

Portfolio

현업 문제 해결을 위한 기술 도입과 자동화에 강점을 가진 개발자입니다

1. 인적사항

3. 2nd Project
(CRN System Web Crawling)

2. 1st Project
(DDS Infra 구축)

4. 3rd Project
(Super Resolution)

인적 사항

김태수

010-7535-8423

kts8421@naver.com

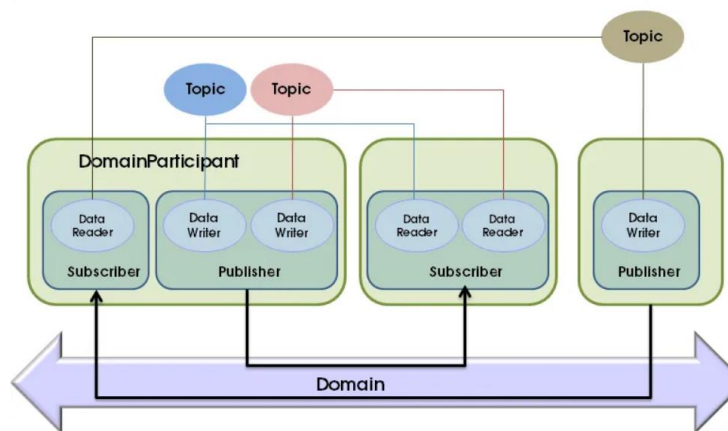
SKILLS



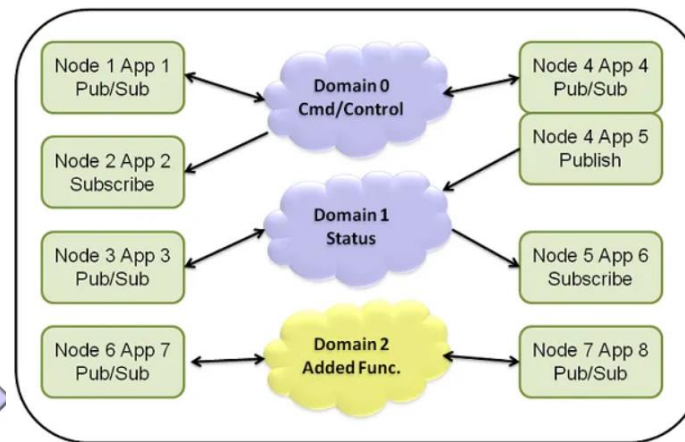
DDS Infra 구축

프로젝트 기본 정보

- 프로젝트명 : DDS(Data Distribution Service) Infra 구축
- 프로젝트 기간 : 2023.06 ~ 2023.09
- 프로젝트 인원 : 4명
- 프로젝트 목표 : 사내의 머신 제어SW APP들은 기계의 기능/모듈별로 App들을 갖고 있고 App들은 Server-Client관계로 이루어진 복잡한 구조를 갖고 있다. APP들 간의 통신을 보다 간소화하고 Socket API의 단점을 극복하기 위해 각각의 SW APP에 DDS 미들웨어를 도입해야 한다.
MA(Monolithic Architecture)에서 MSA(Microservices Architecture)로 시스템 구조를 전환한다.
- 담당 역할 : Python/C++ 기반 DDS Publisher/Subscriber 개발
메시지 손실, 지연, QoS 설정 시나리오별 테스트
DDS Core 이슈 분석 및 패치 제안



<DDS Architecture>



<DDS Domain Structure>

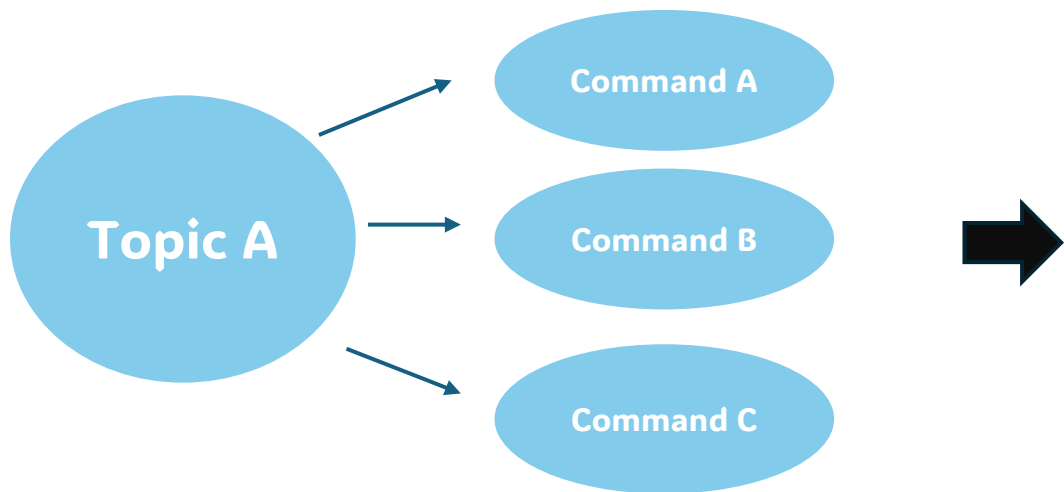
DDS Infra 구축

기술적 문제

- DDS 통신 과정에서 동일한 토픽으로 대용량 데이터를 지속적으로 발행하던 중, 일부 커맨드 메시지가 처리 지연되고 데이터가 유실되는 현상이 발생했습니다.

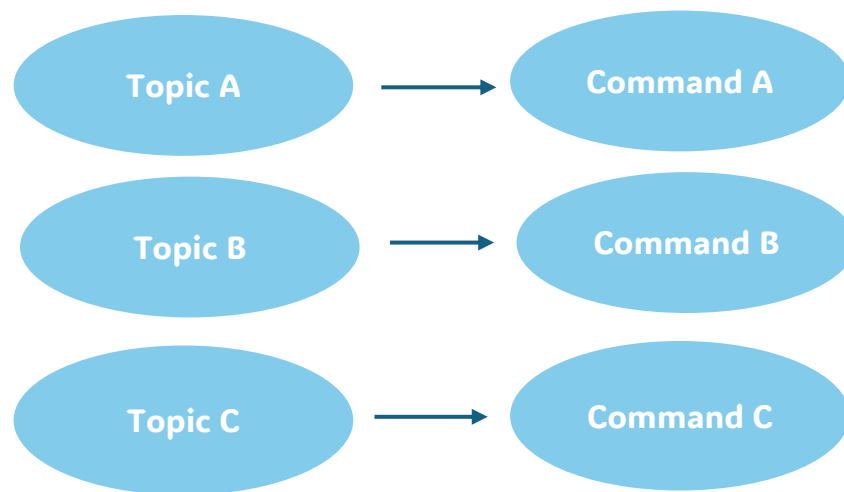
원인

- 이는 하나의 토픽에 모든 유형의 메시지를 집중시킴으로써 발생한 과부하와 큐 처리 지연 문제로 분석되었습니다.



문제 해결 방법

- 문제 해결을 위해 데이터를 별도의 DDS 토픽으로 분리하는 방향으로 설계를 변경하였습니다. 이로 인해 토픽 간 데이터 흐름이 독립적으로 분산되어 지연 시간 최소화과 메시지 유실 방지, QoS 적용의 유연성 확보 등의 효과를 얻을 수 있었습니다.
- 또한, 이 구조를 기반으로 토픽별 처리 구조를 추상화하기 위해 @abstractmethod를 활용한 DDS Handler 및 데이터 parser 인터페이스를 정의하고 토픽별로 상속받은 클래스를 구성해 유지보수와 확장성을 고려하였고 각 토픽 수신 모듈을 별도 스레드로 처리하여 병렬 수신 구조를 구현하였습니다.



DDS Infra 구축

결과 및 성과

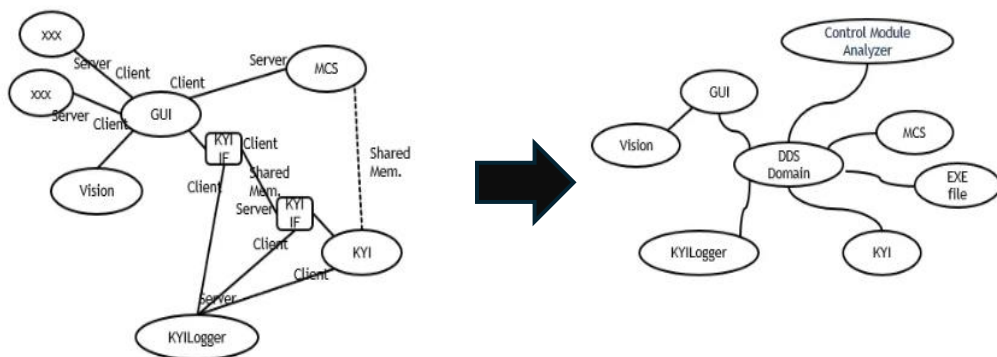
- 통신 모듈 교체 후 시스템 메시지 전달 지연 시간 1ms로 구현
- 시스템 구성 변경 없이 서비스 추가 및 통신 경로 확장 가능해짐
- DDS 미들웨어 안정성 개선에 기여 (Core 버그 발견 및 공유)
- 전체 시스템의 모듈화 및 유지보수 효율성 향상
- 멀티스레딩 기반 수신 모듈(DDS Data Collector) 개발

사용 Skills

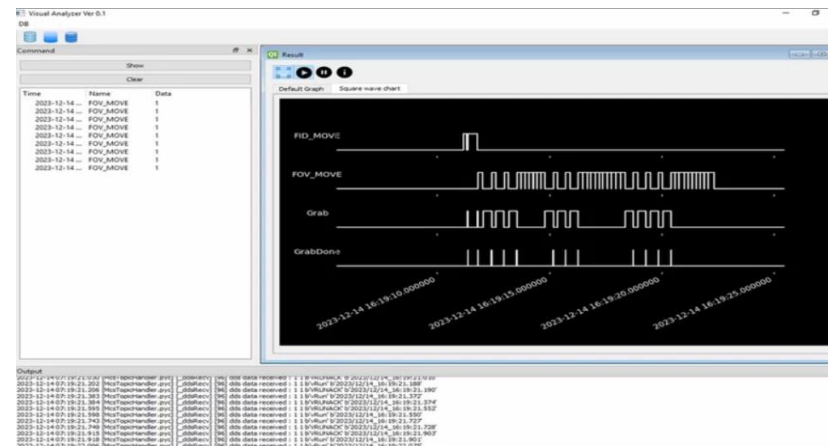
- Python
- C/C++
- DDS API(Data Distribution Service)
- 멀티스레딩/뮤텍스(Multithreading / Mutex)
- JSON

GitHub

<https://github.com/Kim-Tae-Su/Kim-Tae-Su-Data-Analysis-Application>



<MA to MSA : DDS 기반 분산 통신로 구조 설계 변경>



<DDS Data Collector UI>

CRM System Web Crawling 데이터 수집 자동화

프로젝트 기본 정보

프로젝트명 : CRM Web Crawling 데이터 수집 자동화

프로젝트 기간 : 2024.07 ~ 2024.08

프로젝트 인원 : 1명

프로젝트 목표 : CRM(고객관계관리) 시스템 내 장비 및 고객 이력 데이터를 자동으로 수집하여 딥러닝 학습 데이터로 사용하여야 한다.

CRM 시스템은 웹 기반 UI로 구성되어 있어 데이터가 서버에서 직접 접근 가능한 형식(DB, API 등)으로 제공되지 않는다.

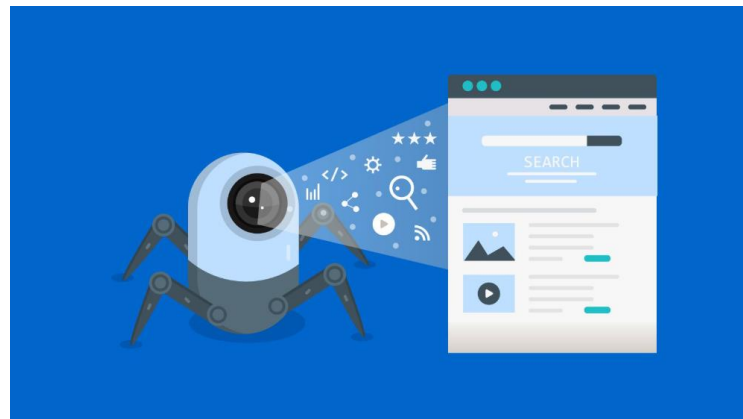
제한된 시스템 구조 속에서 자동화와 대량 수집을 동시에 달성하기 위한 방법으로 웹 크롤링을 선택한다.

담당 역할 : 웹 크롤링 자동화 로직 설계 및 구현

Selenium을 활용한 동적 페이지 대응 및 자동 스크롤 제어

데이터 크기 사전 확인 로직으로 리소스 낭비 최소화 및 성능 최적화

크롤링 안정성 확보 및 HTML 구조 파악, 예외 처리 구현



<Web Crawling>

CRM System Web Crawling 데이터 수집 자동화

기술적 문제

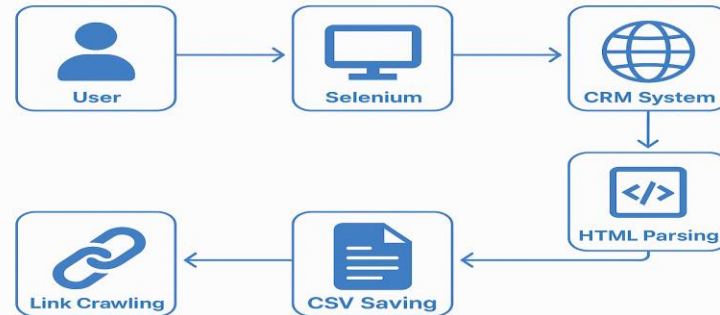
- CRM 시스템은 스크롤을 내릴 때마다 데이터가 동적으로 로드되고, 개별 데이터 링크를 따라갔다가 다시 뒤로 돌아오면 페이지 상태가 초기화되는 문제가 발생하였다..

원인

- 이미지가 일반적인 RGB 0~255 범위 사이의 값을 갖는데 데이터셋의 이미지 픽셀 값의 범위는 각기 달랐기 때문에 발생하였습니다.
(예: [-50, 50], [-200, 200] 등)

문제 해결 방법

- 자동 스크롤 기능을 구현하여 데이터를 로딩 하면서 모든 링크를 한 번에 수집해 CSV 파일로 저장하고 해당 파일을 기반으로 링크를 하나씩 접근하도록 구성했습니다. 동적 로딩 문제와 페이지 초기화 문제를 동시에 해결했습니다.



<Web Crawler Structure>

기술적 문제

- 데이터 파일 크기가 크고 일정하지 않아 서버 리소스 낭비 발생하였다.

원인

- 수집 대상 데이터의 크기가 10MB에서 10GB까지 다양했으며, 크기 확인 없이 일괄 다운로드 시 디스크 용량 및 네트워크 리소스 낭비 가능성이 있었습니다.

문제 해결 방법

- 요청 헤더를 통해 사전에 파일 용량 정보를 조회하고, 사전 정의된 임계값보다 큰 파일은 다운로드에서 제외하는 조건을 추가하여 서버 및 클라이언트 자원 소모를 효율적으로 제어하였습니다.

CRM System Web Crawling 데이터 수집 자동화

결과 및 성과

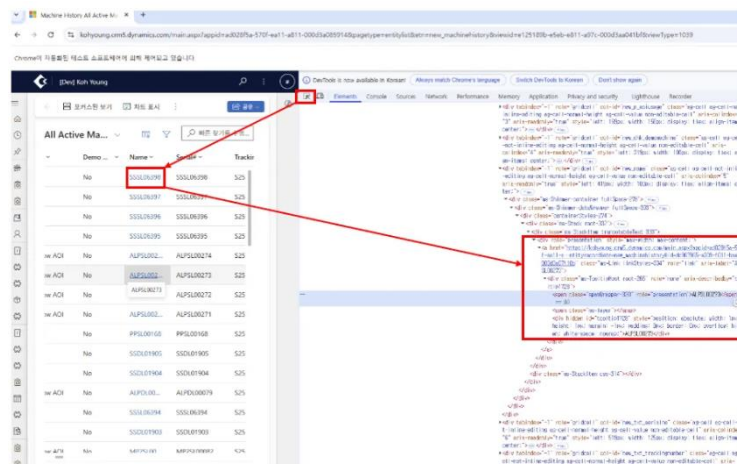
- 동적 로딩 페이지에 대응하는 자동화된 웹 크롤링 시스템 구축
- 수작업 대비 80% 이상의 시간 절약, 대량의 장비 이력 데이터(10MB~10GB급) 안정 수집
- 딥러닝 모델 개발에 필요한 CRM 데이터셋 확보를 통해 데이터 기반 업무 효율성 향상
- Selenium, HTML 구조 분석, HTTP 통신 등 웹 자동화 전반에 대한 실무 역량 강화

사용 Skills

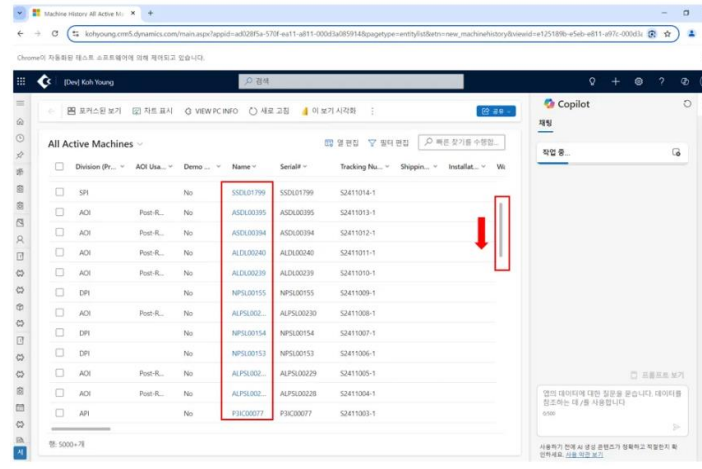
- Python
- Selenium, requests
- HTML / DOM 구조 분석
- HTTP 통신 구조 이해
- CSV 처리

GitHub

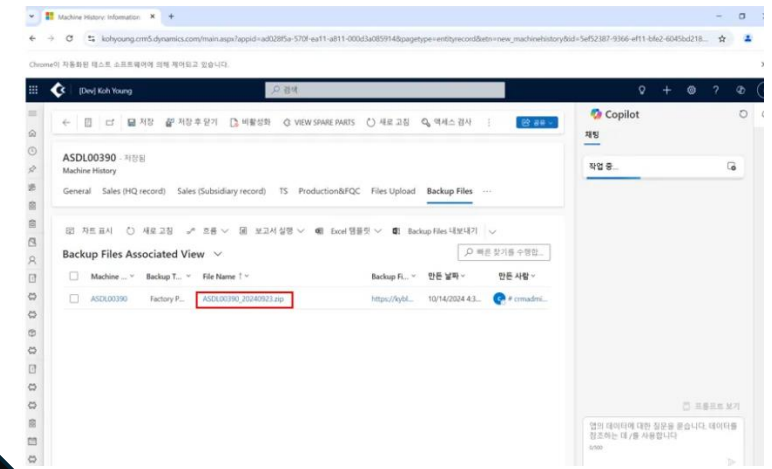
<https://github.com/Kim-Tae-Su/Web-Crawling>



<Find Element>



<Scroll Down>



<Download>

Super Resolution

프로젝트 기본 정보

- 프로젝트명 : Super Resolution
- 프로젝트 기간 : 2024.09 ~ 2025.01
- 프로젝트 인원 : 2명
- 프로젝트 목표 : 장비의 정확한 위치 이동을 위해 Machine Calibration 이미지 파일을 측정하여 활용한다.

하지만 Machine Calibration 측정 과정은 시간이 오래 걸리기 생산 효율성을 저하시키는 문제가 있다.

저해상도 Calibration 데이터를 고해상도로 복원할 수 있는 딥러닝 기반 Super-Resolution(SR) 모델을 개발하여 이러한 문제를 해결한다.

담당 역할

1. 데이터 로딩

- 시퀀스 기반 데이터 로더 설계
- 이미지 파일 NumPy 배열 형태로 변환

2. 데이터 전처리

- 저해상도 및 고해상도 이미지 생성
- 노이즈 제거, Scaling

3. 모델 설계 및 구현

- Super-Resolution 모델 후보 선정
- 모델 구현 및 논문 분석

4. 모델 학습 및 튜닝

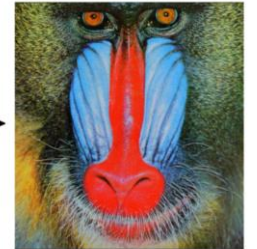
- 모델 학습을 위한 데이터셋 준비 및 분할 (훈련, 검증, 테스트 데이터셋)
- 하이퍼파라미터 튜닝, 최적화 기법 적용

5. 모델 검증 및 테스트

- PSNR, SSIM 등의 지표를 사용하여 모델 복원력 평가
- 고해상도 복원 이미지를 실제 장비 오차 이미지와 비교 분석하여 현실 적용 가능성 검증



Low-resolution image



High-resolution image

Super Resolution

기술적 문제

- 전체 데이터셋 기준으로 정규화를 하면 일부 이미지가 극단적으로 작거나 큰 값으로 왜곡되는 현상이 발생했습니다.
- 학습 시 모델이 특정 이미지에 bias를 가지게 되었고 모델의 성능이 저하되었습니다.

원인

- 이미지가 일반적인 RGB 0~255 범위 사이의 값을 갖는데 데이터셋의 이미지 픽셀 값의 범위는 각기 달랐기 때문에 발생하였습니다.
(예: [-50, 50], [-200, 200] 등)

문제 해결 방법

- 이미지 개별 기준의 Min-Max 스케일링을 적용하여, 각 이미지가 [0, 1] 범위 내에서 적절히 정규화되도록 처리했습니다.
- 모델이 예측한 고해상도 이미지도 각 이미지의 원래 Min/Max 값으로 복원되도록 구성하여 복원 후에도 물리적 해석이 가능하도록 했습니다.



$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

X(min) : 전체 이미지 데이터 값 중의 최소값
X(max) : 전체 이미지 데이터 값 중의 최대값

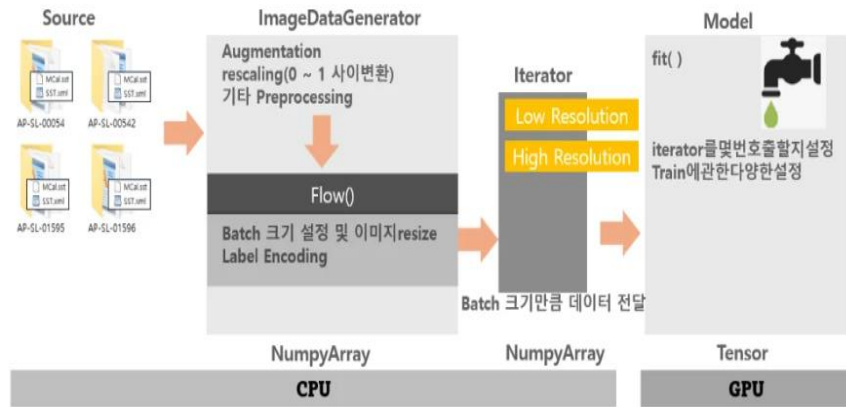


X(min) : 개별 이미지 데이터 값 중의 최소값
X(max) : 개별 이미지 데이터 값 중의 최대값

Super Resolution

결과 및 성과

- Mcal 오차 이미지 파일 생성 시간 약 90% 단축
- 고정밀 장비 보정 정확도 향상
- 총 5종의 SR 모델 설계 및 시뮬레이션 반복을 통해 최적 모델 선별
- SR 기술 기반의 실시간 보정 시스템 초기 기반 구축
- 기술적 제안 → 설계 → 구현 → 적용까지 End-to-End 프로세스 전담



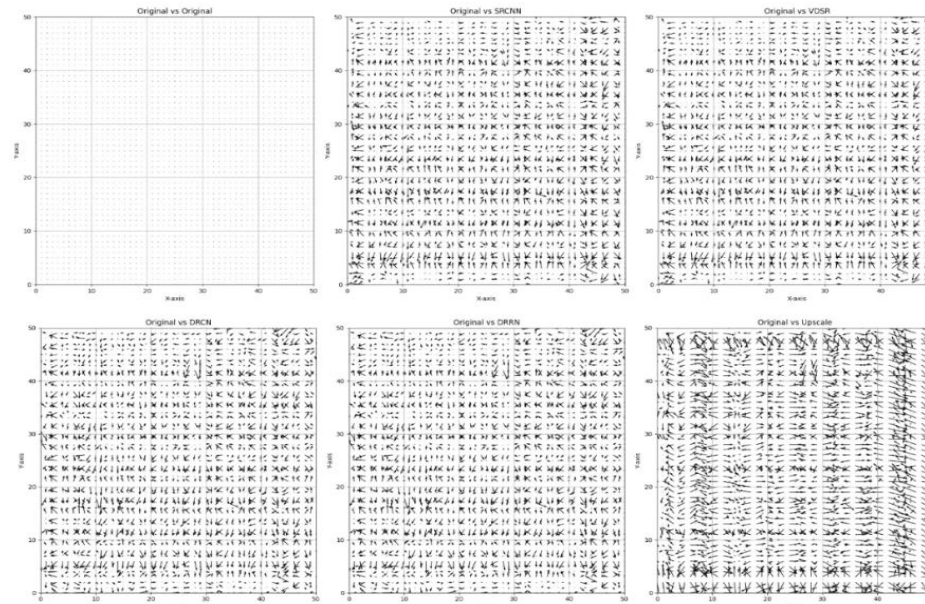
<Data Train Flow>

사용 Skills

- Python
- Keras / Tensorflow
- NumPy
- OpenCV

GitHub

<https://github.com/Kim-Tae-Su/super-resolution-project>



<SRCNN, VDSR, DRCN, DRRN, SRResNET Model Test Result(Error Vector)>

END

Thank

you