AY2025 CREATIVE FACTORY SEMINAR

CFS01 Human Activity Analysis and Recognition using Machine Learning Techniques

Instructors: Prof. Jungpil SHIN

Introduction

Developing systems and services that incorporate AI technology requires more than just pursuing superior technology and functionality. It is essential to consider how to deliver these to users and what kind of experience to create.

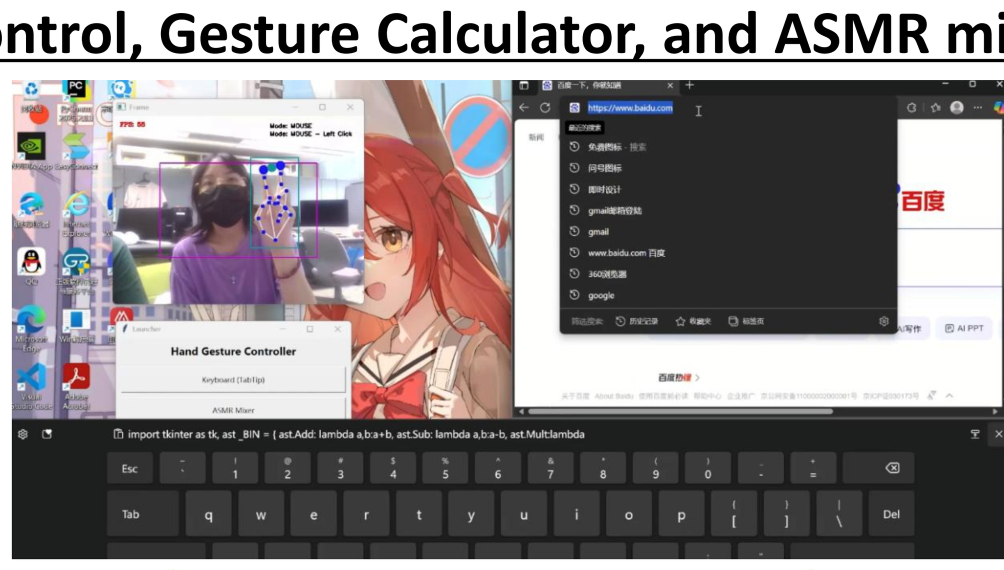
Deep learning-based emotion recognition, person identification and age estimation analyze users' faces and expressions in real time while they use services, enabling dynamic adjustments to user interactions and service delivery based on age and emotional state, thereby enhancing the user experience.

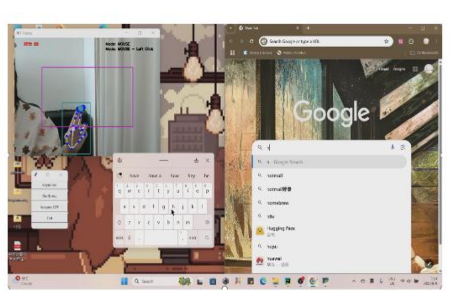
The development of non-contact interfaces also contributes to enhancing the user experience. By utilizing gestures and gaze without requiring touch panels or physical buttons, these interfaces offer significant advantages in terms of “intuitiveness” and “convenience.”

Hand Gesture Mouse: Vision-Based Virtual Mouse Control

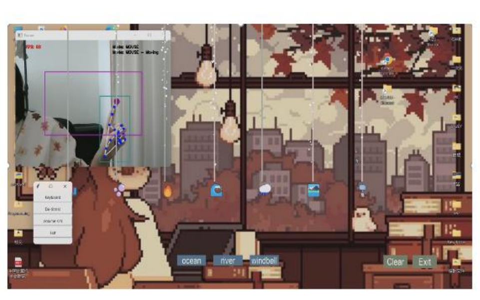
m5291009 Ching Wai Teng, m5291029 Jiang Yi

- Develop a **virtual mouse system** using **hand gestures** detected by a webcam.
- Achieved real-time and stable gesture recognition.
- Additionally, this apps supports operation of features such as **volume control, Gesture Calculator, and ASMR mixer**.

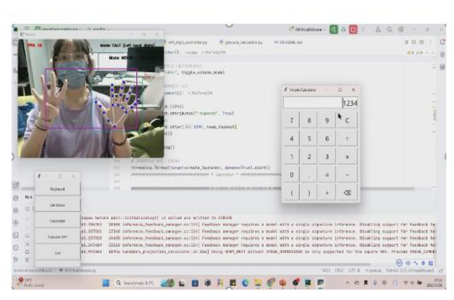




Virtual Keyboard



ASMR Mixer

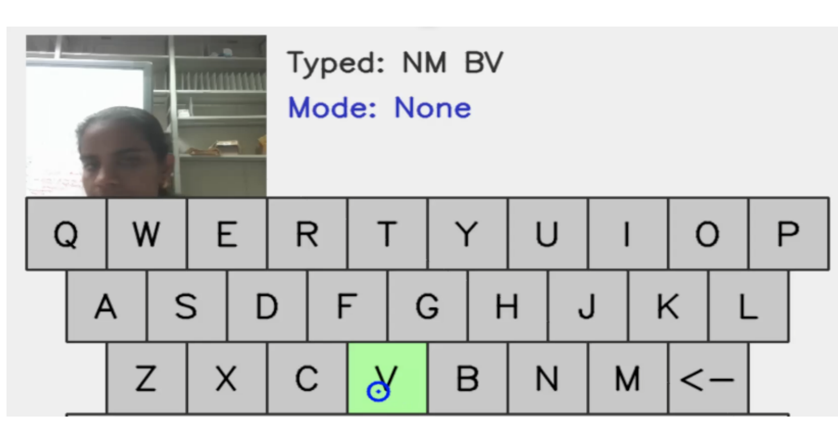


Gesture Calculator

Virtual Handwriting & Eye-Typing Project

m5291001 Saurabh Singh Adhikari, m5291068 Charu Shukla

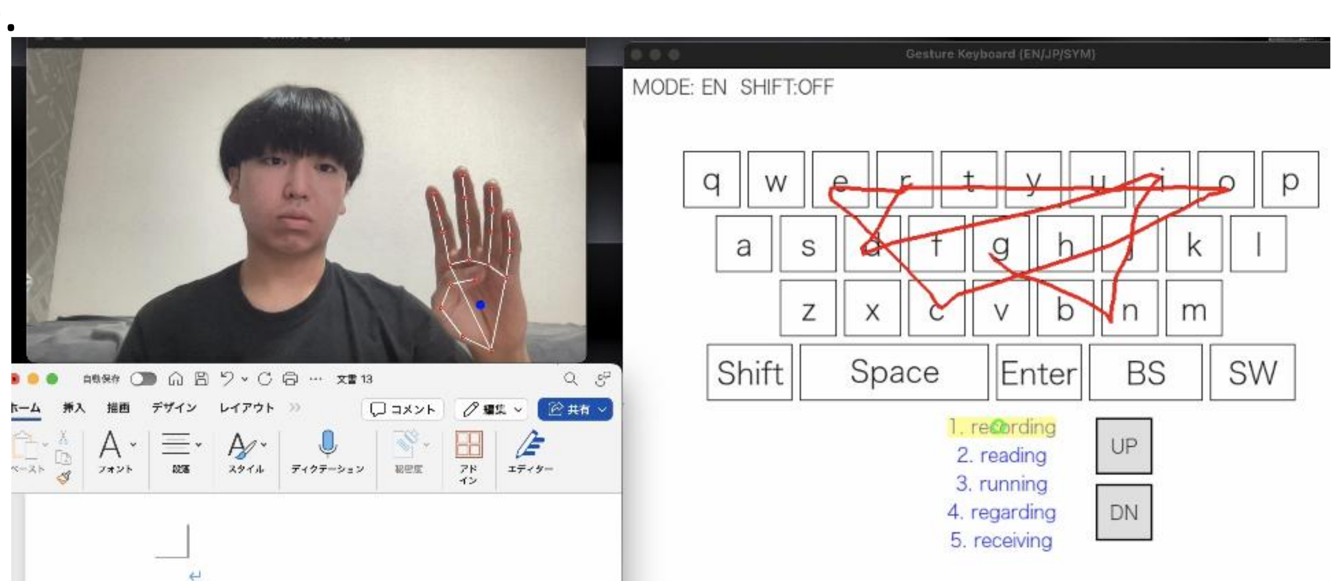
We developed a **real-time multimodal virtual text input system** combining handwriting recognition based on **hand gestures** with **gaze and blink** detection to enable hands-free and accessible communication.



Contactless Mouse Operation and Keyboard Input System Using Mediapipe and PyAutoGUI

M5291014 Ran Fujii, m5291089Takuya Yahagi, m5291092 Tatsuki Yamanaka

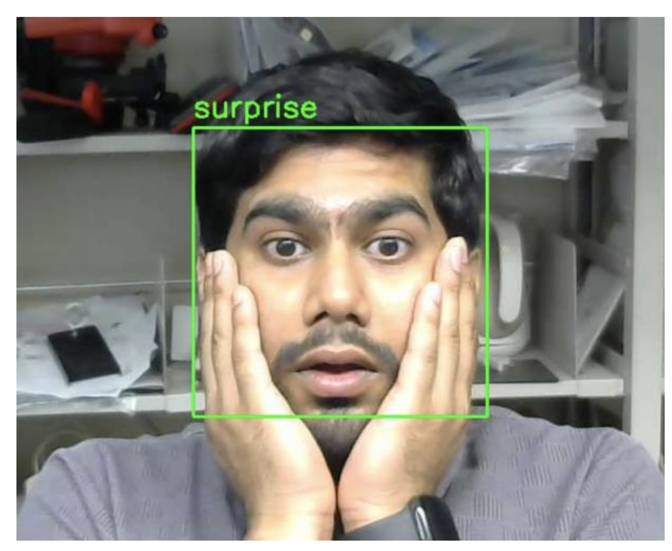
Developing a **contactless operation system** that integrates mouse control and keyboard input using a **camera-based gesture** interface.

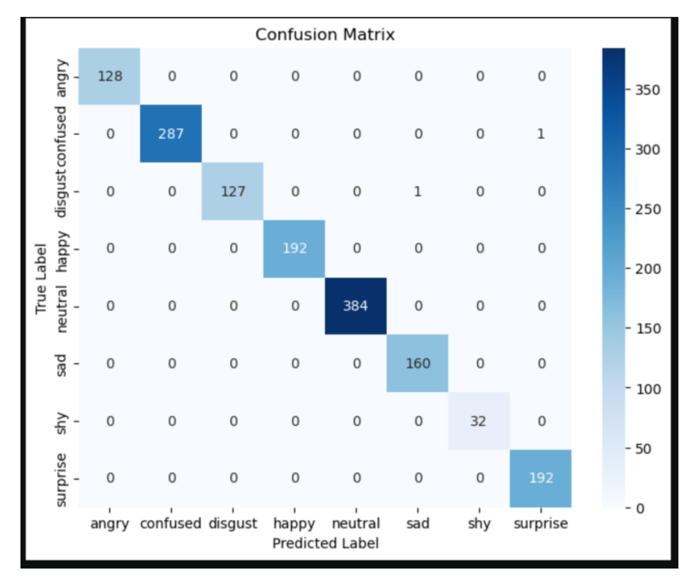


Improved Facial Emotion Recognition by merging and restructuring imbalanced dataset

m5281002 Alvi Ali Haider, m5282006 Jahidul Islam

Proposed approach achieved a **99.87% recognition rate in facial expression recognition** by preprocessing to address imbalanced datasets and utilizing transfer learning techniques with ResNet50.

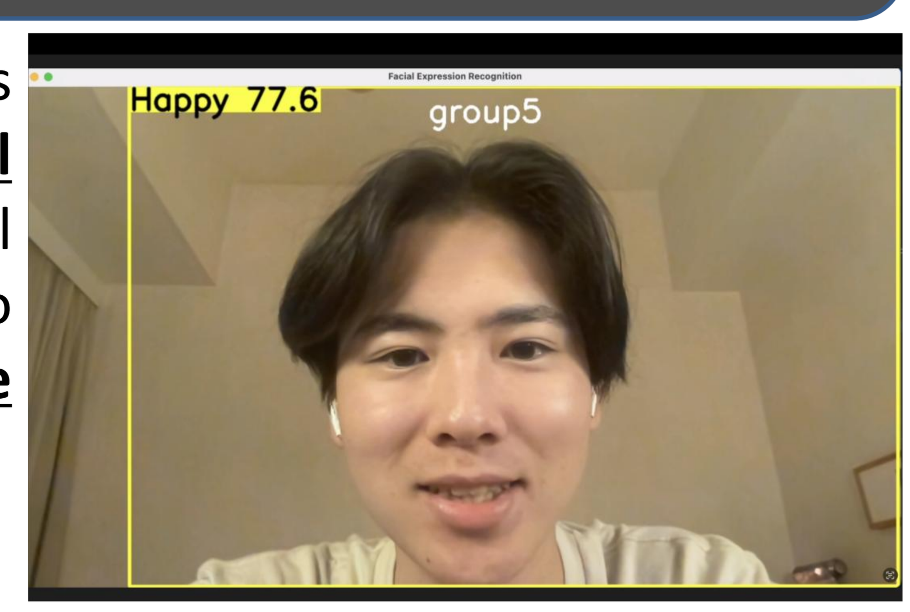




Emotion Recognition Music System

m5281014, Ichikawa Yusei, m5281030, Nakamura Zen, m5291051, Murakami Tatsuya; m5291067, Hoshitaka Shu

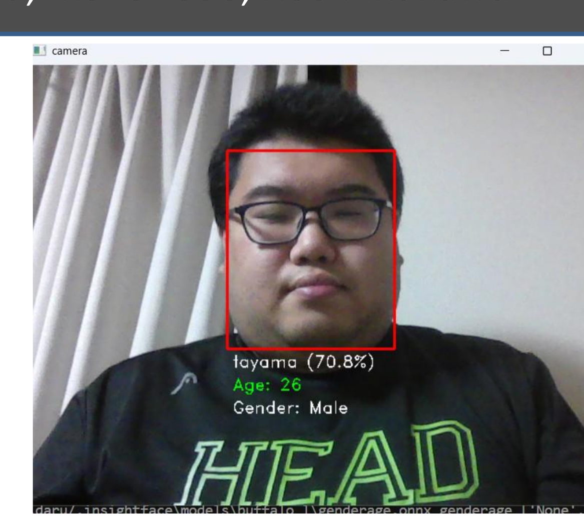
A system that recognizes **emotions and situations in real time** from the facial expressions of up to two people and **modulates the background music** accordingly.



Person Identification and Age Estimation

m5291006, Daichi Azuma, m5291079, Taichi Tayama, m5291036, Reo Kikukawa

This system **identifies person** by comparing images input from the camera with image data registered through ***InsightFace***., and **estimate age** for detected face.



A Real-Time AI Yoga Coaching System

m5291035 Yosuke Kanno, m5291080 Mugi Tsushima, m5291090 Taisei Yakushinji

The system uses ***OpenCV*** and ***MediaPipe*** to **detect and estimate the user's yoga pose**, and **provides feedback** on how consistently it approximates the reference pose angle over a 5-second period.

CFS02 Map and track yourself from the ground and the air!
Self-positioning and 3D surveying with rover and drone

Instructor: Prof. DEMURA, H., OGAWA, Y., YAMADA, R., HONDA, C.

Members:
A0 Poster Making: KAKAZU Kengo(γ), KOMO Yui(δ), KOSHITA Kosei(δ), SATO Risa(δ), UCHISIMA Ayumi(α), WEDAMUNI ARACHCHIGE Randima Nimantha(β), TANAGI Atsuya(β)
Small Poster Making in Japanese: MATSUI Iori(α), YASUI Banri(α), KONNO Daichi(α)
Poster Presentation Talker: WITHANAARACHCHI Tithira Yasaswin(β), IKEDA Yuma(β), KAMINSKI Philipp Michael(γ),
Supporting Materials: ARAI Takaaki(γ), MIYAZAKI Hayate(γ),
(Role: α-> Drone operation/experiment recorder β->GNSS chip installation/drone hardware γ-> 3D shape reconstruction δ-> Self-localization)

Introduction

In recent years, drones have become increasingly prevalent in various fields, including construction and disaster prevention. Drones have made it possible to carry out tasks that were traditionally done by hand more safely and efficiently at a lower cost. One of the key innovations is the high quality of data that can be acquired, as well as its versatility of application. Not only can drones capture high-resolution images and videos from inaccessible locations, they can also analyse this data to generate precise 3D models of terrain and structures.
Self-localisation is essential to generate these 3D models. This means knowing the exact location and orientation of each image or video frame captured by the drone.
The aim of this research is therefore to reconstruct the 3D shape of an object and determine its position from aerial footage captured by a drone. Specifically, we will film an area containing a red cone representing a rover using a drone. From this footage, we will perform 3D shape reconstruction to determine the precise position of the red cone.

Methodology

1. Equip the drone with a GNSS chip.
2. Use the drone to conduct aerial photography of the simulated crater.
3. Reconstruct 3D shapes from drone-captured video footage.
4. Estimate the position of the tip of red cone, representing the rover.

The above steps are carried out in two teams.

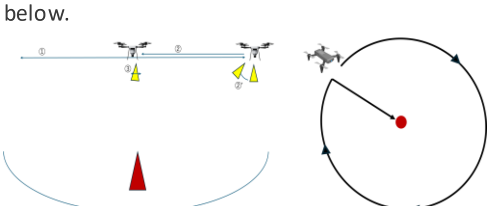
Equipment used: drones & GNSS

MAVIC2 PRO / ZOOM MAVIC3 PRO

Experiment

Aerial photography was conducted using drones at the Robot Test Field in Minamisoma.

It was discovered that flying a drone equipped with a GNSS chip is illegal, so aerial photography was conducted without installing the GNSS chip.
A red cone is placed at the center of the mock crater, and the drone is flown so that the cone is visible in the footage.
The drone was flown as shown in the figure below.

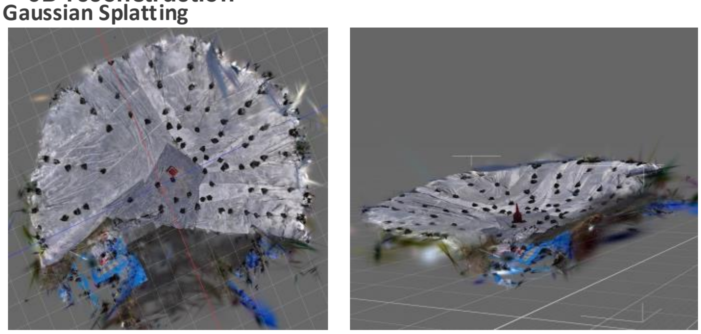


: observation object

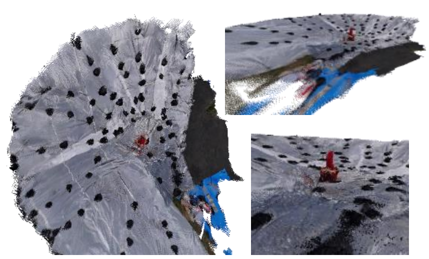
Results & Discussion

・3D-reconstruction

Gaussian Splatting



VGGT



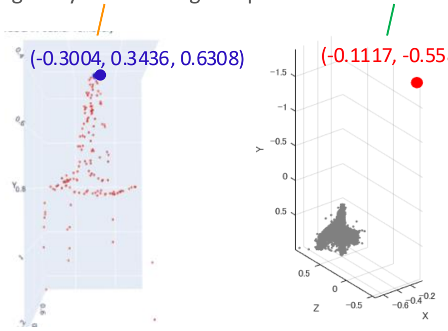
Restoration using VGGT is less effective due to sparse point clouds, making Gaussian Splatting a superior method. However, both methods make the ground appear somewhat flat. We should find a better method for 3D reconstruction that consider features such as crater rims.

・Estimation of the red cone position

The position of the tip of red cone was identified using the 3D reconstruction results from Team A.

Team A :Color-based estimation
Extract the red point cloud and calculate from the center of gravity near the highest point.

Team B : Shape-based estimation
Extract a point cloud shaped like a cone and determine its coordinates.



When point clouds are scatted or distorted, shape-based estimation becomes difficult. In this case, color-based estimation is more suitable.

Conclusion

This survey took consistent aerial photographs of the observation object and used the resulting video and images to restore its three-dimensional shape. We need to find a better reconstruction method. In this case, we estimated the position by treating the red cone as a rover, but we should establish a method for estimating the position of a moving rover.

AY2025 CREATIVE FACTORY SEMINAR (CFS03)

Teleoperation of Robot Arm with Haptic Feedback
ハプティックフィードバック付きロボットアームの遠隔操作

Instructors: Lei JING

SEMINAR INTRODUCTION

Imagine a future where a “copy robot” can replicate human skills with precision and adaptability. Such a robot would not only mirror our movements but also internalize the subtlety of human expertise—whether in assembly lines, healthcare assistance, or delicate agricultural tasks. This vision of a robot that can imitate, learn, and eventually augment human abilities represents one of the grand challenges in robotics and human–machine interaction. It is within this vision that our current seminar is positioned.

In this seminar, we present the progress made by three student groups, each tackling aspects of teleoperation, VR simulation, and imitation learning.

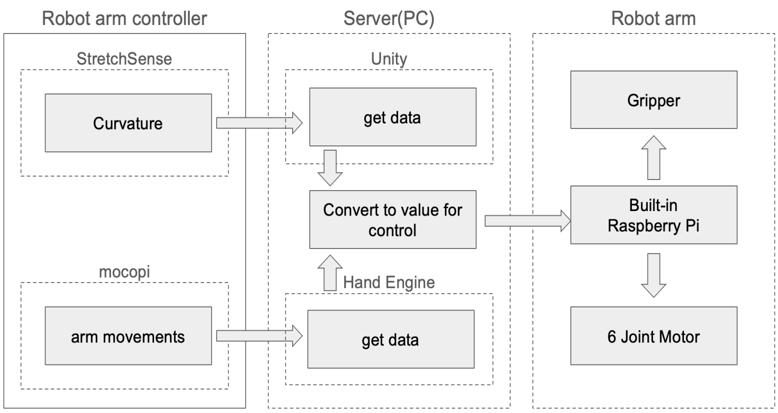
Group A built a teleoperation system with sensors and RealSense tracking.

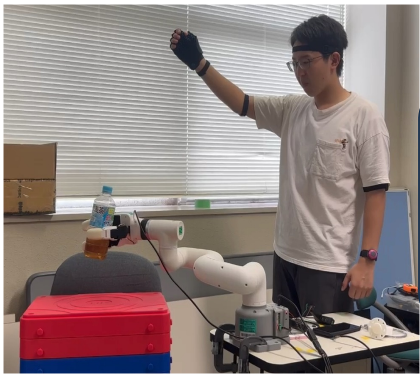
Group B created a VR twin for efficient experiments and data collection.

Group C applied imitation learning to reproduce human actions.

Together, these efforts form a path from teleoperation to VR simulation and machine learning imitation, moving toward robots that can observe, learn, and copy human expertise.

Group A: Robot Arm Control with Human Action

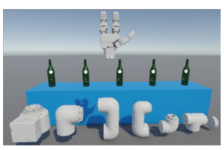




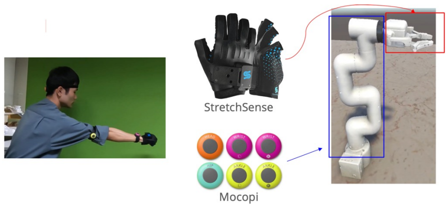
Members:
Shun KUGIMIYA
Ni CAI
Sahad NOUSHAD
Takeshi OKUYASU
OTAKE Toru
SASAKI Keito
Shashank BHARRDWAJ

Group B: VR Twins Control

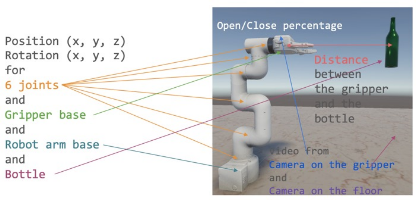
1. Prepare the VR environment



2. Control VR Robot with Real-World Sensors



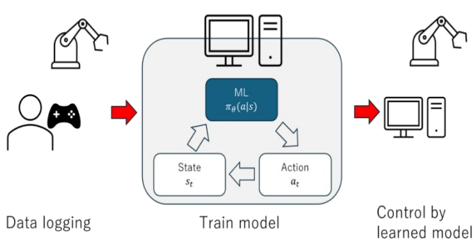
3. Training Data Logging and Formatting



Members:
Kaede TAKAHASHI,
Atsuya WATANABE,
Takase ITO, Hao XU,
Salim YANAGIDA,
Ryota WATANABE,
Cai NI

Group C: Imitation Learning

Behavior Cloning



Members:
Cui Jiaxing
Huang Rui
Yuto Nakazato
Reshma Ramesh
Takuto Odaira
Zhang Boyu
Yang Huaqin

Conclusion

•**Period and Participants**
This seminar was held about 1.5 months (July 2 – August 13) with 20 students organized into three groups.

•**Achievement**
the seminar produced a stepwise roadmap: teleoperation → VR simulation → machine learning–based imitation.

•**Future Work**
- Improve accuracy and robustness of teleoperation systems under varied conditions.
- Scale imitation learning with larger datasets and advanced algorithms (e.g., Transformer-based policies).
- Integrate teleoperation, VR, and imitation learning into a unified framework.

AY2025 CREATIVE FACTORY SEMINAR

Performance Improvement of Applications Using FPGA Boards

Instructors: SAITO Hiroshi, KOHIRA Yukihide, TOMIOKA Yoichi

Objective

To accelerate an application using a field programmable gate array (FPGA) board

Through this CFS, students learn

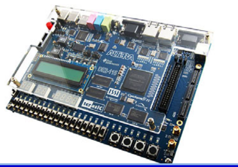
1. How to model an application
2. How to use a design tool
3. How a synthesized circuit works on an FPGA board
4. Evaluation of the developed circuit comparing with CPU execution

FPGA

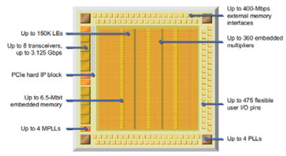
A reconfigurable device

- Designers can change circuit structure freely
- Low design cost
- Well used in embedded systems

Terasic DE2-115 (source:terasic)

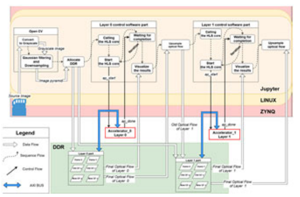


Cyclone IV FPGA (source:Intel)



Pyramid Lucas–Kanade Method

Designer: LIU Zhifei
Motivation: Optical flow is well used to detect a motion of objects
Purpose: Acceleration of the PLK method
System architecture:



Result:

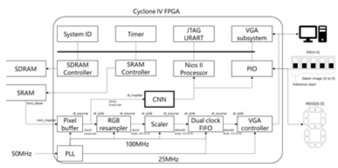
- Board: Pynq-z2

Layer	Resolution	Simulation (ms)	PYNQ-z2 (ms)	12th i7 (ms)
0	80*60	4.3	5.2	6.4
1	160*120	18.9	22.7	26.7
2	320*240	79.0	95.9	112.7
3	640*480	318.0	452.6	472.4

93.2% 100%

Lightweight CNN Implementation for MNIST Digit Classification

Designer: AWAKI Kyoji
Motivation: CNNs require numerous convolution operations, resulting in a massive computational load
Purpose: Acceleration of the CNN for the MNIST Digit Classification
System architecture:



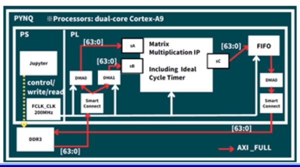
Result:

- Board: DE2-112

Metric	CPU (O3)	Core (Simulation)	FPGA(Actual Device)
Latency / Image	15 μs (baseline)	7.8 μs (−48%)	12.2μs(-19%)

Matrix Multiplication

Designer: HAMADA Shoma
Motivation: Matrix multiplication is a fundamental computational method, especially in high-performance scenarios
Purpose: Acceleration of a matrix multiplication
System architecture:



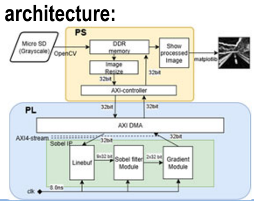
Result:

- Board: Pynq-z1

Target	Total Time	Matrix Calculation Time (ideal)	Clock Cycle
Apple M4 chip	100ns		
11th Gen Intel(R) Core(TM) i7-1195G	207ns		
Pynq-Z1 (This study)	1.453ms (slowest)	110ns	200 MHz

Sobel Filter

Designer: MAETSUBO Kazuki
Motivation: Edge detection is utilized as a preprocessing step in machine learning tasks related to images, such as object detection
Purpose: Acceleration of the Sobel filter
System architecture:

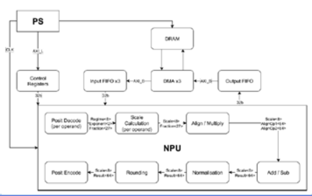


Result:

- Board: Pynq-z1
- CPU time : 26.633 ms 100%
- HW processing time : 5.786 ms 21.7%

Posit-based Numerical Processing Unit

Designer: RAMOS Justin Kyle Nodora
Motivation: FPUs consume significant area & power on FPGAs
Purpose: Acceleration of a posit-based numerical processing unit
System architecture:

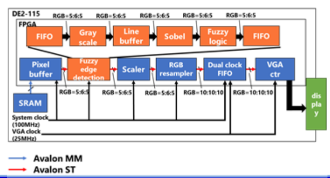


Result:

- Board: Pynq-z1
- Posit NPU (32-bit posit): (23.4 ms)
- Note: Synthesis/simulation
- Software on CPU (32-bit posit) :(60 ms)

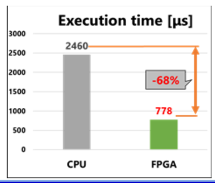
Fuzzy Edge Detection

Designer: SAGA Haruki
Motivation: High-speed processing for edge detection is crucial in edge devices to enable real-time processing
Purpose: Acceleration of the Fuzzy edge detection
System architecture:



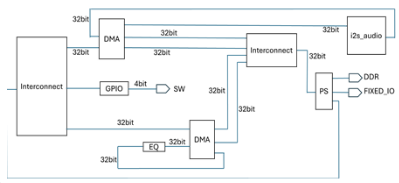
Result:

- Board: DE2-115



Audio Equalizer Using FFT

Designer: SHIOTA Rui
Motivation: Audio applications demand high throughput and low latency
Purpose: Acceleration of an audio equalizer using FFT
System architecture:

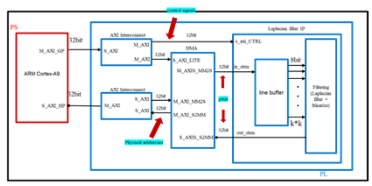


Result:

- Board: Zybo Z7-20
- CPU 1.142 s 100%
- Zybo Z7-20: 0.566 s 49.5%

Laplacian Filter

Designer: WATANABE Kyo
Motivation: Edge detection is essential for real-time system
Purpose: Acceleration of the Laplacian filter
System architecture:



Result:

- Board: Zynq-z1

	Execution time [ms] (5-run average)	
CPU (Core i7, gcc -03)	101.4	100%
FPGA (Zynq, Jupyter)	39.3	38.8%

AY2025 CREATIVE FACTORY SEMINAR

Developing Parallel Big Data Analytical Framework for Knowledge Discovery in Multiple Timeseries Data

Instructors: RAGE Uday Kiran

Introduction

1. Air pollution monitoring in Japan is crucial due to high urban density and industrial activity, generating large multivariate time-series data that challenge traditional analytics.
2. The scale and complexity of AEROS sensor network data make real-time analysis and actionable insights difficult, with existing methods failing to efficiently process large-scale time-series data.
3. To address these challenges, air pollution data from Japan's AEROS network was organized into two tables (Hourly_Observations and Station_Info) on a multinode PostgreSQL cluster, enabling advanced analysis.
4. A distributed Dask + PyTorch framework was built for fast forecasting, combined with LLM-based querying to simplify exploration and uncover hidden patterns.

Proposed Solution

A distributed Dask + PyTorch framework for efficient forecasting, combined with LLM-based querying for intuitive exploration.

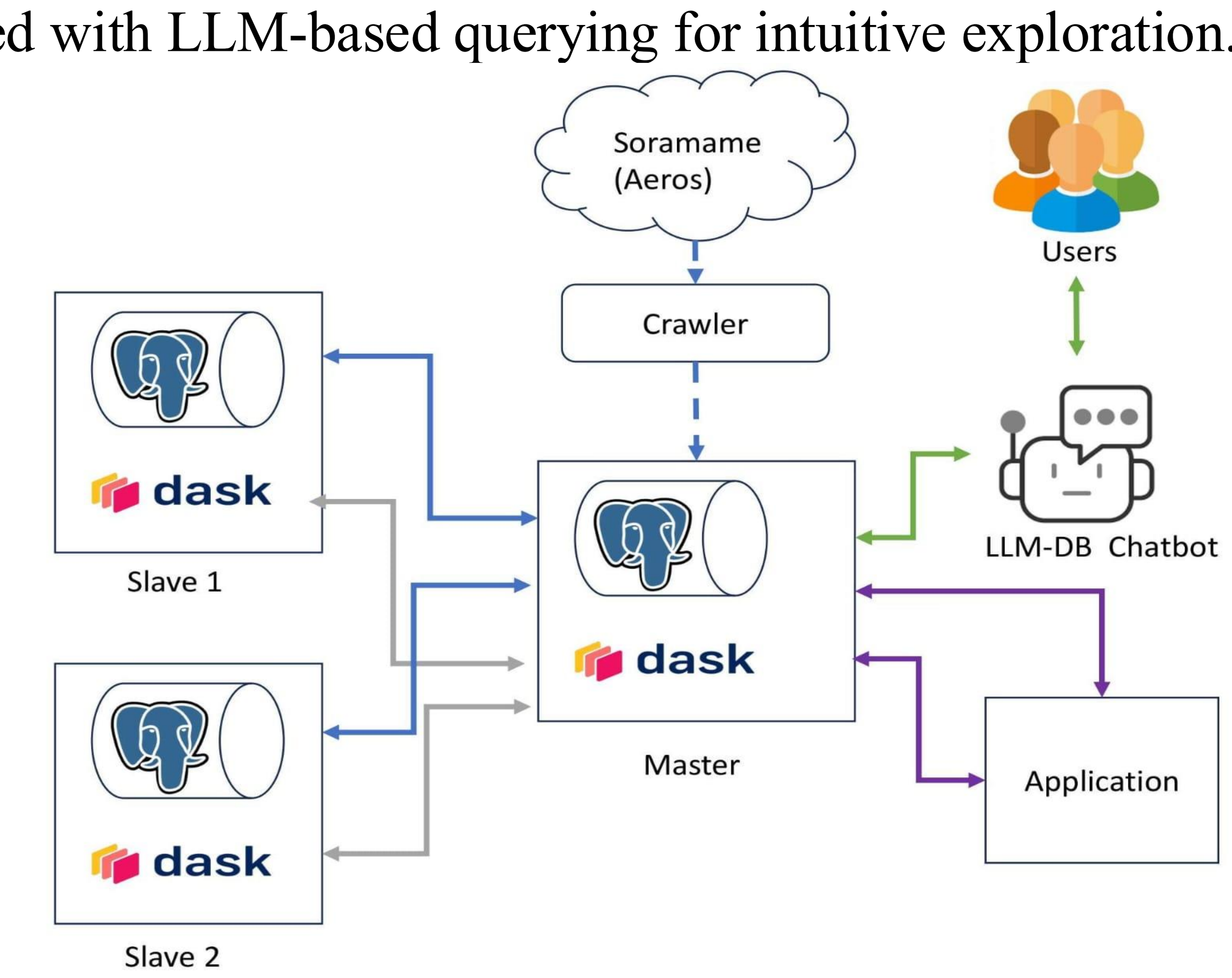


Figure 1: Overall Architecture of the Parallel Big Data Framework

Distributed Forecasting Architecture

1. Developed a scalable Dask framework for parallel processing and fault-tolerant distributed computing, with real-time performance monitoring via dashboards.
2. Designed and evaluated 5 deep learning models (RNN, LSTM, GRU, Bi-LSTM, CNN) for environmental data forecasting, processing data from all sensors simultaneously with unified evaluation and HTML-based reporting.

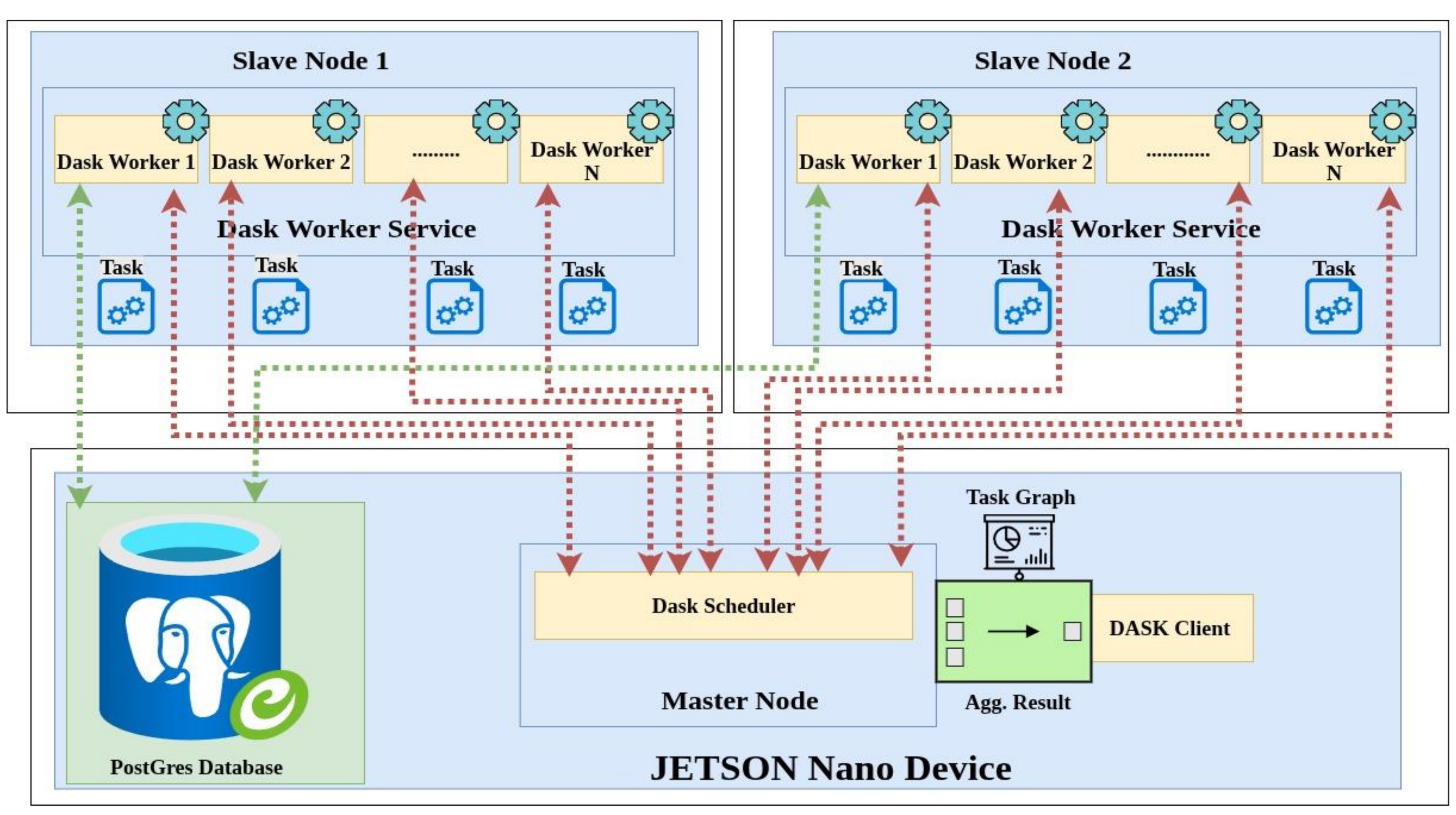


Figure 2: Parallel and Distributed Data Processing Architecture Using Dask

LLM-Based Query Processing Architecture

1. Users submit queries via a responsive interface using React, Tailwind, Vite, and TypeScript. The LLM generates a SQL query, executes it on PostgreSQL, and delivers concise, human-readable answers with the query and data preview.
2. We used 7 LLM models, including DeepSeek-Coder:6.7B and Ollama models (Mistral, Llama3, LLaVA-Phi3, CodeLlama, Gemma, Phi3).

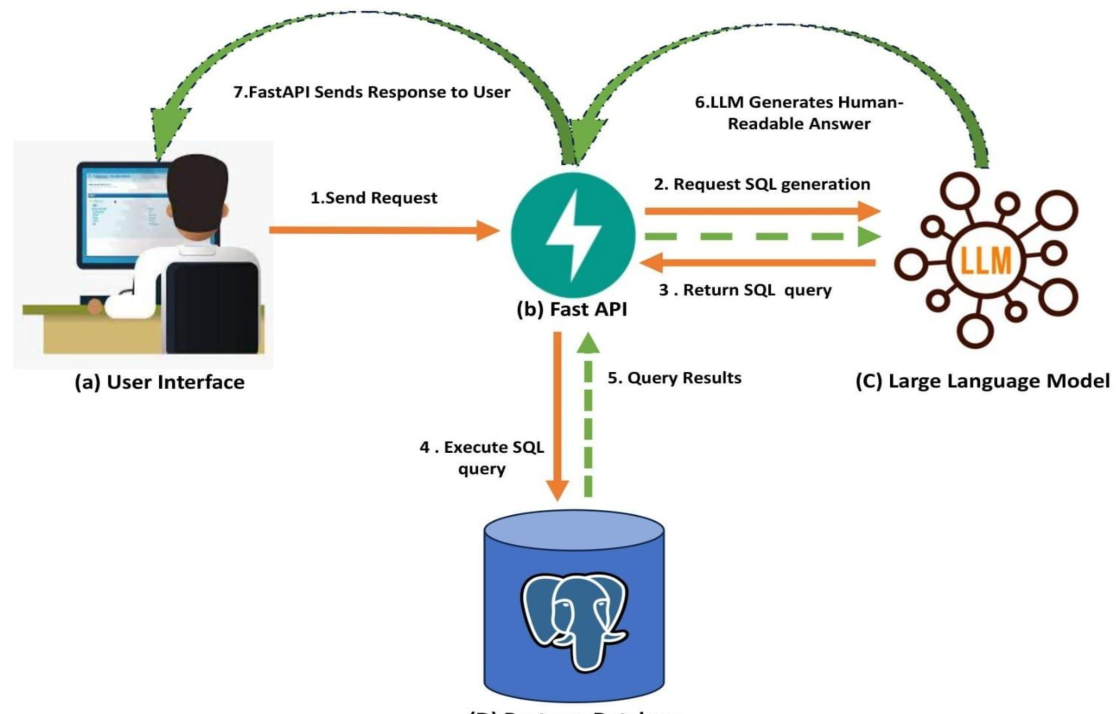


Figure 3 LLM-Based Query Processing Architecture

Results

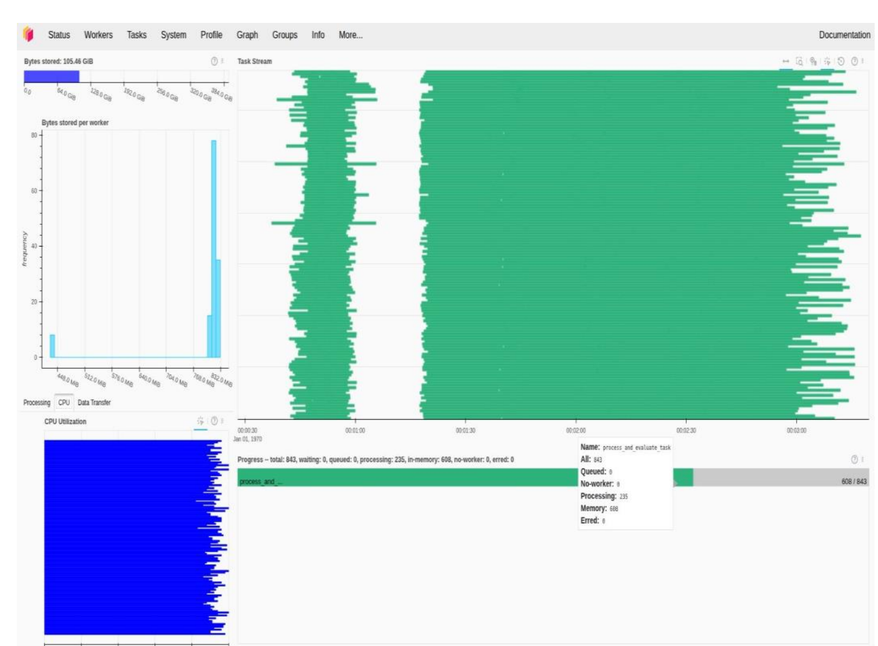


Figure 4: Real time dashboard monitoring

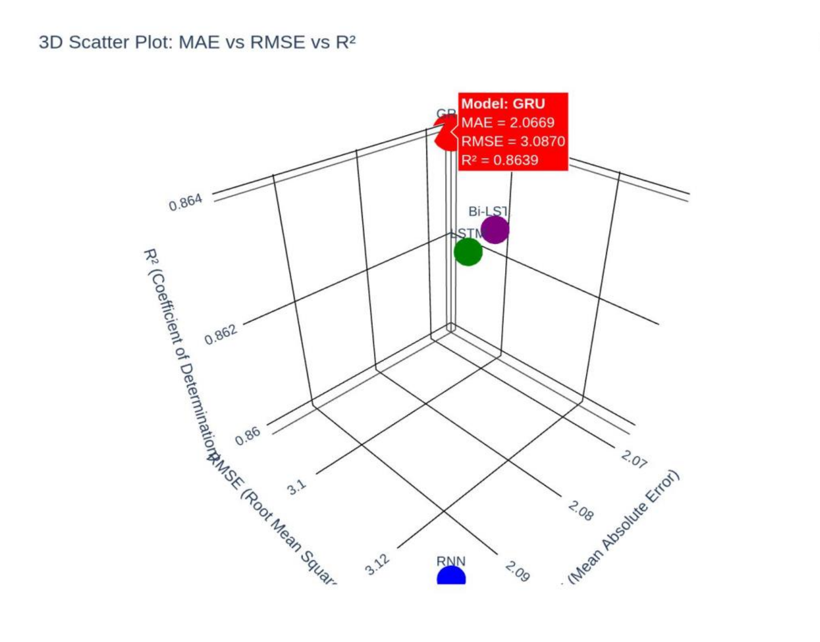


Figure 5: Comprehensive Evaluation

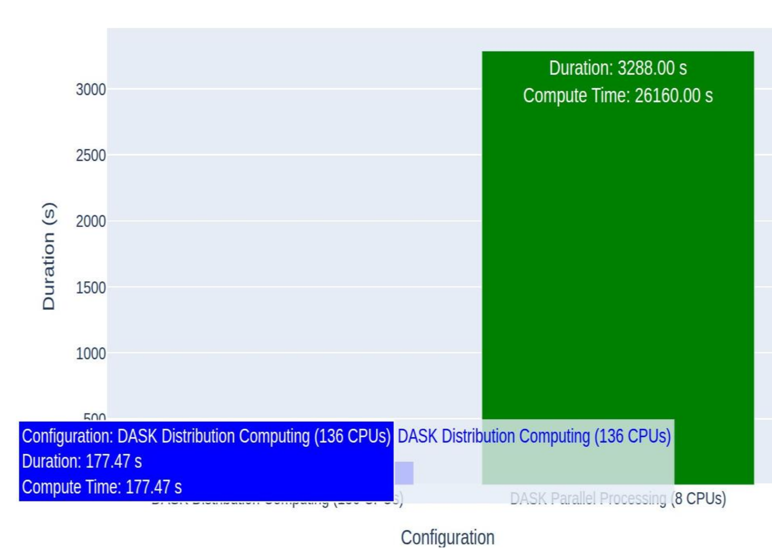


Figure 6: Computation time between DASK parallel vs distribution

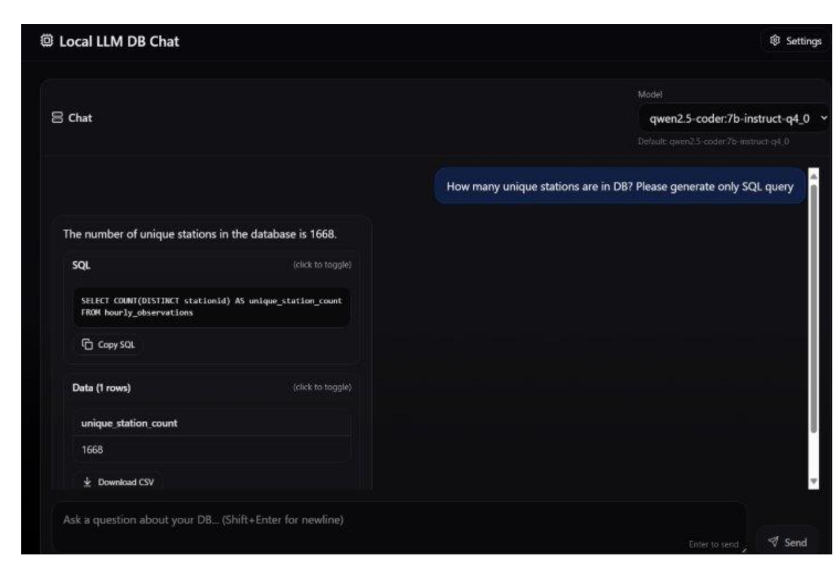


Figure 7: LLM-DB Interactive Interface

Conclusion

This research presents a scalable distributed framework for large-scale air pollution forecasting using DASK and Deep learning. Our system successfully processes all sensor streams concurrently, providing accurate multi-step forecasts while maintaining computational efficiency. The approach demonstrates the power of distributed computing for environmental analytics and enables real-time pollution forecasting at scale.

References

1. Ministry of the Environment, Japan. (n.d.). 環境省大気汚染物質広域監視システム（そらまめくん）. Retrieved September 17, 2025, from https://soramame.env.go.jp/
2. Dask.distributed. (2025, July). Dask.distributed documentation (Version 2025.7.0). Retrieved September 17, 2025, from https://distributed.dask.org/en/stable/
3. PyTorch. (2025). PyTorch Temporal Forecasting Research (Version 1.0). GitHub. https://github.com/pytorch/temporal-forecasting
4. Ollama. (2023). Ollama models (Mistral, Llama3, LLaVA-Phi3, CodeLlama, Gemma, Phi3). Ollama Inc. https://ollama.com