Automated Taxonomic Identification of Marine Plankton

USING A LARGE-LANGUAGE-MODEL AND RETRIEVAL-AUGMENTED GENERATION FRAMEWORK

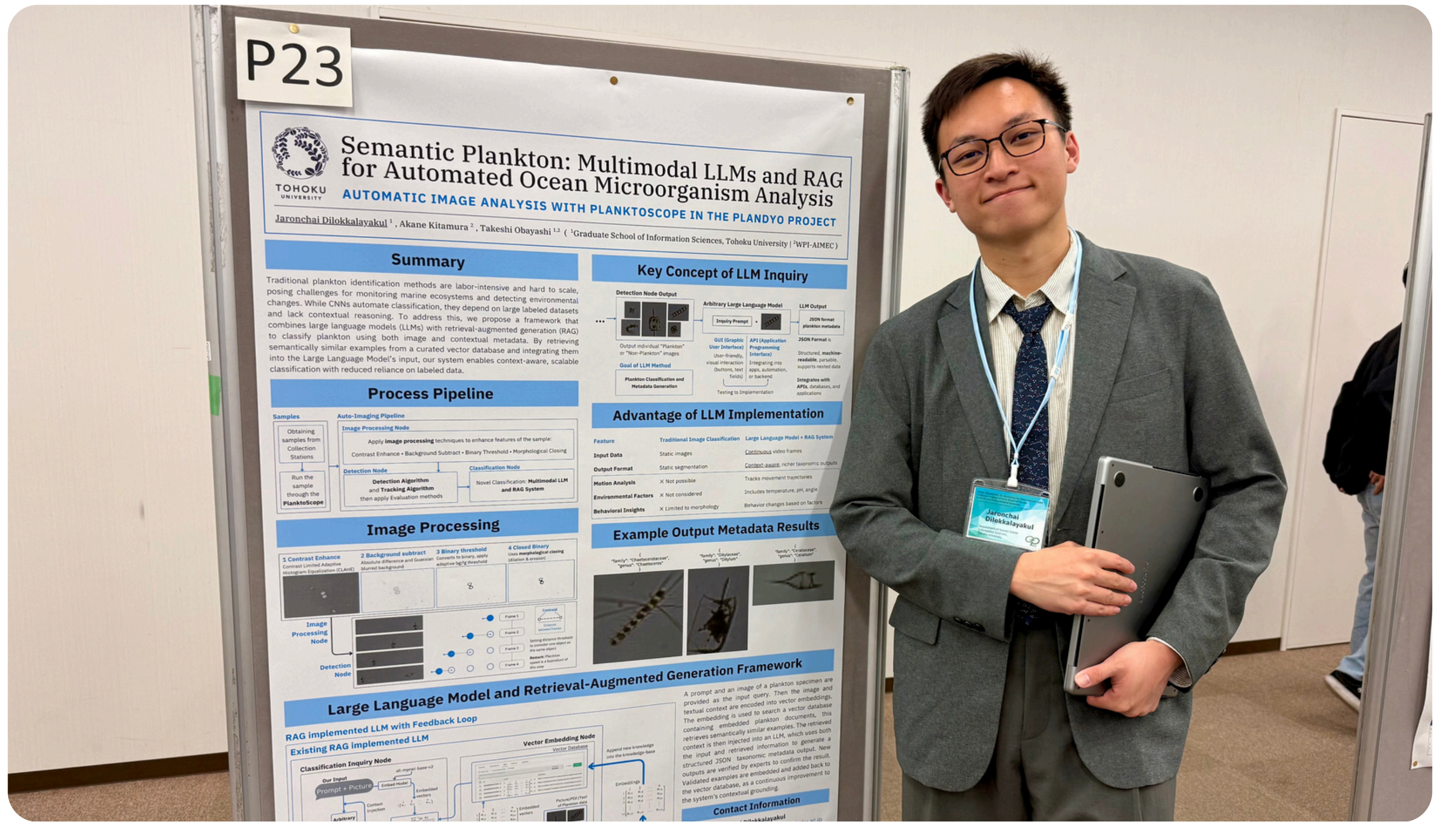
Jaronchai Dilokkalayakul [1] , Akane Kitamura [2] , Takeshi Obayashi [1,2]

[1] Graduate School of Information Sciences (GSIS), Tohoku University

[2] Advanced Institute for Marine Ecosystem Change (WPI-AIMEC), Tohoku University

INTRODUCTION ABOUT ME

JARONCHAI DILOKKALAYAKUL



- ***Graduate School of Information Science***, Information Biology Laboratory, Tohoku University, Japan
- *Data Engineering,* Bachelor of Engineering, International Program, Thai-Nichi Institute of Technology, Thailand

- **IT Engineer, *Info-Bio Lab***
  November 2024 – Present
- **Data Engineer, *IBM***
  May 2024 - October 2024
- **Data Engineer Intern, *IBM***
  June 2023 - November 2023
- Junior AI Researcher, *TNI*
  November 2022 – April 2023
- Computer Laboratory Assistant, *TNI*
  March 2022 – April 2023
- Programming Teaching Assistant, *TNI*
  November 2021 – March 2022

INTRODUCTION AND OVERVIEW

Objective of this project

Develop an **AI-powered analysis pipeline** to identify taxonomy and analyze **plankton trends and their influence on the marine ecosystem** along with LLMs technologies.

Samples

Collection Stations

PlanktoScope

Auto-Imaging Pipeline

Image Processing Node

Apply **image processing** techniques to enhance features of the sample:
Contrast Enhance • Background Subtract • Binary Threshold • Morphological Closing

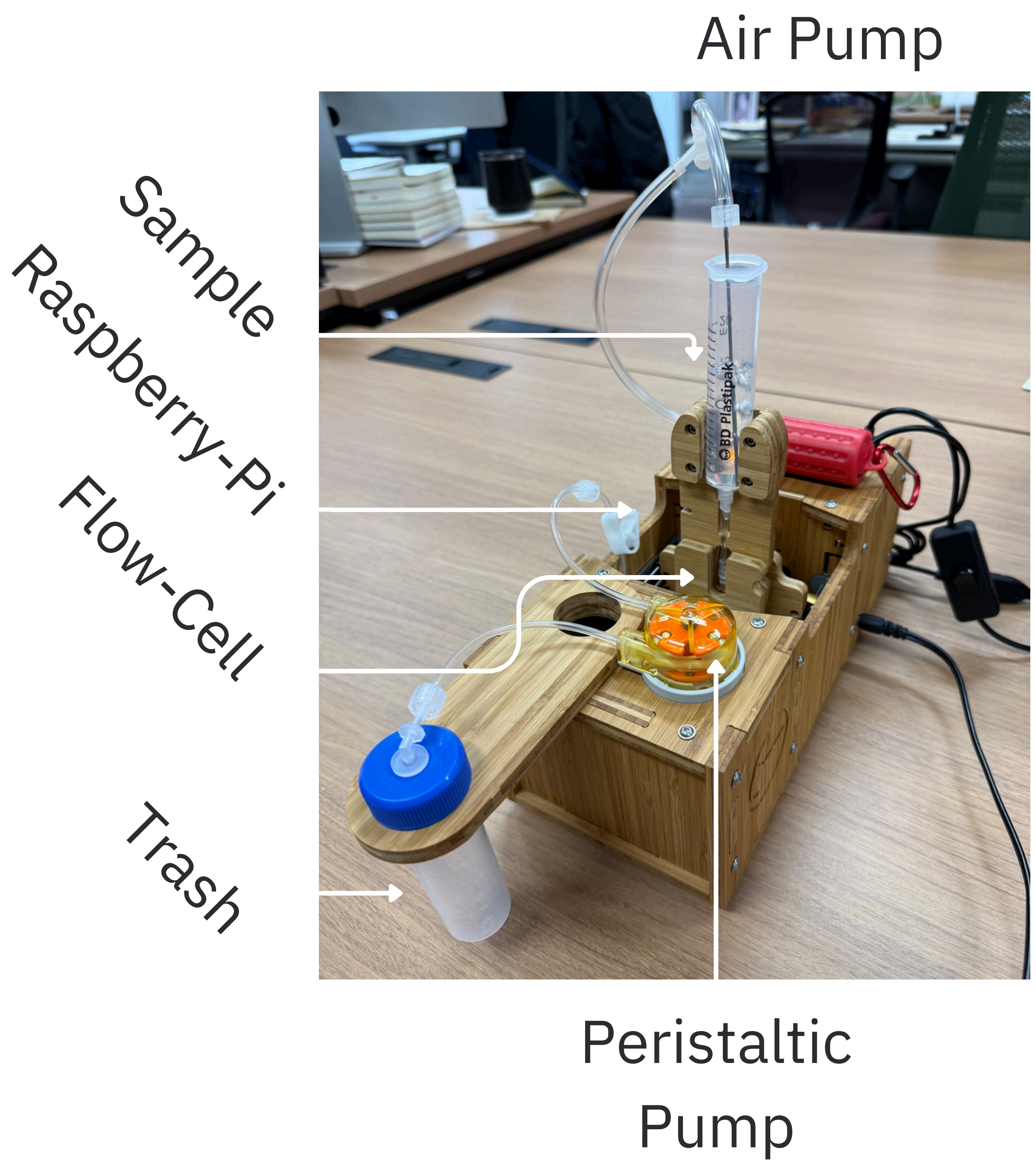
Detection Node

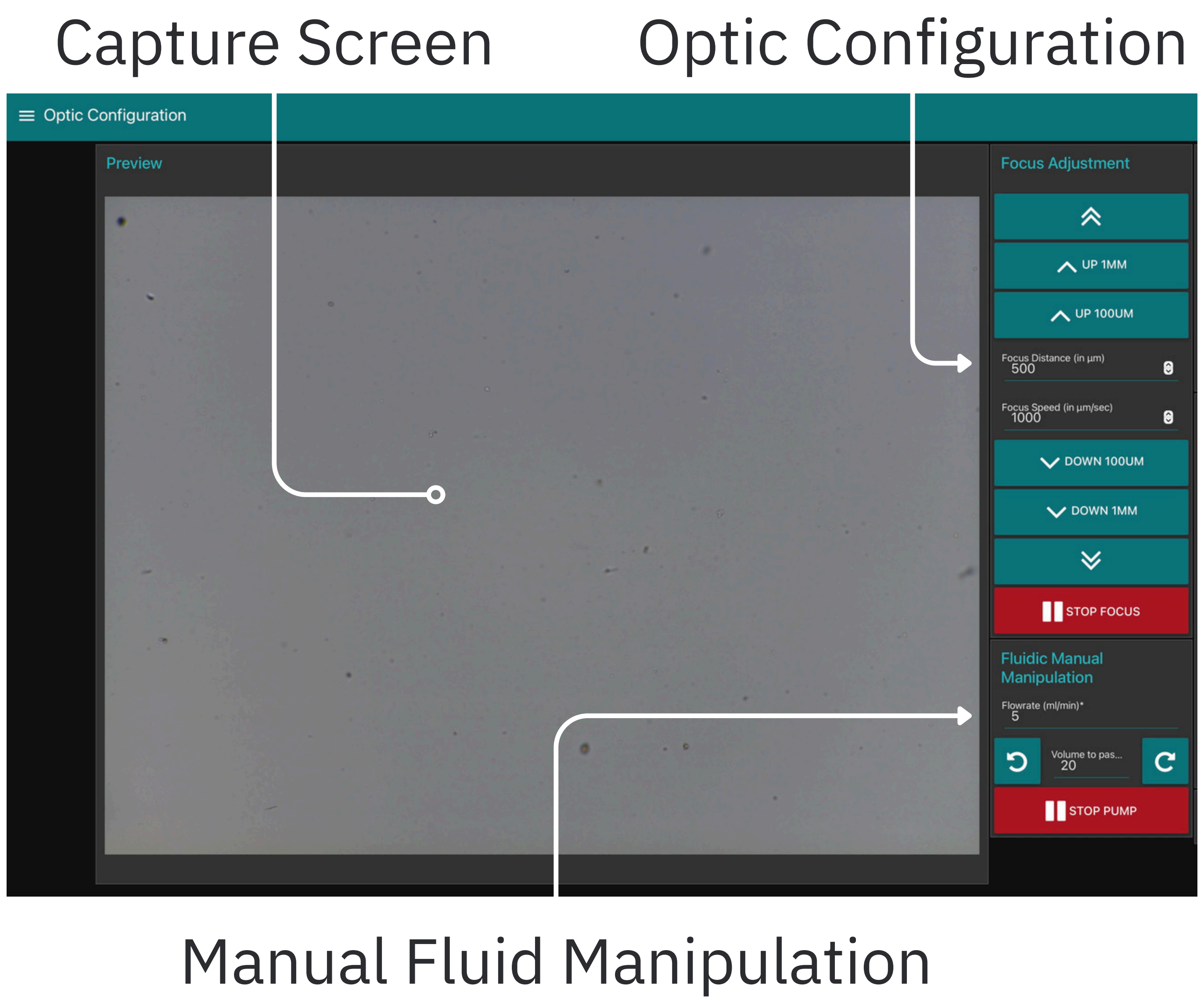
Detection Algorithm
and **Tracking Algorithm**

Classification Node

Multimodal LLM and RAG System

SAMPLE COLLECTION VIA PLANKTOSCOPE

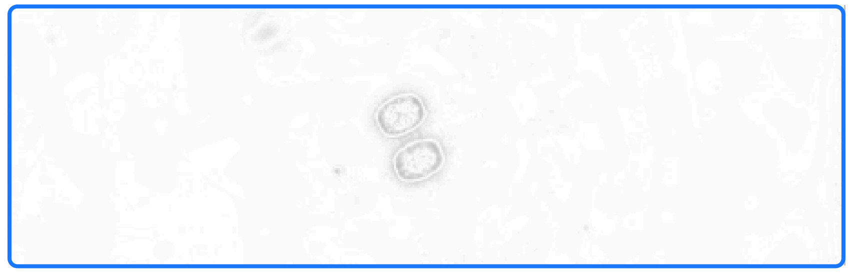




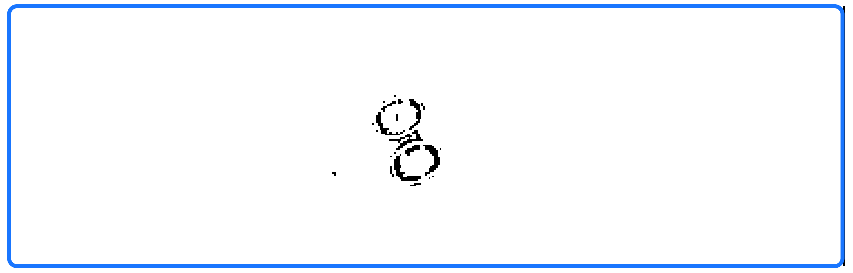
PLANKTON DETECTION ALGORITHM

1 Contrast Enhance
Contrast Limited Adaptive Histogram Equalization (CLAHE)

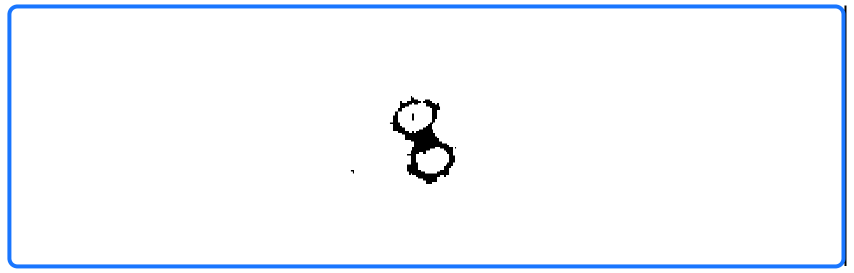
2 Background subtract
Absolute difference and Guassian blurred background

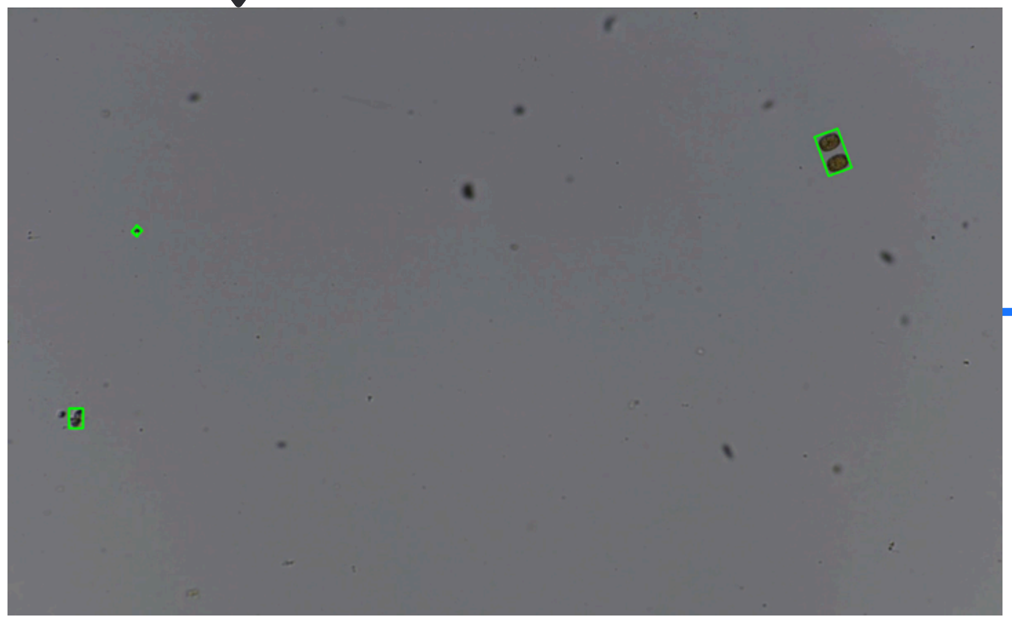


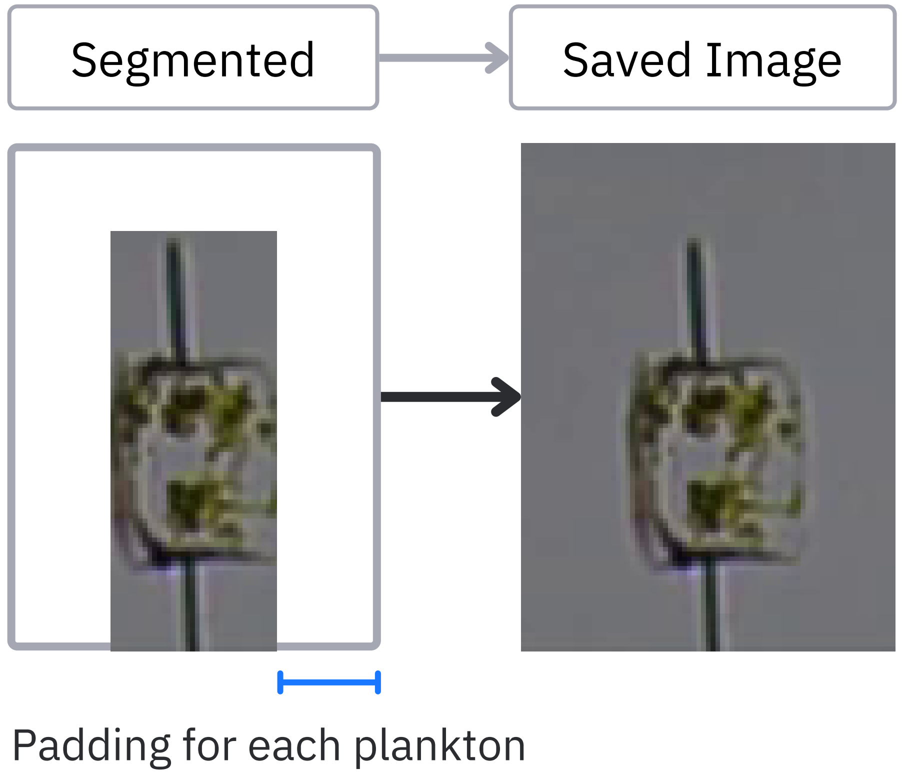
3 Binary threshold
Converts to binary, apply adaptive bg/fg threshold



4 Closed Binary
Uses **morphological closing** (dilation & erosion)

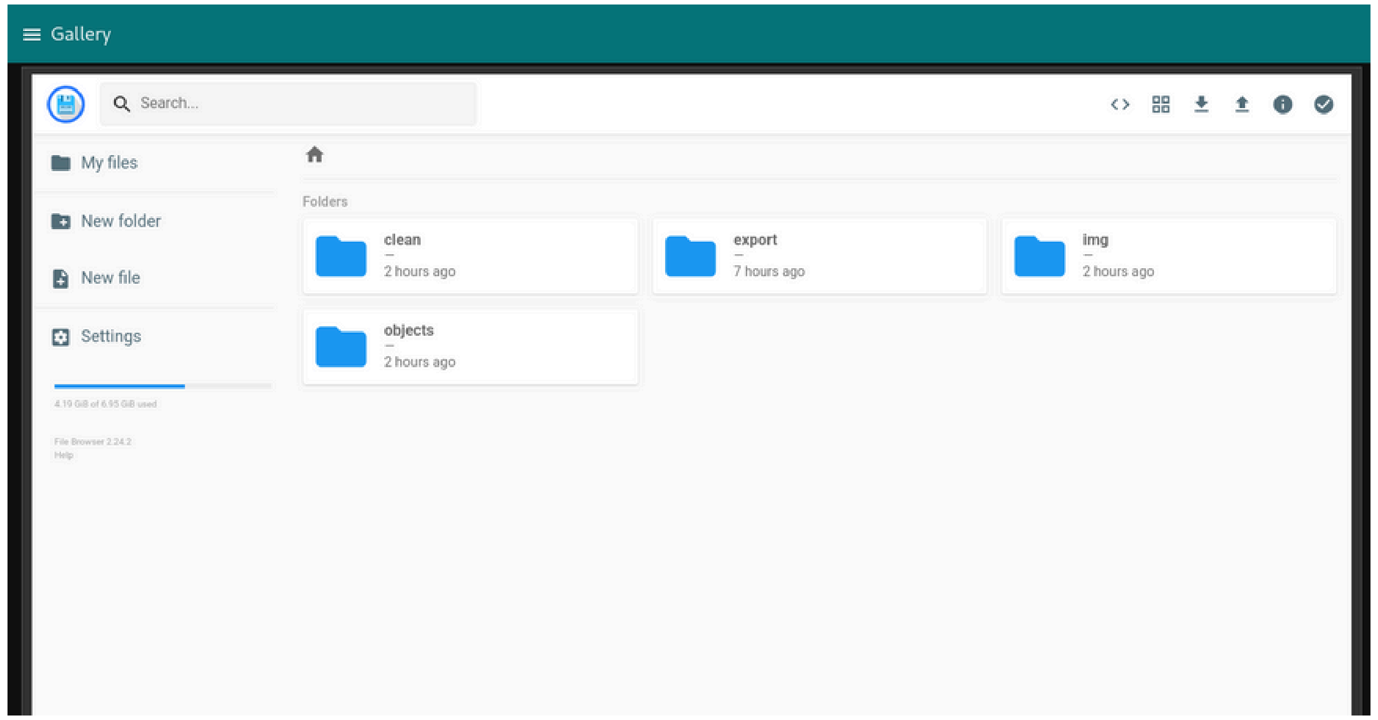






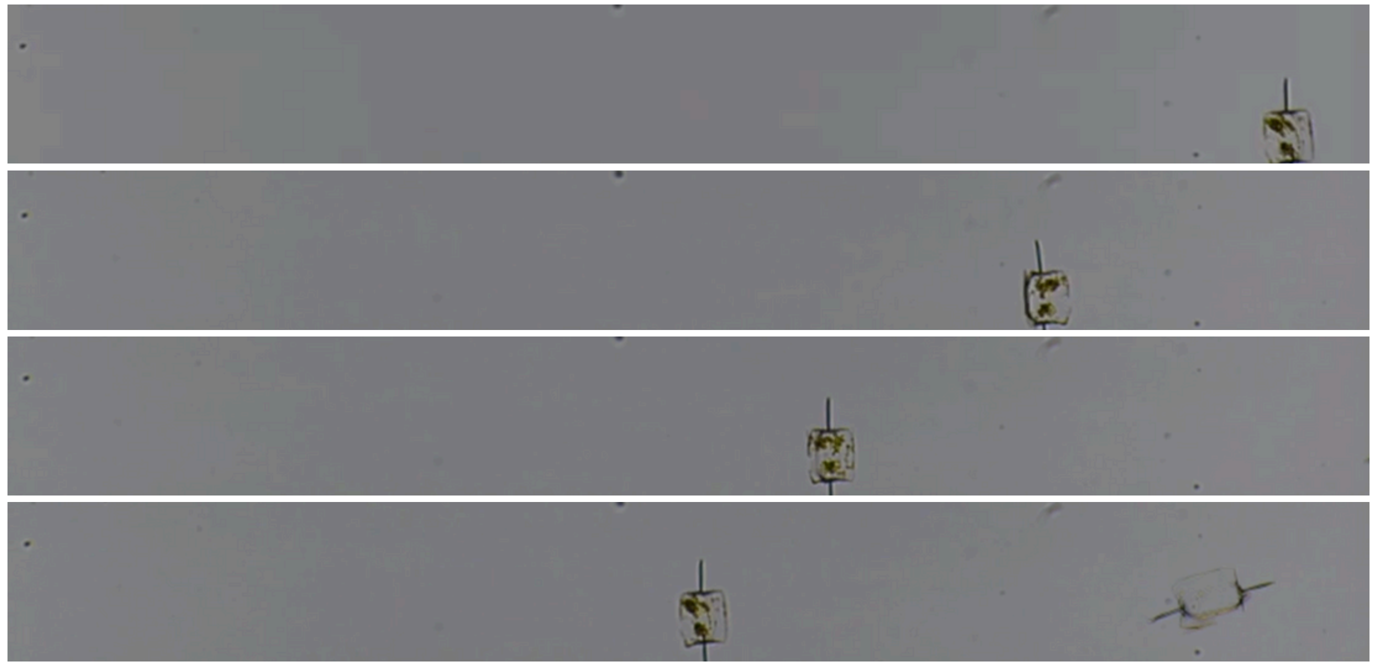
WHY VIDEO?

Planktoscope Processes



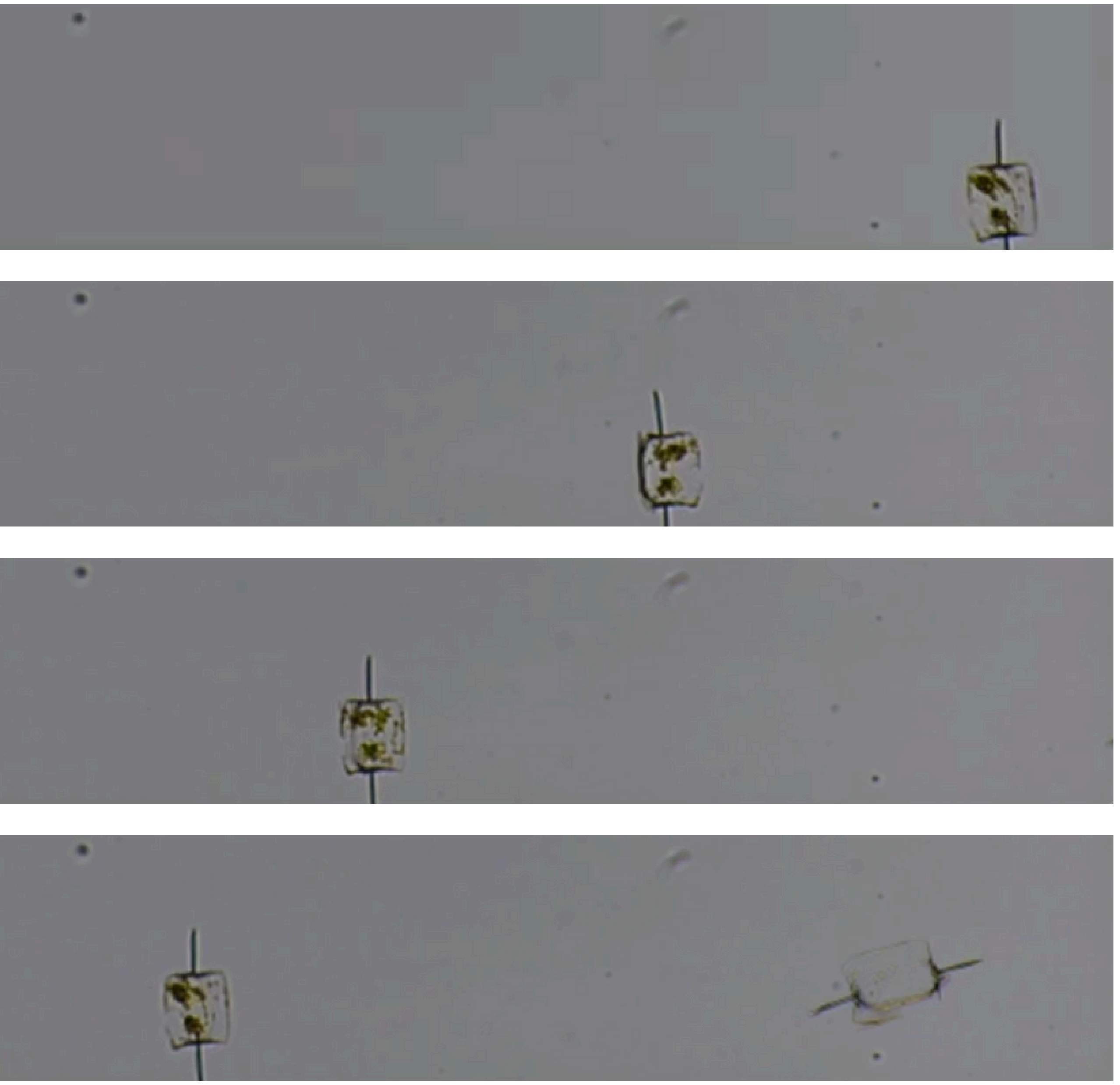
- Images are captured as still frames (no continuity)
- Movement cannot be distinguished
- Limited in detecting subtle moves

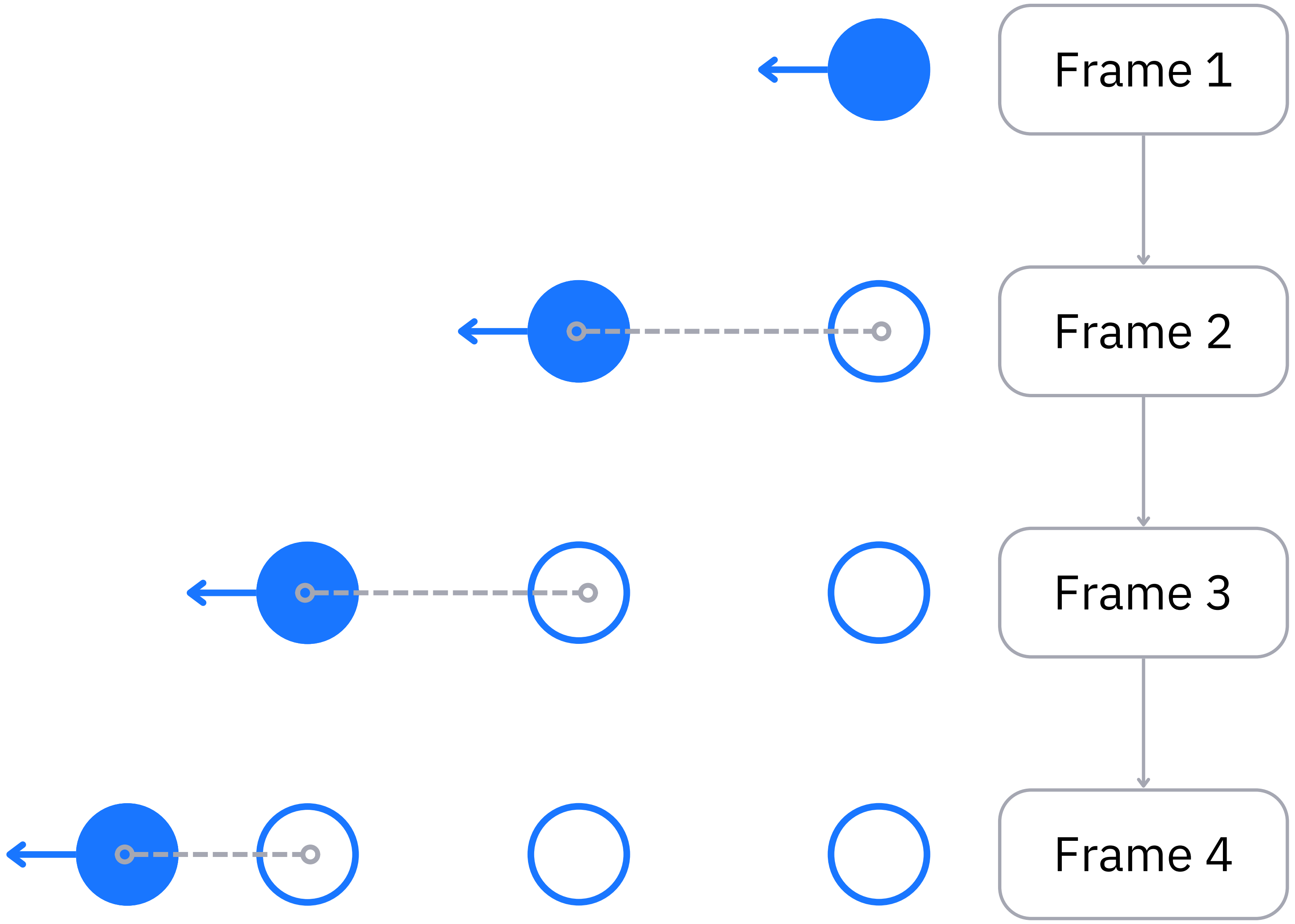
Movie and Movement Method

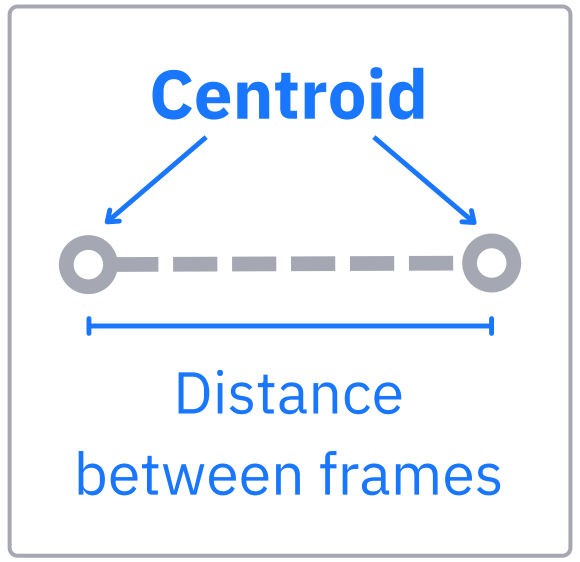


- Uses video to **track motion** across time
- Movement-based filtering
- Captures subtle, real-time **behaviors**
- Flexible: Planktons can rotate, this method captures **multiple angle**

PLANKTON TRACKING ALGORITHM







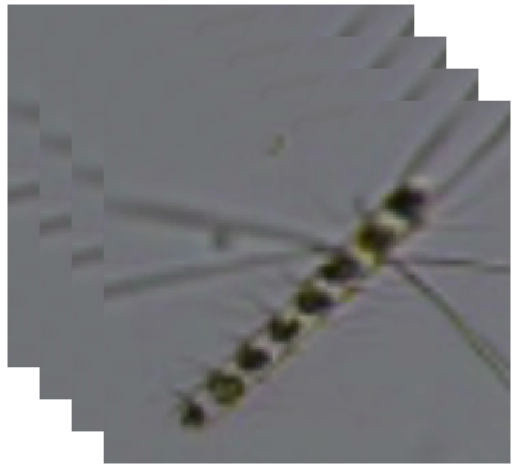
Setting distance threshold to consider one object as the same object

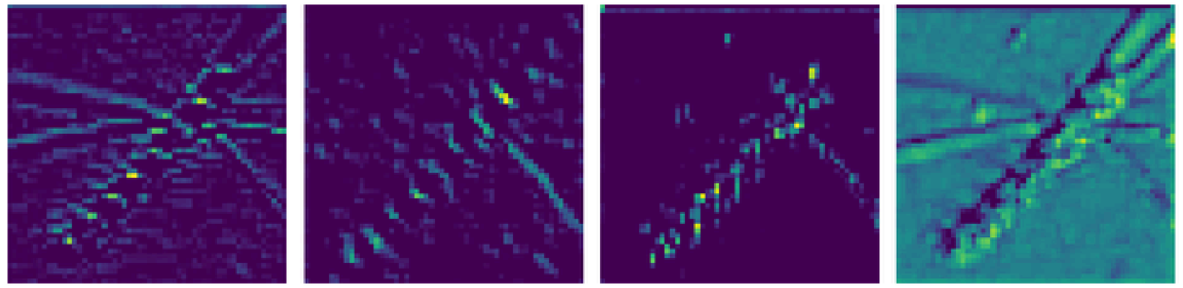
Remark: Plankton speed is a byproduct of this step

GENERATIVE AI AND LARGE LANGUAGE MODEL

Convolutional Neural Network

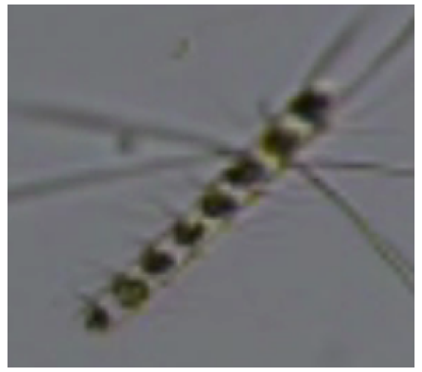
Requires **many labeled images per class** to generalize well, since it learns patterns only from the training dataset and lacks external knowledge or semantic understanding.





Large Language Model + RAG

Requires only **one labeled image**, leveraging prior knowledge from pretraining and external retrieval to classify novel objects with minimal supervision.

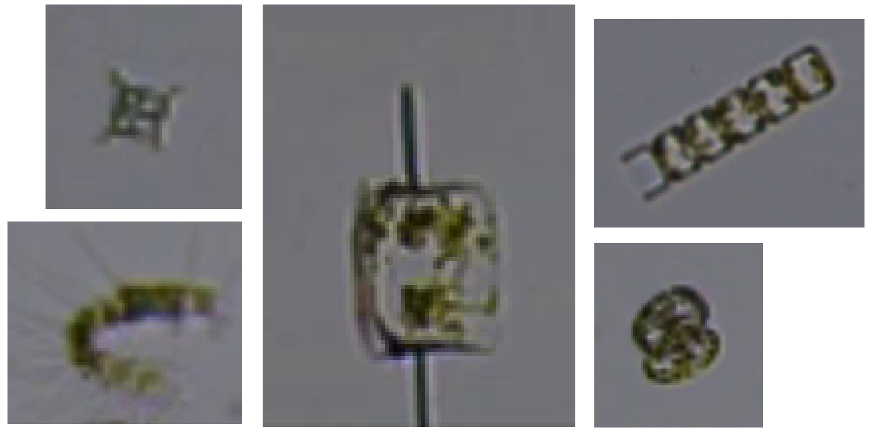


$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & \cdots & n \end{array} \\ 1 & \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \end{bmatrix} \end{array}$$

distance	auto_id	entity
0.838844	458681890305626240	{'family': 'Ceratiaceae', 'genus': 'Ceratium',...

MULTIMODAL LLM IMPLEMENTATION

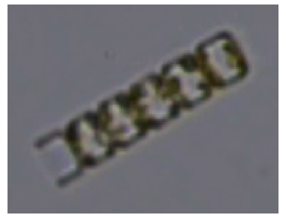
Detection Node Output



Output individual "Plankton" or "Non-Plankton" images

Arbitrary Large Language Model

Inquiry Prompt +



GUI (Graphic User Interface)
User-friendly, Model Testing

API (Application Programming Interface)
App Integration

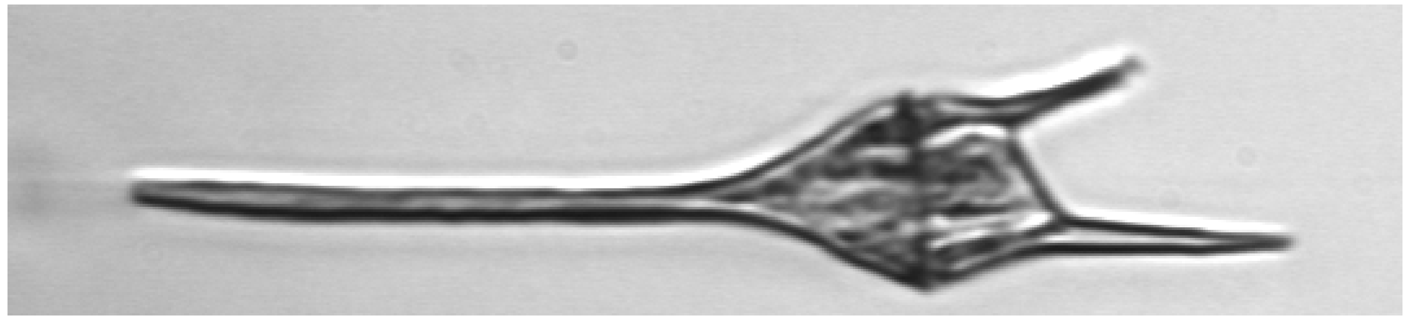
LLM Output

JSON format plankton metadata

Structured, machine-readable, parsable, Integrates with APIs

Input

Inquiry Prompt +



ceratium.png

Output

```
```json
{
  "type": "Ceratium"
}
```
```

Ceratium Furca = Tripos Furca (updated name)

FINE TUNE BY PROMPT ENGINEERING

Structures LLM queries to utilize retrieved metadata and ensures responses are formatted correctly

PERSONA	You are an expert marine biologist with extensive knowledge of plankton taxonomy. You specialize in identifying and classifying various plankton species based on provided image data or textual descriptions.
CONTEXT	You are classifying an organism to determine whether it is plankton. If it is, provide its taxonomic details (family, genus, species). If it is not plankton, return `"none"`. The output must strictly adhere to JSON format for seamless integration into a structured database.
STEPS	**Steps:** 1. Analyze the given image to determine if it belongs to the plankton category. 2. If it is plankton, classify it as accurately as possible and provide the taxonomic details. 3. If it is not plankton, return `"none"` in the response. 4. Ensure the response strictly follows the JSON format.

OUTPUT	**Output Format:**
	{
	"family": "",
	"genus": ""
	}
	If it is **not plankton**, return:
	{
	"genus": "none"
	}

CUE	Classify the given input according to the structured taxonomy above. If the organism is not plankton, return `"none"`. Do not include any additional text or explanation—only the JSON output.

```
image = PIL.Image.open('ceratium.png')

response = client.models.generate_content(
    model="gemini-2.0-flash-thinking-exp-01-21",
    contents=["""You are an expert marine biologist with extensive knowledge of plankton taxono

            You are classifying an organism to determine whether it is plankton. If it is, p

            **Steps:**
            1. Analyze the given data (image, description, or other input) to determine if i
            2. If it is plankton, classify it as accurately as possible and provide the taxo
            3. If it is not plankton, return `"none"` in the response.
            4. Ensure the response strictly follows the JSON format.

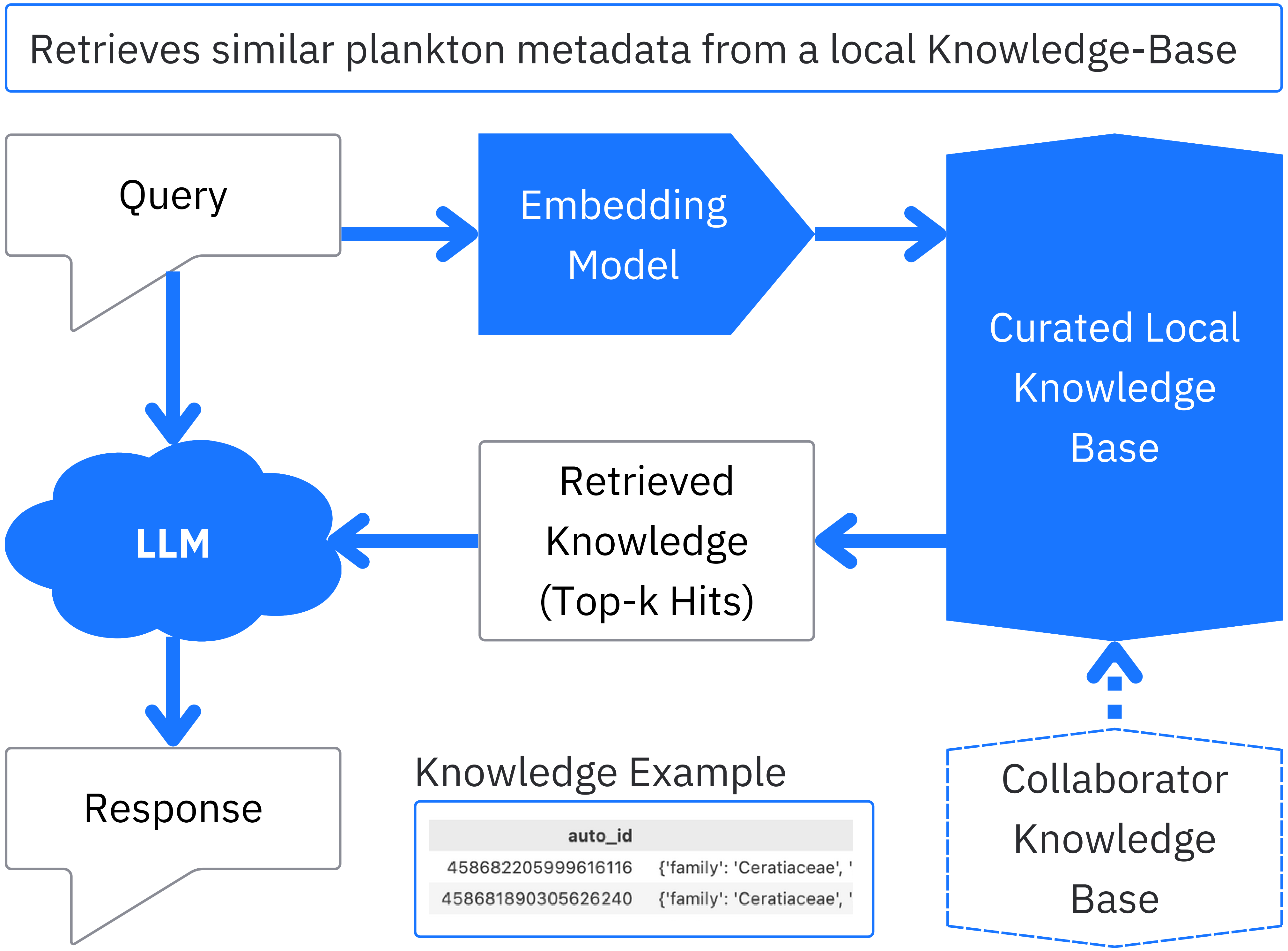
            **Output Format:**
            ```json
            {
              "family": "Calanidae",
              "genus": "Calanus",
              "species": "Calanus finmarchicus"
            }
            ```
            If it is **not plankton**, return:
            ```json
            {
              "classification": "none"
            }
            ```

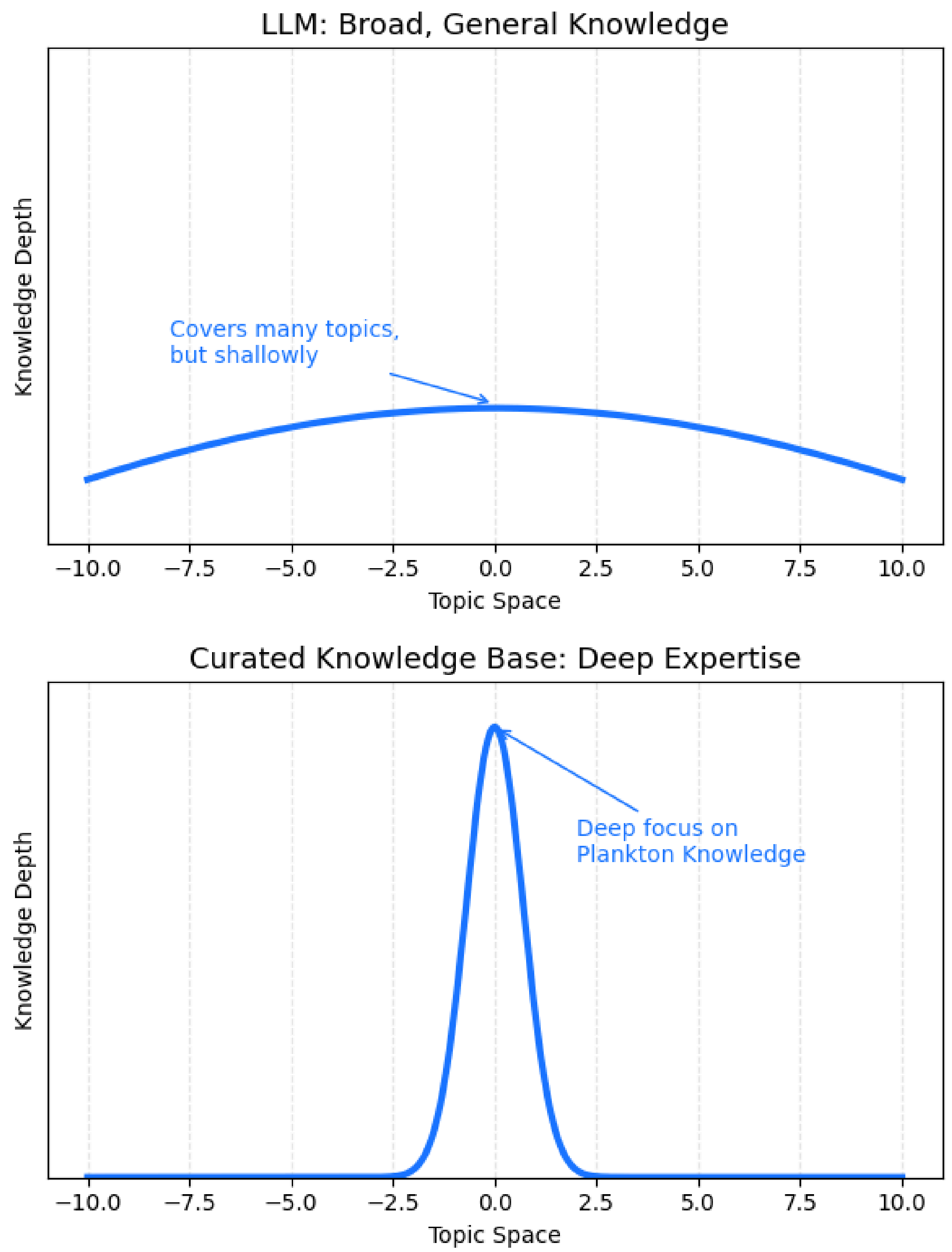
            Classify the given input according to the structured taxonomy above. If the orga
            """,
            image])

print(response.text)
```

[21] ✓ 3.9s

```
```json
{
  "family": "Ceratiaceae",
  "genus": "Ceratium",
  "species": "Ceratium furca"
}
```
```

Correctly formatted JSON output

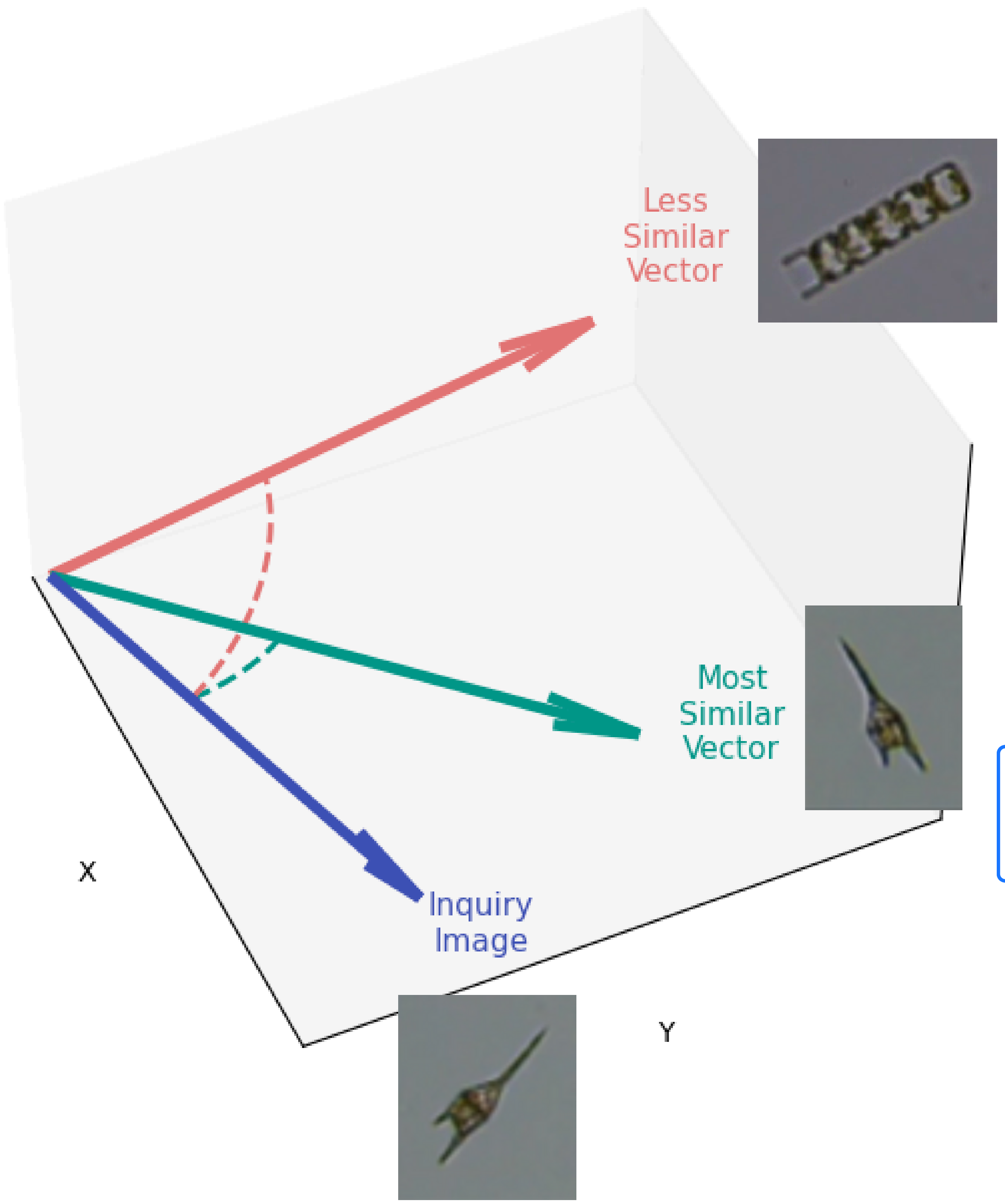
RETRIEVAL - AUGMENTED GENERATION (RAG)





SEMANTIC SEARCHING VIA VECTOR DATABASE

Cosine Similarity in Vector Space



Inquiry Image

Cosine Similarity
0.839

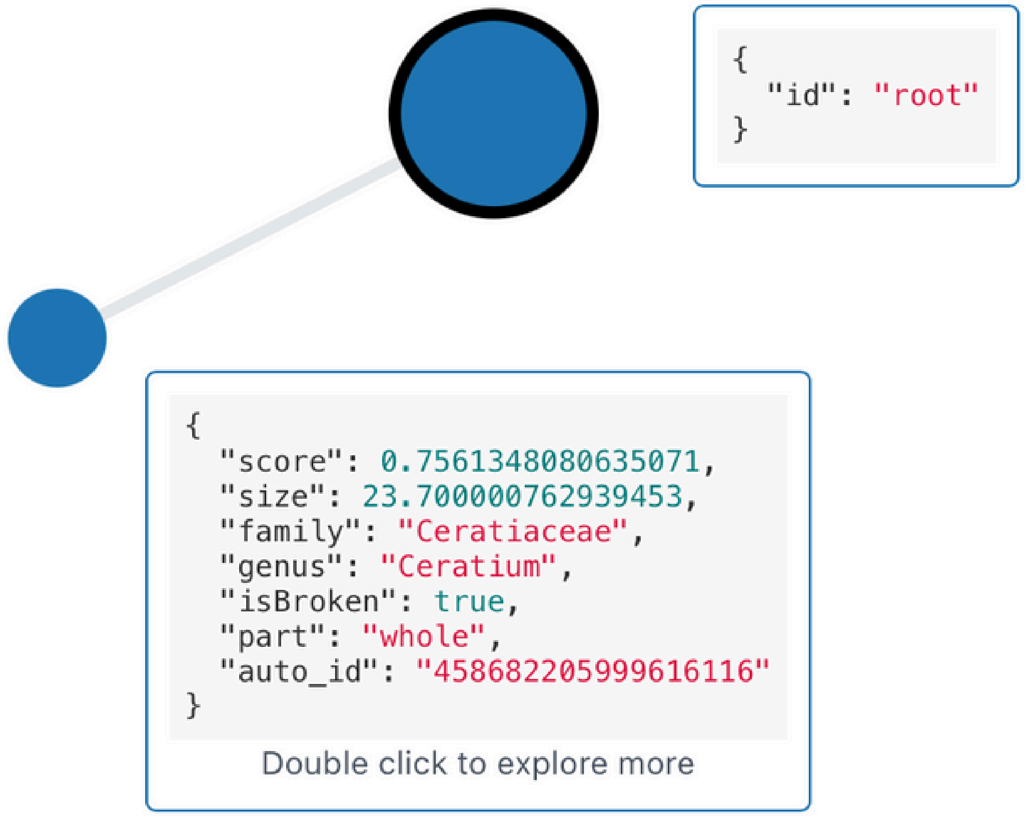
Image from Knowledge-Base

	distance	entity
0	0.838844	{'family': 'Ceratiaceae', 'genus': 'Ceratium',...
1	0.575968	{'family': 'NOISE', 'genus': 'NOISE', 'isBroke...
2	0.574942	{'family': 'NOISE', 'genus': 'NOISE', 'isBroke...
3	0.574366	{'family': 'NOISE', 'genus': 'NOISE', 'isBroke...

BROKEN PLANKTONS HANDLING IN VECTORDB

Test Image

BROKEN



VectorDB Image

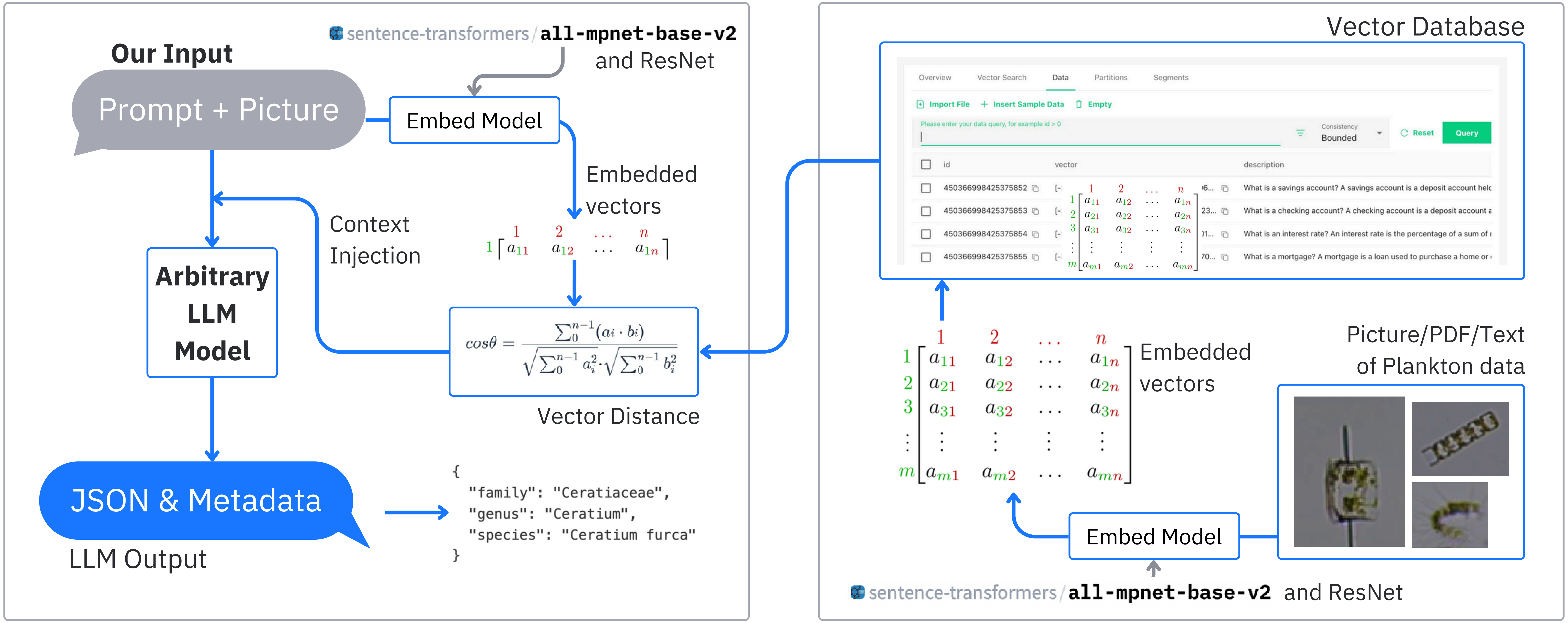
BROKEN

		distance	auto_id	entity
Broken →	0	0.756135	458682205999616116	{'family': 'Ceratiaceae', 'genus': 'Ceratium',...
Not Broken →	1	0.745046	458681890305626240	{'family': 'Ceratiaceae', 'genus': 'Ceratium',...
	2	0.543354	458681890305626201	{'family': 'NOISE', 'genus': 'NOISE', 'isBroke...
	3	0.539729	458681890305626197	{'family': 'NOISE', 'genus': 'NOISE', 'isBroke...

FINE TUNE BY RAG IMPLEMENTATION

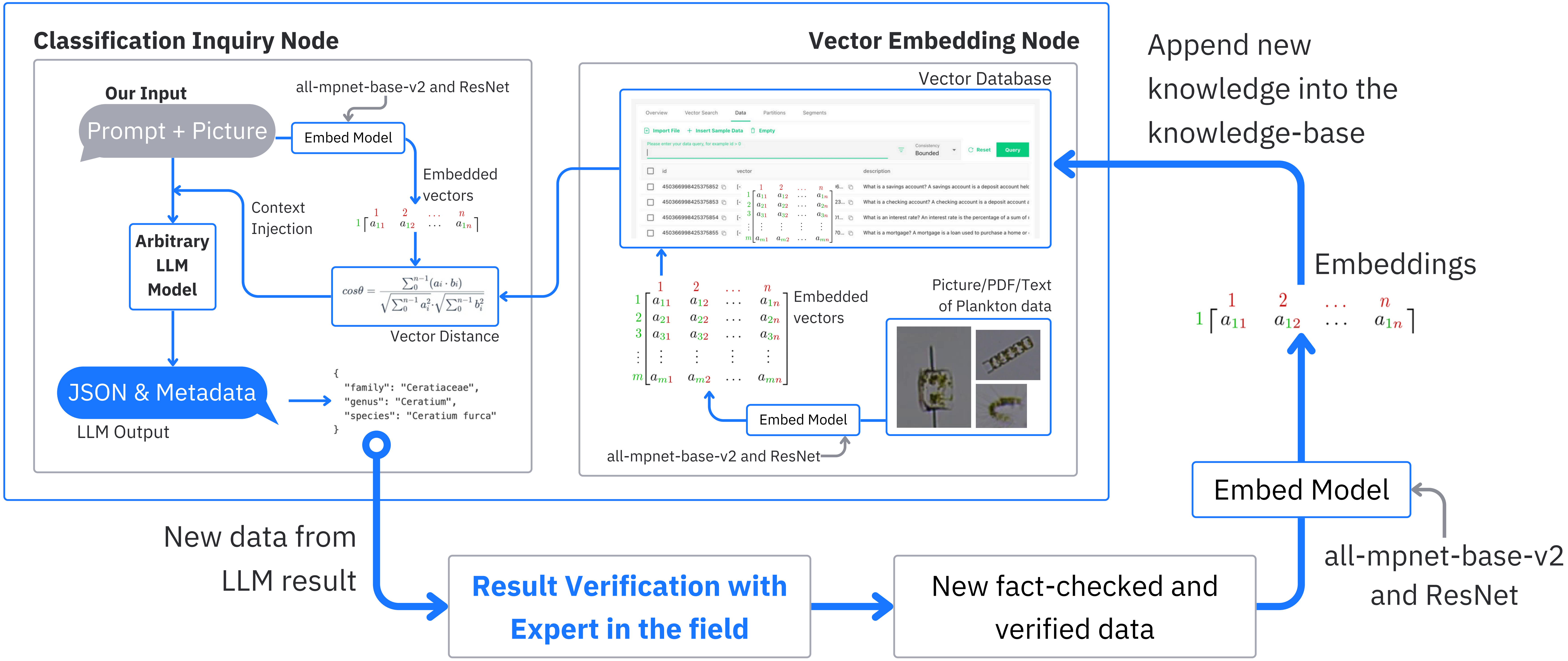
Classification Inquiry Node

Vector Embedding Node

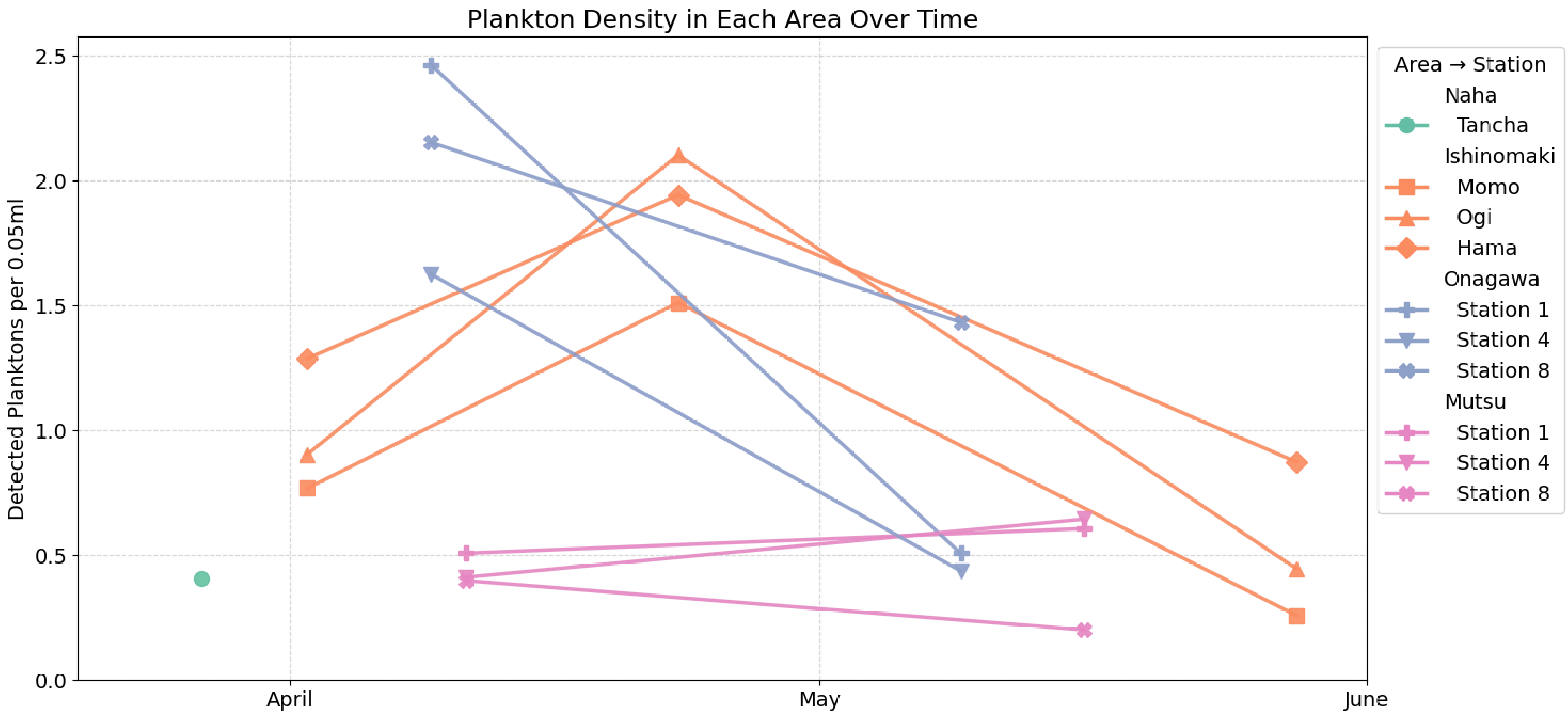


RAG FEEDBACK LOOP IMPLEMENTATION

Existing RAG implemented LLM



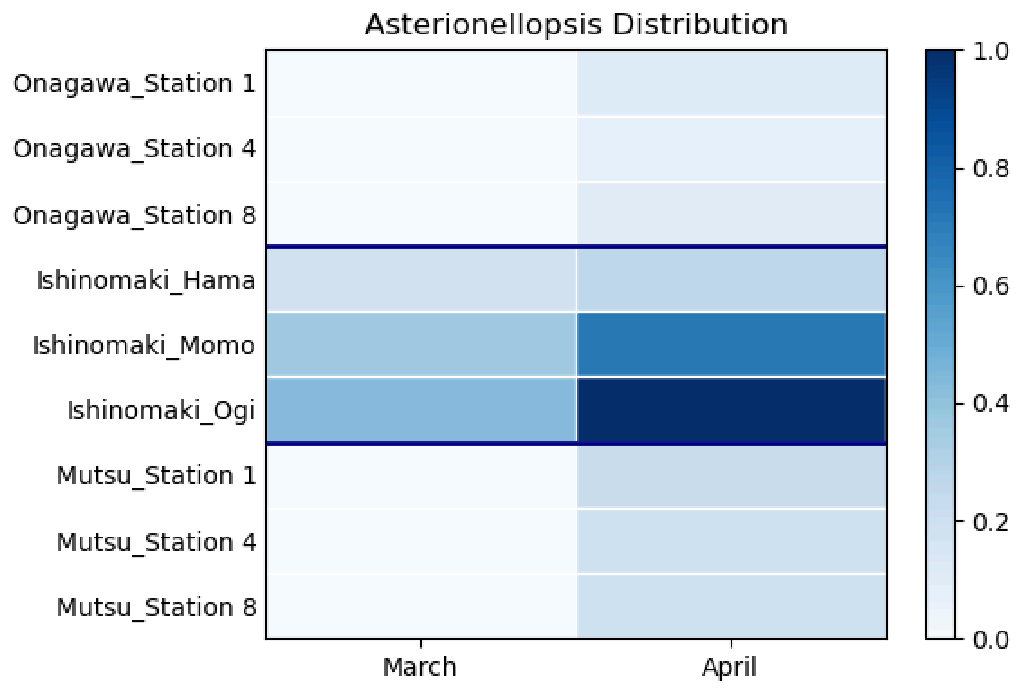
PLANKTON ABUNDANCE FROM MARCH

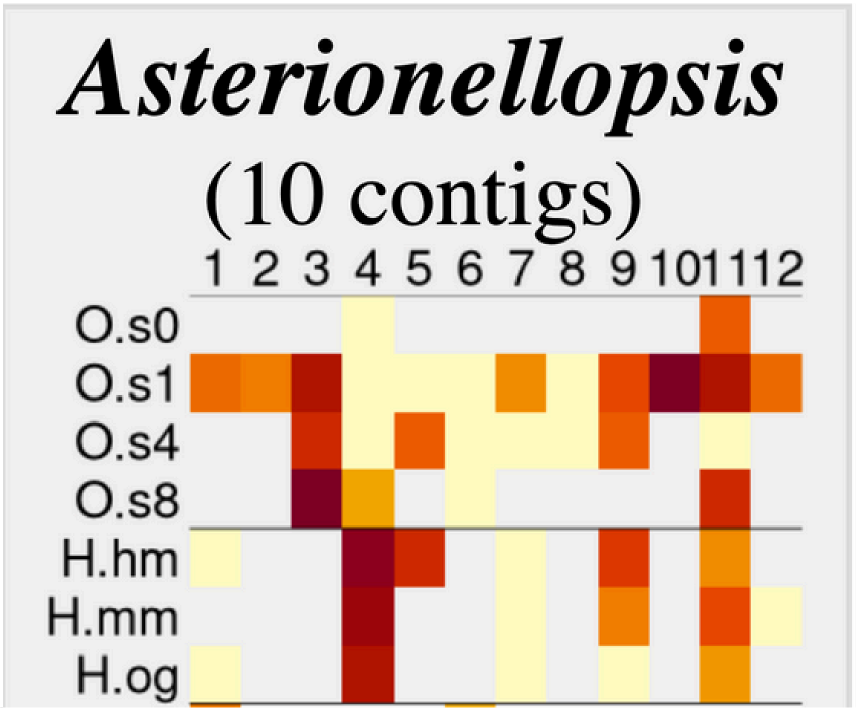


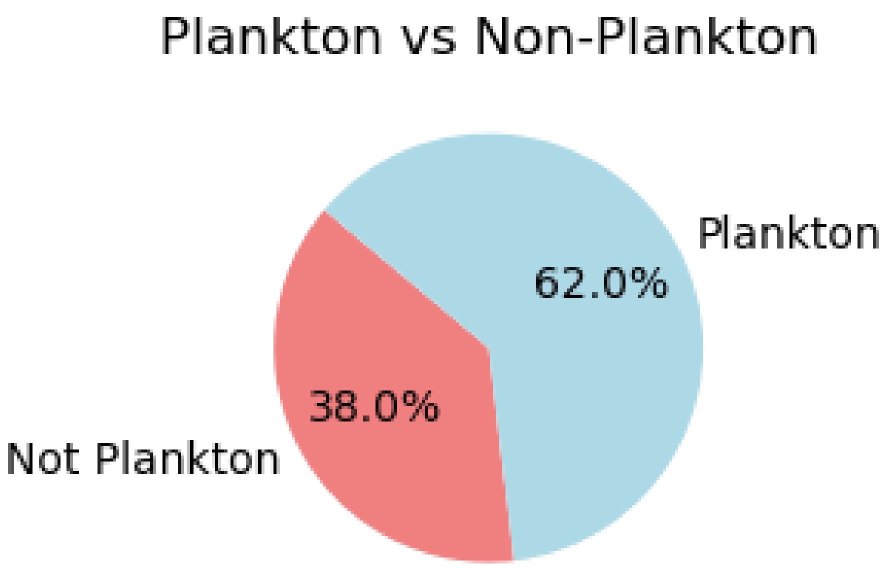
TAXONOMICAL CLASSIFICATIONS

Taxonomy Output

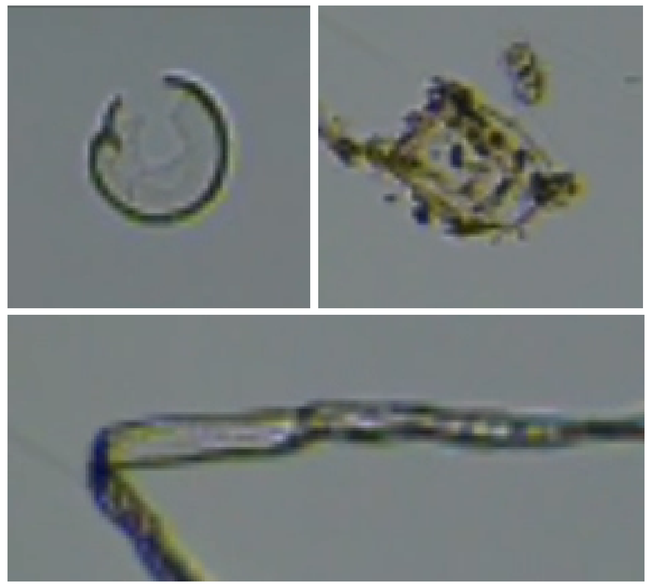
Comparison with PlanDyO Occurrence Data



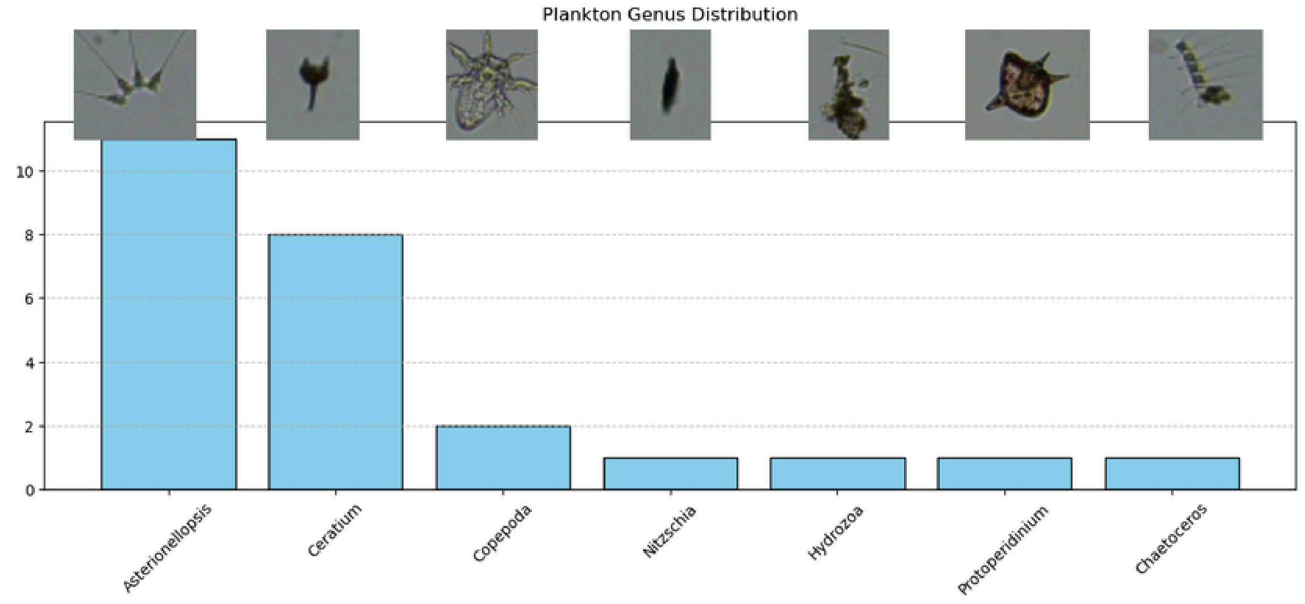




LLM Classification



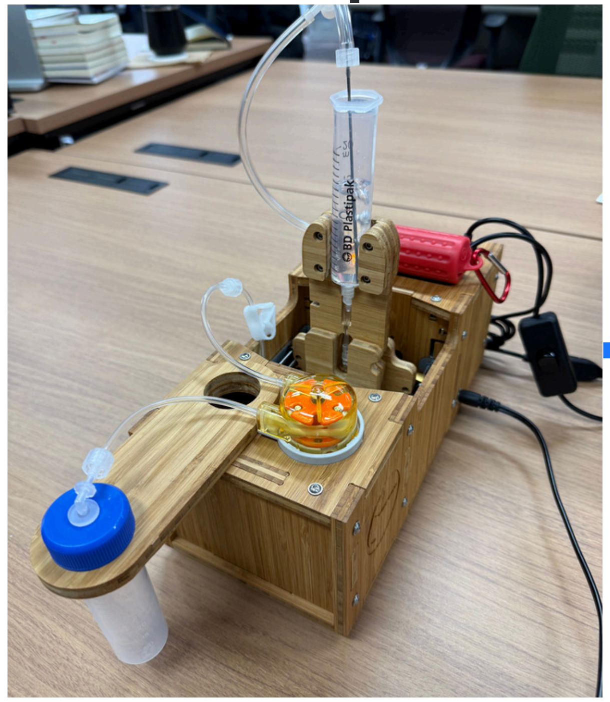
Non-Plankton Examples



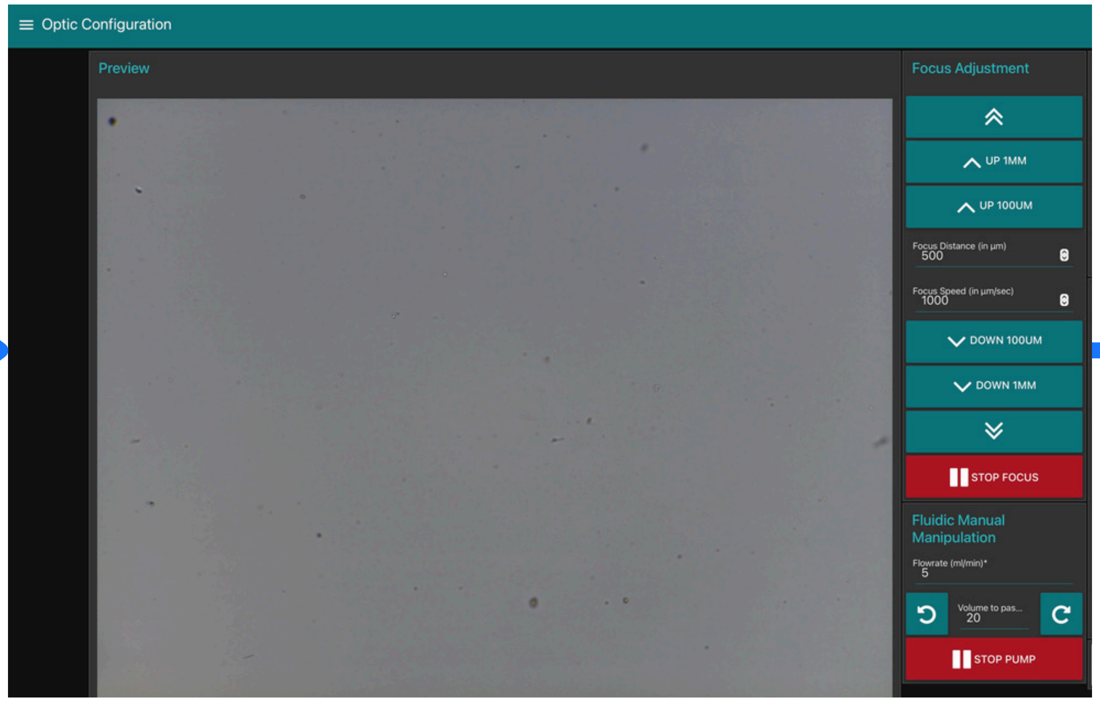
April Mutsu Bay Distribution

AUTOMATED TAXONOMIC PROCESS RECAP

Planktoscope



Plankton Videos



Individual Planktons

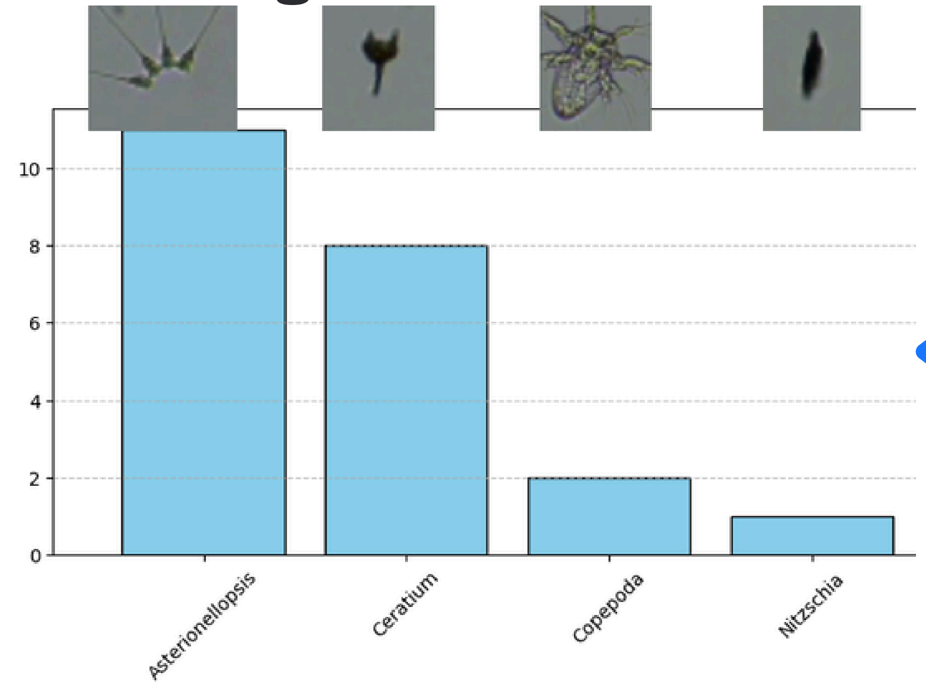
LLM + RAG Classification Framework

Plankton Classifications

{
"family": "Chaetocerotaceae",
"genus": "Chaetoceros"
}

{
"family": "Ditylaceae",
"genus": "Ditylum"
}

Monitoring



Merci beaucoup pour votre attention

Next step for this project

- **Scalable** platform for marine plankton monitoring
- Non-Plankton Analysis in addition

Potential Applications in Related Fields

- Detection and monitoring of Harmful Algae Blooms
- Identify plankton species impacting fish farms/aquaculture

Jaronchai Dilokkalayakul [1] , Akane Kitamura [2] , Takeshi Obayashi [1,2]
[1] Graduate School of Information Sciences (GSIS), Tohoku University
[2] Advanced Institute for Marine Ecosystem Change (WPI-AIMEC), Tohoku University

jaronchai.com

APPENDIX

Appendix

INTRODUCTION TO TERMINOLOGIES

Planktoscope:
Hardware and software for quantitative imaging of plankton samples

Image Processing:
Analyzing, transforming, and optimizing images by modifying pixel values, patterns, and structures to extract meaningful information.

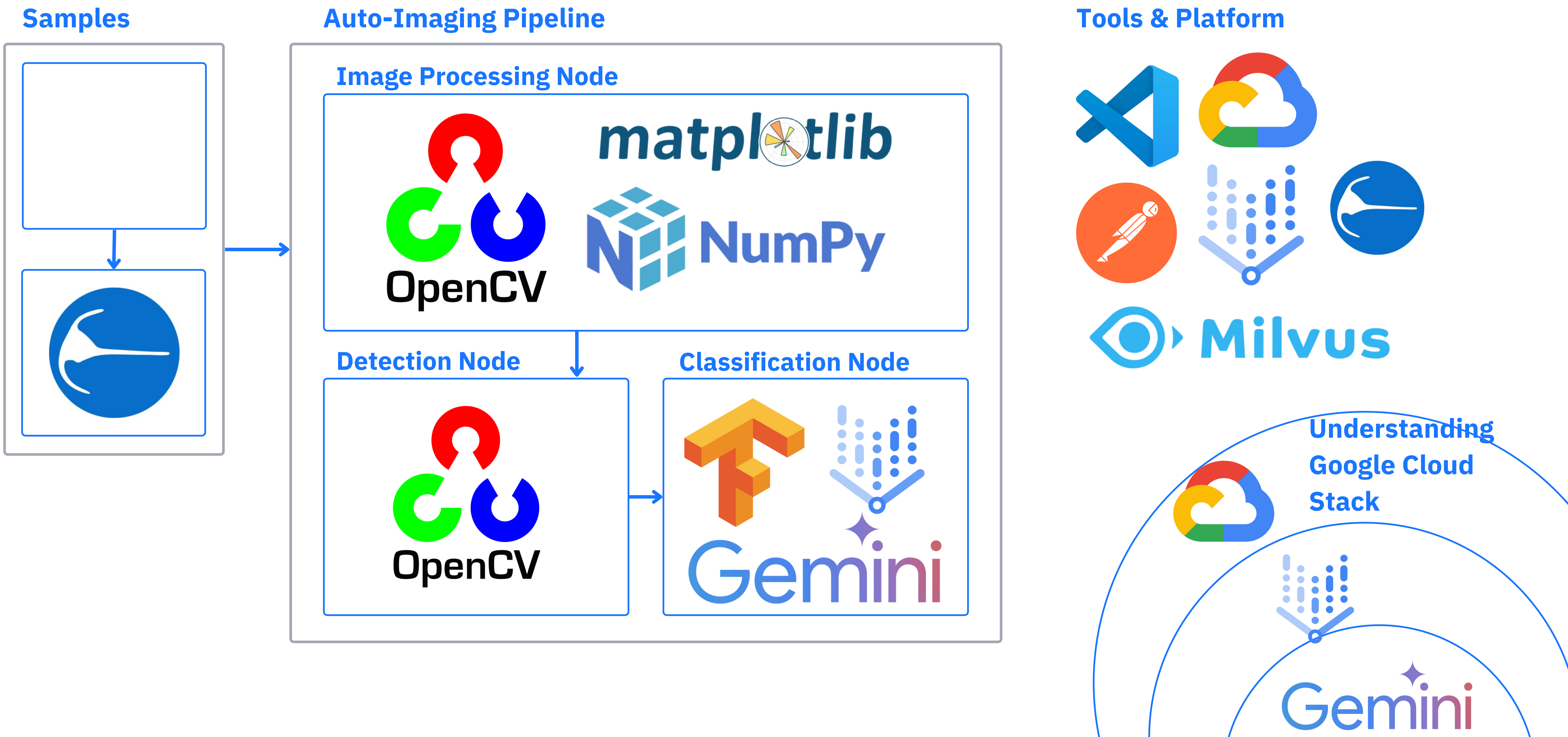
Multimodal LLM (Large Language Model):
A Multimodal LLM processes and **understands multiple data types** (text, images, audio, etc.) to generate context-aware responses of desired format.

Vector Embedding:
Vector embedding **converts data (text, images) into numerical vectors**, allow similarity searches in high-dimensional space for data retrieval

Semantic Similarity
Measures how similar meanings of words, phrases, or texts are based on context, often using embeddings or language models to quantify closeness.

Appendix

TECHNOLOGY STACK



Appendix

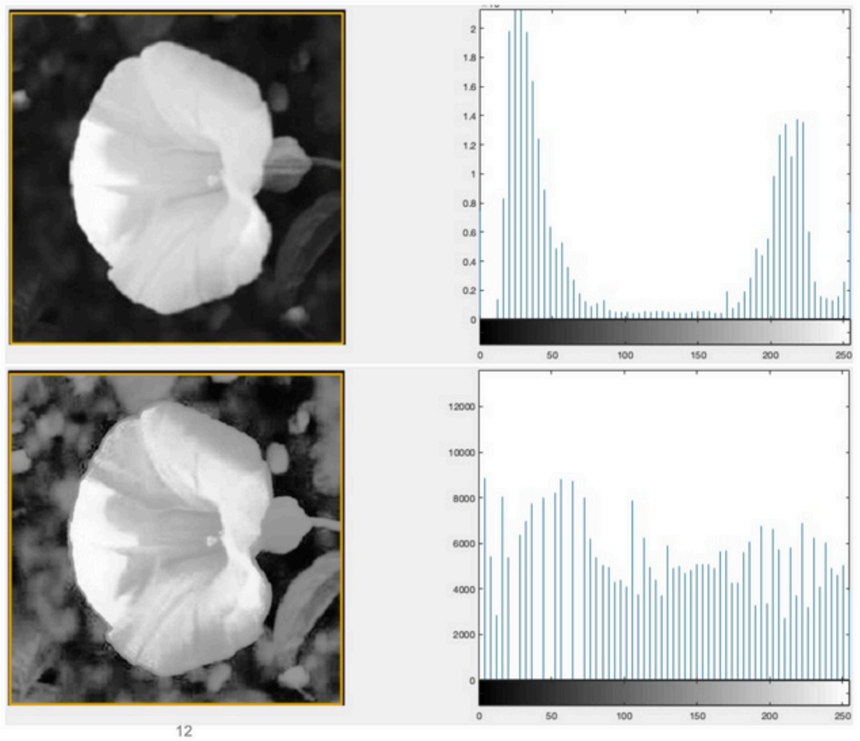
HISTOGRAM EQUALIZATION AND CLAHE

Histogram Equalization

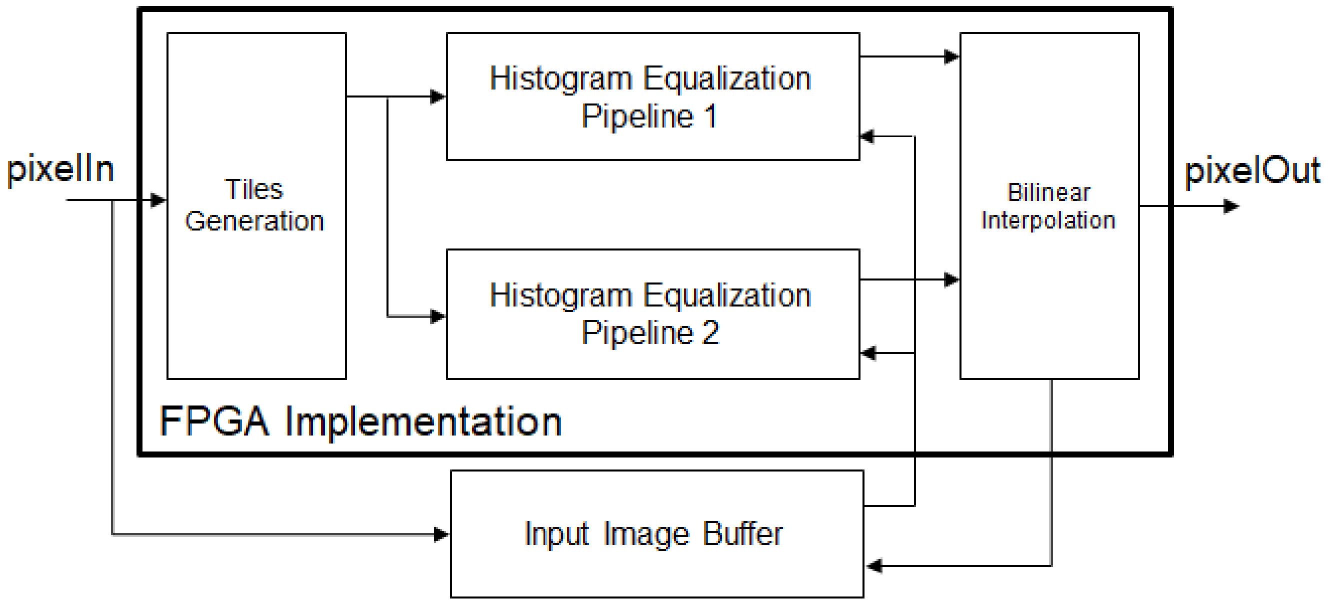
- A histogram processing method to adjust the contrast of the image to have an evenly distributed intensity throughout the range.

Original

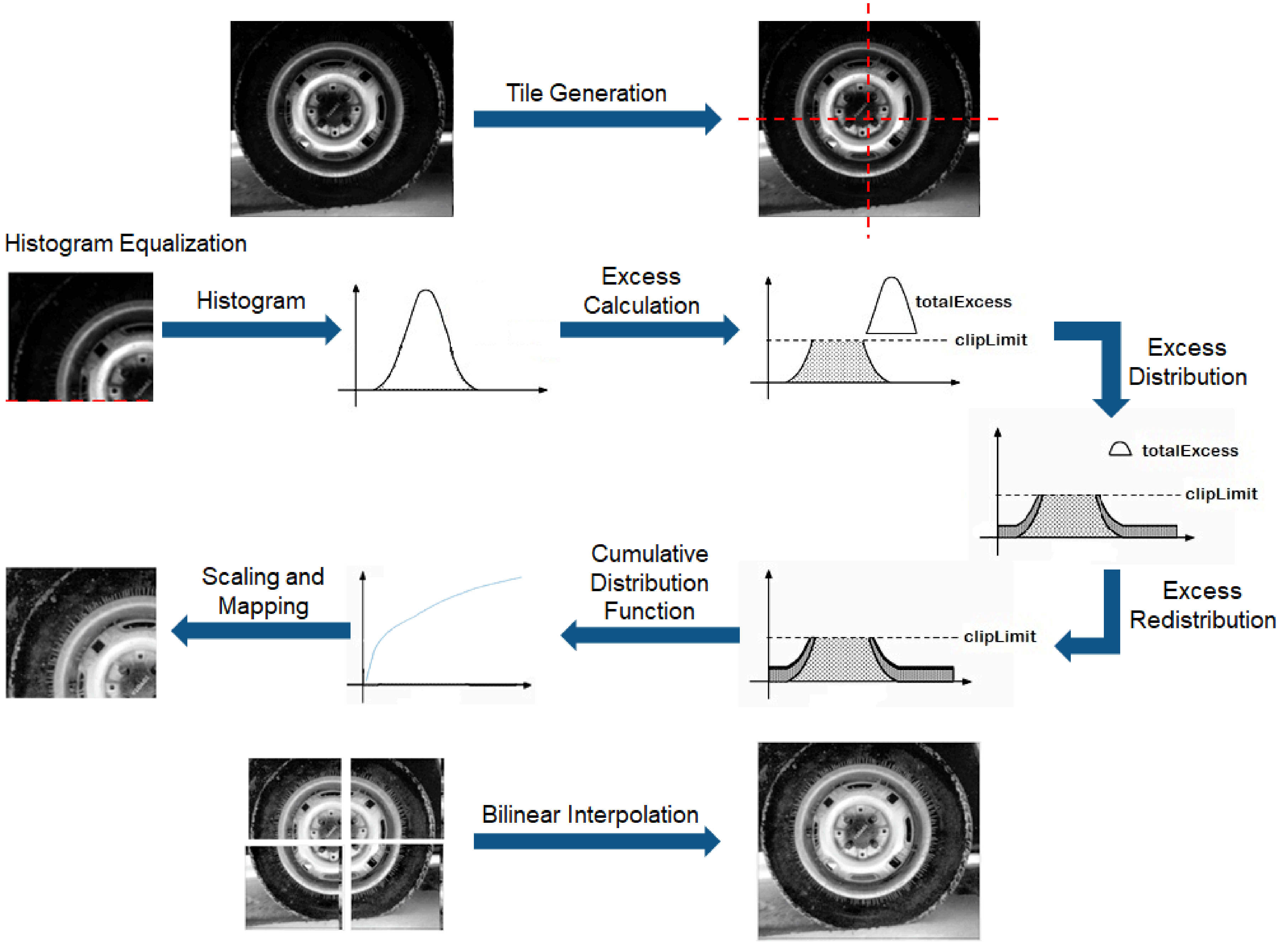
Histogram Equalization



Gonzalez, R. C. (2018). Digital image processing (4th ed.). Pearson.



Contrast Limited Adaptive Histogram Equalization



MathWorks. (n.d.). Contrast Limited Adaptive Histogram Equalization. Retrieved February 18, 2025, from https://www.mathworks.com/help/visionhdl/ug/contrast-adaptive-histogram-equalization.html

Appendix

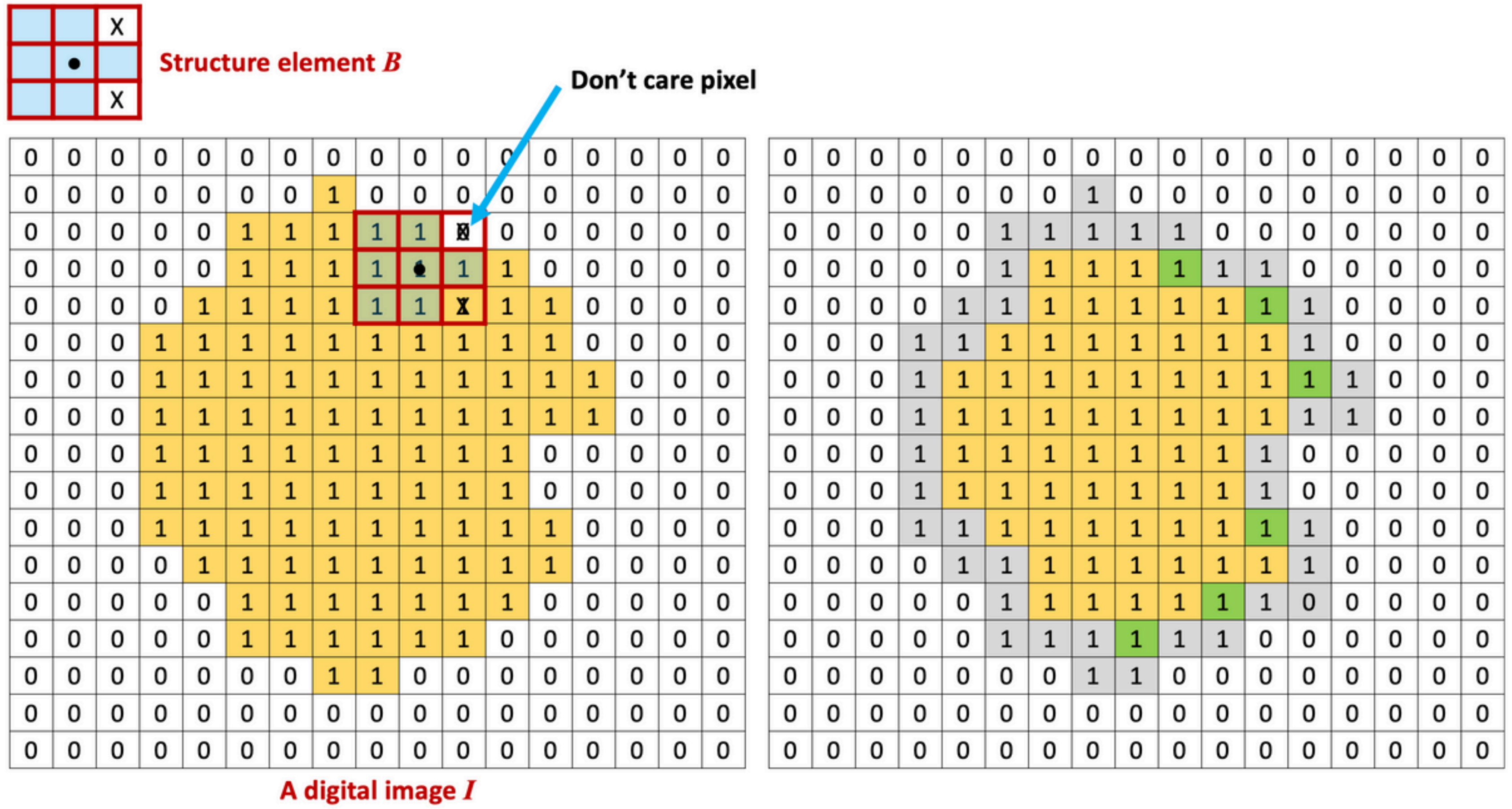
MORPHOLOGICAL OPERATORS ON BINARY IMAGE

Erosion

- An **operator** in the area of mathematical morphology.

$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

  - The set of all points z such that B is contained in A



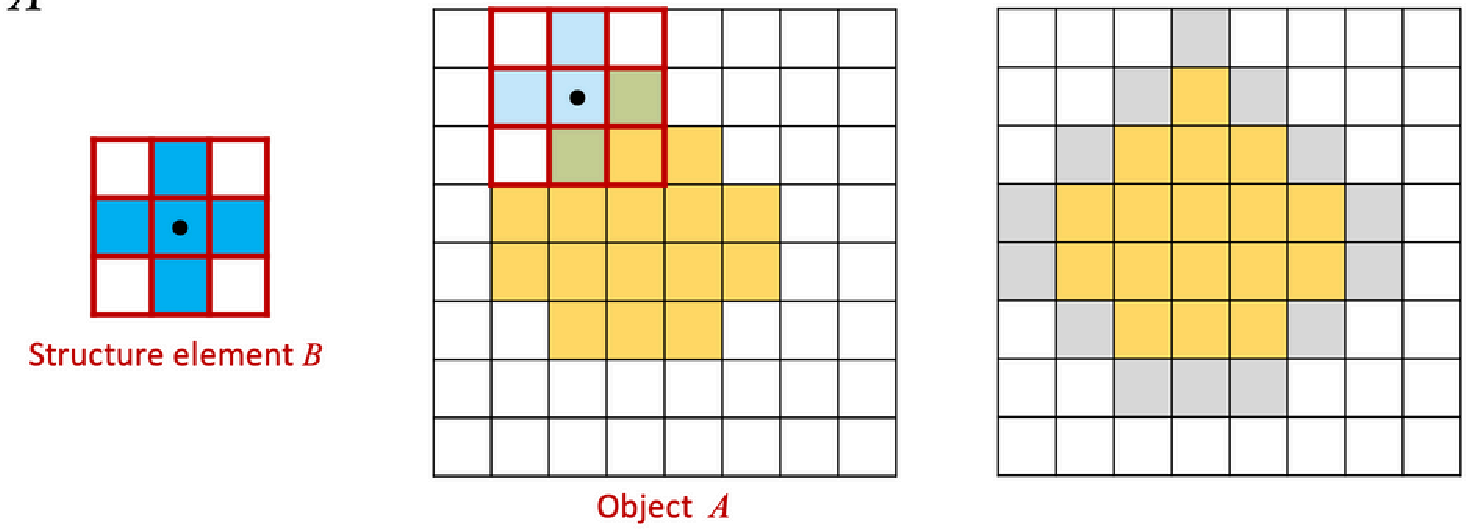
Gonzalez, R. C. (2018). Digital image processing (4th ed.). Pearson.

Dilation

- An **operator** in the area of mathematical morphology.

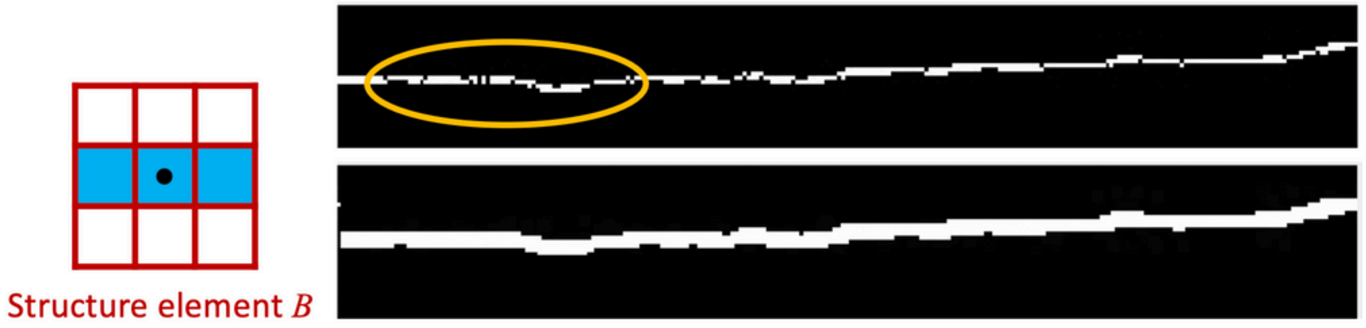
$$A \oplus B = \{z \mid (B)_z \cap A \neq \emptyset\}$$

  - The set of all points z such that B is overlapped at least one element of A



Erosion

Dilation



Appendix

ABOUT MORPHOLOGICAL CLOSING

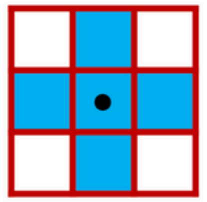
$$A \bullet B = (A \oplus B) \ominus B$$

Dilation: $\oplus$ — Erosion: $\ominus$

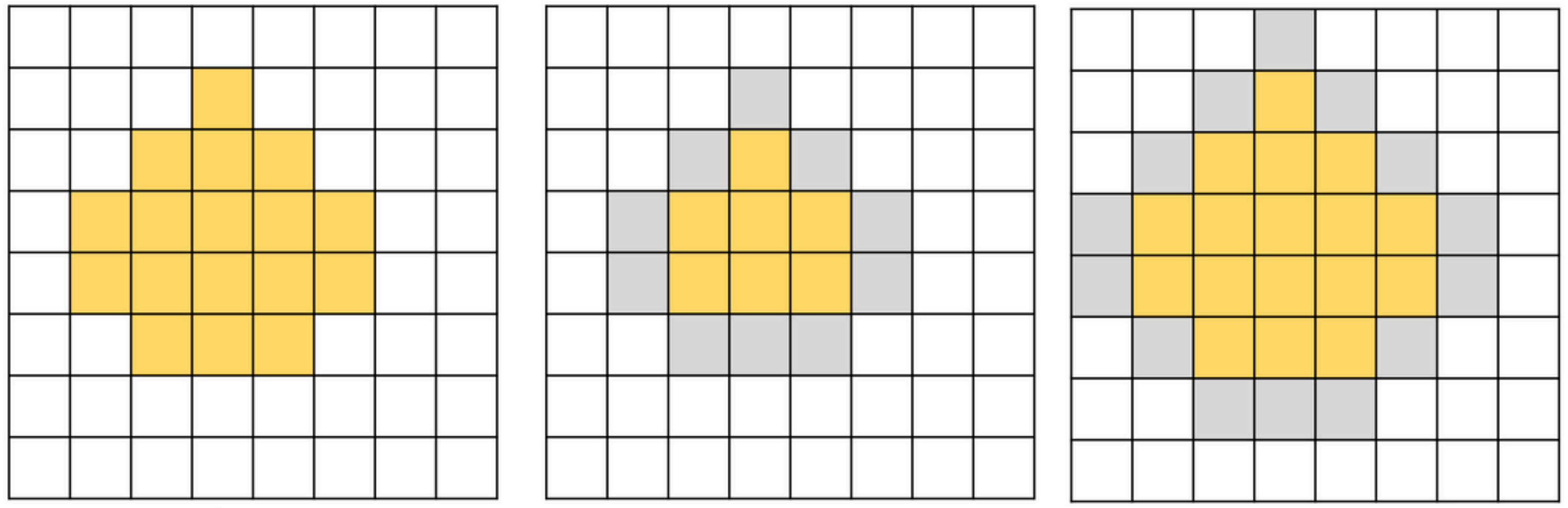
0	0	1	1	1	1	1	0	0
0	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	0	0

Structure element B

Object A — $A \oplus B$ — $(A \oplus B) \ominus B$

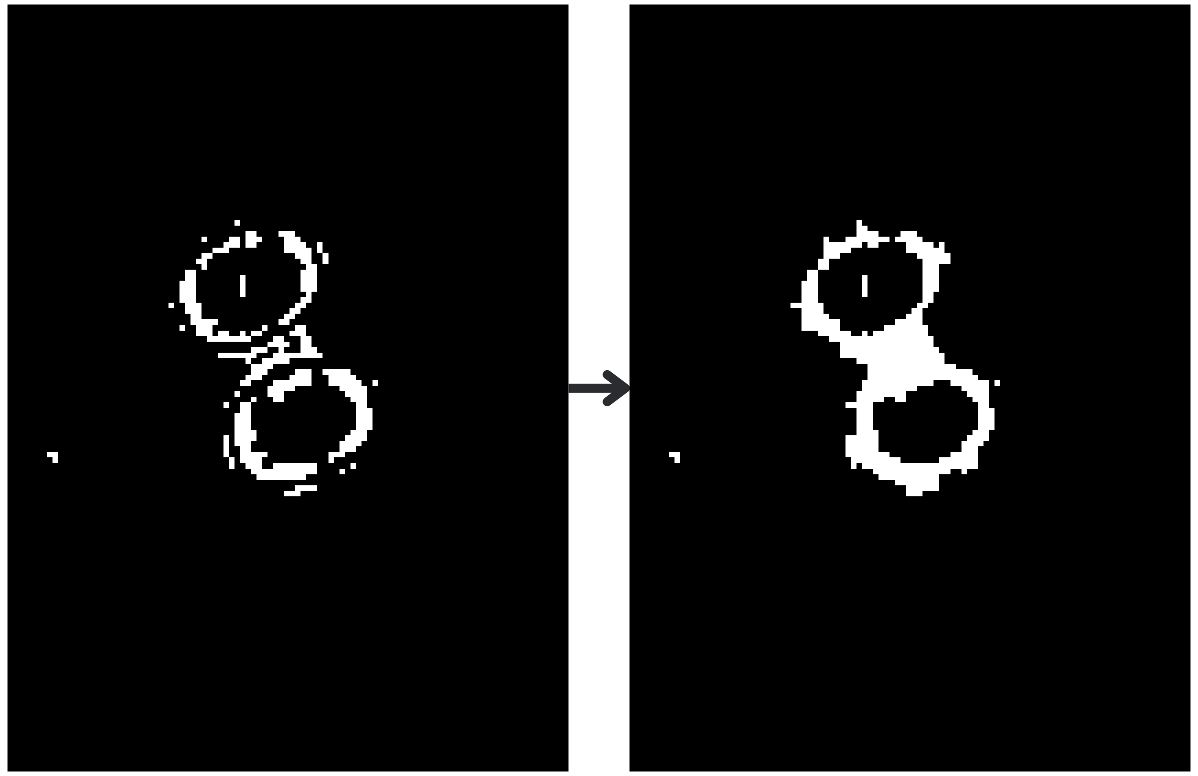


Structure element B



Object A — Result of erosion — Result of dilation

Actual Image from the Pipeline



Result of Binary Threshold → Apply Closing Method

Figures: *Gonzalez, R. C. (2018). Digital image processing (4th ed.). Pearson.*

Appendix

MODEL PROVIDER SELECTION

Vertex AI's Gemini

Token-based pricing | Modality-based pricing

Model	Type	Price	Price with Batch API
Gemini 2.0 Flash	1M Input tokens	$0.15	$0.075
	1M Input audio tokens	$1.00	$0.50
	1M Output text tokens	$0.60	$0.30
Gemini 2.0 Flash Lite	1M Input tokens	$0.075	$0.0375
	1M Input audio tokens	$0.075	$0.0375
	1M Output text tokens	$0.30	$0.15

	Free Tier	Paid Tier, per 1M tokens in USD
Input price	Free of charge	$0.10 (text / image / video) $0.70 (audio)
Output price	Free of charge	$0.40
Context caching price	Free of charge	$0.025 / 1,000,000 tokens (text/image/video) $0.175 / 1,000,000 tokens (audio) Available February 24, 2025
Context caching (storage)	Free of charge, up to 1,000,000 tokens of storage per hour Available February 24, 2025	$1.00 / 1,000,000 tokens per hour Available February 24, 2025
Tuning price	Not available	Not available
Grounding with Google Search	Free of charge, up to 500 RPD	1,500 RPD (free), then $35 / 1,000 requests
Used to improve our products	Yes	No

OpenAI's GPT

Text tokens

Price per 1M tokens · Batch API price

Model	Input	Cached input	Output
gpt-4o ↳ gpt-4o-2024-08-06	$2.50	$1.25	$10.00
gpt-4o-audio-preview ↳ gpt-4o-audio-preview-2024-12-17	$2.50	-	$10.00
gpt-4o-realtime-preview ↳ gpt-4o-realtime-preview-2024-12-17	$5.00	$2.50	$20.00
gpt-4o-mini ↳ gpt-4o-mini-2024-07-18	$0.15	$0.075	$0.60
gpt-4o-mini-audio-preview ↳ gpt-4o-mini-audio-preview-2024-12-17	$0.15	-	$0.60
gpt-4o-mini-realtime-preview ↳ gpt-4o-mini-realtime-preview-2024-12-17	$0.60	$0.30	$2.40
o1 ↳ o1-2024-12-17	$15.00	$7.50	$60.00
o3-mini ↳ o3-mini-2025-01-31	$1.10	$0.55	$4.40
o1-mini ↳ o1-mini-2024-09-12	$1.10	$0.55	$4.40

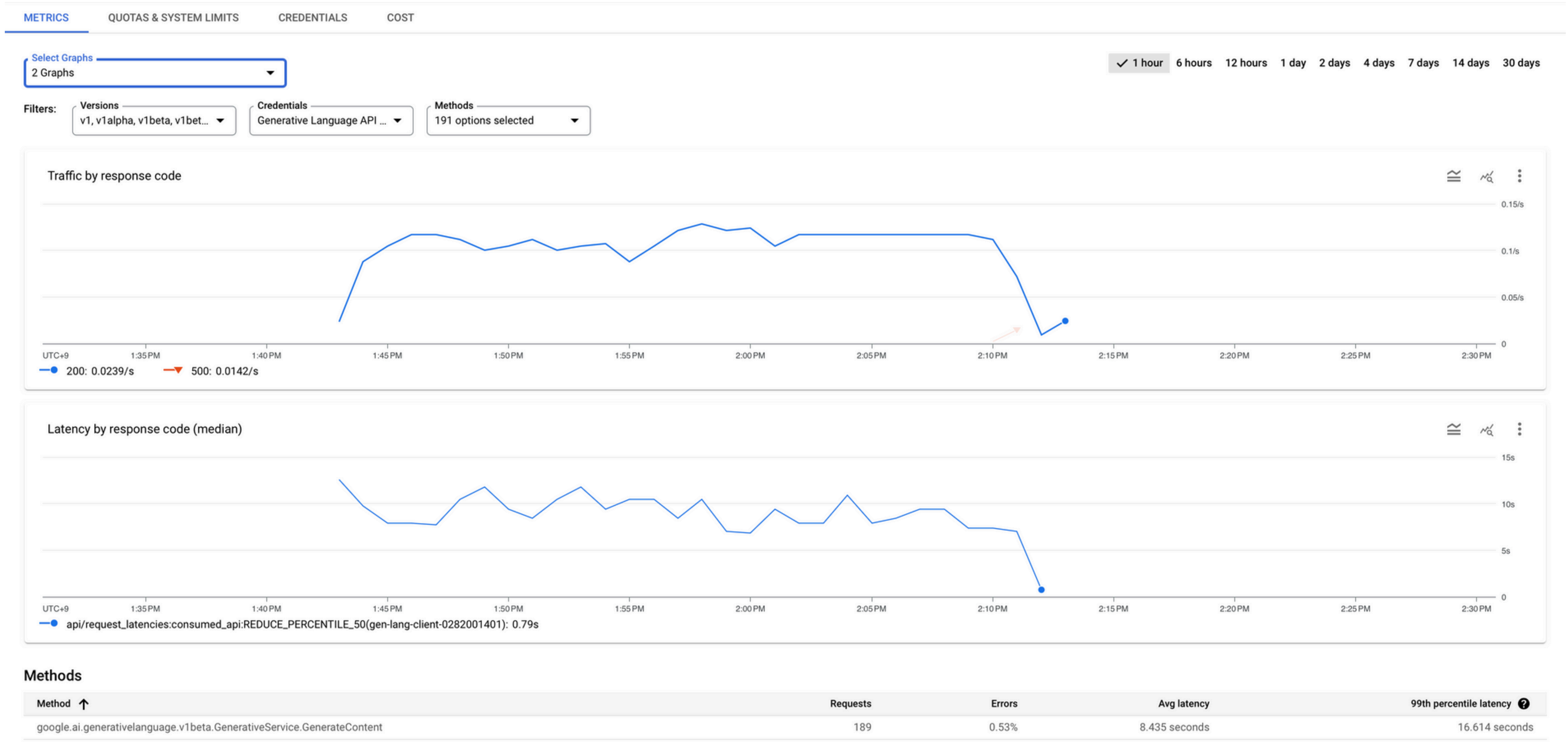
https://platform.openai.com/docs/pricing
https://ai.google.dev/gemini-api/docs/pricing

Appendix

MODEL SELECTION AND USAGE QUOTA LIMIT

Model Selection

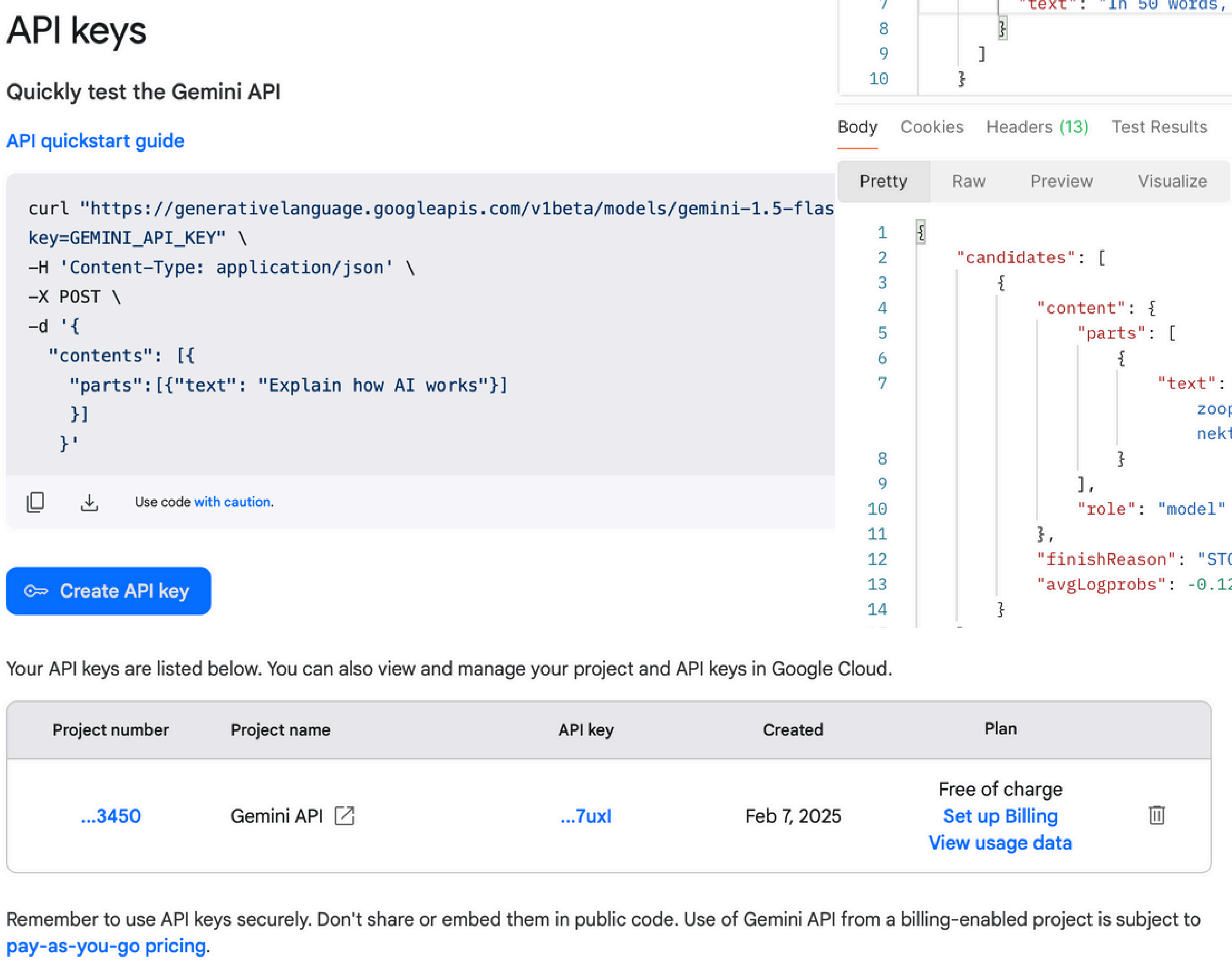
Model	Inputs	Outputs	Use case
Gemini 2.0 Flash `gemini-2.0-flash-001`	Text, Code, Images, Audio, Video, Video with Audio, PDF	Text, Audio (private preview), Images (private preview)	Workhorse model for all daily tasks. Strong overall performance and supports real-time streaming Live API.
Gemini 2.0 Pro `gemini-2.0-pro-exp-02-05`	Text, Images, Video, Audio, PDF	Text	Strongest model quality, especially for code & world knowledge; 2M long context.
Gemini 2.0 Flash-Lite `gemini-2.0-flash-lite-preview-02-05`	Text, Images, Video, Audio, PDF	Text	Our cost effective offering to support high throughput.
Gemini 2.0 Flash Thinking `gemini-2.0-flash-thinking-exp-01-21`	Text, Images	Text	Provides stronger reasoning capabilities and includes the thinking process in responses.

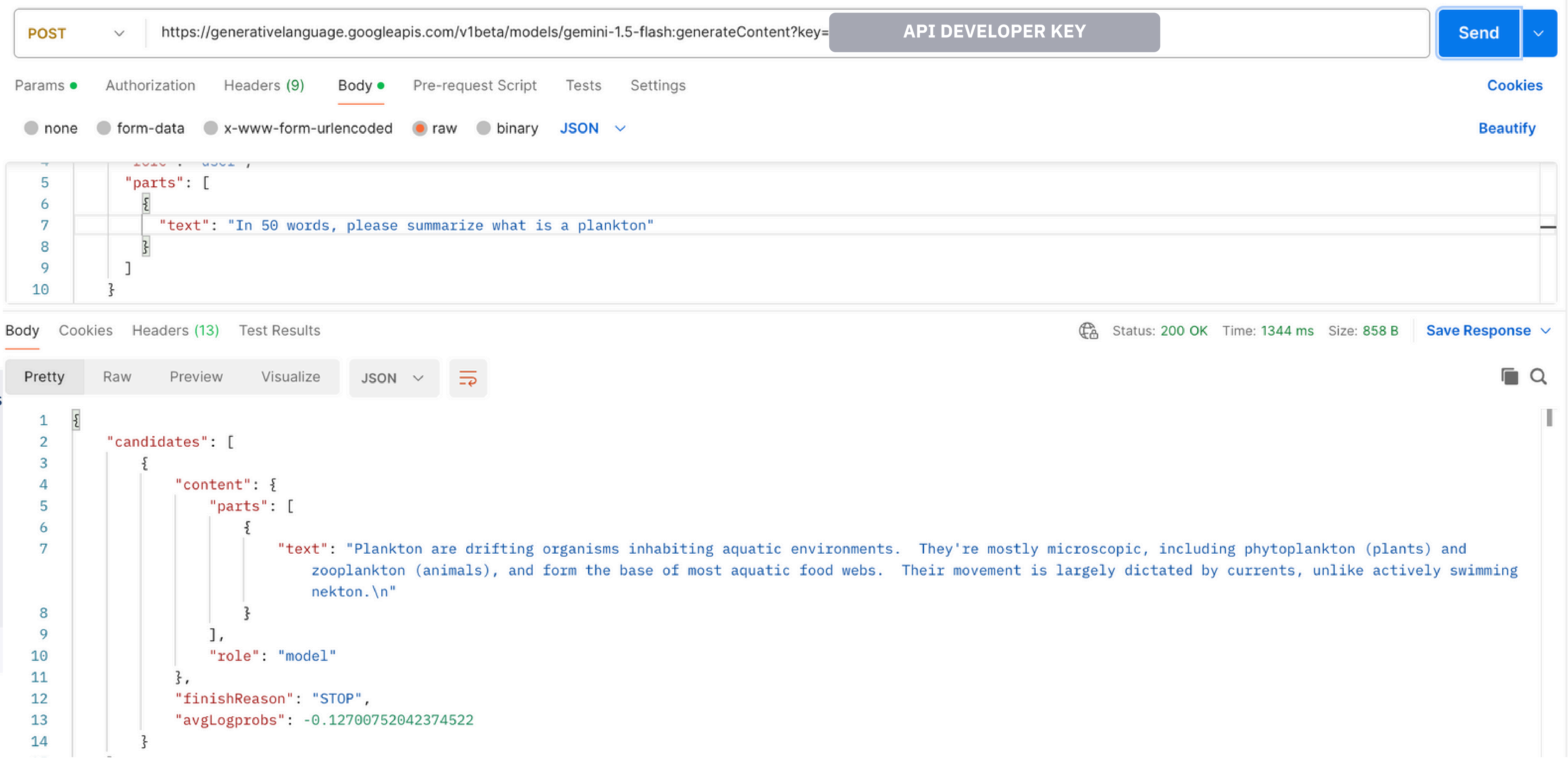


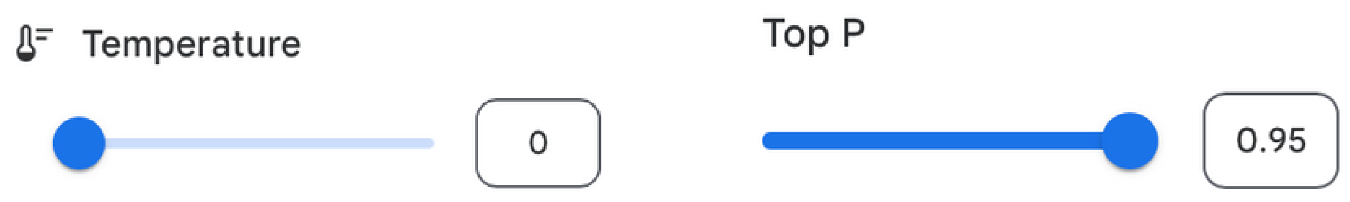
☐	Name	Type	Dimensions (e.g. location)	Value	Current usage percentage	Current usage	Adjustable
☐	Request limit per model per day for a project in the free tier	Quota	model : gemini-2.0-flash-exp	1,500	27.07%	406	Yes
☐	Request limit per model per day for a project in the free tier	Quota	model : gemini-1.5-flash	1,500	0.13%	2	Yes

Appendix

VERTEX AI API AND TESTING VIA POSTMAN







Allowed creativity in the response

Probability threshold for top-p sampling

Appendix

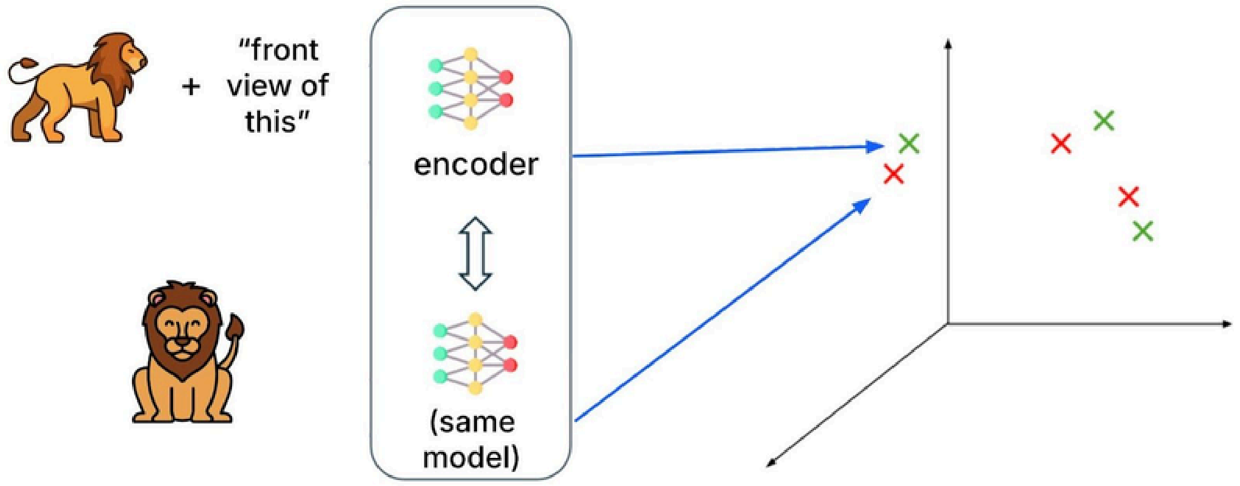
GENERATIVE AI EVALUATION

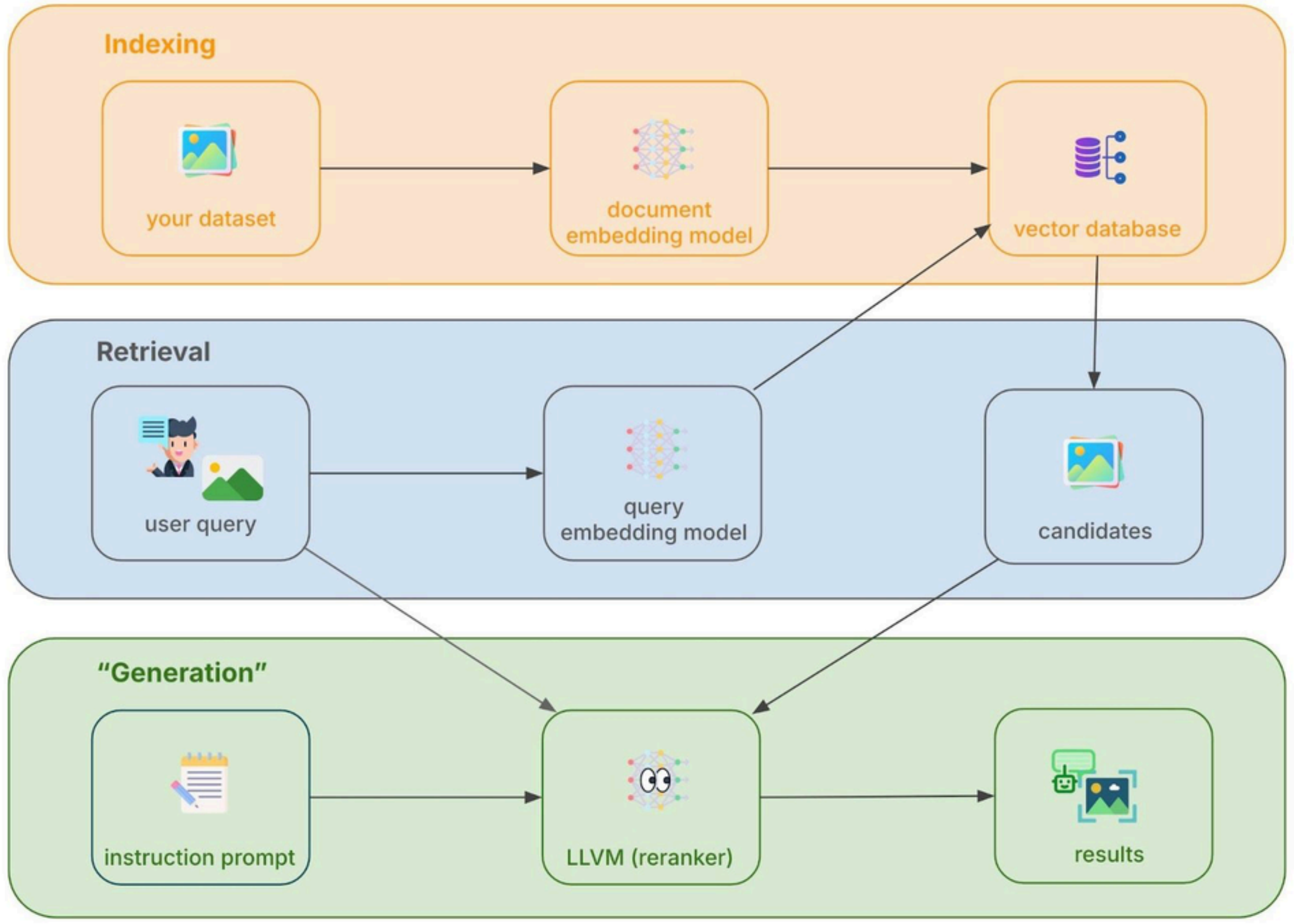
Metric	Importance	Evaluation Method
Accuracy	Ensures correct species identification	Comparison with labeled & **Human Evaluation**
Confidence Scores	Avoids unreliable classifications	Model's probability scores
Generalization	Tests performance on unseen species	Evaluate on unseen plankton images
Speed & Efficiency	Ensures practical use in research	Measure processing time per image
Bias & Hallucination	Prevents incorrect classifications	Cross-check AI output with expert labels

Human evaluation is important for accuracy, misclassifications
+ refine the model by feedback loop with expert knowledge

Appendix

MULTIMODAL SEMANTIC SEARCH IMAGES + TEXT





- Self-Supervised Learning: Uses web image pairs and foundation models to generate training data with 36.7M triplets.
- **Open-Ended Retrieval: Supports complex search intents beyond visual similarity.**
- Diverse Intent Handling: Interprets various search instructions in large-scale tests.

https://arxiv.org/abs/2403.19651

TRIPLETS consisting of three components:

- Query Image – The starting image for the retrieval task.
- Instruction – A description specifying how the retrieved image should relate to the query image.
- Target Image – The image that best matches the query image based on the given instruction.

Appendix

FREQUENTLY USED SEMANTIC VECTOR DISTANCE

Euclidean Distance

$$d(a,b) = d(b,a) = \sqrt{\sum_{i=0}^{n-1}(b_i - a_i)^2}$$

Measures the length of a segment that connects 2 points. It's the most commonly used distance metric and is very useful when the data are continuous.

Inner Product

$$p(A,B) = A \cdot B = \sum_{i=0}^{n-1} a_i \cdot b_i$$

IP is more useful if you need to compare non-normalized data or when you care about magnitude and angle.

If you use IP to calculate similarities between embeddings, you must normalize your embeddings. After normalization, the inner product equals cosine similarity.

Cosine Distance

$$cos\theta = \frac{\sum_0^{n-1}(a_i \cdot b_i)}{\sqrt{\sum_0^{n-1} a_i^2} \cdot \sqrt{\sum_0^{n-1} b_i^2}}$$

Uses the cosine of the angle between two sets of vectors to measure how similar they are. The cosine similarity is always in the interval [-1, 1]. The larger the cosine, the smaller the angle between the two vectors, indicating that these two vectors are more similar to each other. By subtracting their cosine similarity from 1, you can get the cosine distance between two vectors.

Appendix

BASIC PRIMER FOR VECTOR EMBEDDING IN RAG

Transforming Data into Vectors

Output Vector represents semantic embedding

Input Data	Embedding model	Output Vector
I **LOVE** PLANKTONS	→	0.82
I **LIKE** PLANKTONS	→	0.70
I **HATE** PLANKTONS	→	0.24

Euclidean Distance

$$d(A,B) = \sqrt{(v_1 - v_2)^2}$$

$$d(A,B) = |v_1 - v_2|$$

Distance represents **semantic similarity**

1 Dimension Euclidean Space

Shortest distance means <u>most similar</u>

Distance = | 0.82–0.70 | = **0.12**

Distance = | 0.70–0.24 | = **0.46**

Longest distance means <u>least similar</u>

Distance = | 0.82–0.24 | = **0.58**

Output vectors in this slide are for example only

Appendix

VECTOR DATABASE FOR PLANKTON IMAGES

Schema | Vector Search | Data | Partitions | Segments | Properties

Name	planktons_f4f8d5	Status	Loaded
Description	Plankton collection	Replica	1
Created Time	12/06/2025, 21:00:47	Entity Count	52

Features: Auto ID; Consistency: Bounded; Mmap Settings

Field	Type	Nullable	Default	Index Name	Index Type	Index Parameters	Description
auto_id PK	Int64	×	--		Scalar Index	--	--
image_vector	FloatVector(1280)	×	--	image_vector	IVF_FLAT(COSINE)	nlist: 128	--
family	VarChar(64)	×	--		Scalar Index	--	--
genus	VarChar(64)	×	--		Scalar Index	--	--
isBroken	Bool	×	--		Scalar Index	--	--
part	VarChar(64)	×	--		Scalar Index	--	--
size	Float	×	--		Scalar Index	--	--

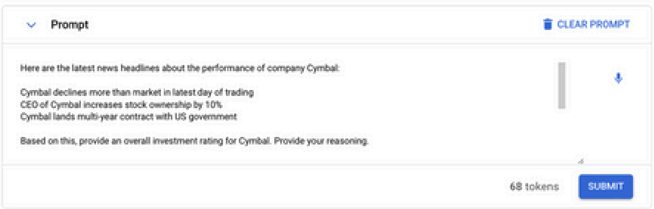
Appendix

FINE TUNING GEN-AI BY LAYER OF INTERACTION

User Interface Layer

Skills Required:
Natural Language Understanding

- Model Configuration Layer
- Application Logic Layer
- Model Architecture Layer
- Pre-training Layer
- Evaluation & Monitoring Layer



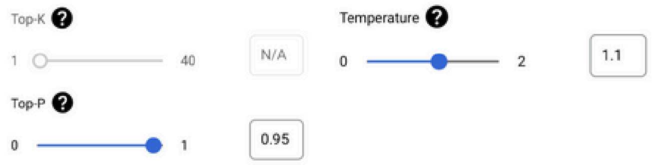
Prompt Engineering
Crafting specific prompts to guide the AI's response. This includes using structured templates, keywords, and context-setting within the prompt.

Example: To get a detailed story, instead of asking "Tell me a story," you might ask "Tell me a detailed and thrilling adventure story set in a medieval fantasy world."

Model Configuration Layer

Skills Required:
Understanding LLM parameters

- Application Logic Layer
- Model Architecture Layer
- Pre-training Layer
- Evaluation & Monitoring Layer



Parameter Tuning
Adjusting hyperparameters like temperature, top-k sampling, or top-p (nucleus) sampling to control the **diversity** and **creativity** of the generated content.

Top-k: Narrows choices to top-k tokens, then samples based on probability
Top-p : Chooses from tokens whose combined probability reaches a threshold p
The most important parameter is the temperature, it **affect the hallucination level**.

Application Logic Layer

Skills Required:
Programming (with/without LLM knowledge), Application Development – **Exception handling**.

- Model Architecture Layer
- Pre-training Layer
- Evaluation & Monitoring Layer

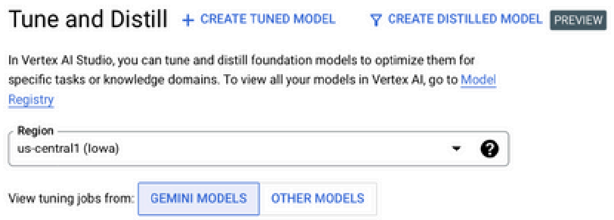
Rule-based Application Level Adjustments
Applying rules or heuristics to modify AI outputs based on the application's needs. Example: Automatically append "Please provide more details." to any user query detected as too vague.

Post-processing Filters
Implementing filters to refine or alter the AI's output after generation, such as spell-checking, grammar correction, or content moderation

Model Architecture Layer

Skills Required:
Able to utilize tuning and distillation tools

- Pre-training Layer
- Evaluation & Monitoring Layer



Transfer Learning
Leveraging pre-trained models and fine-tuning them on specific tasks or datasets. This includes methods like using a pre-trained Bison or GPT model and fine-tuning it on a new dataset.

Distillation
Creating smaller, efficient models (student models) that approximate the performance of larger, pre-trained models (teacher models) through a distillation process.

Pre-training Layer

Skills Required:
Able to train LLM models

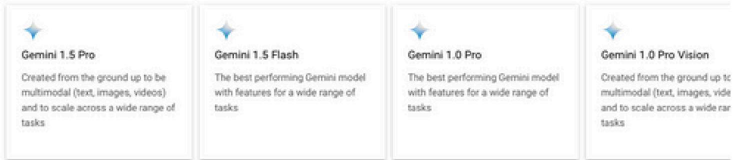
- Evaluation & Monitoring Layer

Domain-specific Pre-training
Pre-training the model on a large corpus of domain-specific data before fine-tuning it on task-specific data. For example, pre-training on medical literature for a healthcare application.

Example: Pre-training a language model on a corpus of biomedical research papers to enhance its ability to understand and generate medical literature.

Evaluation & Monitoring Layer

Skills Required:
Understand domain specific knowledge, to evaluate model performance



Continuous Evaluation
Implementing a feedback loop for continuous evaluation of the model's performance using metrics like accuracy, and user satisfaction.

A/B Testing
Conducting A/B tests to compare different versions of the model or configurations to determine the most effective approach.

Presentation slides are reused from my *GenAI Layers of Interaction* slides

End of Presentation!

Jaronchai Dilokkalayakul [1] , Akane Kitamura [2] , Takeshi Obayashi [1,2]
[1] Graduate School of Information Sciences (GSIS), Tohoku University
[2] Advanced Institute for Marine Ecosystem Change (WPI-AIMEC), Tohoku University

jaronchai.com