EDGE AI & BACKEND ENGINEER

최지웅.

분야	미들웨어 / 백엔드 / MLOps
현재	Visualization, Interaction, and Semantic Intelligence Lab 학부연구생
이전	INNODEP Backend Intern
학교	숭실대학교 컴퓨터학부, 2027.02 졸업 예정
연락	contact.jiwoong.choi@gmail.com

CASES

01	GPU 실험 파이프라인 자동화	VIS Lab →
02	단일 GPU 학습의 시간·메모리 효율 개선	VIS Lab →
03	JetPack 업그레이드 후 카메라 탐색 실패 해결	INNODEP →
04	화재 감지 오탐 개선	INNODEP →
05	자연어 인물 검색 재현율 개선	INNODEP →
06	장비 사양별 모델 탑재 전략	INNODEP →
07	에이전트 기반 시나리오 QA 도입	INNODEP →
08	AI 에이전트 기반 선행연구 검증	VIS Lab →
09	학습 서버 외부 노출 축소	INNODEP →
10	연구실 서버 보안 업데이트 자동화	VIS Lab →
11	음성 기반 LLM 대화 시뮬레이션 백엔드	SAEGIL →

Cases

01

GPU 실험 파이프라인 자동화

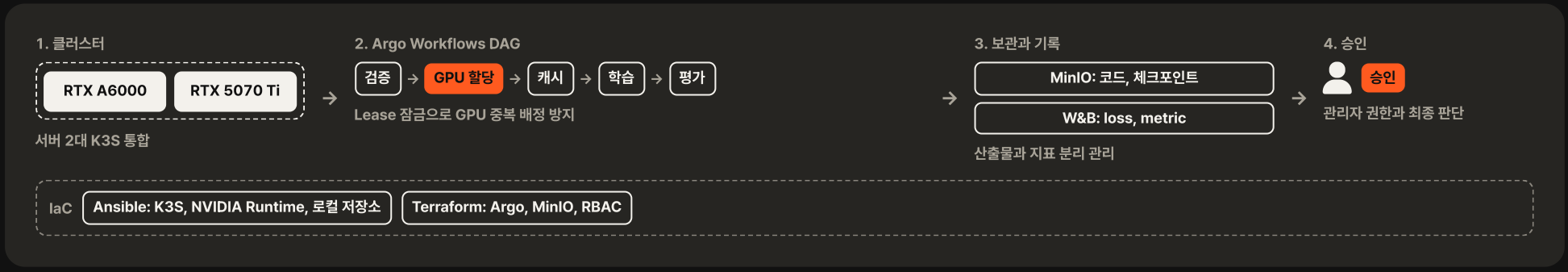
● VIS Lab
연구실 GPU 실험 인프라

문제 비전언어 모델 연구와 함께 GPU 서버 구성, 장애 점검, 결과 수집을 모두 담당했습니다. 서로 다른 두 컴퓨터의 RTX A6000과 RTX 5070 Ti를 개별로 사용해 실험이 순차 실행되었고 대기 시간이 길었습니다.

접근 두 서버를 K3S 클러스터로 구성하고 Argo Workflows로 검증, 학습, 평가를 하나의 파이프라인으로 연결했습니다. 동시에 시작된 실험이 같은 GPU에 중복 배정되는 문제는 상태 조회만으로 막기 어려워 Kubernetes Lease 잠금을 적용했습니다. 산출물은 MinIO, 지표는 W&B로 분리하고 인프라는 Ansible과 Terraform으로 코드화했습니다.

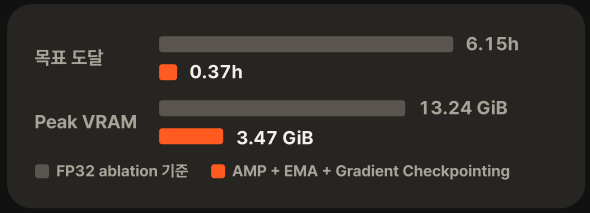
결과 순차 실행 대비 전체 실험 시간을 약 40% 단축했습니다. 동일한 환경을 재구성할 수 있게 되었습니다. 관리자 권한과 최종 판단은 승인 단계로 분리했습니다.

K3S · Argo Workflows · Kubernetes Lease · MinIO · W&B · Ansible · Terraform



02

VIS Lab
인물 검색 학습 효율 논문



단일 GPU 학습의 시간·메모리 효율 개선

문제 Edge 장비의 온디바이스 학습처럼 고성능 GPU를 쓰기 어려운 환경에서는 학습 시간과 GPU 메모리가 함께 제약됩니다. 이 문제를 단일 GPU의 시간과 메모리 제약으로 정의하고, 텍스트 기반 인물 검색 모델을 비용 효율적으로 학습하는 방법을 다뤘습니다.

접근 모델 구조는 그대로 두고 목표 정확도에 빠르게 도달하는 것을 과제로 정의했습니다. 학습률 조정과 경량화 기법은 성능 회복이 느려 EMA를 적용했고, 계산 시간은 AMP로, activation 메모리는 Gradient Checkpointing으로 줄였습니다. AMP, EMA, GC의 8가지 조합을 모두 평가했고, 시간과 메모리 중 어느 제약이 더 큰지에 따라 적합한 구성이 달라진다는 것을 확인했습니다.

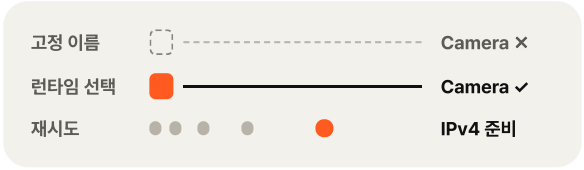
결과 CUHK-PEDES에서 CLIP ViT-B/16을 RTX 5070 Ti 한 장으로 학습하면서 AMP, EMA, Gradient Checkpointing을 함께 적용했습니다. 검증 R@1 68.19% 목표에 도달하는 시간을 FP32 ablation 기준 6.15h에서 0.37h로 줄였고, 학습 중 최대 GPU 메모리를 13.24 GiB에서 3.47 GiB로 73.8% 줄였습니다. 최고 검증 체크포인트는 0.85h에 R@1 71.06%였습니다. ICFG-PEDES와 RSTPReid에 대한 근거는 제한적입니다.

CLIP · AMP · EMA · Gradient Checkpointing

논문 설명 보기 ↗ · 논문 PDF ↗

03

INNODEP
Edge 장비 관제 시스템



JetPack 업그레이드 후 카메라 탐색 실패 해결

문제 JetPack 업그레이드 이후 일부 Edge 장비에서 ONVIF 카메라 자동 탐색이 실패했습니다. 버전에 따라 NIC 이름이 달라지지만 use-go/onvif는 특정 이름을 전제로 Probe를 전송하고 있었습니다.

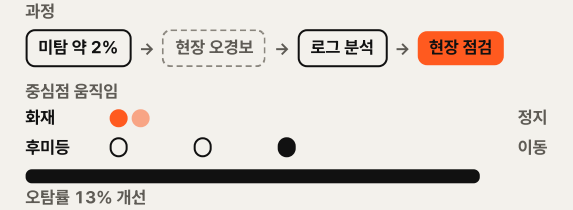
접근 이름 변경만으로는 다른 장비에서 문제가 재발할 수 있어 라이브러리를 수정했습니다. 런타임 네트워크 조건에 따라 NIC를 선택하도록 했습니다. DHCP 장비의 재부팅 직후 실패는 서비스가 네트워크보다 먼저 기동되어 발생했기에, 준비 상태 확인 후 지수 백오프로 재시도하도록 변경했습니다.

결과 JetPack 버전과 NIC 이름에 관계없이 동일한 Go 미들웨어 로직으로 카메라 탐색과 DB 등록이 완료되도록 개선했습니다.

Go · ONVIF · WS-Discovery

04

INNODEP
관제 시스템 KISA 인증



화재 감지 오탐 개선

문제 KISA 인증은 고객사 피드백 기반 보정을 넘어 공인 기준으로 신뢰성을 증명하는 과정이었습니다. DETR 파인튜닝으로 미탐률을 약 2%까지 낮췄지만, 초기 배포 모델은 고객사 지하주차장에서 차량 후미등을 화재로 인식하는 오경보를 반복했습니다.

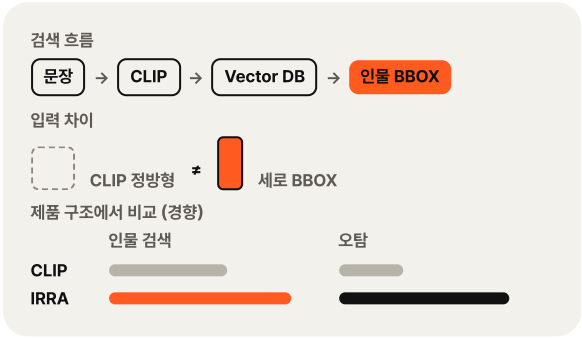
접근 임계값 재조정은 미탐과 오탐의 반복을 초래할 수 있다고 판단해 탐지 로그를 분석했습니다. 실제 화재의 BBOX 중심점은 고정되고 후미등은 지속적으로 이동한다는 차이를 근거로, 이동하는 붉은 객체의 화재 의심 점수를 낮추는 규칙을 적용했습니다.

결과 오탐률을 13% 개선했습니다. 설치 현장 점검에서 취합한 운영 담당자 의견을 추가 개선에 반영했고 실험 환경과 재현 방법을 문서화해 후속 담당자에게 인계했습니다.

DETR · KISA 인증 · 현장 점검

05

INNODEP
자연어 인물 검색 고도화



자연어 인물 검색 재현율 개선

문제 CLIP 기반 자연어 인물 검색의 정확도가 낮았습니다. CLIP은 정방형 이미지로 학습됐고 카메라가 검출하는 인물 영역은 세로형이었습니다. 이 형태 차이를 원인으로 가정했습니다.

접근 입력 형태와 크기, 모델 크기, 연산 정밀도별 성능을 측정하고 팀원 대상 블라인드 테스트로 체감 품질을 검증했습니다. 실제 CCTV 데이터의 개인정보와 법적 제약을 고려해 합성 데이터셋으로 국소 특징을 학습했습니다.

결과 재현율을 약 40% 높였습니다. IRRA(CVPR 2023)를 제품에 적용해 인물 검색 성능이 향상됐습니다. 제품은 DETR 검출 이미지에서 대상을 선택합니다. 이 구조에서는 CLIP보다 오탐이 크게 증가해 원인을 분석했습니다.

CLIP IRRA Vector DB DETR

06

INNODEP
Edge 장비 모델 탑재



장비 사양별 모델 탑재 전략

문제 판매 장비마다 GPU 사양과 지원 연산 정밀도가 달라, 단일 모델을 모든 제품에 적용하기 어려웠습니다.

접근 Jetson Nano, Jetson Orin, DGX Spark(Dell Pro Max)에서 모델 크기와 정밀도별 정확도 및 자원 사용량을 비교했습니다.

결과 보급형에는 연산 효율 중심 모델, 고사양에는 정확도 중심 모델, 고성능 장비에는 더 큰 모델을 탑재하는 전략을 제안했고 제품에 즉시 적용 가능한 전략으로 평가받았습니다.

CLIP Jetson DGX Spark

07

INNODEP
관제 시스템 PoC 품질



에이전트 기반 시나리오 QA 도입

문제 PoC와 기능 개발이 병행되며 약 5%의 버그가 발생했습니다. QA 요청서 기반 검증은 변경 속도를 따라가지 못했고 테스트 코드는 유지 비용 문제로 방치되어 있었습니다.

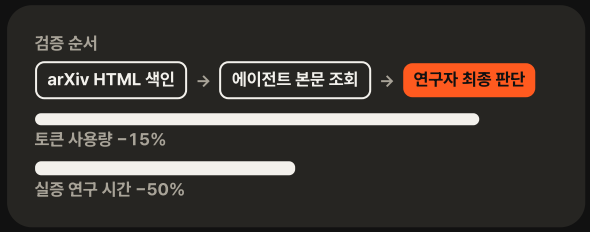
접근 잦은 로직 변경으로 단위 테스트는 적합하지 않다고 판단해 CI 파이프라인에 Claude Code GitHub Action 기반 에이전트를 연결하고 Playwright로 주요 기능의 실제 동작을 검증했습니다. 비용이 높은 에이전트 검증은 배포 시 보장해야 할 기능에 한정하고 기존 정적 검사와 병행했습니다.

결과 E2E 테스트의 복잡도를 낮춰 개발 공수를 줄이고 QA팀과 별도로 팀 내부에서 품질을 확인하는 체계를 마련했습니다.

Claude Code Action Playwright CI

08

VIS Lab
비전언어 모델 연구



AI 에이전트 기반 선행연구 검증

문제 비전언어 모델 연구의 선행연구 검토와 실험 분석에 AI 에이전트를 활용하면서 반복적인 문맥 설명과 불필요한 자료 탐색으로 토큰 소모가 컸습니다.

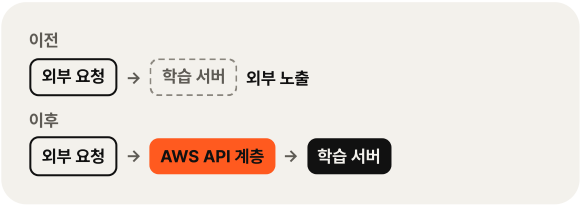
접근 HTML을 지원하는 arXiv 논문의 링크를 색인으로 정리해 에이전트가 본문을 조회하도록 했습니다. 최종 판단은 원문과 실행 결과를 대조해 내렸습니다.

결과 동일한 검증 범위에서 토큰 사용량을 15% 이상 절감하고 실증 연구 시간을 약 50% 단축했습니다.

AI Agent · arXiv

09

INNODEP
DETR 파인튜닝 서비스



학습 서버 외부 노출 축소

문제 온프레미스 서버에서 DETR 파인튜닝 서비스를 운영했고 외부 요청도 그대로 수신했습니다. 외부 연동이 늘수록 내부 학습 환경의 노출 위험도 커졌습니다.

접근 외부 요청이 AWS API 계층을 거쳐 내부 학습 서비스로 전달되도록 라우팅을 재구성하고 외부 연동 계층과 내부 학습 서비스의 역할을 분리했습니다.

결과 온프레미스 서버의 외부 노출 범위를 축소하고 외부 연동이 확대되어도 내부 학습 환경을 보호할 수 있는 구조를 마련했습니다.

AWS · DETR

10

VIS Lab
연구실 서버 운영



연구실 서버 보안 업데이트 자동화

문제 연구실 서버 10대를 SSH로 개별 접속해 관리하는 데 많은 시간과 비용이 들었습니다. apt 자동 업데이트로 NVIDIA 드라이버가 변경됐지만 서버는 재부팅되지 않았습니다. 디스크의 라이브러리와 메모리의 커널 모듈 버전이 불일치해 컨테이너 오류가 발생한 사례도 있었습니다.

접근 Ansible을 도입해 unattended-upgrades로 보안 업데이트만 자동 설치하고 GPU 스택(NVIDIA 드라이버, CUDA)과 버전이 고정된 Docker는 대상에서 제외했습니다. 재부팅은 실험 중단을 막기 위해 자동화하지 않았습니다.

결과 재부팅 대기와 서비스 재시작 대기 상태를 리포트로 확인하고 변경 사항은 사전 검토한 뒤 적용합니다. 결정 근거는 ADR로 기록했습니다.

Ansible · unattended-upgrades · ADR

11

● SAEGIL
캡스톤경진대회, TEAM New World
2025.03 — 2025.07, 4인 개발팀 AI 백엔드 담당

음성 기반 LLM 대화 시뮬레이션 백엔드



문제 북한이탈주민이 정착 과정에서 마주칠 상황을 음성 대화로 연습하는 기능이 필요했습니다. 팀의 주력 기술은 Spring Boot였지만, OpenAI API 연동은 Python 생태계가 가장 적합했습니다.

접근 AI 기능은 Python FastAPI 서버로 분리하고, AI 도구를 활용한 개발 방식을 도입해 직접 구현했습니다 (LLM 서버 커밋 25개 중 24개). Whisper로 음성을 텍스트로 바꾸고, GPT-4o-mini로 답을 만든 뒤, OpenAI와 ElevenLabs TTS로 다시 음성으로 들려주는 파이프라인을 만들었습니다. Assistants API의 thread로 대화 문맥을 유지했습니다. 실행이 60초를 넘으면 취소하고, 실패하거나 만료되면 예외로 처리했습니다. Spring 서버에서는 대화 기록을 시나리오별로 저장하고, 시나리오 프롬프트는 팀원들과 함께 설계했습니다.

결과 이후 Spring AI로 통합해 별도 서버 없이 운영되는 구조로 단순화했습니다. 앱이 API 키 없이 Realtime API와 gpt-4o-audio 음성 대화를 쓰도록 임시 토큰 발급도 구현했습니다. 앱은 데모 목적으로 스토어에 출시했습니다.

Python · FastAPI · OpenAI Whisper · GPT-4o-mini · Assistants API · TTS · Realtime API · ElevenLabs · Spring Boot · Spring AI

Papers

● VIS Lab · Journal Paper

Improving Time-to-Target Performance in Text-Based Person Search

Jiwoong Choi (제1저자), Hyunjoo Song (교신저자)
School of Computer Science and Engineering, Soongsil University

단일 GPU의 시간과 메모리 제약 아래에서 텍스트 기반 인물 검색 모델을 학습하는 방법을 다룹니다. AMP, EMA, Gradient Checkpointing(GC)의 8가지 조합을 모두 평가하고, 목표 정확도에 도달하는 시간(time-to-target)과 학습 중 최대 GPU 메모리를 함께 측정했습니다.

0.37h
검증 R@1 68.19% 목표 도달 시간, FP32 ablation 기준 6.15h

3.47 GiB
학습 중 최대 GPU 메모리, 13.24 GiB 대비 73.8% 감소

71.06%
최고 검증 R@1, 0.85h 시점

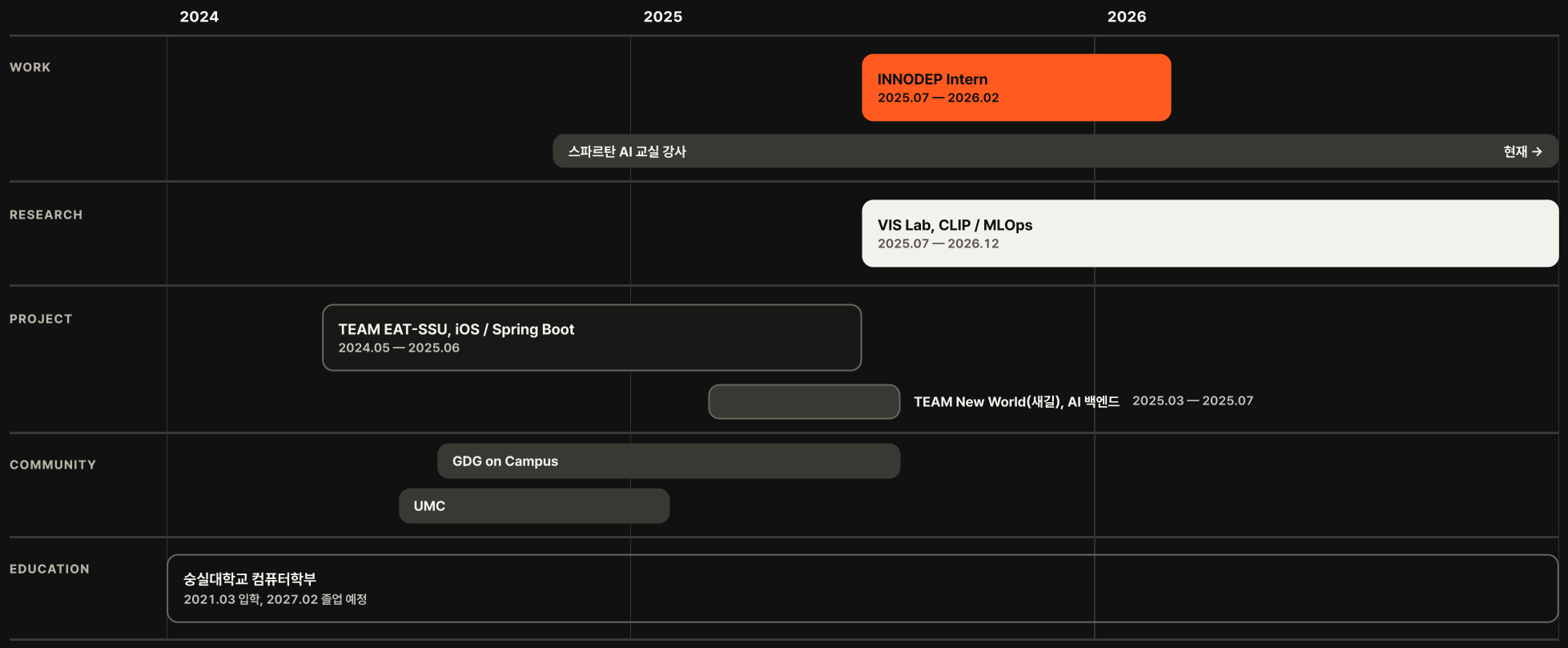
CUHK-PEDES에서 단일 GPU로 측정한 결과이며, ICFG-PEDES와 RSTPReid에 대한 근거는 제한적입니다.

Text-based person search · Time-to-target · Memory-efficient training · Mixed precision training · Exponential moving average · Gradient checkpointing

논문 설명 보기 ↗ · 논문 PDF ↗ · Code ↗

Timeline

2024.01 — 2026.12



Places

INNODEP
2025.07 — 2026.02

VUNex 사업부 Backend / AI Backend Intern

VUNex 제품 소개 ↗

VIS Lab
2025.07 — 2026.12

Visualization, Interaction, and Semantic Intelligence Laboratory 학부연구생, 비전언어 모델 연구

연구실 홈페이지 ↗

TEAM EAT-SSU
2024.05 — 2025.06

교내 식단 서비스, iOS / Spring Boot

Tuist 도입, 위젯 개발, 모니터링, CI/CD 고도화, DB 마이그레이션. 다국어 식단 기획, 활동 이후 후속 기수가 구현해 출시

서비스 소개 ↗ GitHub ↗

CES 2025
2025.04

우수학생 해외연수, Las Vegas

GDG on Campus
2024.08 — 2025.07

Soongsil 개발자 커뮤니티

웹사이트 ↗

UMC
2024.07 — 2025.01

University MakeUs Challenge

Instagram ↗

스파르탄 AI 교실
2024.11 — 현재

다과목 강사

Blog ↗

여름발명학교
2024.07 — 2024.08

공학교육봉사, 수업 기획 및 진행

웹사이트 ↗

숭실대학교
2021.03 — 2027.02

컴퓨터학부, 졸업 예정

SEOUL, KR

Contact

contact.jiwoong.choi@gmail.com ↗

github.com/jayn2u ↗

PDF 다운로드 ↓