

Yojun Moon (문요준)

yojun313@postech.ac.kr github.com/yojun313 [linkedin.com/in/yojun-moon](https://www.linkedin.com/in/yojun-moon) yojun313.github.io

Education

POSTECH (Pohang University of Science and Technology) — Mar 2023 – Present
B.S. in Computer Science and Engineering

Work Experience

Samsung Fire & Marine Insurance — Jul 2026 – Aug 2026

AI Engineer Intern, AI Development Part

Built two computer vision modules for insurance claim data, each delivered as an end-to-end pipeline with a web service.

- Automatic License Plate Masking for Dashcam Video:
 - Delivered a module that masks every vehicle license plate in dashcam footage, reaching 99.0% recall and 0.98 F1 on an 11.6K-image evaluation set.
 - Sustained 27+ fps end-to-end, shipped as a web service covering upload, processing, and download.
- Detection of Generative-AI Document Forgery:
 - Built FinForge, a 6,000-image forgery dataset and Korean insurance claim forms, and fine-tuned TruFor (CVPR 2023) on it.
 - Raised forgery localization F1 from 0.51 to 0.97 and document-level classification accuracy from 55% to 93%, holding 0.88 F1 on low-quality images.
- Skills: PyTorch, Ultralytics YOLO, OpenCV, FFmpeg, FastAPI

Korean National Police University — FPEI (Future Public Safety Engineering Institute) — Nov 2023 – Present

Research System Developer

Core developer of the institute's research platform: 8 services on a shared FastAPI backend, deployed in production at knpu.re.kr/systems (github.com/yojun313/knpu).

- Data Collection — CRAWLER, an asynchronous, queue-based crawler with proxy IP rotation, gathering about 1,500 articles and 60,000+ comments per hour from Naver News, and also supporting Naver Blog, Naver Cafe, and YouTube.
- Analysis — STATISTICS (statistical analysis and visualization), NETWORK (social network extraction and visualization from natural language), and KEMKIM (future signal prediction), drafted by MANAGER, the research resource and big data platform.
- AI Legal Complaint Drafting — built an LLM-based pipeline that unifies complaints meeting legal requirements and received a Letter of Appreciation from the Korean National Police University (Apr 2024).
- Skills: Python, FastAPI, MongoDB, vLLM, PyQt, PM2

POSTECH HAIV Lab — Jan 2026 – Feb 2026

Undergraduate Researcher

Researching ACCIDENTOR, a multi-VLM framework that estimates traffic accident fault ratios from dashcam video.

- Skills: Python, PyTorch, Vision-Language Models

Startup Team (founded by a POSTECH CSE alumnus) — Feb 2025 – Jul 2025

Backend Developer Intern

Developed the backend of a social networking service and supported data-driven product decisions.

- Key Contributions:
 - Developed the SNS backend with NestJS.
 - Built web crawlers and analyzed the collected data.
 - Completed the 2025 I-Corps program, including 3 weeks of entrepreneurship training at UC Berkeley (KIC Tech Frontier Program).
- Skills: NestJS, TypeScript, Python, MongoDB, Crawler

Projects

PCSS (POSTECH Computer Scientist Search) — Jan 2025 – Mar 2026

In-house service for the POSTECH CSE department. Faculty candidate search service, developed on commission.

- Key Implementations:
 - Parsed DBLP XML dumps and stored normalized publication records in MongoDB.
 - Identified Korean-name authors at scale with an LLM API.
 - Reduced the search for Korean authors of top-venue papers within a given year and venue range from hours to seconds (several hundred times faster).
- Deployed at pcss.postech.ac.kr
- Skills: Python, MongoDB, LLM APIs

Commeet — Aug 2026 – Present

In-house service for the POSTECH CSE department. Group scheduling service that collects everyone's availability and assigns the meeting times.

- Key Implementations:
 - Automatic time slot assignment — reduced the conflict-free schedule to a degree-constrained bipartite matching and solved it optimally with min-cost max-flow, maximizing the number of people scheduled while packing them into as few sessions as possible.
 - Contact management — a reusable address book for frequent invitees, covering both registered users and people who have not signed up.
 - Professor-only tools — consultation records in the four categories by POVIS, with per-student notes, a drawn signature, and a printable form, plus student guidance fee settlement with receipt upload. All records are encrypted at rest, as they hold student personal data.
- Deployed at commeet.postech.ac.kr
- Skills: FastAPI, MongoDB (Motor), Jinja2, JavaScript

POPO — Jan 2026 – Feb 2026

Campus web service operated by PoApper, POSTECH's student developer club.

- Developing the backend with NestJS.
- Skills: NestJS, TypeScript

Activities

POSTECH Broadcasting Station (PBS) — Mar 2023 – Dec 2025

Member of the Programming & Production Department.

- Filmed, edited, and live-broadcast campus entertainment programs.

Programming Tutor — Jan 2025 – Present

Teaching Python and C to middle and high school students.

- Topics include object-oriented programming, MongoDB, and FastAPI.

Skills

Programming Languages	Python, C, C++, TypeScript, JavaScript
AI / Machine Learning	PyTorch, Ultralytics YOLO, OpenCV, vLLM, LLM APIs
Backend	FastAPI, NestJS, MongoDB
Tools / Miscellaneous	Docker, Git, Linux, PyQt, FFmpeg

1. Samsung Fire & Marine Insurance — Computer Vision Modules

Jul 2026 – Aug 2026 · AI Engineering Internship, AI Development Part

[Company Website](https://www.samsungfire.com) · samsungfire.com

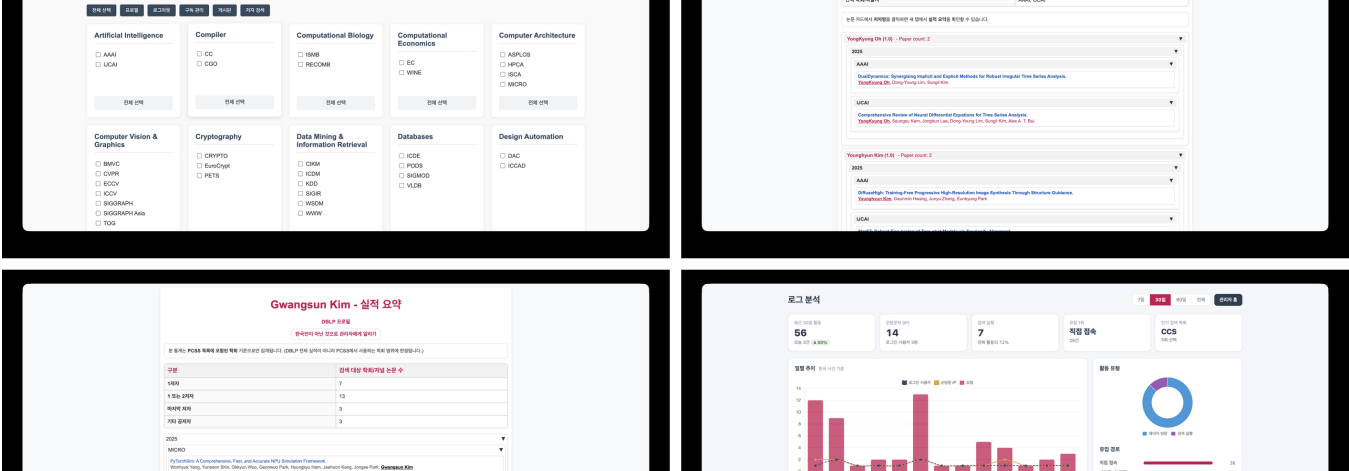
Automatic License Plate Masking

[Website](https://masking.yojun.dev) · masking.yojun.dev

I developed a module that automatically masks license plates in dashcam videos. The pipeline uses YOLO to detect vehicles, crops and enlarges the vehicle regions, and passes them to a separate license plate detector. I fine-tuned the plate detector with image degradation such as blur and downscaling, then improved processing speed through optical flow tracking, batch inference, and parallel processing of video segments.

The detector achieved 99.0% recall and approximately 0.98 F1 on the image test set. The video pipeline processed more than 27 frames per second in the development GPU environment. I also built a web interface for file uploads, processing settings, result previews, and downloads.

Technologies: Python, PyTorch, Ultralytics YOLO, OpenCV, FFmpeg, FastAPI



Generative-AI Document Forgery Detection

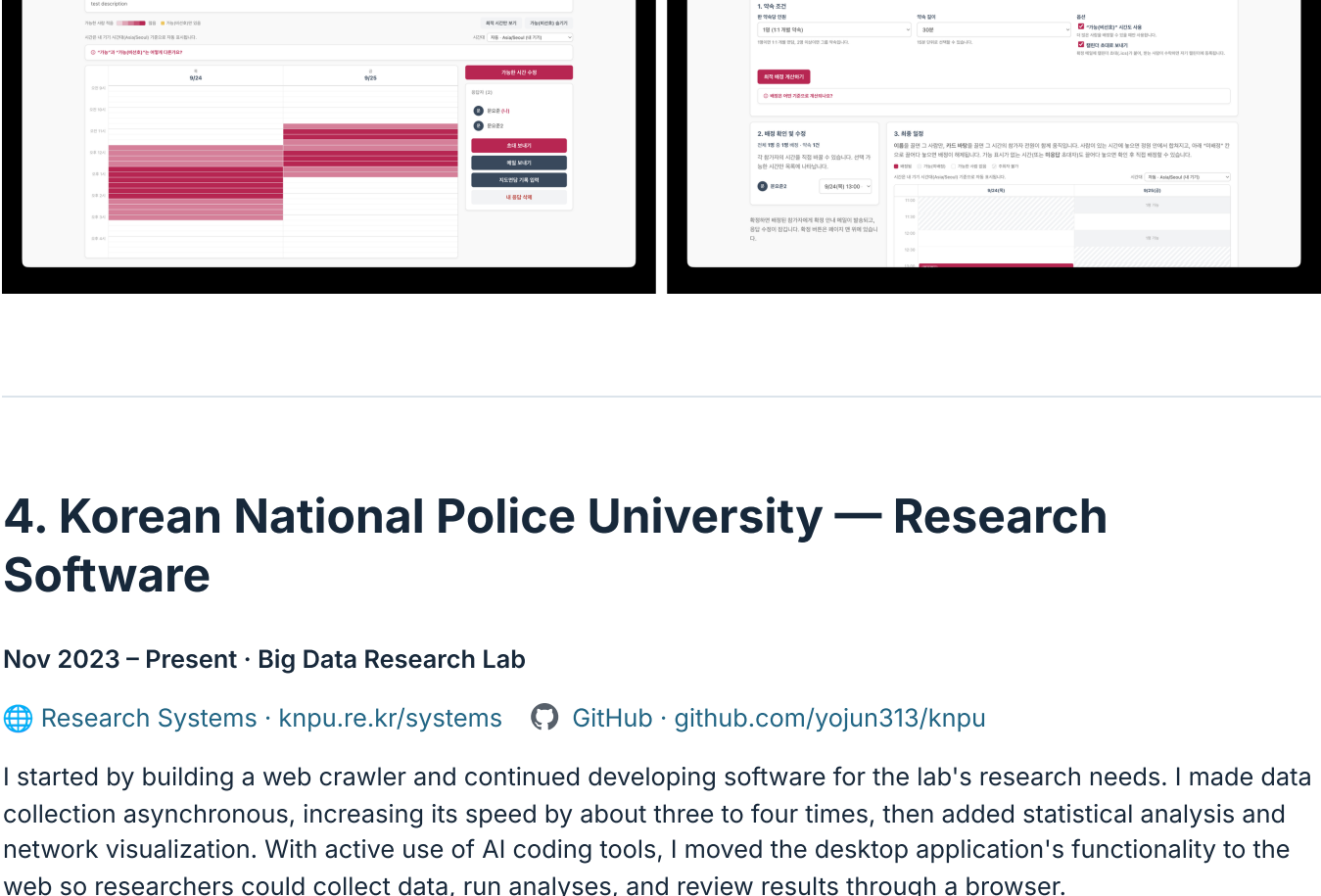
[Detection Service](https://detection.yojun.dev) · [forge.yojun.dev](https://detection.yojun.dev)

[Dataset Viewer](https://datasetviewer.docforgery.knpu.re.kr) · [docforgery.knpu.re.kr](https://datasetviewer.docforgery.knpu.re.kr)

I worked with another intern to develop a module that detects document regions edited with generative AI. We used eight image-generation models to build a dataset of 6,000 images, comprising 3,000 forged documents and their corresponding originals. We applied TruFor, which combines visual information with subtle noise patterns. I led the model experiments and testing, sequentially fine-tuning the forgery localization and document classification components.

Document-level accuracy improved from about 55% to 93% on the test set, while the best localization F1 on clean images reached approximately 0.97. A separate model trained with blur, noise, and recompression achieved approximately 0.88 localization F1 on degraded images. We built a web service that displays suspicious regions and forgery scores. I also added a dataset viewer for browsing original and forged document pairs, comparing edited regions, and reviewing sample metadata.

Technologies: Python, PyTorch, TruFor, Generative AI APIs, FastAPI



2. PCSS — Faculty Candidate Search

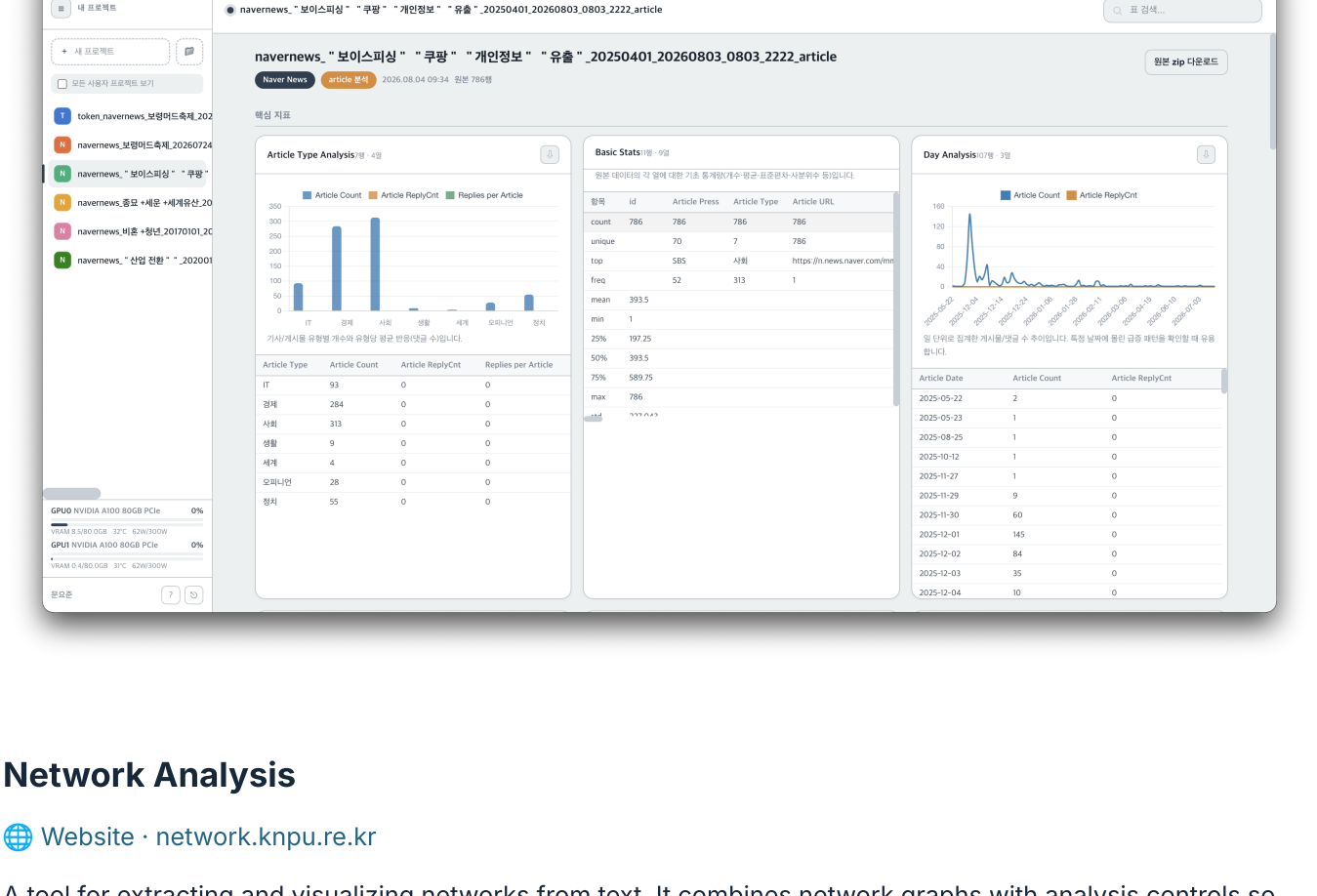
Jan 2025 – Mar 2026 · Commissioned by the POSTECH CSE Department

[Website](https://pcss.postech.ac.kr) · pcss.postech.ac.kr

I developed PCSS to reduce the time professors spent manually searching conference pages for Korean researchers. I parsed DBLP XML data, stored publication and author records in MongoDB, and used an LLM API to identify authors whose names were likely Korean.

Users can select conferences and a year range to view candidate authors and publication statistics. A task that previously required hours of manual browsing became a query that returns results from the prepared database in seconds. I also implemented monthly data updates and email notifications for new papers from selected conferences.

Technologies: Python, MongoDB, LLM APIs



3. Commeet — Meeting Scheduling and Record Management

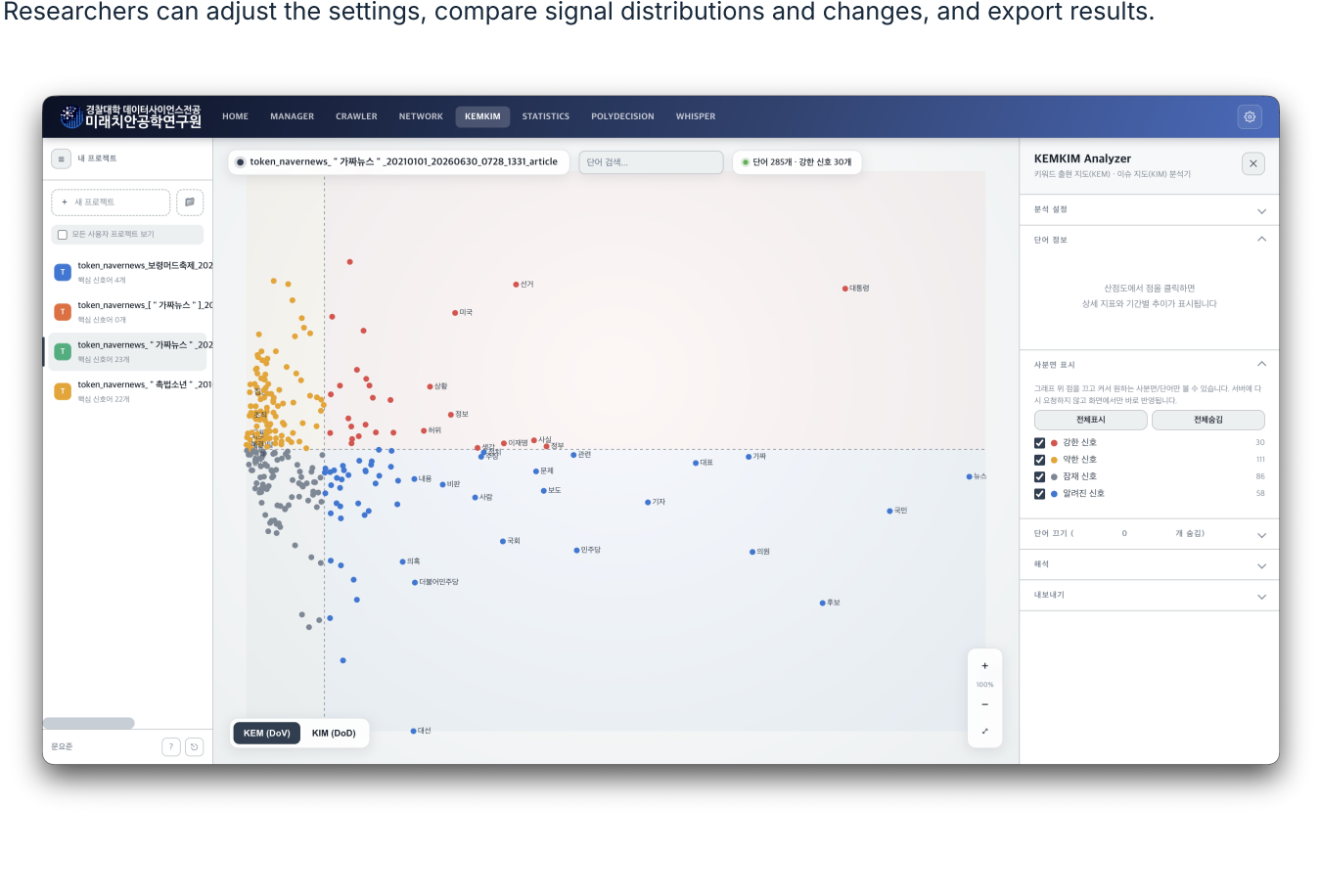
Aug 2026 – Present · Service for the POSTECH CSE Department

[Website](https://commeet.postech.ac.kr) · commeet.postech.ac.kr

Commeet calculates meeting schedules from participants' availability. I modeled scheduling as a constrained bipartite matching problem and used min-cost max-flow to schedule as many participants as possible while minimizing the number of sessions.

I also added an address book for frequent invitees and tools for professors to manage consultation records. These features support the workflow from arranging meetings to writing and organizing student records afterward.

Technologies: FastAPI, MongoDB, Motor, Jinja2, JavaScript



4. Korean National Police University — Research Software

Nov 2023 – Present · Big Data Research Lab

[System](https://knpu.re.kr/systems) · knpu.re.kr/systems

[GitHub](https://github.com/yojun313/knpu) · github.com/yojun313/knpu

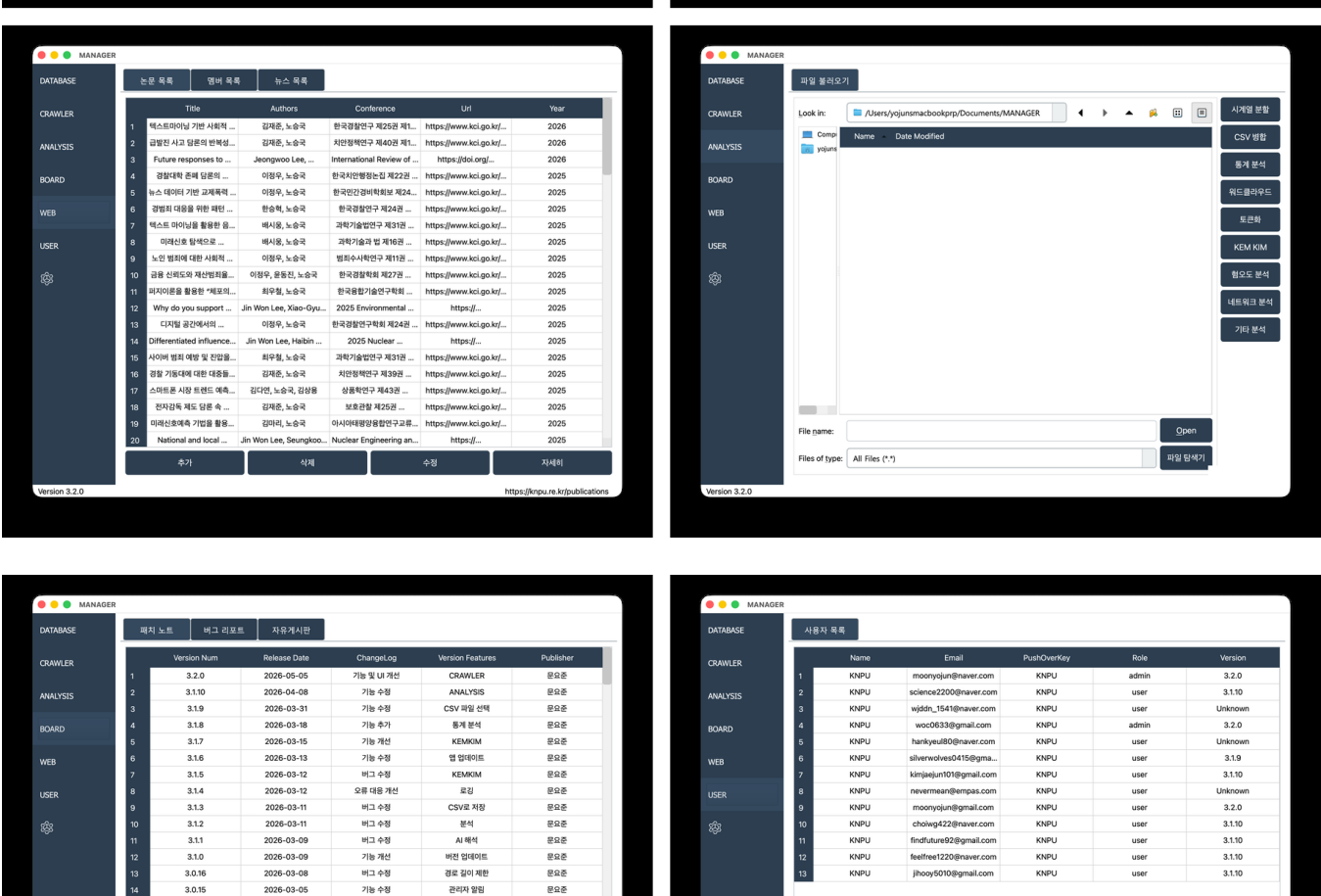
I started by building a web crawler and continued developing software for the lab's research needs. I made data collection asynchronous, increasing its speed by about three to four times, then added statistical analysis and network visualization. With active use of AI coding tools, I moved the desktop application's functionality to the web so researchers could collect data, run analyses, and review results through a browser.

Technologies: Python, FastAPI, MongoDB, PyQt, vLLM, PM2

Web Crawler

[Website](https://crawler.knpu.re.kr) · crawler.knpu.re.kr

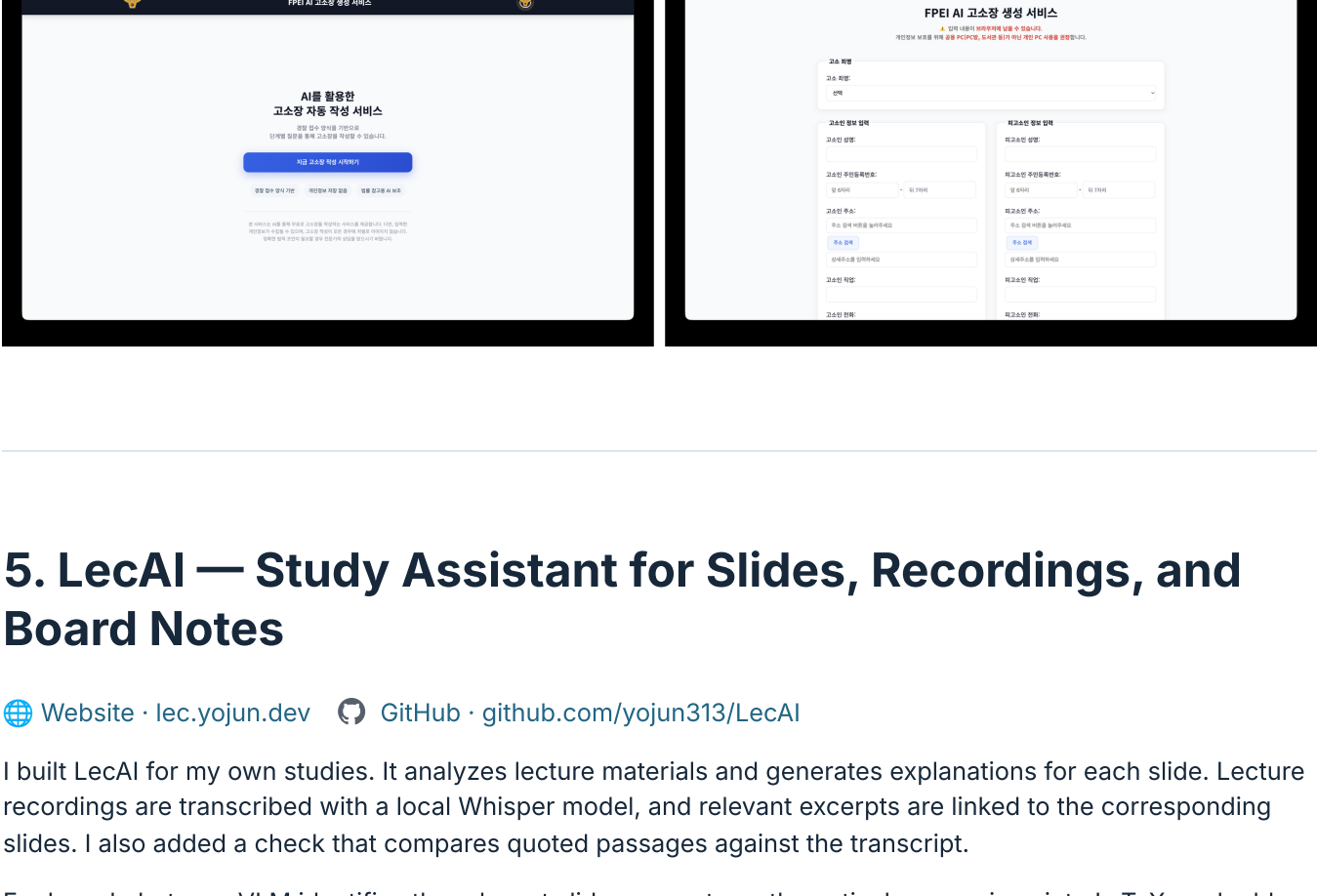
A tool for collecting data from Naver News, Blog, Cafe, and YouTube. I implemented asynchronous task queues and proxy IP rotation, with a web interface for monitoring collection jobs and reviewing the results.



Statistical Analysis

[Website](https://statistics.knpu.re.kr) · statistics.knpu.re.kr

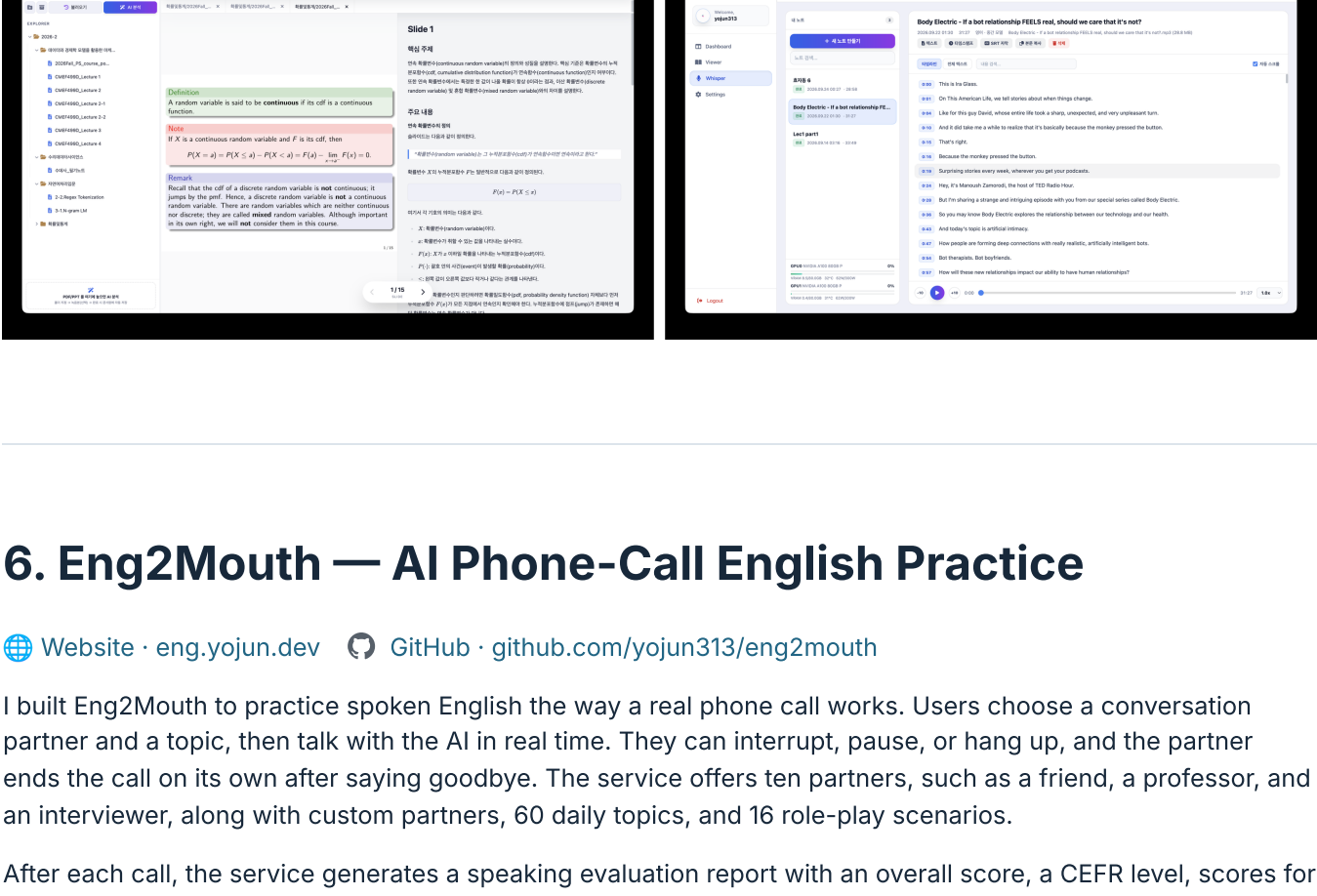
A tool for loading research data, running statistical analyses, and visualizing results. Researchers can select variables and analysis settings, then review tables and charts together.



Network Analysis

[Website](https://network.knpu.re.kr) · network.knpu.re.kr

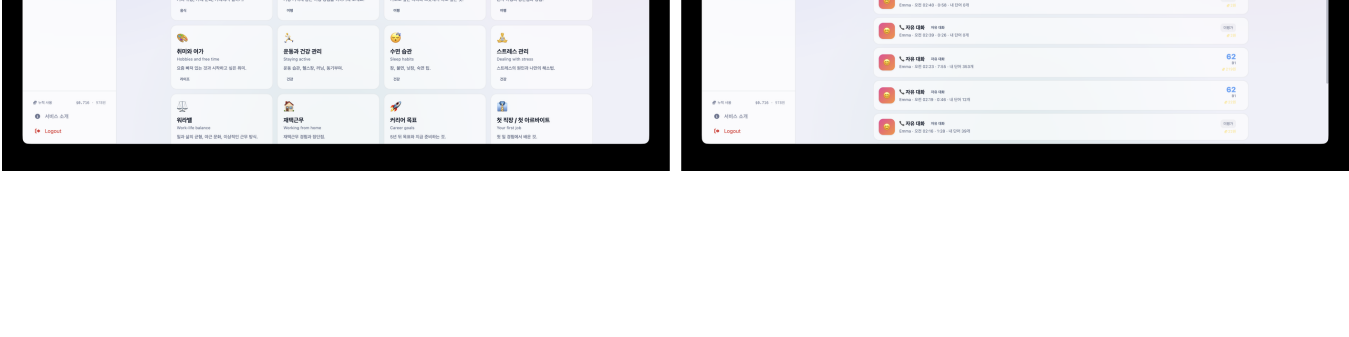
A tool for extracting and visualizing networks from text. It combines network graphs with analysis controls so researchers can explore connections between words and entities.



Keyword Signal Analysis

[Website](https://kemkim.knpu.re.kr) · kemkim.knpu.re.kr

I implemented the KEM/KIM methodology as software to support research analysis. The tool calculates indicators from keyword frequencies, document counts, and time weights, then classifies keywords. Researchers can adjust the settings, compare signal distributions and changes, and export results.



Research Data Manager — MANAGER

[Website](https://manager.knpu.re.kr) · manager.knpu.re.kr

I developed a desktop tool that brings together data collection, analysis, and research resource management. It provides database browsing, collection jobs, analysis functions, and user management in one environment. I later extended the research functionality to web applications.

5. LecAI — Study Assistant for Slides, Recordings, and Board Notes

[Website](https://lecai.yojun.dev) · lecai.yojun.dev

[GitHub](https://github.com/yojun313/LecAI) · github.com/yojun313/LecAI

I built LecAI for my own studies. It analyzes lecture materials and generates explanations for each slide. Lecture recordings are transcribed with a local Whisper model, and relevant excerpts are linked to the corresponding slides. I also added a check that compares quoted passages against the transcript.

For board photos, a VLM identifies the relevant slide, converts mathematical expressions into LaTeX, and adds step-by-step explanations. I use the tool to review lecture slides, the instructor's spoken explanations, and board notes together.

Technologies: Python, LLM APIs, VLM, Whisper, FFmpeg, LaTeX

6. Eng2Mouth — AI Phone-Call English Practice

[Website](https://eng.yojun.dev) · eng.yojun.dev

[GitHub](https://github.com/yojun313/eng2mouth) · github.com/yojun313/eng2mouth

I built Eng2Mouth to practice conversing the way a real phone call works. Users choose a conversation partner and a topic, then talk with the AI in real time. They can interrupt, pause, or hang up, and the partner ends the call on its own after saying goodbye. The service offers ten partners, such as a friend, a professor, and an interviewer, along with custom partners, 60 daily topics, and 16 role-play scenarios.

After each call, the service generates a speaking evaluation report with an overall score, a CEFR level, scores for fluency, grammar, vocabulary, organization, and interaction, corrected sentences, and useful expressions. Saved expressions can be reviewed with flashcards, and a dashboard tracks call history and score trends. Users connect their own OpenAI or Gemini API key. The server uses the key only to issue temporary tokens and never sends it to the browser.

Technologies: OpenAI Realtime API, Gemini Live API, real-time voice, PWA

