

생성형 AI를 이용한

VMS 표출문자 자동생성 시스템 개발계획

2025. 6. 18(수)

□ 개 요

- 생성형 인공지능(AI) 기술을 활용하여 기존의 VMS 운영 글자체에 다양성을 부여하여 예쁘게 표출하고 DOT 불량 회피 알고리즘을 적용하여 특정 불량 부분을 최소화한 글자를 생성하여 표출하는 시스템 개발

□ 현 실 태

- DoT 불량시 글자 깨짐 현상 발생
 - 개별, 한줄, 여러줄(1글자 무조건 Shift)

□ 개선방안

- 생성형 AI를 이용하여 깨진 문자 인식 및 인근 LED로 재생성
 - Dot 이상 상태 에러유무 파악 : BMP file,
 - BMP File의 에러 부분 제외한 폰트 및 색상으로 AI 문자 생성
 - 한글 글자수 : 2,350 완성형코드 KS X1001(11,172 유니코드)

□ 시행방안

- 운영대상 : 본부별 개발 예상(서버 1식, S/W 1식)
- 운영방법 : 소형 VMS(2*2, 1*4) 2개소 시범 설치 및 운영
- 소요예산 : 40백만원(여성기업 수의)

□ 추진사항

- 특허등록 완료(특허 제 10-2807998호)
- S/W : 개발중, H/W : 업체 협의중

□ 향후계획

- 시스템 개발(25.8월) ⇒ 시범운영 및 의견수렴(25.9월) ⇒ 확대방안 마련



1. 깨진 상태로 문안 표출(현재)

소통원활

2. 깨진 부분을 수정(크기, 두께, 색상)하여 생성된 문안 표출(개발)

소통원활

소 : ‘ㅅ’ 부분을 두껍게 생성

통 : 글자를 약간 작게(폰트 축소), 우측으로 Shift

원 : ‘ㅇ’ 부분을 약간 위로 Shift

활 : 글자를 약간 작게, ‘ㅏ’와 ‘ㄹ’을 좌측으로 Shift

3. 생성형 AI Model(GAN, CGAN, Pix2Pix, Transformer 등)로 개발하여 적정 문구를 생성하고 사용자가 적정문구를 선택하여 표출할 수 있는 시스템 개발

4. Dot 이상상태 파악하는 BMP 파일 설명 및 샘플

BMP 파일을 통해 Dot의 이상 상태를 시각적으로 확인할 수 있습니다. 정상적인 Dot(픽셀)은 검정색(Black)으로 나타나며, 이상이 있는 경우 해당 색상에 따라 다르게 표현됩니다. 아래는 각 Dot 상태에 대한 설명입니다:

- 정상 상태 : 해당 픽셀에 이상이 없을 경우 검정색(0,0,0)으로 표시
- Red Dot 불량 : 빨간색(R)이 고장난 경우 빨간색(255,0,0)으로 표시
- Green Dot 불량 : 녹색(G)이 고장난 경우 녹색(0,255,0)으로 표시
- Blue Dot 불량 : 파란색(B)이 고장난 경우 파란색(0,0,255)으로 표시
- 화이트 불량 : R, G, B가 모두 불량일 경우 하얀색(255,255,255)으로 표시

5. 2.2.2.7 도트 감시 및 제어 기능(9쪽), 라. LED Dot 감시기능(14쪽)의 기능과 관련하여 개선된 기능으로 개발하고자 함.

6. GAN 관련 Model 비슷한 것

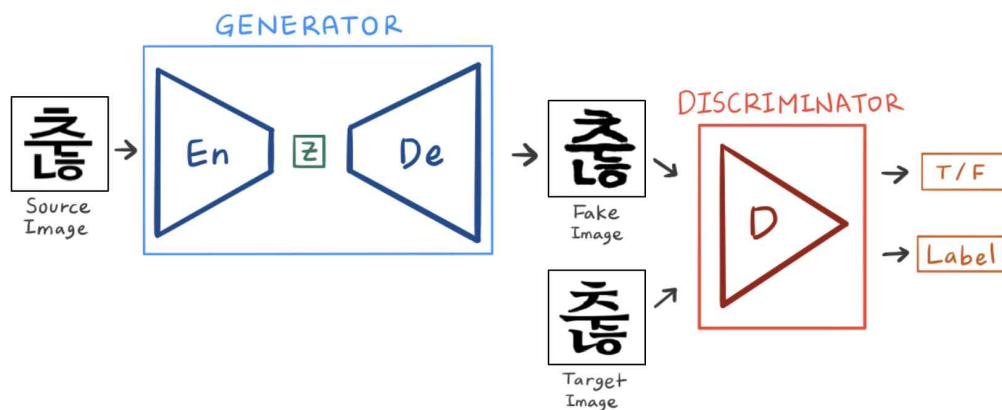
Wrinie의 입력 데이터는 고딕체의 글자이고, 출력해야 하는 데이터는 스타일이 변환된 새로운 글자입니다.

즉, 입력 데이터가 noise인 Original GAN 과는 비슷하면서도 명확하게 다릅니다.

이렇게 Generator 부분은 이미지를 낮은 차원의 벡터로 Mapping 시키는 Encoder와 다시 이미지로 복원하는 Decoder, 두 가지로 구성됩니다.

Decoder 구조만 있는 Original GAN 과는 Generator의 구조와는 다르죠.

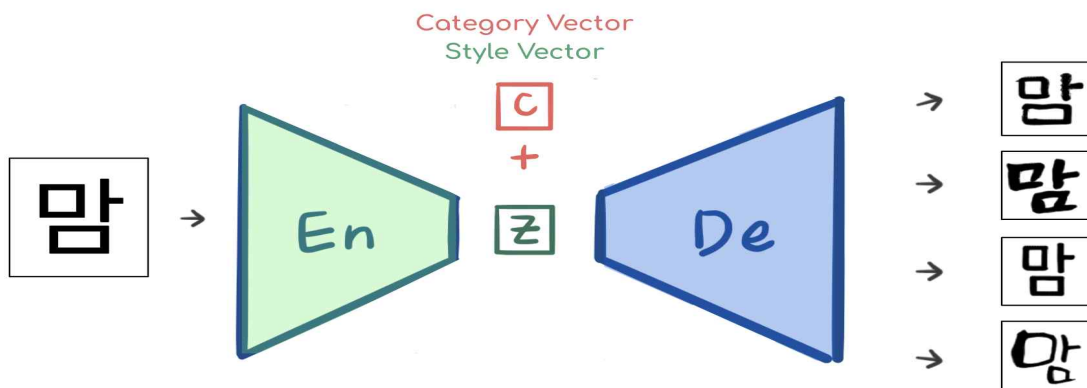
Wrinie



Wrinie의 Generator 부분에서는 고딕체를 새로운 글씨체로 바꾸는, 일명 "스타일 변환 (Style Transfer)" 이 이루어져야 합니다. 이를 위해서는 모델에게 변환하기를 원하는 폰트의 카테고리에 대해 입력해 줄 필요가 있습니다.

이 "폰트 카테고리"는 Category Vector, 또는 Style Vector 라는 이름으로 벡터 형태로 모델에 입력됩니다. 그림과 같이 Encoder 를 거쳐 이미지가 낮은 차원에 매핑된 z벡터에 붙여서 입력합니다.

Encoder로 이미지 특징 추출이 끝난 후 Decoder에 들어가기 전, z벡터와 함께 Decoder에 입력됩니다. 따라서 어떤 Style Vector를 입력해주느냐에 따라, 내가 원하는 폰트로 변환을 할 수 있게 됩니다.



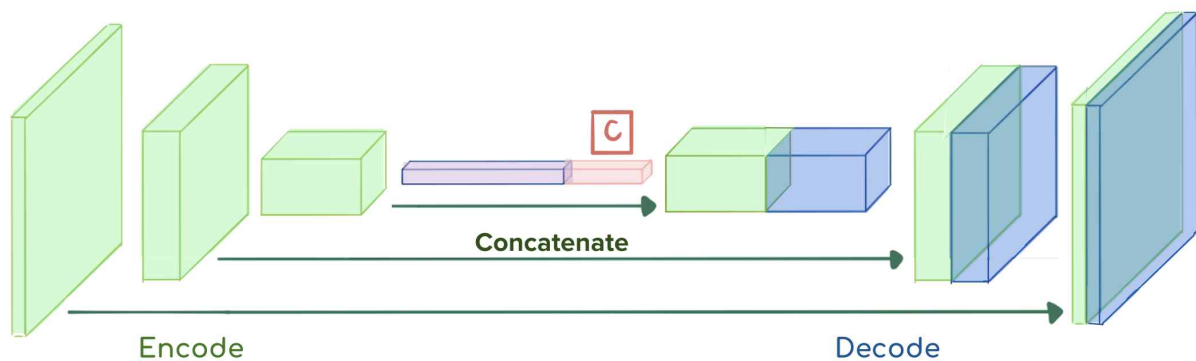
여기서 Style Vector를 Encoder 의 특징 추출이 끝난 후 입력하는 것은 다음과 같은 의미를 가집니다:

Encoder는 폰트와 관계없이 고딕체 글자를 글자 종류에 따라서만 특징을 효과적으로 추출하는 데에만 집중합니다. 예를 들어, '맘'이라는 글자를 입력받으면 폰트를 고려할 필요 없이 '왼쪽 위에는 ㅁ이 있고, 오른쪽 위에는 ㅂ가, 그리고 밑에는 ㅁ이 있다' 는 특징만을 잘 추출할 수 있도록 글자 특징 추출기로써의 역할에 충실하도록 학습합니다. 반면, Decoder는 Encoder가 잘 추출한 z벡터와 함께 style vector를 입력받으면 글자를 각 style에 따라 복원하도록 학습됩니다.

즉, 이렇게 Encoder와 Decoder의 역할을 분명히 구분함으로써 각자는 자기가 맡은 일을 더 잘할 수 있게 하는 효과를 가지게 됩니다.

Generator의 구조를 3D 형태로 한 번 확인

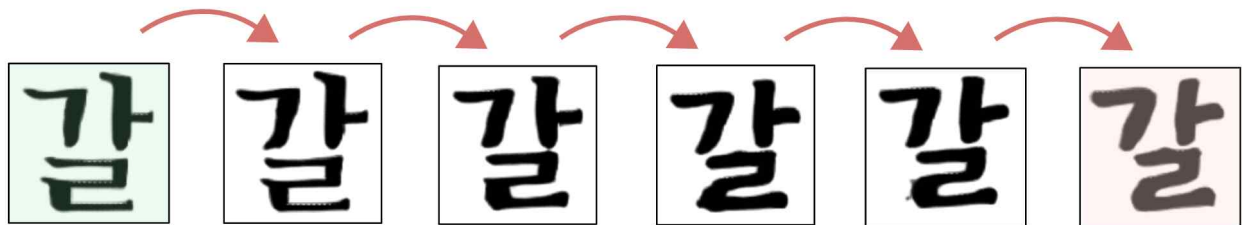
Encoder 와 Decoder, 그리고 Style Vector 까지 모두 확인하면 다음과 같습니다.



확인할 수 있는 구조는 Encoder에서 단계적으로 축소되어 가는 벡터들을 모두 Decoder에 단계적으로 입력하는, [UNet] 구조입니다.

UNet 구조는 이미지를 복원해야 하는 Decoder에게 Encoder에서 추출되는 정보를 추가로 입력해 줌으로써 약간의 도움을 주는 것으로 생각할 수 있습니다.

폰트와 폰트 간의 Interpolation이 이루어진 모습



같은 폰트이지만 얇은 글자와 두꺼운 글자, 두 가지를 모두 모델이 학습했다면 모델은 두 가지 두께 사이의 모든 두께의 글자를 생성해낼 수 있을 것입니다

2-1. 아이템의 개발/개선 준비현황

○ 아이템 기획 및 추진경과

(준비현황) S/W 개발을 위한 시스템 설계 및 AI모델(Deep Learning) 선정, 개발중
H/W 설계, 시제품 제작단계

(제조협력) 제조사와 개발 관련 미팅 5회 이상

(특허출원) 기술보호를 위한 특허출원중(출원번호 10-2024-0161044 도정암)

○ 깨진 상태로 문안 표출(현재)

소통원활

○ 깨진 부분을 수정(이동, 크기, 두께, 색상 등)하여 생성된 문안 표출(개발)

소통원활

소 : 'ㅏ' 부분을 두껍게 생성

통 : 글자를 약간 작게(폰트 축소), 우측으로 Shift

원 : 'ㅇ' 부분을 약간 위로 Shift

활 : 글자를 약간 작게, 'ㅏ' 와 'ㄹ'을 좌측으로 Shift

○ DOT 이상상태 파악하는 BMP 파일 및 샘플

BMP 파일을 통해 Dot의 이상 상태를 시각적으로 확인할 수 있다. 정상적인 Dot는 검정색(Black)으로 나타나며, 이상이 있는 경우 해당 색상에 따라 다르게 표현됩니다.

- 정상 상태 : 해당 픽셀에 이상이 없을 경우 검정색(0,0,0)으로 표시
- Red Dot 불량 : 빨간색(R)이 고장난 경우 빨간색(255,0,0)으로 표시
- Green Dot 불량 : 녹색(G)이 고장난 경우 녹색(0,255,0)으로 표시
- Blue Dot 불량 : 파란색(B)이 고장난 경우 파란색(0,0,255)으로 표시
- 화이트 불량 : R, G, B가 모두 불량일 경우 하얀색(255,255,255)으로 표시

==> 특정색이 불량일 경우 그 색을 제외한 조합으로 표출 가능













○ 문제 분석


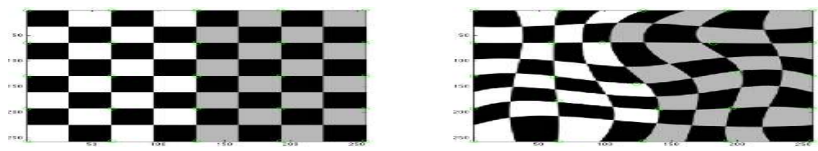
VMS 패널의 이상 상태에 따라 발생한 노이즈(noise)를 회피 또는 포함하는 새로운 이미지를 생성하면서, 기존의 글자 내용을 유지하는 것이 목표임.

구 분	설 명
Adaptive Text Reconstruction	노이즈 상태를 고려하면서 텍스트를 재구성하는 과제로서, 적응형은 VMS 패널의 변화 상태(노이즈)를 동적 반영한다는 의미
Noise-Aware Text Preservation	노이즈를 인지하면서 기존 텍스트를 유지하는 과제로서, 노이즈를 회피하거나 포함하면서도 텍스트 본질적 정보를 보존한다는 의미
Robust Text Rendering	노이즈가 발생해도 텍스트를 강건하게 보존한다는 표현으로서, "Robust"는 패널의 이상 상태에도 텍스트 내용을 유지함을 의미
Fault-Tolerant Text Generation	노이즈를 견디며 텍스트를 생성하는 과제로서, "Fault-Tolerant"는 VMS 패널의 고장 또는 이상 상태를 포함한다는 점을 반영
Pattern-Constrained Text Reconstruction	VMS 패널의 고정된 패턴(노이즈)을 고려하는 텍스트 재구성으로서, 고정된 패턴을 반영하며 텍스트 내용을 유지
Context-Preserving Text Adaptation	기존의 텍스트 내용을 유지하면서 노이즈를 포함한 새로운 환경에 적응하는 작업을 의미로서, 텍스트의 의미와 구조를 변경하지 않고 적응한다는 점을 강조
Noise Integration for Text Consistency	텍스트 일관성을 유지하면서 VMS의 노이즈 패턴을 통합하는 과제로서, 노이즈 통합과 텍스트 유지의 목표를 반영

○ 이미지 변형

이미지의 다양한 변형 방식의 부분적 적용을 통해 텍스트 이미지에서 내용 변화를 최소화 하며 노이즈를 회피함.

기 능	표 출	
Translation (이동)		
Scaling (확대축소)		
Shearing (기울임)		
Rotation (회전)		
Affine Transform(복합 변환)		
Perspective Transformation (원근 변환)		

Seam Carving (술기 제거)	
Warping (워핑, 왜곡)	

- Affine Transformation은 이동, 회전, 확대 축소, 기울임을 복합적으로 수행함.
- Perspective Transformation은 3차원 원근감을 주거나 제거하는 방식으로 사다리꼴의 직사각형화 또는 역변환을 포함함.
- Seam Carving (술기 제거)은 이미지의 중요도를 나타내는 에너지 값을 기준으로 중요하지 않은 부분을 순차적으로 잘라내어 이미지 크기를 축소함. 줄어든 이미지를 노이즈를 회피하는 방향으로 삽입하여 늘리는 방식의 적용을 검토함.
- Warping (워핑, 왜곡)은 Geometric translation의 한 종류로서, 이미지의 제어점 (control points)들을 기준으로 픽셀들을 새로운 위치로 매핑하여 이미지의 모양을 변형함. 노이즈 안쪽 공간에 제어점들을 설정하고 노이즈 주변으로 확장함으로써 노이즈를 회피함.

2-2. 아이템 실현 및 구체화/고도화 방안

- o 개발 목표 : VMS 패널 오류에도 불구하고 표출된 글자의 의미를 유지할 수 있도록 변경 이미지를 생성하는 AI 엔진을 개발
 - 가. VMS 상태 판단 기술 (표출중인 글자, 패널 노이즈 상태)
 - 나. 노출중인 글자의 가독성 확인 기술 (유사성 AI)
 - 다. 노이즈를 흡수할 수 있는 글자 이미지 생성 기술 (생성형 AI)
 - 라. 최선의 이미지 선택 기능 제공
- o DoIoT GPU 서버에서 JupyterLab을 운영중 : fargus.doiot.co.kr:18888/lab
 - Google colab 및 chatGPT를 활용하여 Python으로 AI 모델 개발 중
 - Gan, c-Gan, Pix2Pix, Tranformer 등과 모델 비교 분석하여 최적 모델로 개발
 - 다양한 환경시험(Hyper Parameter 등)을 통한 성능 비교
- o 차별화 전략 : 기존 제품과 신규 개발 제품의 차별성 강화로 경쟁력 확보

구분	기 준	추 가
글자체	8개 : 고딕체(나눔고딕), 한길체, 궁서체, 명조체(나눔명조), 돋움체, 굴림체, 바탕체, 전각문자	예쁜 꾸밈체 4개 생성
운영방식	-	Server Type : 여러 대의 VMS MCU 제어 PC Type : 독립된 한 대의 VMS MCU 제어

< 최종 산출물 작성(전체 사업단계) >

구분	최종 산출물	수량	완료 일정
단기(협약 기간 내)	생성형 AI 적용 SW	1	~'25.12.31.
	VMS 시제품(4글자)	1	~'25.12.31.
	서버	1	~'25.12.31.

o 개발 Flowchart : VMS 조정 알고리즘이 실제 서비스에서 운영될 기능으로 알고리즘에서 사용하는 AI 모델은 노이즈 강건 이미지 생성 알고리즘 학습으로 이루어지며 사용될 학습 데이터는 문자 데이터와 오류 데이터 생성 및 검증 방식으로 진행합니다.

가. VMS 패널 데이터 조정 알고리즘 : 개발된 시스템에서 텍스트 이미지 개선이 이루어지는 절차임.

- 1) VMS 패널 데이터 획득 : 현재 표출될 내용과 패널고장 상태를 획득
- 2) 원시 데이터 처리 : AI 처리를 위한 원시 데이터 준비 작업
- 3) 원시 데이터 검증 : 현재 표출되는 오류 이미지 생성 및 고장분석
- 4) 오류 완화 : 생성형 AI의 개선 이미지 생성 및 가독성 검토
- 5) 결과 최적화 : 개선 이미지 TOP 4 선정

나. 노이즈 강건 이미지 생성 알고리즘 학습 : 개선 이미지를 만드는 생성형 AI와, 생성된 이미지의 가독성을 확인하는 검증 AI 모델을 생성하는 절차임.

- 1) 학습데이터 준비 : 마련된 학습데이터를 로딩
- 2) 이미지 생성기 및 이미지 판별기 학습 : 학습데이터로 모델 학습
- 3) 생성된 모델 배포

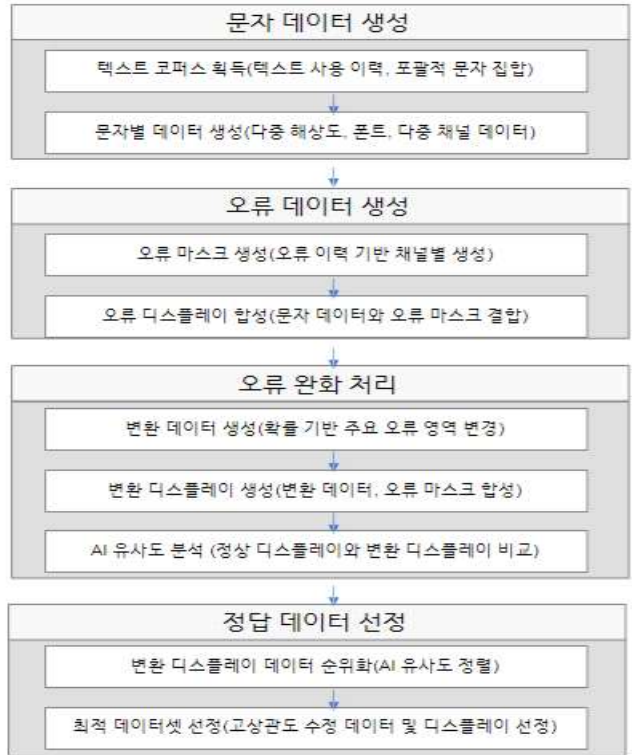
다. 학습 데이터 생성 : 고장 패턴, 표출될 이미지, 정답 데이터를 생성하는 절차임.

- 1) 문자 데이터 생성 : 텍스트 사용 이력 기반의 문자 데이터 생성
- 2) 오류 데이터 생성 : 패널 임의 오류 데이터 생성
- 3) 오류 완화 처리 : 표출 오류 상태 및 임의 변환 데이터 생성
- 4) 정답 데이터 선정 : 변환 디스플레이 데이터 순위화 및 최적화

VMS 패널 데이터 조정 알고리즘



학습 데이터 생성 알고리즘



노이즈 강건 이미지 생성 알고리즘 학습

