The role of humans in the AI era

: Why does AI do some tasks while humans do others?

Jieung Kim
Dept. of Computer Science and Engineering, Yonsei University
2026-07-11

Dentist

Nurse

Police officer

Farmer

Mechanic

In the AI era, what should we do?

Judgment

Problem framing

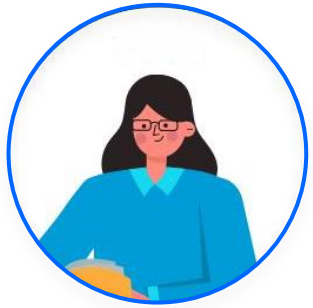
Critical thinking

Verification

Creativity

Adaptability

Teacher



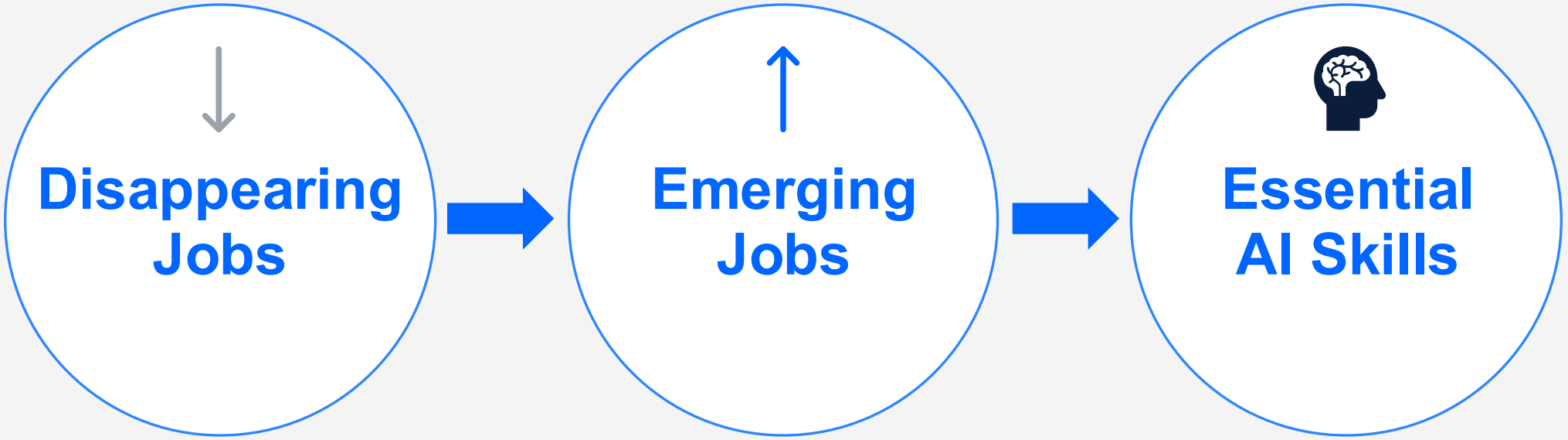
Postman

Chef

Doctor

Baker

The usual story in AI career talks



…And the same advice: “Be more human”

Creativity

Empathy

Judgment

Adaptability

The same story, everywhere you look

TED

AI Is Coming for Your Job. Now What?

Vlad Tenev, Jan 2026

CNN Business

AI isn’t ‘taking’ your job — here’s what’s happening

Lisa Eadicicco, May 2026

Forbes

10 Careers AI Will Replace In The Next 5 Years

Bryan Robinson, Jul 2025

TED

The Future of Work: AI-Proofing Your Career

TED · Matthew Wells

World Economic Forum

92 million jobs displaced by 2030

Future of Jobs Report 2025

LinkedIn

AI won’t take your job — someone who uses AI will

Christina Wodtke, Mar 2025

Two problems with the usual answer

↻ **The ‘safe list’ keeps getting rewritten**

? **Vague advice you can’t act on**

Two ways to read the same story

AUTOMATION

Automation

A concrete answer — “here’s how.”

VS

HUMANITY

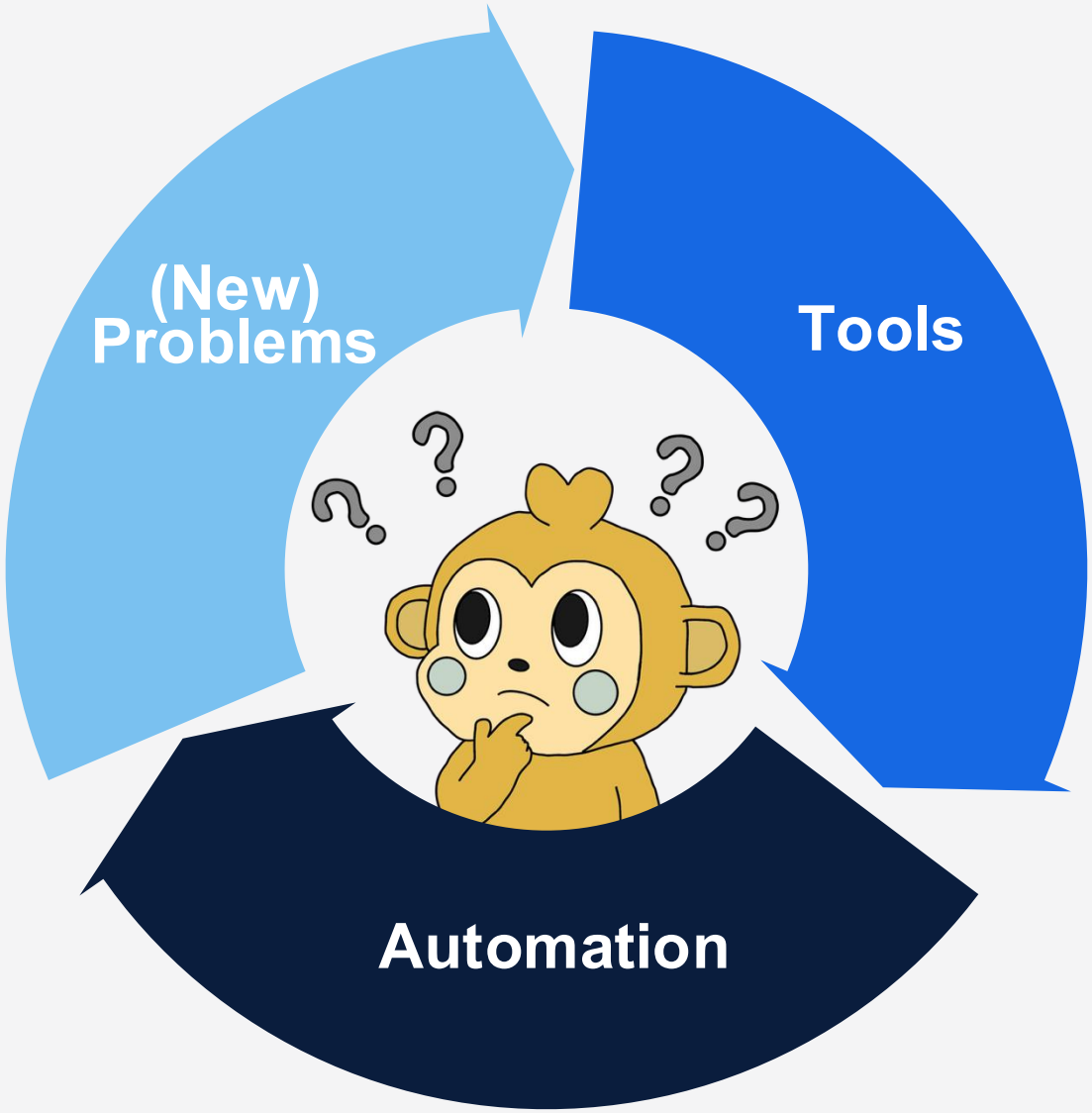
Human Value

Generic tips — “so what?”

Let’s understand it, not just fear it

So how does automation happen?

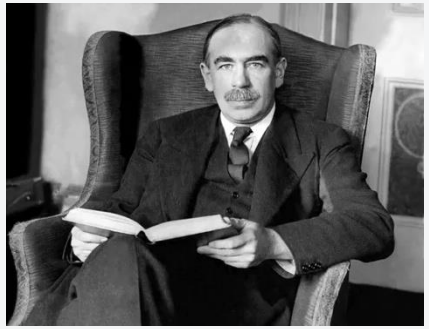
Every wave follows the same loop:



A question repeated for 200 years

1811
Industrial Revolution

1830
Keynes



1990
Computers

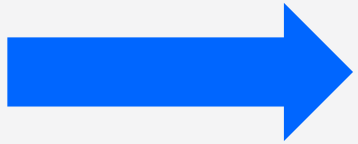
…And still the same question today

2000
The Internet

2020
AI

So I want to change the question

Will AI (the new automation) change the role of humans?



What does AI try to automate?

What makes that automation possible?

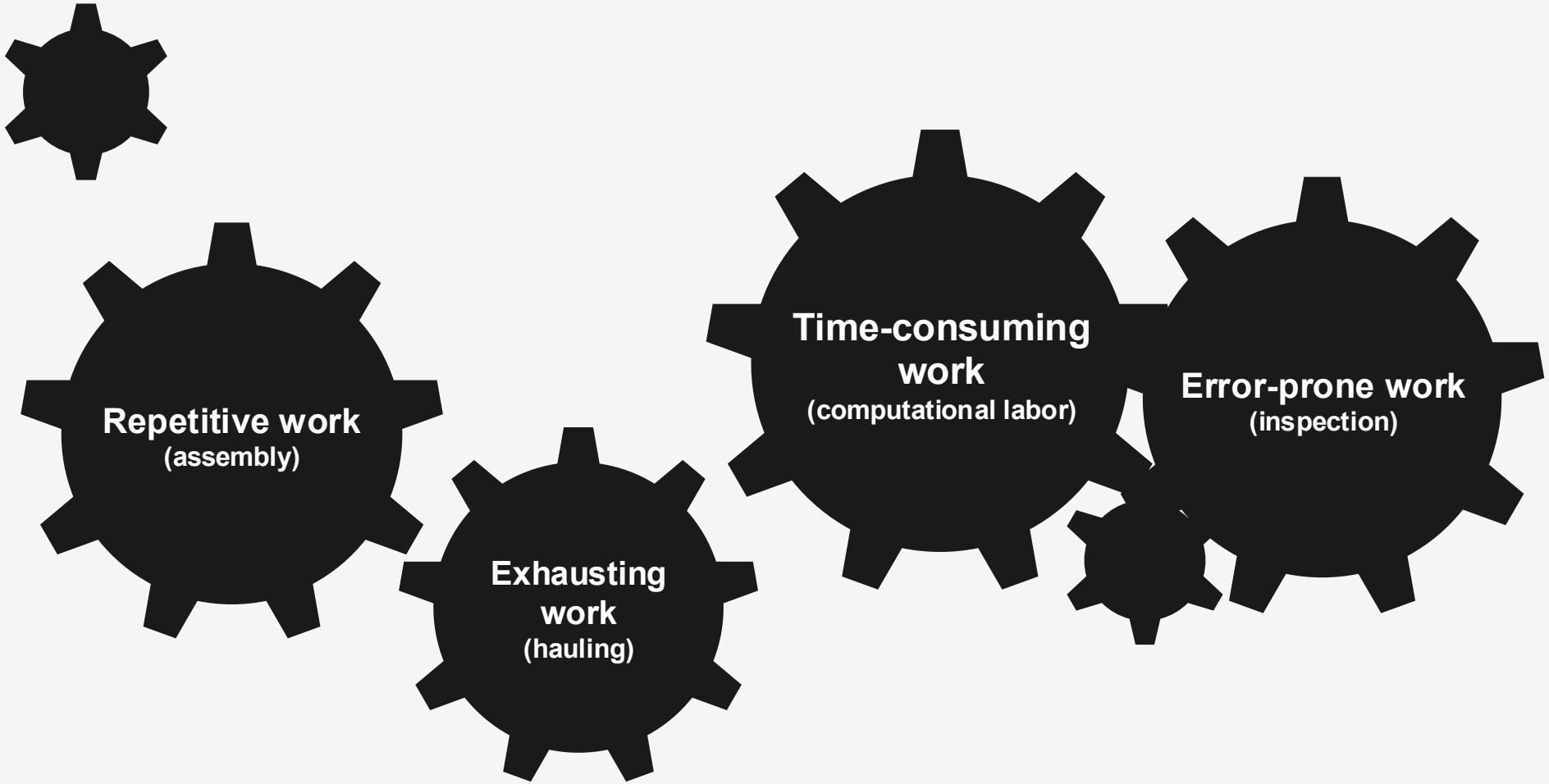
Today's lecture map

1. Automation
2. Criteria for Automation
3. The Role of Humans
4. Experience & Final Lessons

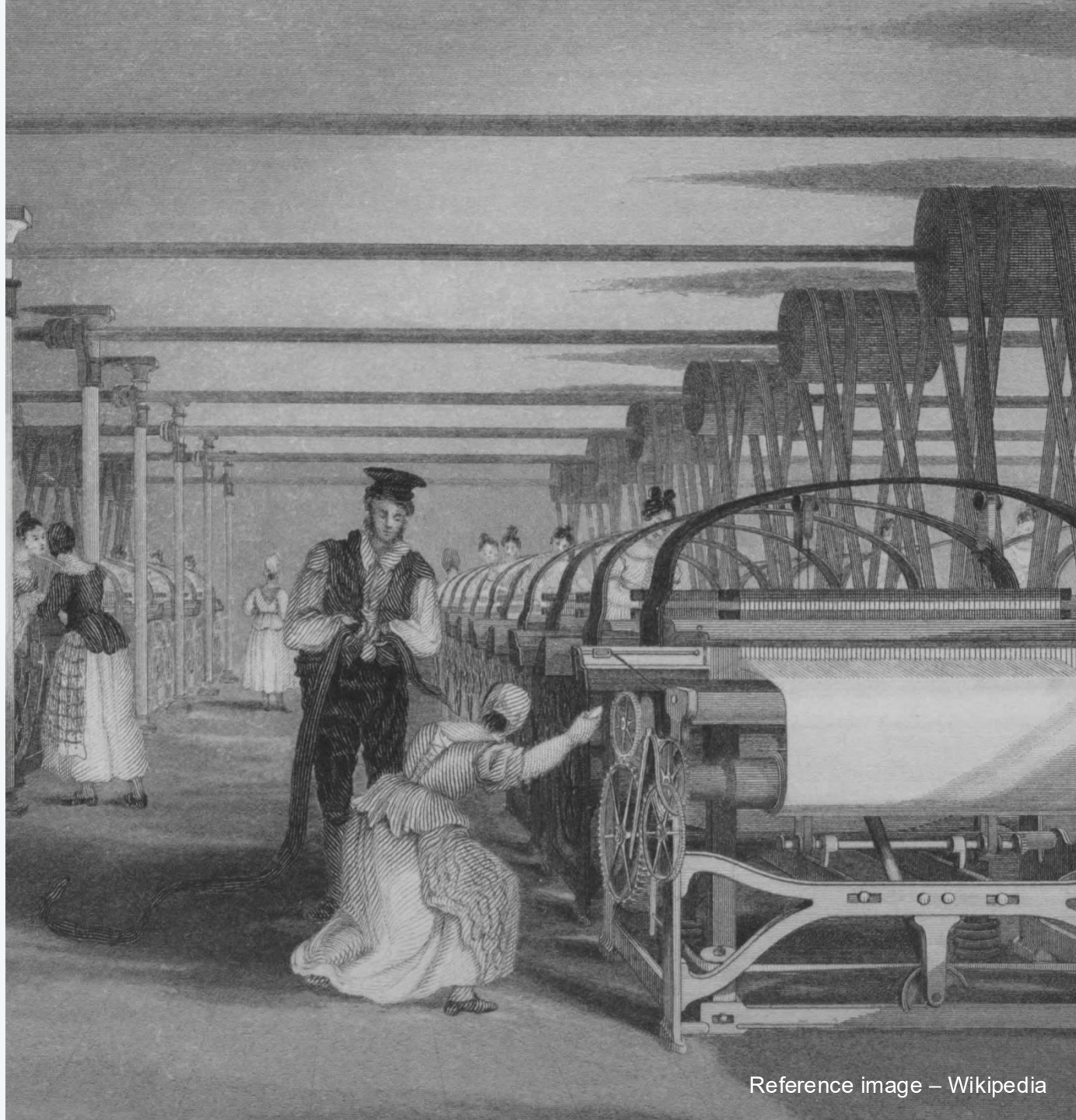
1

Automation

Why do we want automation?



The Industrial Revolution = automation of muscle



Reference image – Wikipedia

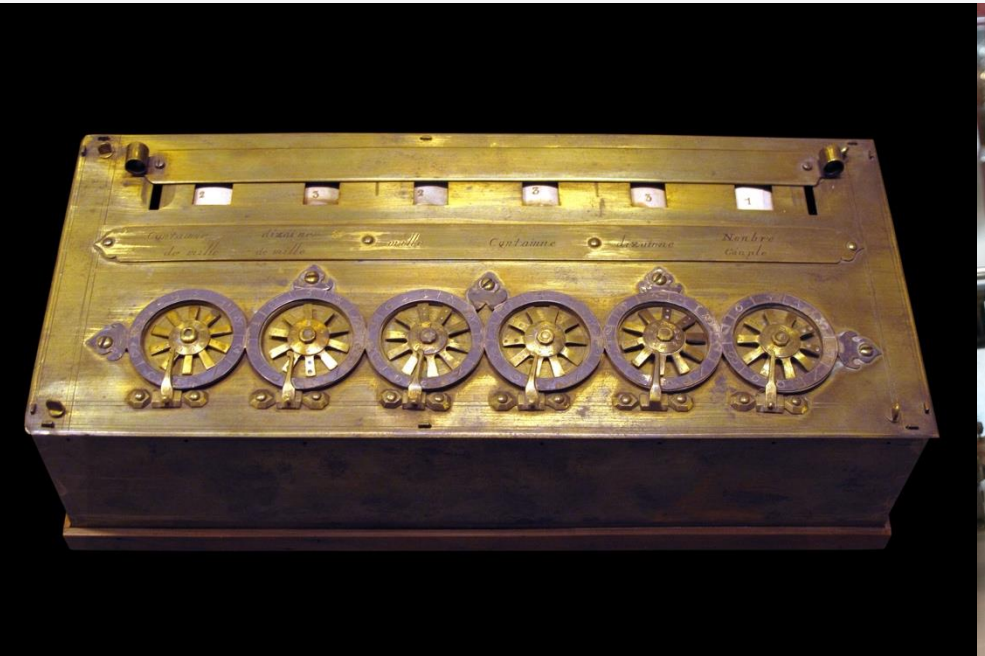
Calculation was labor, too

Numerical labor

Prediction

Optimization

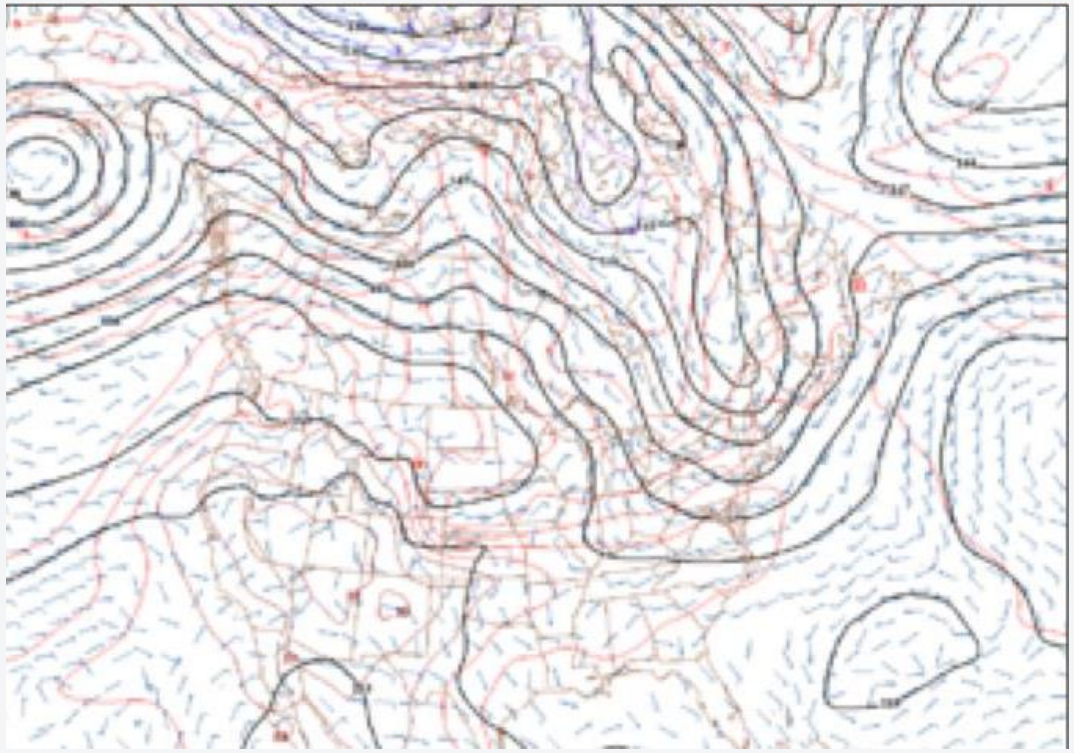
“Computer”
originally the name of a job
*computational labor done by people



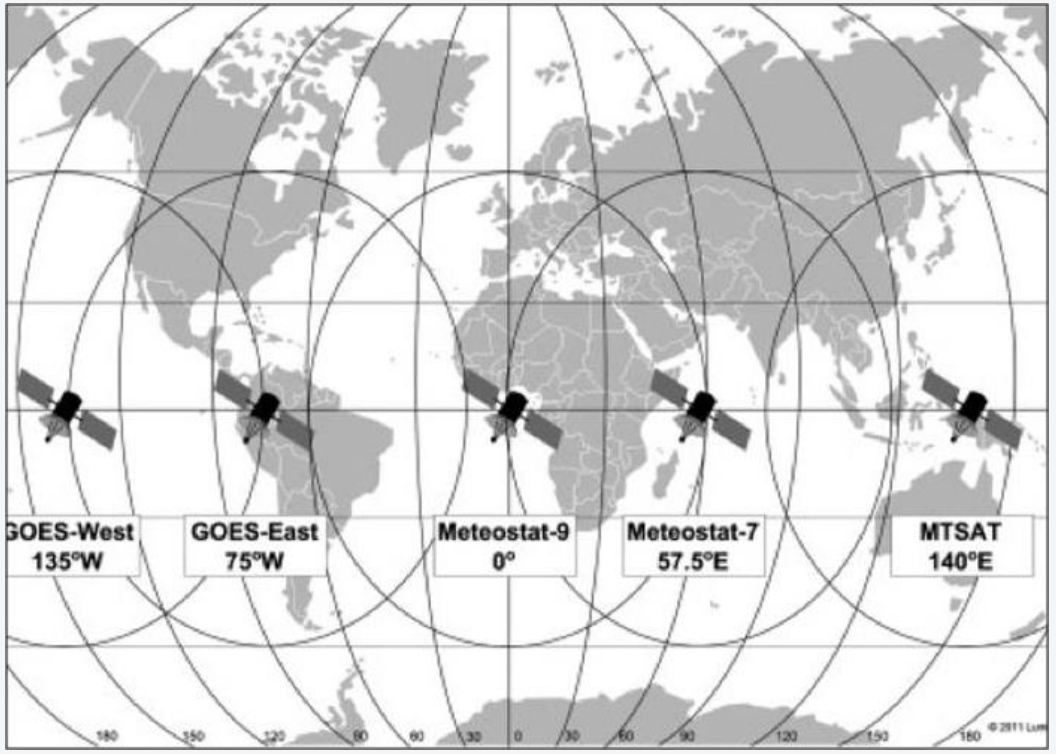
Reference images – Wikipedia

Computers automated calculation

- Large-scale computation
- Fast operations
- Accurate repetition
- Automatic processing



Reference image – Wikipedia

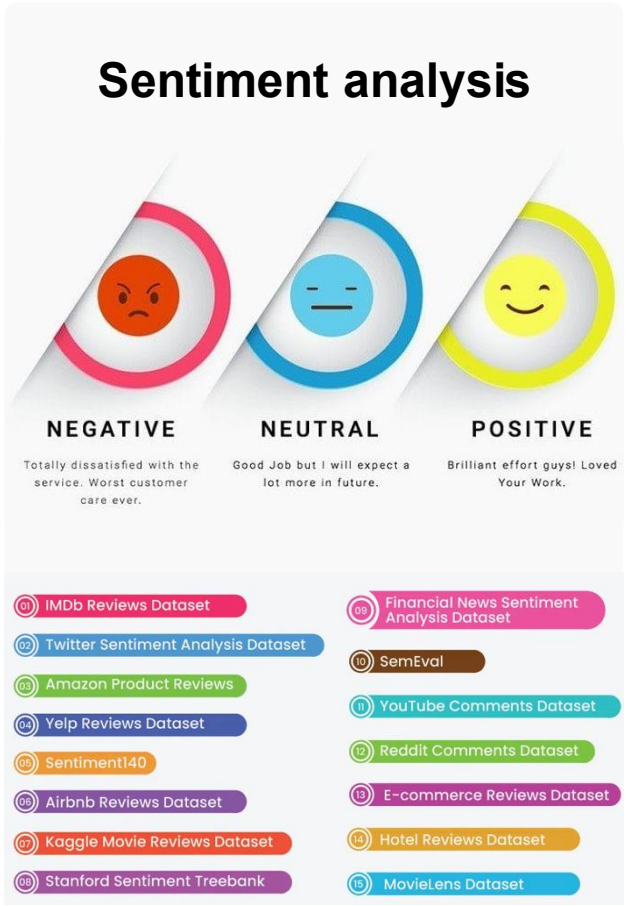


Reference image – ScienceDirect

But humans don't only calculate

We recognize objects

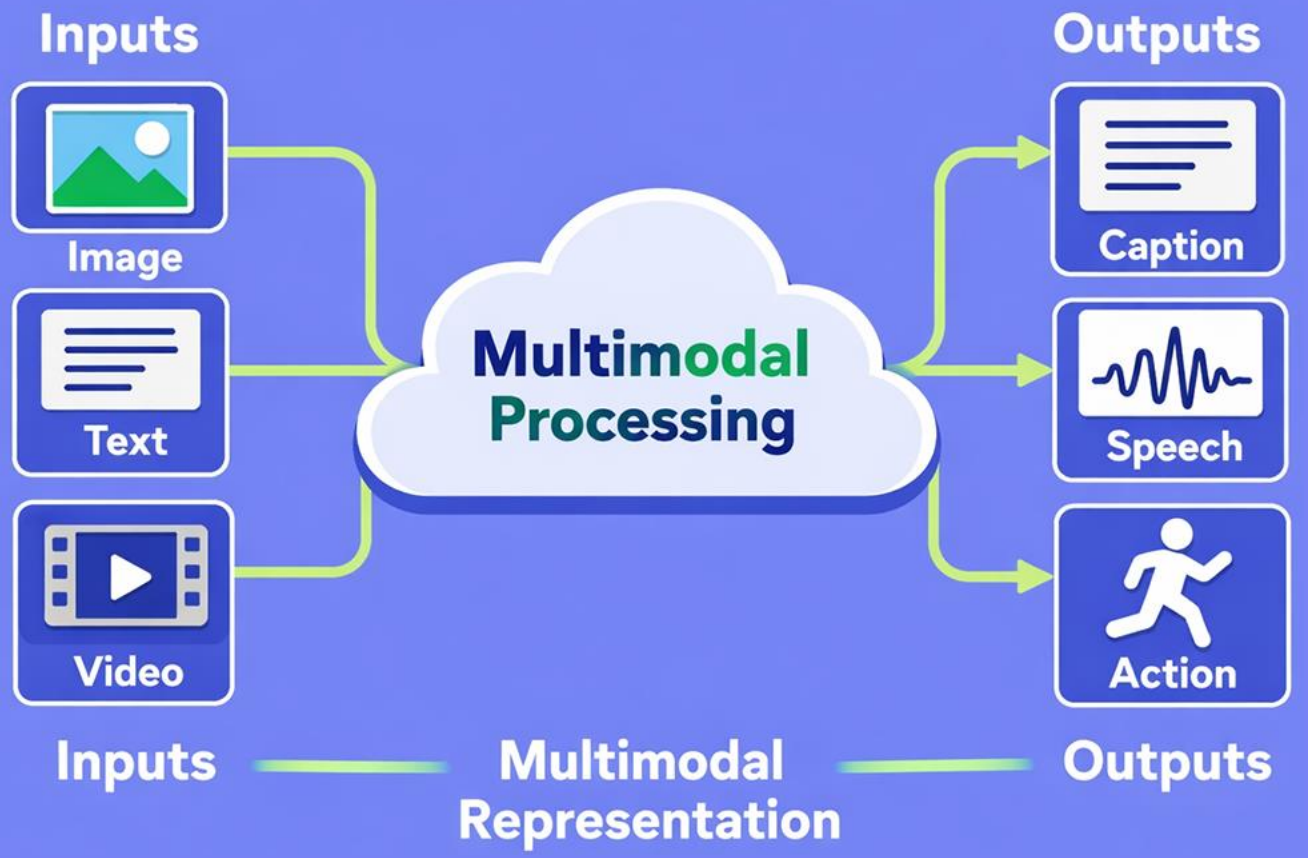
We understand language



We interpret situations

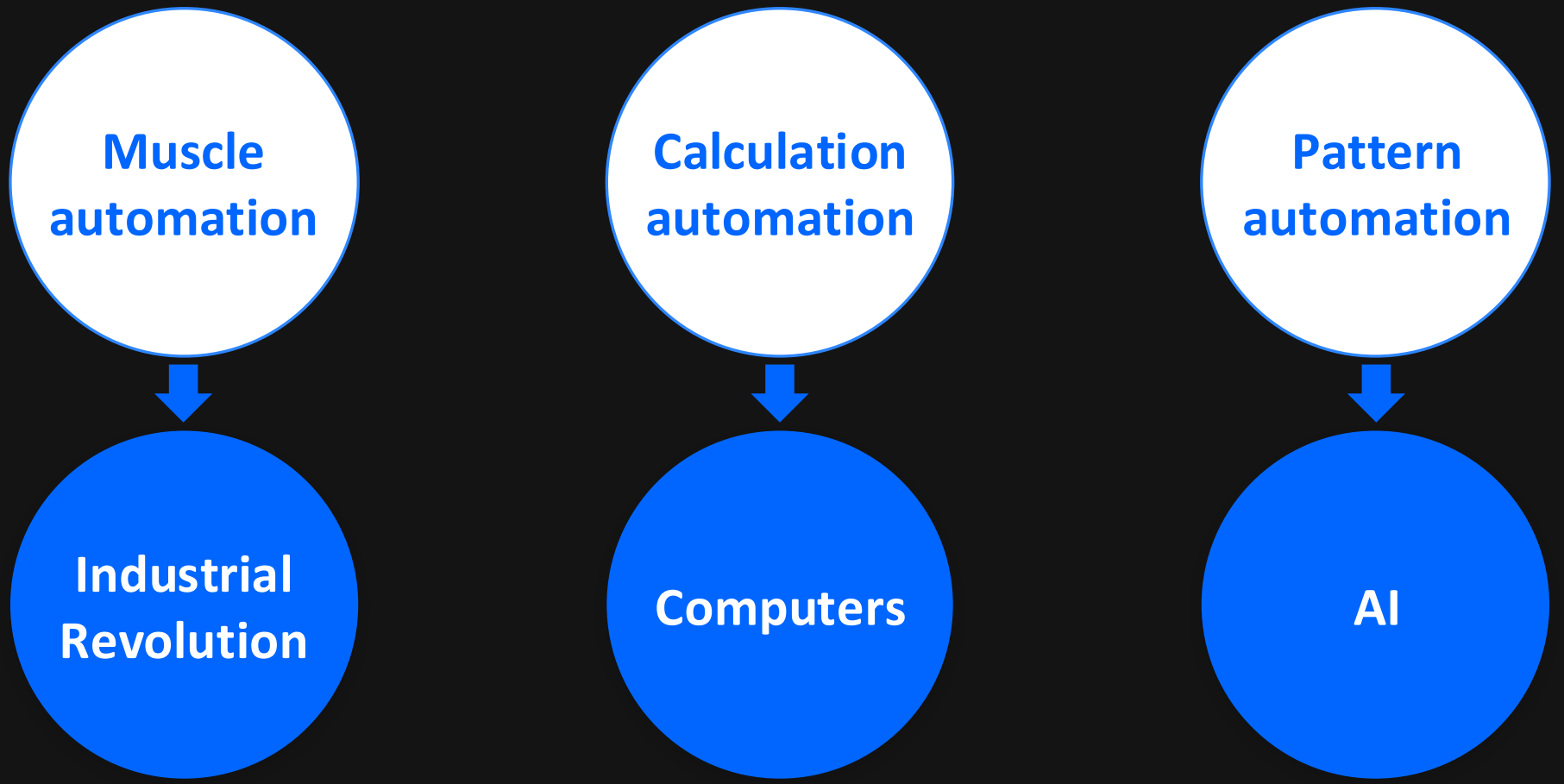
What AI does well — what deep learning changed

- Learning patterns from data
- Images / speech / text
- Fast inference
- Problems that can be evaluated

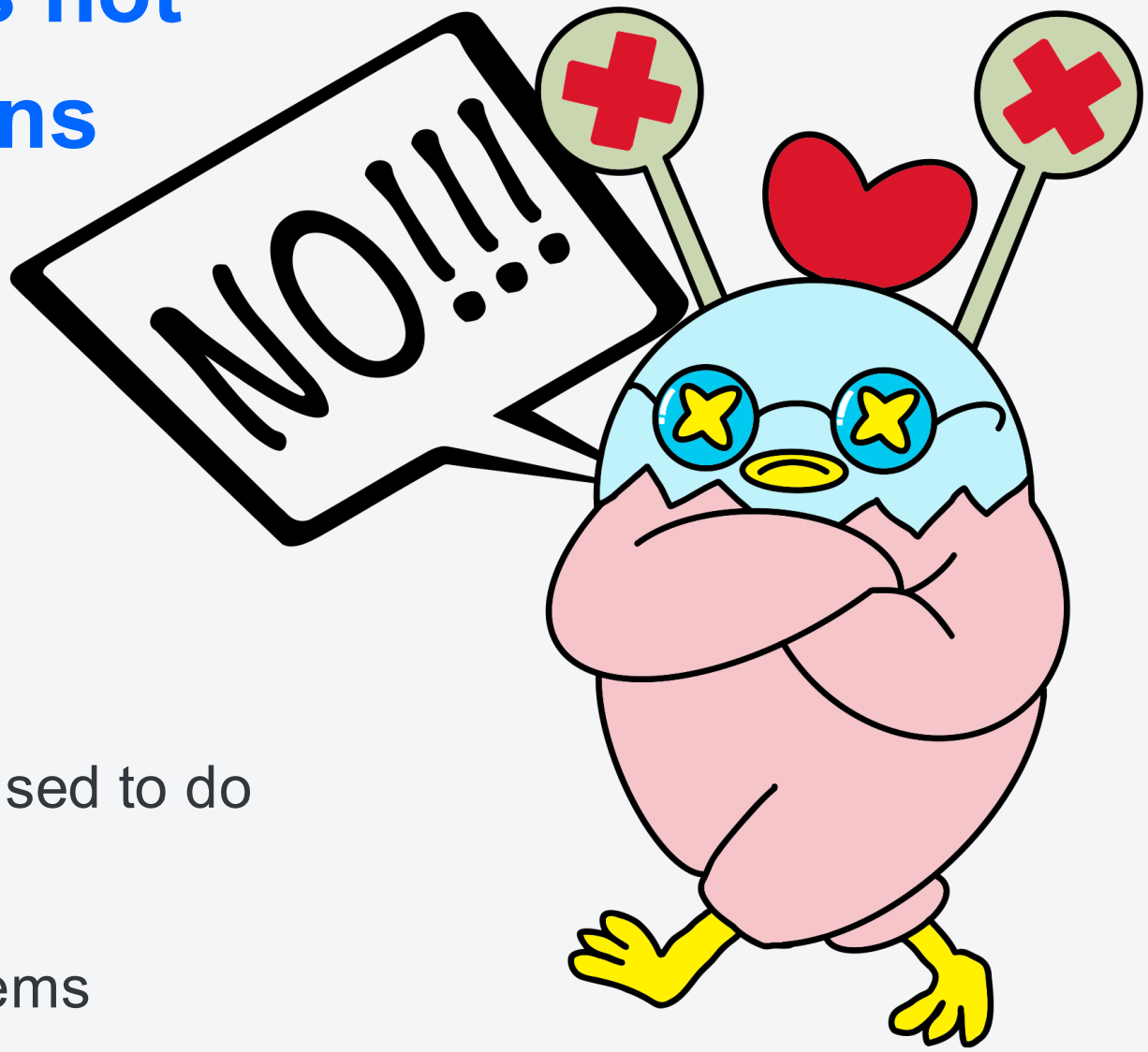


AI is a pattern-automation technology

Artificial Intelligence (AI) = Pattern automation


Muscle automation
Industrial Revolution
Calculation automation
Computers
Pattern automation
AI

Historically, automation is not about eliminating humans

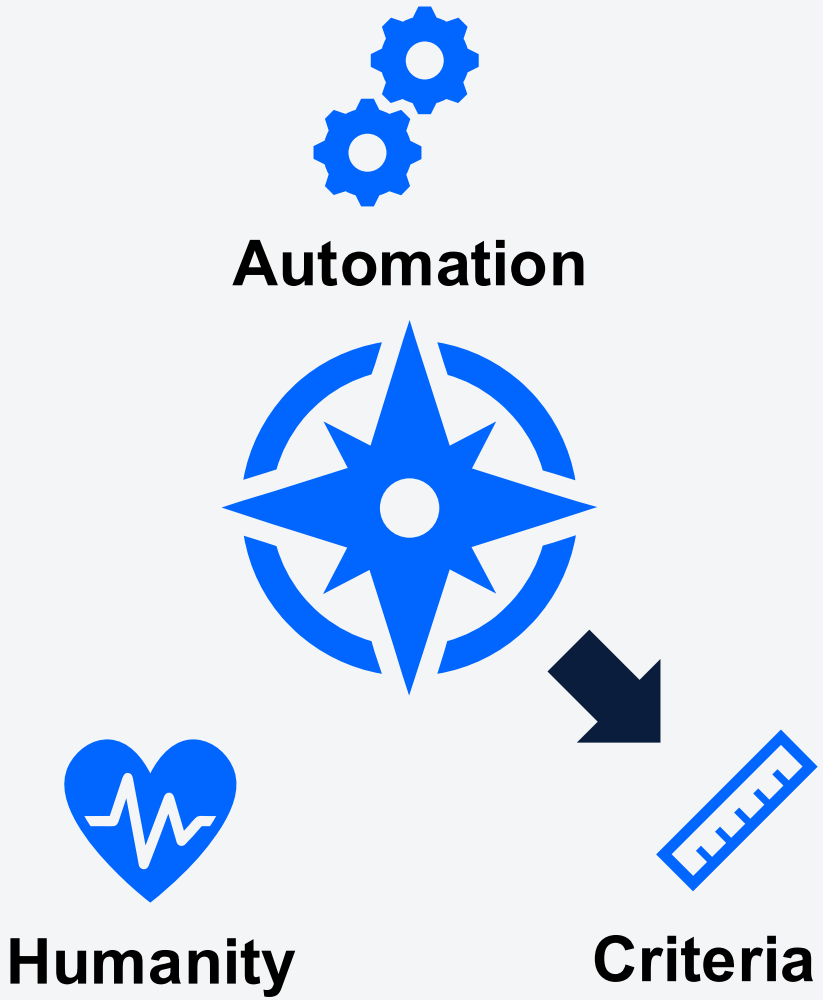


- Machines take over the work humans used to do
- Human roles are reorganized
- Humans move on to higher-level problems

Therefore, we again must change the question

Q. Where is the boundary of automation?

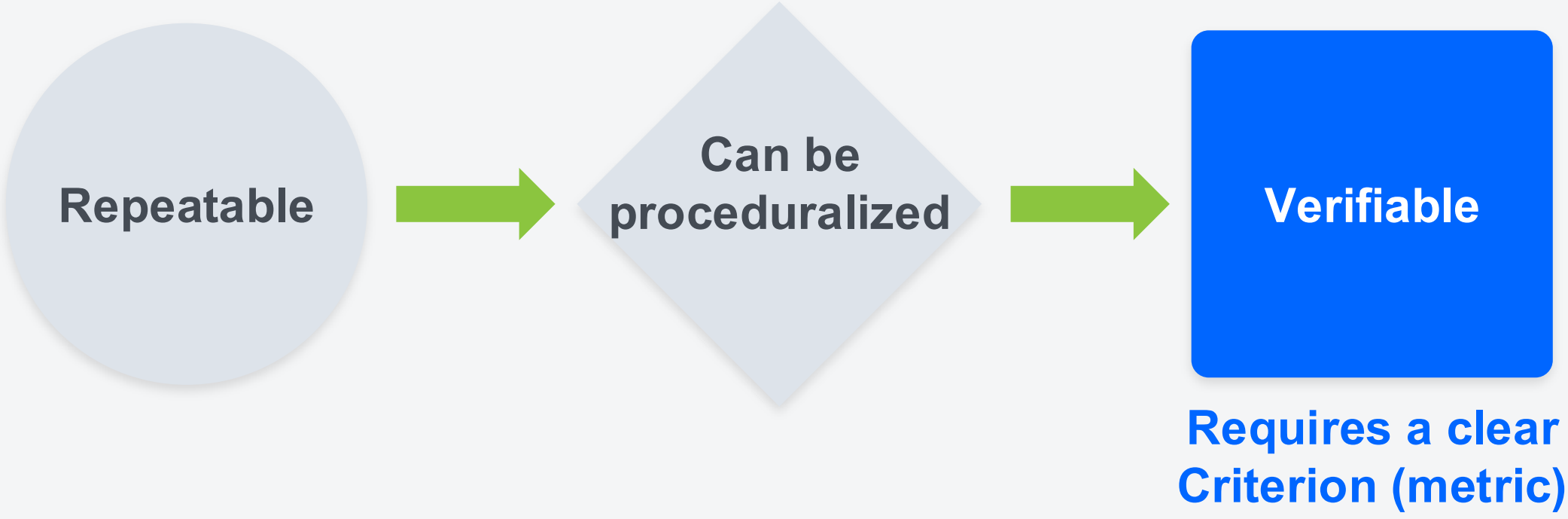
Q. Why are some problems automated while others remain difficult?



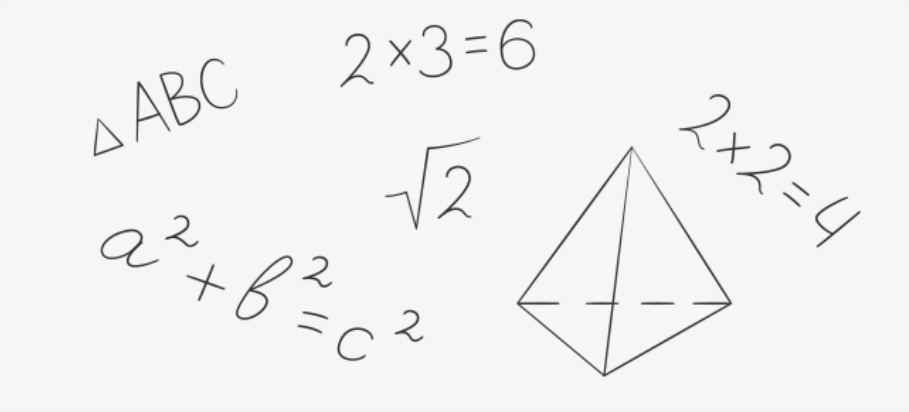
2

Criteria for Automation

The essence of automation



Problems with a criterion



VS

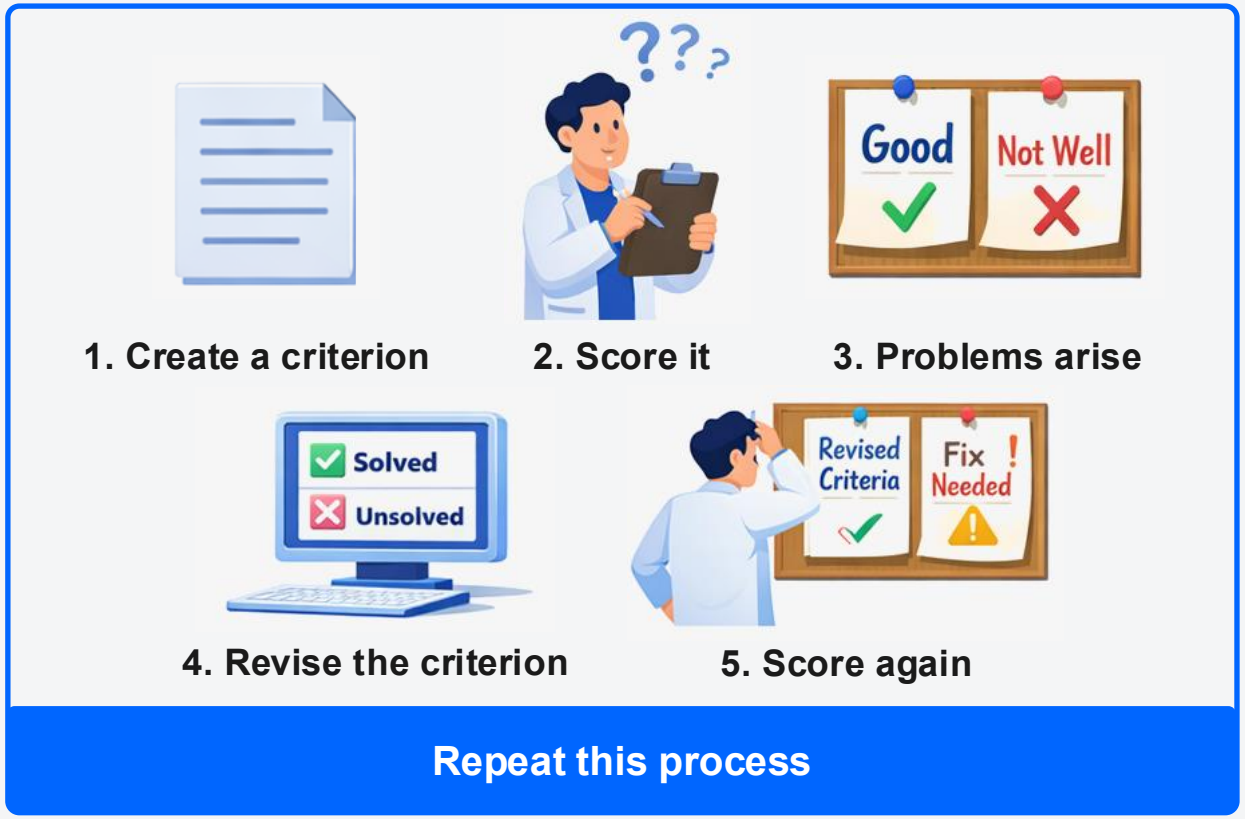
Problems without a criterion

Most problems in the world are on the right

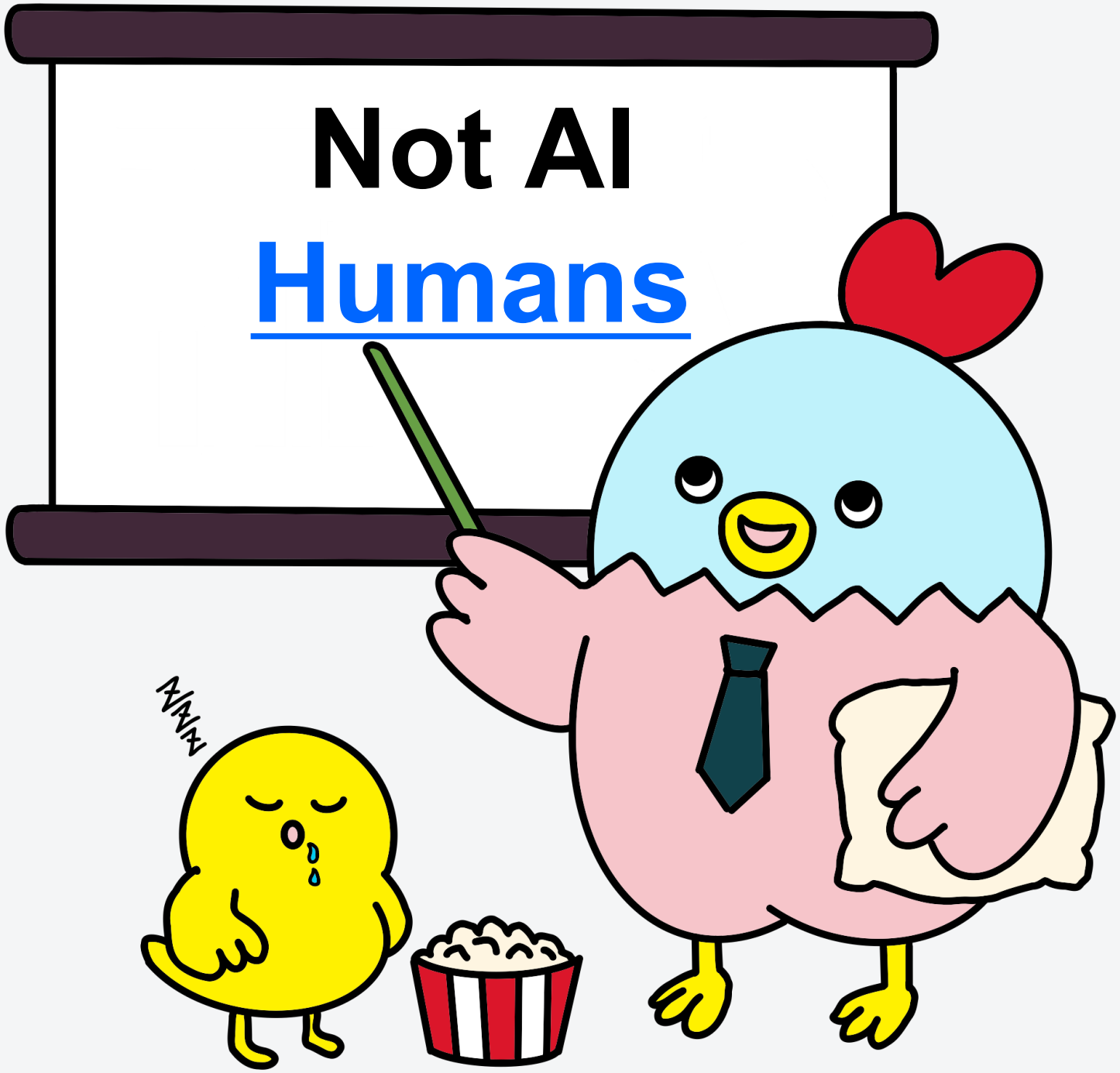
So we create criteria

Problems where criteria are hard

e.g., open-ended questions



Who creates and modifies the criteria?

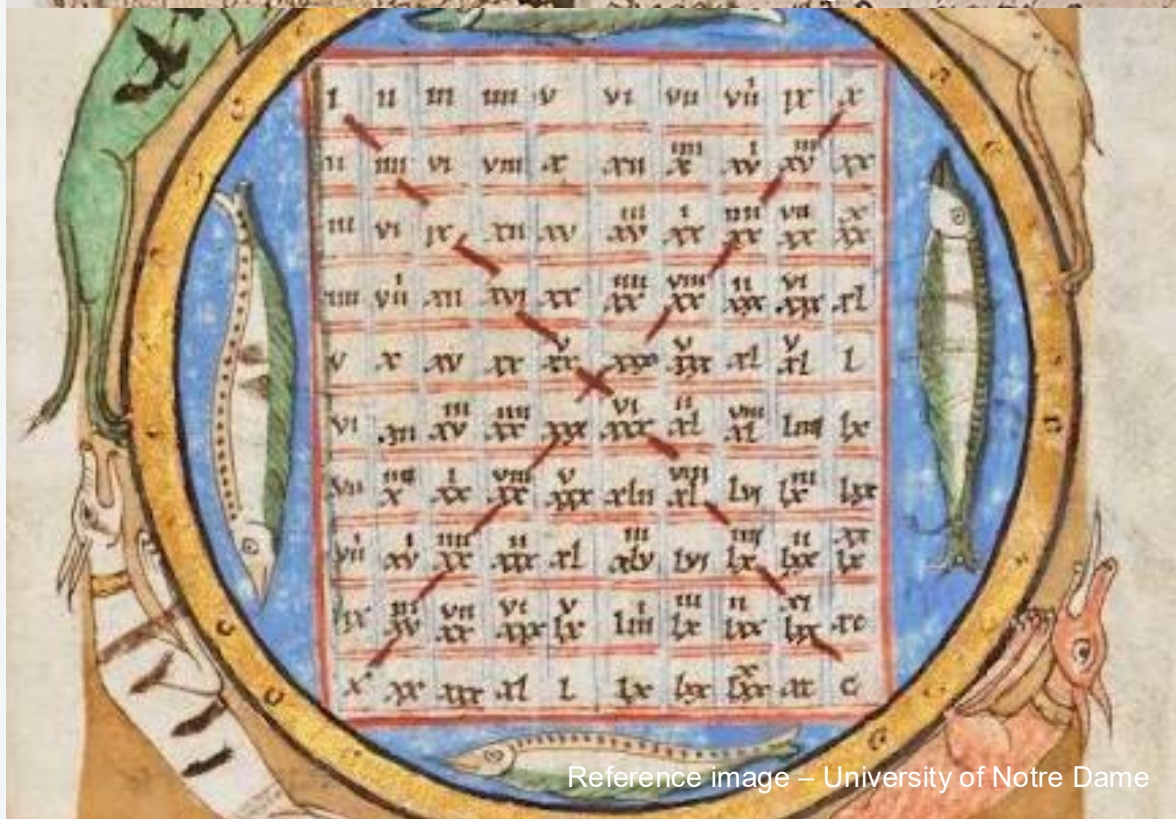


3

The Role of Humans

Was arithmetic easy from the start?

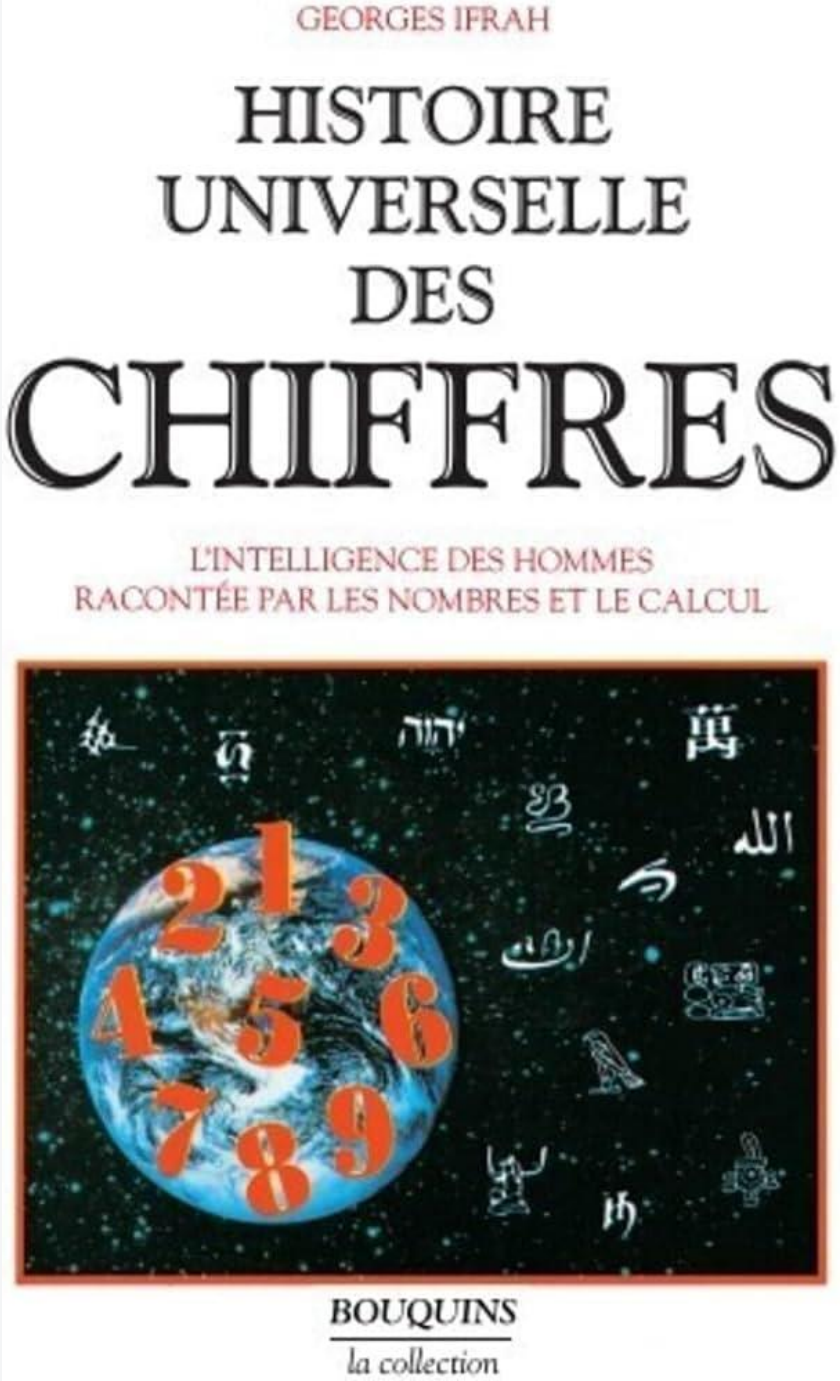
Reference image – The British Museum



Reference image – University of Notre Dame

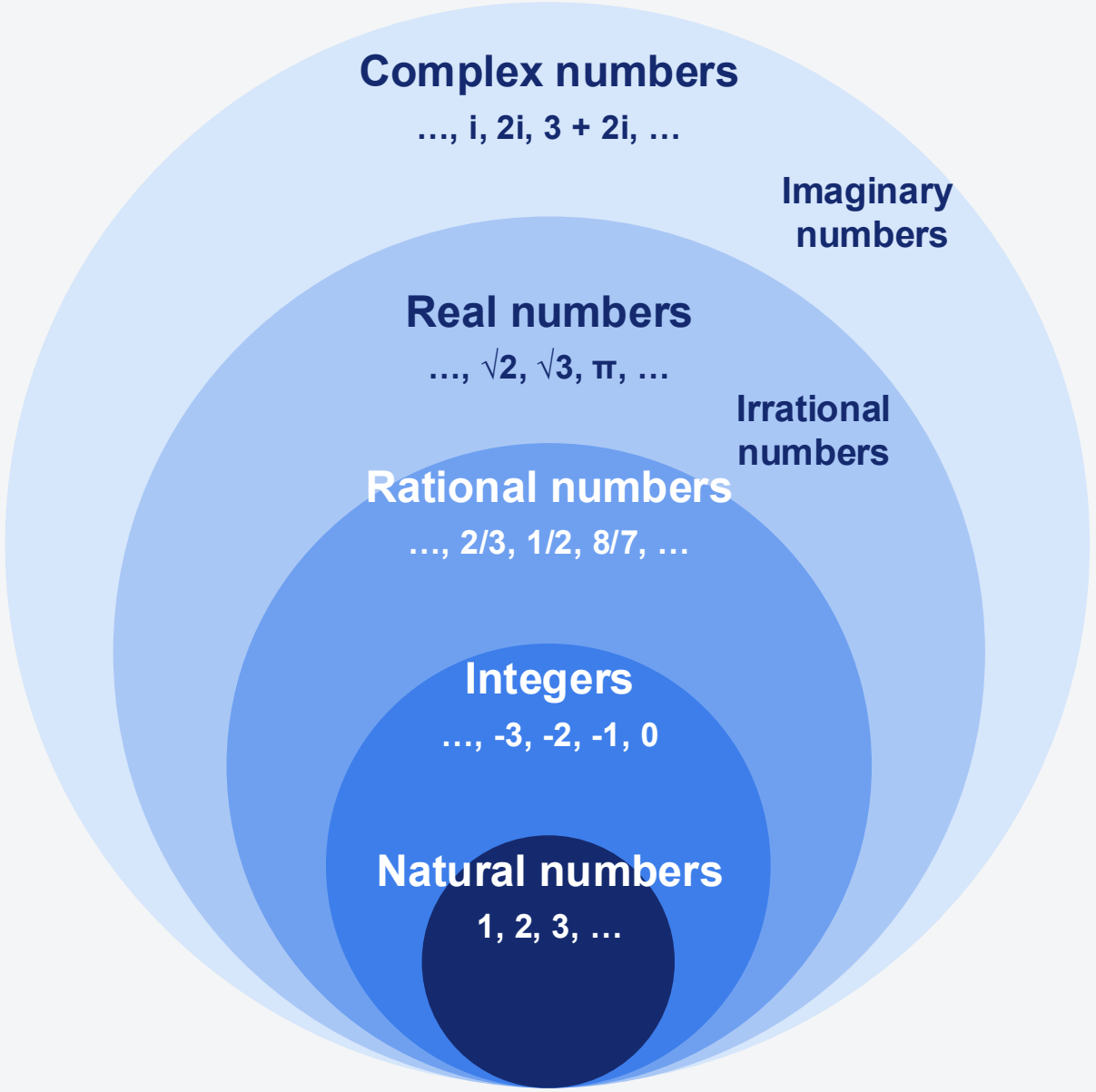
1 2 3 4 5 6 7 8 9

Numbers were the invention of a new criterion



Reference image – Amazon

Numbers kept expanding



Making the complex world computable

Expanding the criteria

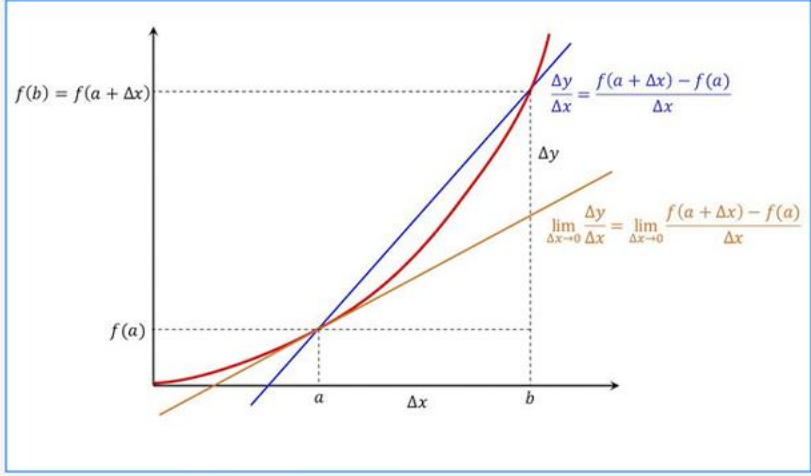
Enables numerical analysis

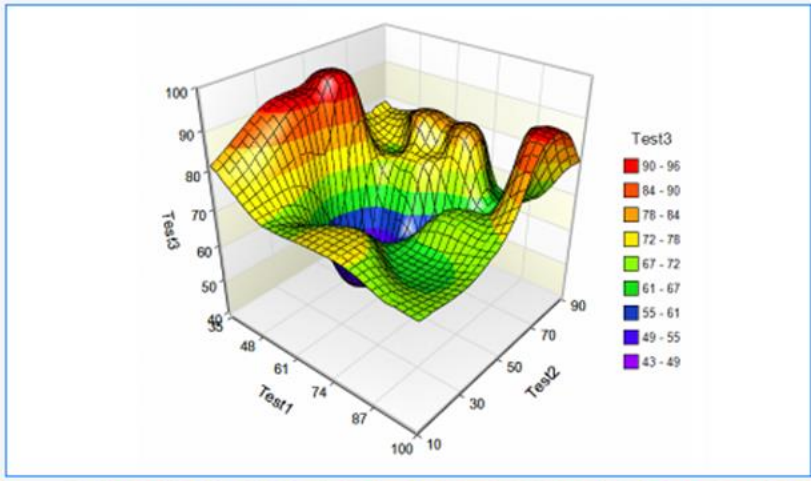
Automates the complex world

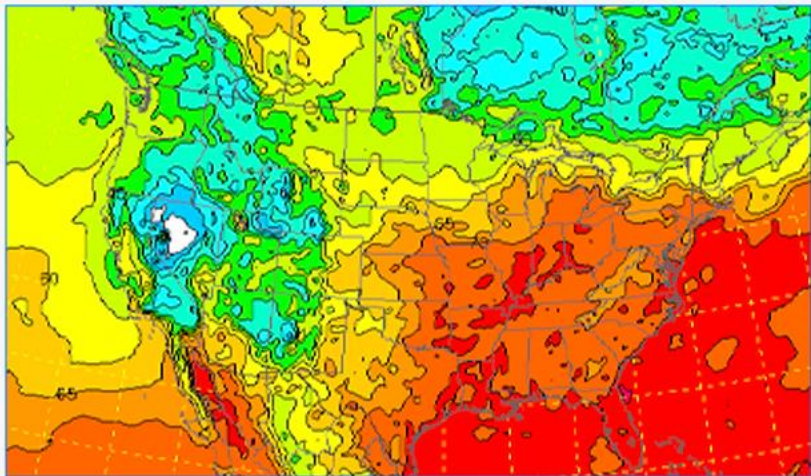
Differential equations

Numerical approximation

Computer computation







Then came things that never existed

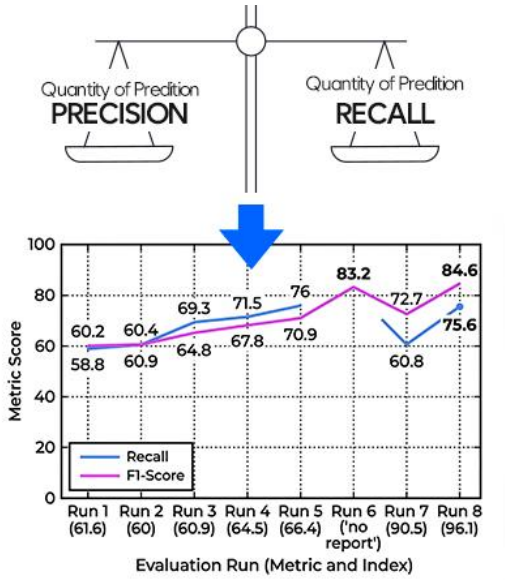
The Internet

Search engines

Virtual worlds

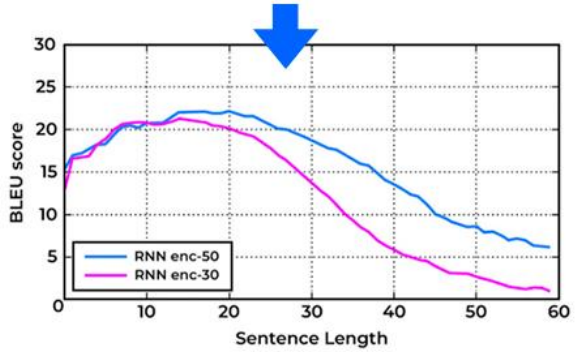
AI has the same structure

F1 Score

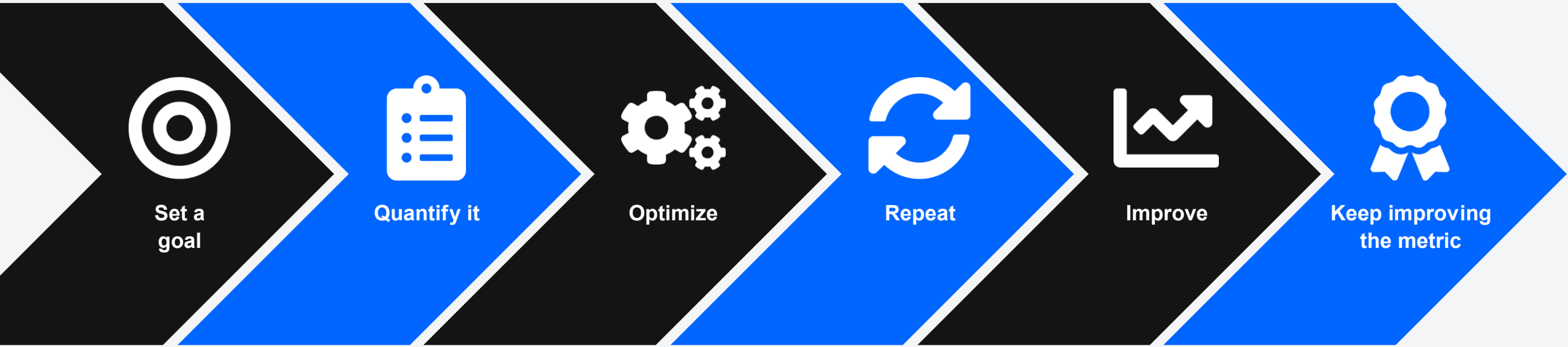


BLEU Score

$$\text{BLEU}^{*} = \prod_{n=1}^{N} p_n^{w_n}$$



The essence of AI: criterion (metric) + optimization



But there are still difficult problems

Reference image – Wikipedia

Which fairness?

Demographic parity	Equal opportunity	Group fairness	**Individual fairness**

Definition 1 — Individual Fairness

M is fair $\Leftrightarrow$ no pair (x, x') satisfies all of:

① $x_i = x'_j$ for all $i \in A \backslash P$ (same non-protected attributes)

② $x_j \neq x'_j$ for some $j \in P$ (differ in a protected attribute)

③ $M(x) \neq M(x')$ (different prediction)

➔ **same except protected attributes → same prediction**

2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)

Fairify: Fairness Verification of Neural Networks

Sumon Biswas
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA
sumonb@cs.cmu.edu

Hridesh Rajan
Dept. of Computer Science
Iowa State University
Ames, IA, USA
hridesh@iastate.edu

***Abstract*—Fairness of machine learning (ML) software has become a major concern in the recent past. Although recent research on testing and improving fairness have demonstrated impact on real-world software, providing fairness guarantee in practice is still lacking. Certification of ML models is challenging because of the complex decision-making process of the models. In this paper, we proposed *Fairify*, an SMT-based approach to verify individual fairness property in neural network (NN) models. Individual fairness ensures that any two similar individuals get similar treatment irrespective of their protected attributes e.g., race, sex, age. Verifying this fairness property is hard because of the global checking and non-linear computation nodes in NN. We proposed sound approach to make individual fairness verification tractable for the developers. The key idea is that many neurons in the NN always remain inactive when a smaller part of the input domain is considered. So, Fairify leverages white-box access to the models in production and then apply formal analysis based pruning. Our approach adopts input partitioning and then prunes the NN for each partition to provide fairness certification or counterexample. We leveraged interval arithmetic and activation heuristic of the neurons to perform the pruning as necessary. We evaluated Fairify on 25 real-world neural networks collected from four different sources, and demonstrated the effectiveness, scalability and performance over baseline and closely related work. Fairify is also configurable based on the domain and size of the NN. Our novel formulation of the problem can answer targeted verification queries with relaxations and counterexamples, which have practical implications.**

***Index Terms*—fairness, verification, machine learning**

I. Introduction

Artificial intelligence (AI) based software are increasingly being used in critical decision making such as criminal sentencing, hiring employees, approving loans, etc. Algorithmic fairness of these software raised significant concern in the recent past [1–8]. Several studies have been conducted to measure and mitigate algorithmic fairness in software [9–18]. However, providing formal guarantee of fairness properties in practice is still lacking. Fairness verification in ML models is difficult given the complex decision making process of the algorithms and the specification of the fairness properties [19–22]. Our goal in this paper is to enable the verification in real-world development and guarantee fairness in critical domains.

Albarghouthi et al. and Bastani et al. proposed probabilistic techniques to verify *group fairness* [20, 22]. Group fairness property ensures that the protected groups (e.g., male-vs-female, young-vs-old, etc.) get similar treatment in the prediction. On the other hand, *individual fairness* states that any two similar individuals who differ only in their protected attribute get similar treatment [21, 23]. Galhotra et al. argued that group fairness property might not detect bias in scenarios when same amount of discrimination is made for any two groups [23], which led to the usage of individual fairness property in many recent works [1, 8, 23–25]. John et al. proposed individual fairness verification for two ML classifiers, i.e., linear classifier and 2) kernelized classifier e.g., support vector machine [21], which is not applicable to neural networks.

We propose *Fairify*, the technique to verify individual fairness of NN models in production, i.e., already trained models. Both abstract interpretation [26–30] and satisfiability modulo theories (SMT) based techniques [31–33] have shown success in verifying different properties of NN such as robustness. However, fairness verification of NN on real-world models has received little attention. We adopted SMT based verification since it enables practical benefits, e.g., solving arbitrary verification query and providing counterexamples. Urban et al. proposed Libra, an abstract interpretation based *dependency fairness* certification for NN [27]. The approach can not check relaxation of fairness queries and does not provide counterexamples in case of a violation. Fairify on the other hand provides configurable options to the developers enabling fairness verification of NN in practice.

Verifying a property in NN is challenging mainly because of the presence of non-linear computation nodes i.e., activation functions [27, 32]. With the size of the NN, the verification task becomes harder and often untractable [34, 35]. Many studies have been conducted to verify NN for different local robustness properties [29, 31, 36]. However, the individual fairness property requires global checking which makes the verification task even harder and existing local property verifiers can not be used [37, 38]. To that end, we propose a novel technique that can verify fairness, i.e., provide satisfiability (SAT) with counterexample, or show UNSAT in a tractable time and available computational resource. Fairify takes a trained NN and the verification query as input. If the verifier can verify within the given timeout period, the output should be SAT (violation) or UNSAT (certification). When the verifier is unable to show any proof within the timeout, the result is UNK (unknown). Our evaluation shows that using state-of-the-art SMT solver, we cannot verify the fairness property of the NN models in days; however, Fairify can verify most of the verification queries within an hour. Fairify makes the

1558-1225/23/$31.00 ©2023 IEEE
DOI 10.1109/ICSE48619.2023.00134

But can art have a criterion?

Is this painting beautiful?

Is this brush texture good?

Reference image – Wikipedia

Even art can be made into criteria

Stroke style

Abstraction

Color (HSV)

$$cartoon = G(photo, \alpha),$$
$$\alpha = \{\alpha_s\,(stroke), \alpha_a(abstraction)\}$$

$$L = L_{texture} + L_{color}$$

Interactive Cartoonization with Controllable Perceptual Factors

Namhyuk Ahn[1] Patrick Kwon[1] Jihye Back[1] Kibeom Hong[2] Seungkwon Kim[1]
[1] NAVER WEBTOON AI [2] Yonsei University

Figure 1. **Interactive cartoonization.** The proposed cartoonization method allows user interaction over texture and color and can generate diverse outputs that meet users' demands. Leftmost: source photography and an exemplar image of the target cartoon.

Abstract

Cartoonization is a task that renders natural photos into cartoon styles. Previous deep cartoonization methods only have focused on end-to-end translation, which may hinder editability. Instead, we propose a novel solution with editing features of texture and color based on the cartoon creation process. To do that, we design a model architecture to have separate decoders, texture and color, to decouple these attributes. In the texture decoder, we propose a texture controller, which enables a user to control stroke style and abstraction to generate diverse cartoon textures. We also introduce an HSV color augmentation to induce the networks to generate diverse and controllable color translation. To the best of our knowledge, our work is the first deep approach to control the cartoonization at inference while showing profound quality improvement over to baselines.

1. Introduction

Cartoons gain steep popularity in a recent, and the number of cartoon creators have also increased. The universal workflow of cartoon painters is as follows: character drawing, which is then composed into a background scene. Post-processing such as shading is added afterward. Professional tools [1, 3] provide helpful plugins to assist the artist. Despite this, cartoon creation still remains an arduous task even for the more skilled creators.

We follow the observation that cartoon-styled scene generation has received notable attention. Many artists convert real-world photographs into cartoon styles to utilize as a background scene, dubbed as *image cartoonization*. This allows creators to more focus on effective decisions in making cartoons, such as character generation. It is shown that deep learning-based cartoonization approach is able to produce cartoon-stylized output with a prominent quality, that is possible to be utilized in real service production [5, 6, 19].

However, the previous deep methods skip the intermediate procedures of cartoon-making processes, thus disabling the creators from controlling outputs. The artists follow a series of structured steps when creating a cartoon background from a photo (Figure 2). **1)** Color stylization, where the author changes the color both locally and globally. Sky synthesis is performed along with this procedure. **2)** Texture stylization, where additional sketch lines are drawn, and fine details are selectively removed to achieve the different

Good writing and good painting

Example:

Good writing — Writing that meets fairness criteria

Good painting — Painting that meets brushwork criteria

KEY TAKEAWAY

In short, automation is
ultimately the history of finding good criteria

We are still creating criteria today

Expansion of the number system

- Numbers
- Introduction of zero
- Extended numbers

- Numerical analysis

Expansion of AI evaluation criteria

- F1
- Fairness
- ?

- ?

The people who matter in the AI era

create the criteria for automation

AI optimizes the criterion

Humans invent it

4

Experience and Final Lessons

Formal Computing and AI Lab.

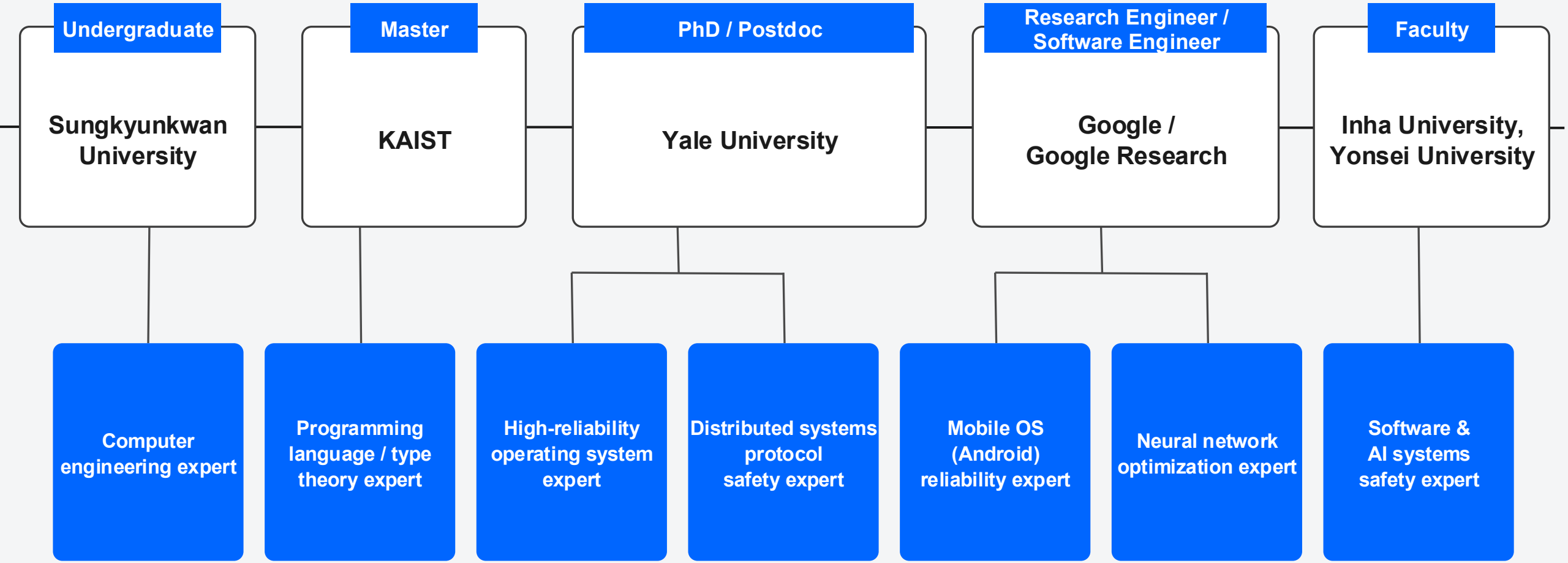
Assistant Professor

Jieung Kim is a tenure-track assistant professor in Department of Computer Science and Engineering, College of Computing, Yonsei University

Jieung Kim worked at **Google (Research - Personal AI Privacy & Security - and Core ML - Model Optimization - teams)** and **Inha University** (Assistant professor in Computer Engineering, College of Software & Convergence) before joining Yonsei University. Before that, he received his degrees from **Yale University (Ph.D)**, **KAIST (M.S)**, and **Sungkyunkwan University (B.S.)**.

My career kept changing

My works also kept changing



But there is one thing that never changed

Undergraduate	Master	PhD / Postdoc	Research Engineer / Software Engineer	Faculty
Sungkyunkwan University	KAIST	Yale University	Google / Google Research	Inha University, Yonsei University

🔒 **Making complex systems trustworthy**

🤝 **Building “criteria for safety” based on logic and math**

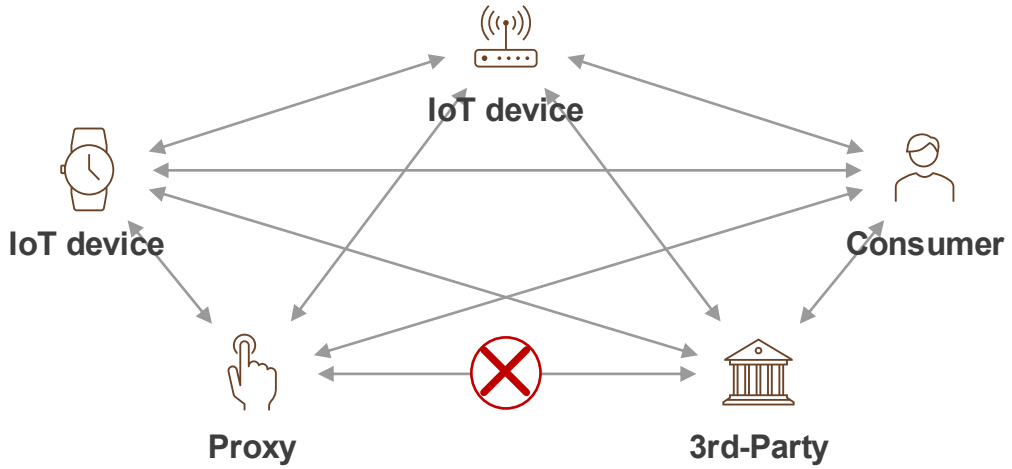
- Computer engineering expert
- Programming language / type theory expert
- High-reliability operating system expert
- Distributed systems protocol safety expert
- Mobile OS (Android) reliability expert
- Neural network optimization expert
- Software & AI systems safety expert

The problem that I worked on

Distributed systems

- Backbone of several internet services

✕ **Failures can occur**



The criterion that I built

The question is how to judge

the correctness of general distributed protocol

GenPullSuccess

$$\frac{O_{pull}(log, nid) = Ok(time, cid) \quad timeOf(cid) < time \quad noOwnerAt(log, time) \quad (cid \in caches(log) \vee cid = root(log))}{O_{pull} \vdash pull(nid) : log \longrightarrow log \bullet Pull^{+}(nid, time, cid)}$$

GenPullPreempt

$$\frac{O_{pull}(log, nid) = Preempt(time) \quad time \notin dom(owners(log))}{O_{pull} \vdash pull(nid) : log \longrightarrow log \bullet Pull^{*}(nid, time)}$$

GenMethodInvocation

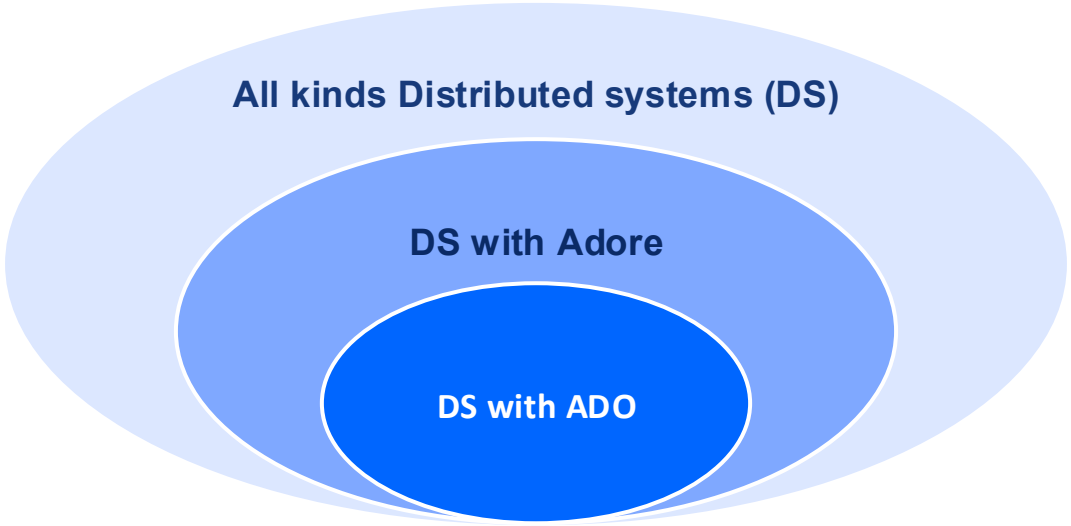
$$\frac{cids(log)[nid] \in caches(log)}{\vdash M(nid) : log \longrightarrow log \bullet Invoke^{+}(nid, M)}$$

GenPushSuccess

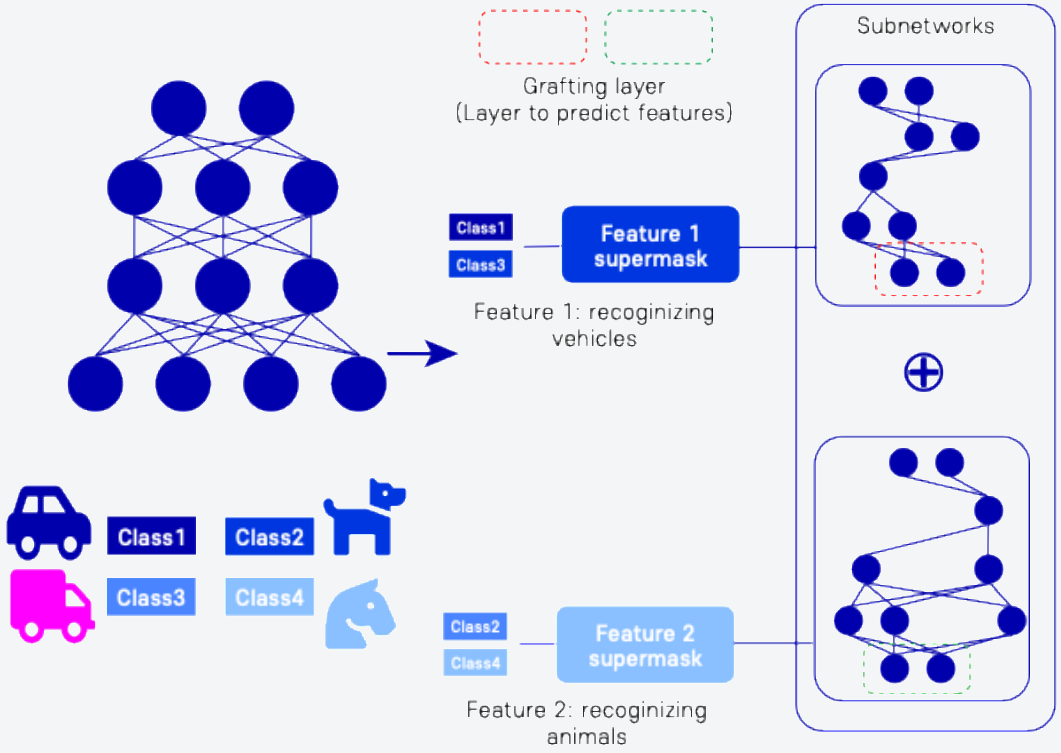
$$\frac{O_{push}(log, nid) = Ok(ccid) \quad nidOf(ccid) = maxOwner(log) = nid \quad timeOf(ccid) = timeOf(cids(log)[nid]) \quad ccid \in caches(log)}{O_{push} \vdash push(nid) : log \longrightarrow log \bullet Push^{+}(nid, ccid)}$$

Extending the criterion: ADO → Adore

	ADO (OOPSLA 2021)	Adore (PLDI 2022)
Failure model	Benign (crash / network)	Benign + reconfiguration
Key advance	Fault-aware atomic object; exposes key failures	Dynamic membership change (reconfiguration)



The problem I'm currently solving



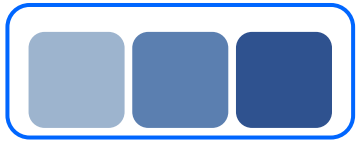
The criterion I'm currently building

Definition 1 (Decompositionality). *F_θ is $(\varepsilon, \tau, m, c, \eta)$-decomposable w.r.t. cover $\mathcal{P}$ if some $F_\theta^{\downarrow}$ satisfies (i) (ε, τ)-boundary-aware semantic fidelity and (ii) operational separability, i.e. per-module target movement $\mathrm{Mov}(k) \geq m$, cross-collateral $\mathrm{Coll}(k) \leq c$, and reduction at least η.*

A new criterion in the making

Structural divergence

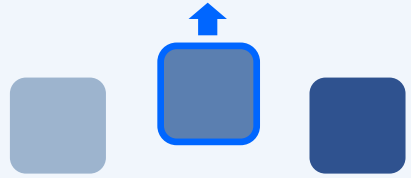
Do the parts look different inside?



Single for all

Operable separability

Can you operate on one part by itself?



Separate criteria for each

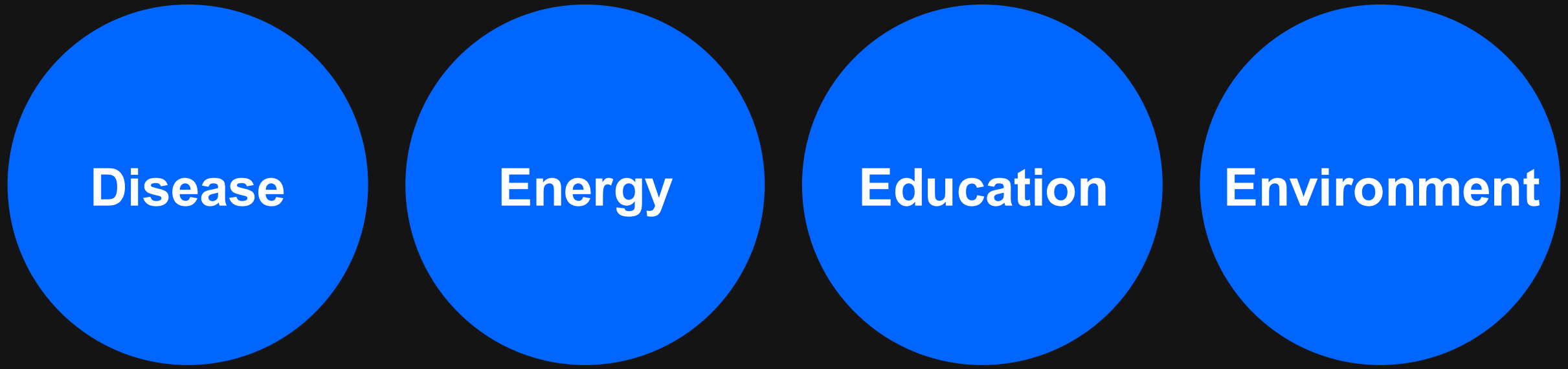
Technology keeps changing

New technology

New systems

New problems

But the problems remain.



AI optimizes criteria

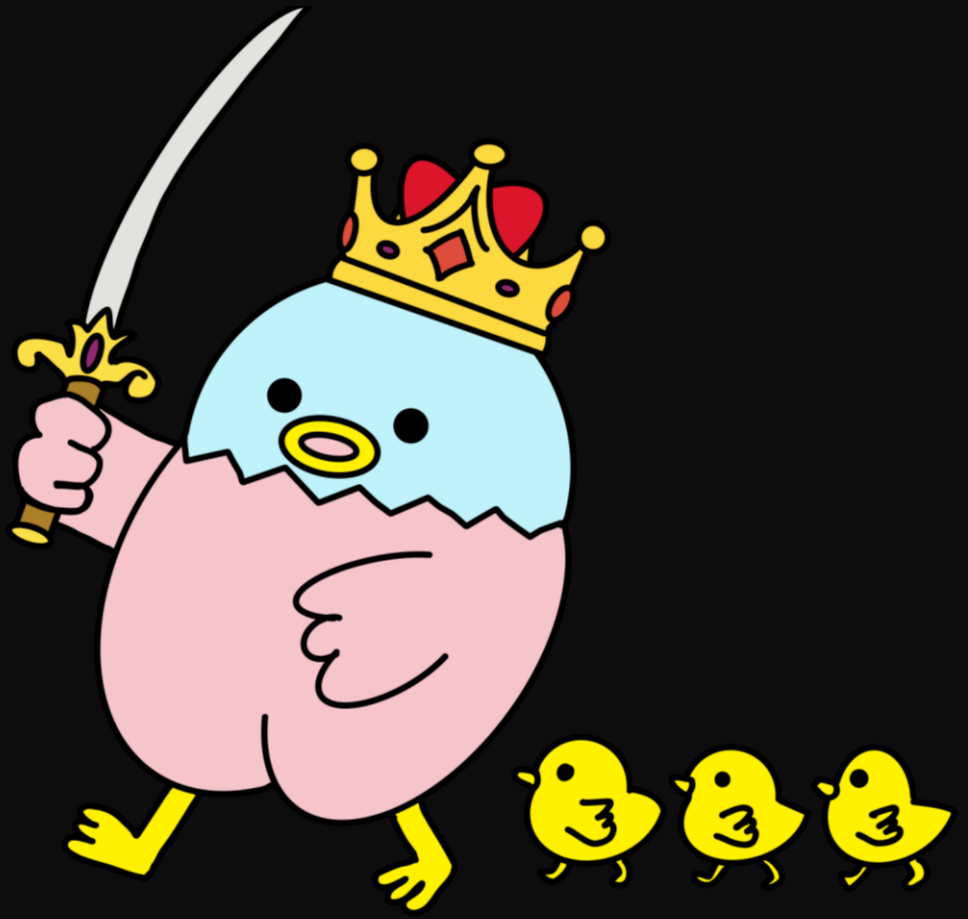
but it is people

who create and

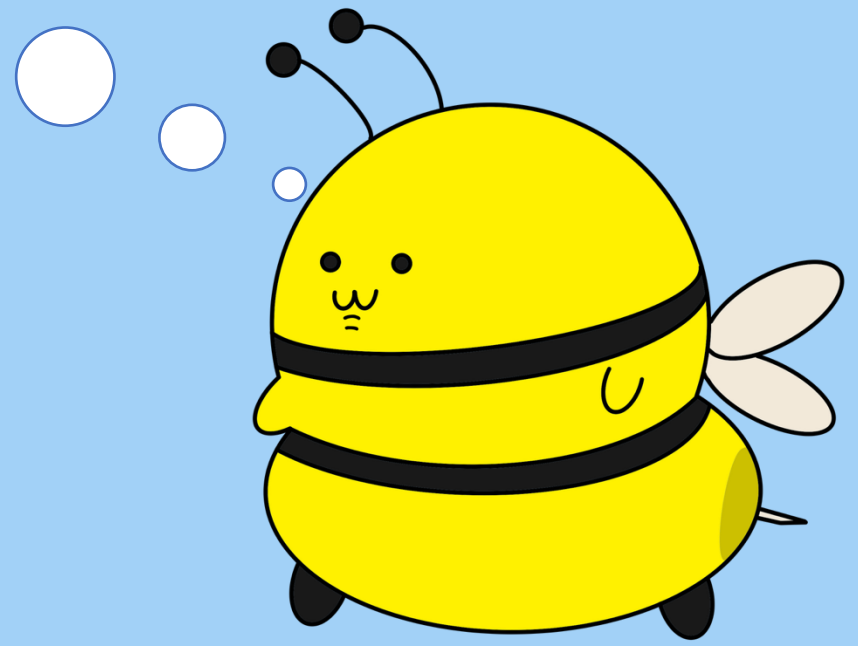
modify them

Techniques & Jobs — don't chase them

Problems — dig deep into them



Thank you



If you have any questions in the future, contact me:

via jieungkim@yonsei.ac.kr (for official works)

or **@hasutudio** (Insta for casual chats)