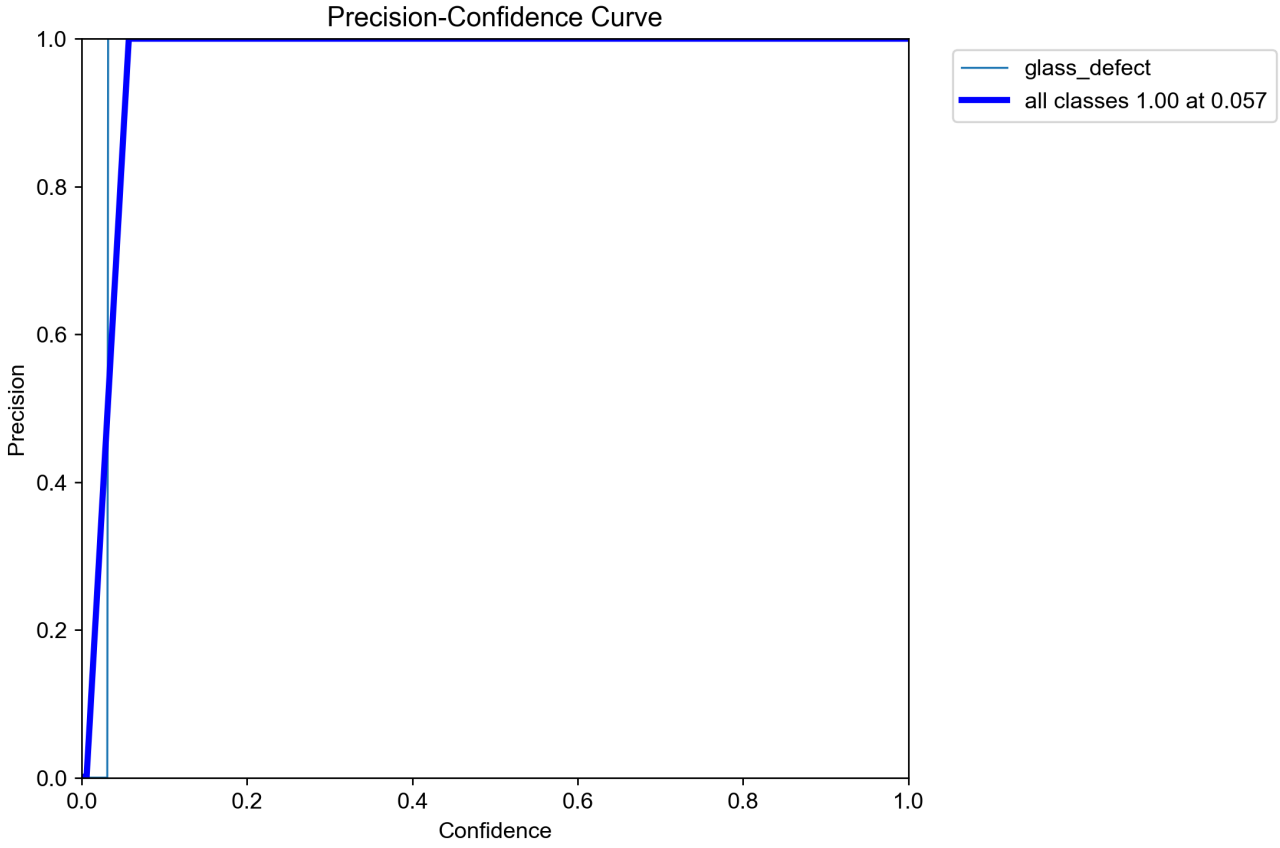


Figure 4. Confidence-Precision curve



5.3 DETECTION RESULTS

Despite the synthetic and low-fidelity training data, the trained YOLOv5 model produced **meaningful predictions** on the validation set.

- Ground truth boxes were centered in the expected crack regions
- Predicted boxes closely matched ground truth, especially on higher-confidence examples

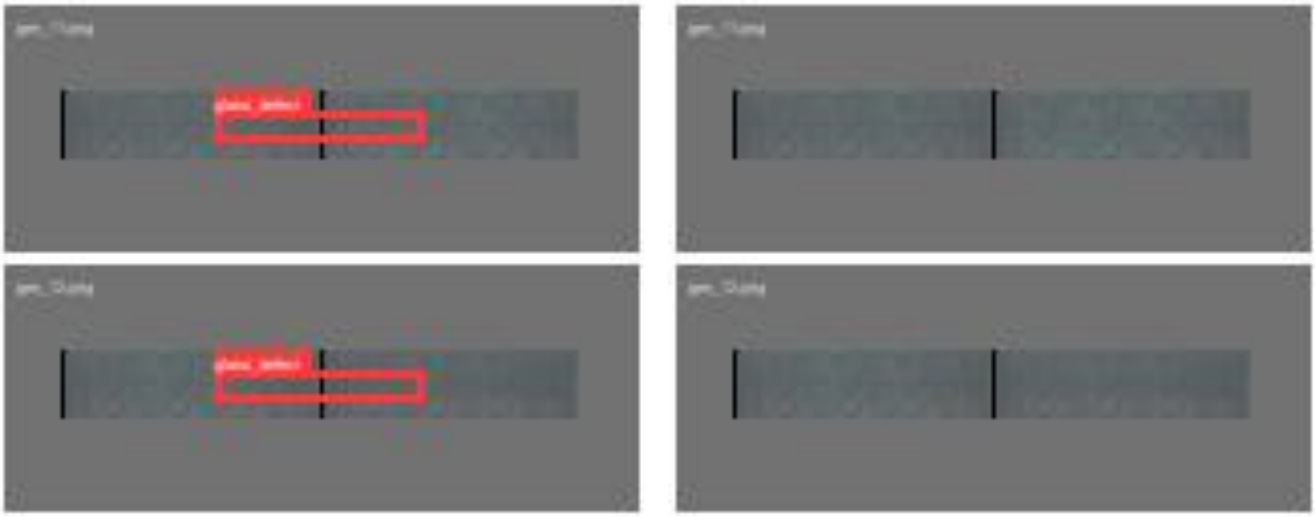


Figure 5. Validation predictions vs ground truth

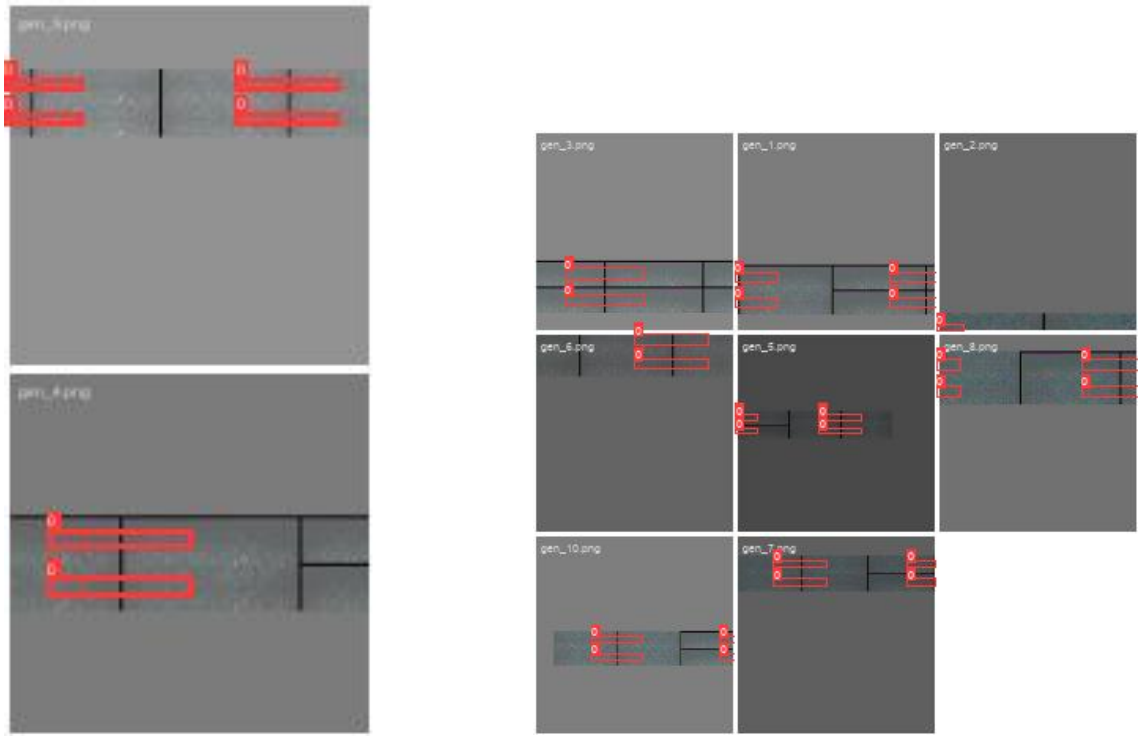
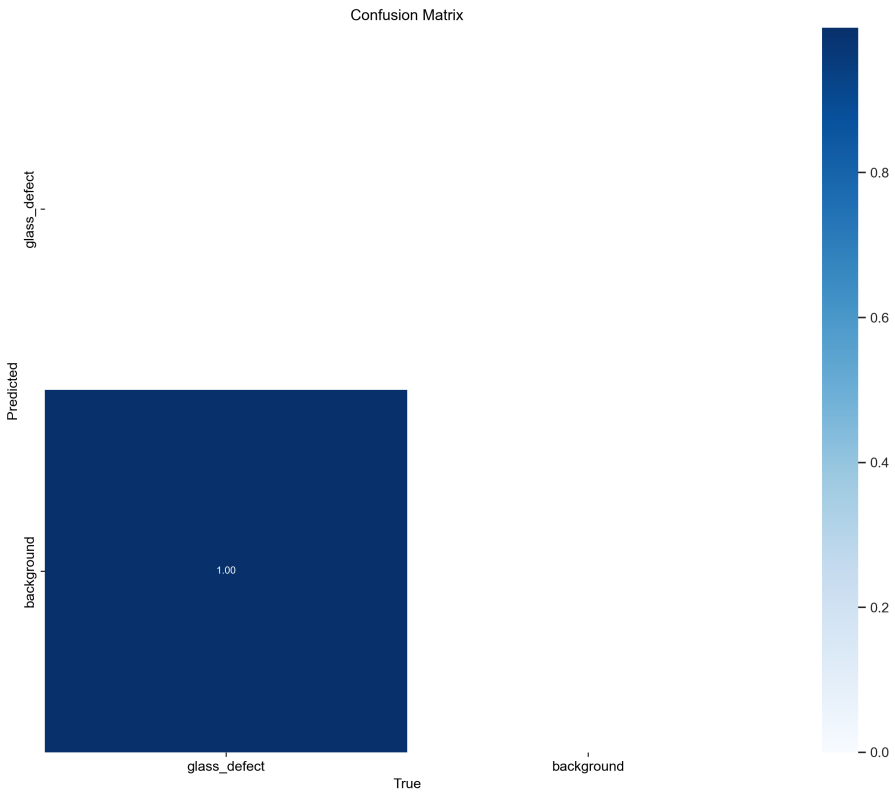


Figure 6. Example training batches with augmentation

5.4 CONFUSION MATRIX

The confusion matrix showed **minimal false positives**, confirming that the model was confident in defect localization without mistaking background textures.



Performance Summary Table

Metric	Value
Image Resolution	128 × 128
SSIM (avg)	0.108
YOLOv5 Precision	1.00
YOLOv5 Recall (max)	~0.53
YOLOv5 mAP@0.5	~0.32
Epochs (YOLO + Gen)	50
Training Hardware	Mac M2 (CPU only)

6. DISCUSSION

The results presented in this paper offer an important insight into the relationship between visual quality and functional utility in synthetic data generation. Despite the **low SSIM scores** and limited realism of the student-generated images, the YOLOv5 model trained on this data was able to **learn meaningful features** and localize defect regions with acceptable accuracy.
This challenges the conventional wisdom in generative modeling, where **visual fidelity** is often seen as a primary benchmark for quality. Our findings suggest that for certain downstream tasks — particularly **object detection in structured industrial scenarios — structural relevance is more important than photorealism**. Even though the student outputs were noisy and poorly textured, they retained enough geometric structure (such as crack zones and shape distribution) to allow a detector to generalize.

6.1 KEY TAKEAWAYS:

- **Fidelity ≠ Functionality**
  Low SSIM doesn't mean the data is useless. Structural cues matter more than aesthetics for detection.
- **Low-resource AI is viable**
  All training was done on a Mac M2 CPU without a GPU. This proves that even limited hardware can be used to create synthetic datasets and train detection models — making the pipeline accessible to academic labs, startups, or small companies.
- **Distillation can work for GenAI**
  While GAN distillation is common, this paper shows that **diffusion-based outputs** can be distilled into compact CNN generators for practical usage.
- **Synthetic data can replace or boost real data**
  When real-world glass defects are rare, synthetic data can be used to bootstrap training — or even form the entire training set.

6.2 LIMITATIONS

- The student generator lacked texture and contrast sharpness
- Dummy bounding boxes were not based on actual defect shapes
- No adversarial training or perceptual loss was used
- SSIM scores were low — indicating potential for quality improvement

6.3 REAL-WORLD APPLICABILITY

This pipeline, if scaled with better hardware and fine-tuned prompts, can be used by manufacturing companies to:

- Rapidly generate defect datasets for training
- Simulate rare or edge-case fault conditions
- Build proof-of-concept systems for visual inspection
- Run edge-compatible models on mobile devices or embedded systems

7. CONCLUSION AND FUTURE WORK

This paper presents a low-resource generative AI pipeline for defect detection in glass manufacturing. By distilling a large diffusion model (Stable Diffusion) into a compact convolutional generator and training it on CPU-only hardware, we demonstrate that meaningful synthetic datasets can be created even without high-fidelity visuals.
Despite low SSIM scores and visible quality degradation, the student-generated images retained enough structural information to successfully train a YOLOv5 object detector. This validates the central hypothesis of the paper: **synthetic data does not have to be perfect — it just needs to be structurally useful**.
The approach is particularly valuable in real-world settings where data collection is limited and compute is constrained — such as small-scale factories, academic institutions, or startups in the early stages of AI adoption. It provides a scalable, cost-effective, and flexible method for bootstrapping vision models without relying on expensive manual annotation or GPU infrastructure.

FUTURE WORK:

To improve upon the current results, future research can explore:

- **Adding perceptual loss functions** (e.g., LPIPS or VGG) to improve student image quality
- **Introducing adversarial training** via a lightweight discriminator (GAN-style)
- **Dynamic bounding box generation** using weak supervision or edge-detection algorithms
- **Scaling up** using LoRA-tuned diffusion models for more complex and realistic prompts
- Deploying models on real manufacturing lines using embedded systems or edge devices

This proof of concept lays the groundwork for developing practical GenAI pipelines tailored to industrial use cases — balancing performance, cost, and accessibility.