Metal Defect Synthesis PoC

Product Requirements Document (PRD)

항목	내용
프로젝트명	Metal Defect Synthesis PoC
버전	2.0 (LlamaGen + Halton-MaskGIT 전환)
작성일	2025-12-12
작성자	박유미
상태	PoC 완료 (v2.0 업그레이드)

1. 프로젝트 개요

1.1 배경 및 목적

제조업 품질 관리에서 결함 이미지 데이터의 부족은 AI 기반 자동 검사 시스템 구축의 핵심 장애 요소입니다. 특히 희귀 결함의 경우 실제 데이터 수집이 현실적으로 어렵고, 기존 데이터 증강 기법(회전, 뒤집기 등)은 결함의 다양성을 충분히 반영하지 못합니다.

본 프로젝트는 **LlamaGen VQGAN + Halton-MaskGIT** 기반 생성 모델을 활용하여 금속 표면 결함 이미지를 합성하는 PoC(Proof of Concept)를 수행하고, 기술적 가능성과 한계를 검증합니다.

1.2 v1.0 → v2.0 주요 변경사항

구분	v1.0 (taming)	v2.0 (LlamaGen)
VQGAN	taming-transformers	LlamaGen VQ-16
Codebook 차원	256	8 (더 효율적)
MaskGIT	직접 구현 (from scratch)	DiT-style + AdaLayerNorm
Transformer 크기	~12M params	~69M params (Small)
클래스 조건	단순 임베딩 더하기	AdaLN (Adaptive LayerNorm)
Sampler	Confidence 기반	Halton Sequence (ICLR 2025)

1.3 기대 성과

- LlamaGen VQGAN + Halton-MaskGIT 파이프라인의 금속 결함 도메인 적용 가능성 검증
- 인페인팅 기반 결함 삽입 기능 구현 및 Gradio 데모 제작
- 기술적 한계점 도출 및 개선 방향 제시

2. 시스템 구조

2.1 파일 구조

파일명	설명
metal_defect_synthesis_llamagen_PoCFinal_.ipynb	LlamaGen VQGAN Fine-tuning
metal_defect_HaltonMaskGIT_PoCFinal_.ipynb	Halton-MaskGIT 학습
metal_defect_gradio_demo_LlamaGen_Halton_PoCFinal_.ipynb	Gradio 데모 (통합)
data/merged_dataset_200x200/	통합 데이터셋 (6개 결함 클래스)

2.2 데이터셋 구성

데이터셋	이미지 수	비고
NEU-DET	1,440장	6개 결함 클래스
SD-saliency-900	900장	클래스 매핑 적용
X-SDD	319장	클래스 매핑 적용
합계 (merged_dataset)	2,659장	8배 증강 → 21,272 토큰

2.3 결함 클래스

클래스명	이미지 수
crazing (균열)	240장
inclusion (개재물)	540장
patches (패치)	662장
pitted_surface (피팅)	240장
rolled-in_scale (압연 스케일)	303장
scratches (스크래치)	674장

3. 모델 아키텍처

3.1 LlamaGen VQGAN

LlamaGen VQGAN은 이미지를 이산 토큰으로 변환하는 Vector Quantized Autoencoder입니다.

구성요소	스펙	비고
모델	VQ-16	16x 다운샘플링
Codebook 크기	16,384	토큰 어휘 크기
Codebook 차원	8	taming 대비 32배 압축
입력 해상도	256×256	RGB 이미지
Latent 크기	16×16 = 256	토큰 시퀀스 길이

3.2 Halton-MaskGIT Transformer

DiT(Diffusion Transformer) 스타일의 양방향 Transformer로, AdaLayerNorm을 통해 클래스 조건을 주입합니다.

하이퍼파라미터	값	비고
Hidden Dimension	512	Small 설정
Layers	12	Transformer 블록 수
Attention Heads	8	Multi-head attention
MLP Ratio	4.0	SwiGLU FFN
Total Parameters	~69M	v1.0 대비 5.7배 증가
Dropout	0.1	정규화

주요 구성요소

- **AdaLayerNorm:** 클래스 조건을 scale/shift 파라미터로 변환하여 정규화 레이어에 주입
- **SwiGLU FFN:** Swish 활성화 함수와 GLU 게이팅을 결합한 피드포워드 네트워크
- **QK Normalization:** Query 와 Key 에 RMSNorm 적용으로 학습 안정성 향상
- **Weight Tying:** 토큰 임베딩과 출력 헤드 가중치 공유로 파라미터 효율성 향상

4. 학습 결과

4.1 LlamaGen VQGAN Fine-tuning 결과

지표	Before (ImageNet)	After (Fine-tuned)	변화
SSIM	0.7297	0.7350	+0.0053 (+0.73%)
Edge IoU	0.2027	0.2242	+0.0215 (+10.6%)

분석: SSIM은 소폭 개선, Edge IoU는 유의미하게 개선됨. taming 대비 Edge 보존 능력이 향상되어 결함의 경계선이 더 선명하게 재구성됨.

4.2 Halton-MaskGIT 학습 결과

지표	최종 값 (Epoch 100)	비고
Cross Entropy Loss	6.77	수렴 정체 상태
Token Accuracy	~6.6%	낮은 정확도
Training Time	~100분 (100 epochs)	A100 GPU 기준

4.3 학습 설정 비교

구분	LlamaGen VQGAN	Halton-MaskGIT
Epochs	50	100
Batch Size	8	32
Learning Rate	4.5e-6 (Decoder)	1e-4 (초기)
Optimizer	Adam	AdamW
Scheduler	-	Halton Scheduler
GPU	A100 80GB	A100 80GB

5. 한계점 및 문제점

5.1 MaskGIT 학습 수렴 문제

- **높은 Loss 값:** 최종 Loss 가 6.77 로, 이론적 최적값(~4.0) 대비 상당히 높음
- **낮은 Token Accuracy:** ~6.6% 정확도는 랜덤 추측(~0.006%)보다는 높지만, 품질 있는 생성에는 부족
- **사전학습 부재:** ImageNet 사전학습 없이 2,659 장(증강 후 21,272 장)으로 69M 모델 학습은 데이터 부족

5.2 생성 품질 문제

- **비정상 패턴 생성:** 결함 특성과 무관한 노이즈 패턴이나 아티팩트 생성
- **클래스 간 차이 미미:** 서로 다른 결함 클래스임에도 유사한 패턴 생성 (클래스 조건 학습 실패)
- **텍스처 일관성 부족:** Inpainting 영역과 기존 영역 간 경계가 부자연스러움

5.3 Loss 분석 및 원인 추정

Loss ~6.7은 Cross Entropy 기준 perplexity $\exp(6.7) \approx 812$에 해당하며, 이는 모델이 각 토큰 위치에서 약 812개의 선택지 중에서 불확실하게 예측한다는 의미입니다.

원인 분석

1. **데이터 부족:** 69M 파라미터 모델 대비 21,272 샘플은 약 1/3,000 수준으로 심각하게 부족
2. **클래스 불균형:** 클래스당 240~674 장의 불균형이 AdaLN 조건 학습을 방해
3. **학습 iteration 부족:** 100 epochs × 664 batches ≈ 66,400 iterations (논문은 1.5M iterations 권장)

6. 개선 방안

6.1 단기 개선안 (현재 아키텍처 유지)

1) **학습 시간 증가:** 500+ epochs 또는 200,000+ iterations 까지 학습 연장
2) **Learning Rate 조정:** Warmup 적용 및 더 작은 초기 LR (5e-5) 시도
3) **모델 크기 축소:** Tiny (23M) 설정으로 데이터 대비 모델 복잡도 감소
4) **클래스 밸런싱:** Weighted sampling 으로 소수 클래스 오버샘플링

6.2 중장기 개선안

1) **추가 데이터 확보:** MVTec AD, GC10-DET, DAGM 등 공개 데이터셋 통합 (목표: 10,000+ 원본)
2) **Two-stage 학습:** 1 단계 unconditional → 2 단계 conditional fine-tuning

Two-Stage 학습이란?

[1 단계: Unconditional]

입력: 마스킹된 토큰

출력: 원래 토큰 예측

목표: "이미지가 어떻게 생겼는지" 일반적인 패턴 학습

[2 단계: Conditional Fine-tuning]

입력: 마스킹된 토큰 + 클래스 라벨 (예: "scratch", "crack")

출력: 해당 클래스의 토큰 예측

목표: "이 클래스의 이미지는 어떻게 생겼는지" 학습

7. 주요 함수 문서

7.1 LlamaGen VQGAN (Notebook 1)

함수/클래스명	설명
VQ_models['VQ-16']	LlamaGen VQGAN 모델 (codebook_size=16384, codebook_embed_dim=8)
NLayerDiscriminator	PatchGAN Discriminator (n_layers=3)
LPIPS	Perceptual Loss 계산 (VGG-based)

7.2 Halton-MaskGIT (Notebook 2)

함수/클래스명	설명
MaskGITTransformer	DiT-style Transformer (vocab=16,385, seq=256, hidden=512, layers=12, heads=8)
TransformerBlock	AdaLN + Attention + SwiGLU FFN 블록
get_mask_schedule()	Arccos 기반 마스킹 비율 계산 (Halton-MaskGIT)
mask_tokens()	입력 토큰을 [MASK] (16384)로 치환
build_halton_mask()	2D Halton sequence로 256개 토큰 순서 생성
HaltonSampler.sample()	CFG 기반 iterative decoding (num_steps=32)

7.3 Gradio Demo (Notebook 3)

함수명	설명
inpaint_image()	Halton sampler 기반 인페인팅 (num_steps, temperature, cfg_weight 지원)
encode_to_tokens()	LlamaGen VQGAN으로 이미지 → 256 토큰 변환
decode_from_tokens()	256 토큰 → 256×256 이미지 복원
visualize_mask_on_image()	마스크 영역을 이미지 위에 시각화 (빨간 오버레이)

8. 용어 정의

용어	정의
LlamaGen VQGAN	8차원 codebook을 사용하는 효율적인 Vector Quantized Autoencoder
Halton-MaskGIT	Halton sequence 기반 스케줄링을 적용한 Masked Image Transformer (ICLR 2025)
AdaLayerNorm	조건 정보를 scale/shift 파라미터로 변환하여 정규화에 적용하는 기법
SwiGLU	Swish 활성화 + GLU 게이팅을 결합한 피드포워드 네트워크
Halton Sequence	저불일치(low-discrepancy) 준난수 시퀀스로, 공간적 균일성 보장
CFG	Classifier-Free Guidance: 조건부/무조건부 예측을 결합한 생성 기법
Weight Tying	임베딩 레이어와 출력 레이어의 가중치를 공유하는 파라미터 효율화 기법

9. 버전 이력

버전	일자	작성자	변경 내용
1.0	2025-12-11	박유미	초안 작성 (taming VQGAN + 직접 구현 MaskGIT)
1.1	2025-12-11	박유미	피드백 반영 (증강 8배, MaskGIT 설정 수정)
2.0	2025-12-12	박유미	LlamaGen VQGAN + Halton-MaskGIT 전환, 한계점 분석 추가