AI Factory Austria AI:AT

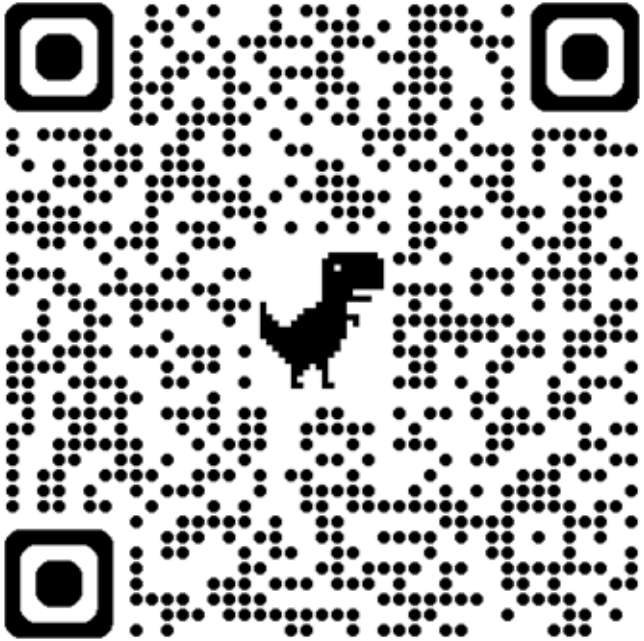
Modern Computer Vision & Knowledge Distillation

Shaping the Future of AI

Dejan Đukić
Senior AI/ML Engineer, TetraScience

10.03.2026

Welcome!



Intro form; 30s

Agenda

Day 1: What's Possible

1. Intro
2. What Actually Goes Wrong — Pitfalls that cost real money
3. How It Works — CLIP, ViT, and zero-shot detection
4. SAM3 Deep Dive — Prompts, failures, and the negation paradox
5. Distillation — From foundation model to edge device
6. Q&A + Day 2 Setup

- The slides & the recording will be shared

Agenda

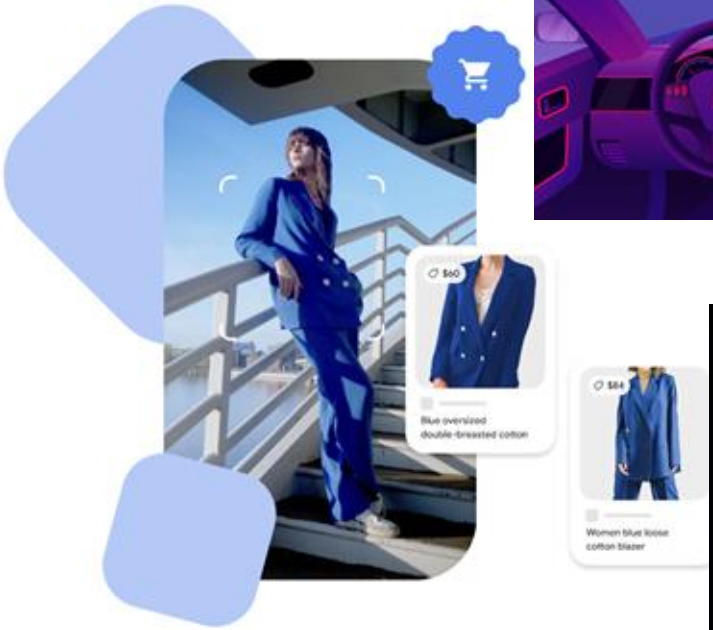
Day 2: Hands on

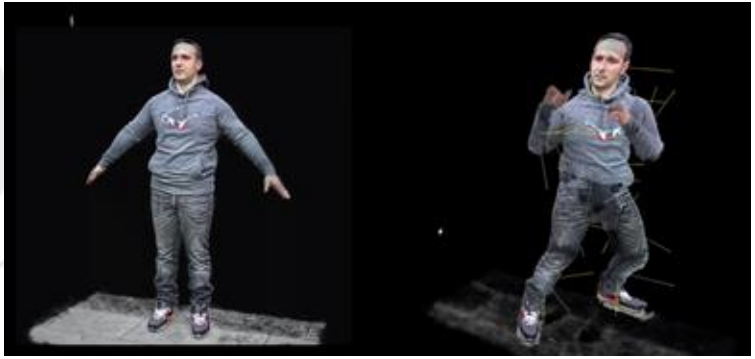
- Applying concepts introduced in day 1
- Computer vision challenge / experiment end-to-end

Overall workshop goals:

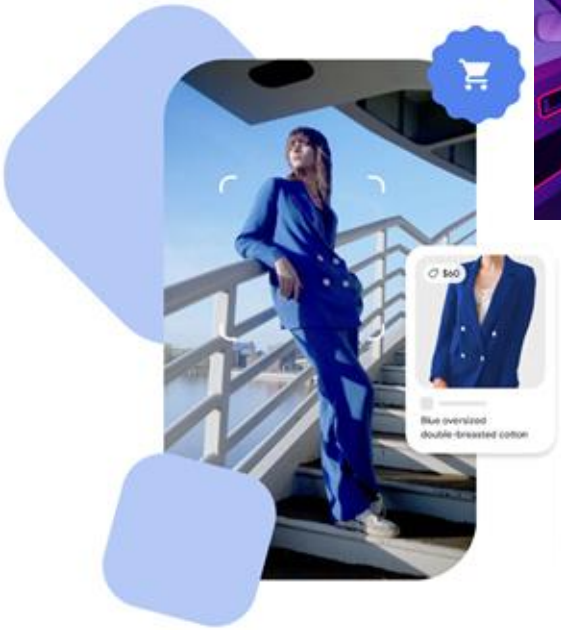
- Understand the intuition for what is ‘under the hood’ of modern computer vision
- Understand how to go from 0 to 0.8 in a computer vision project in a day
    - 80/20 rule

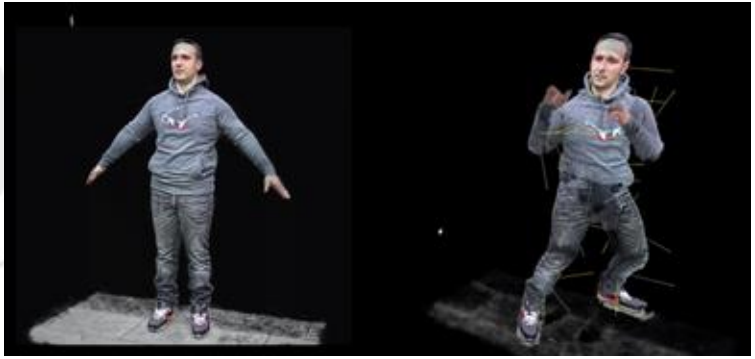
Computer vision



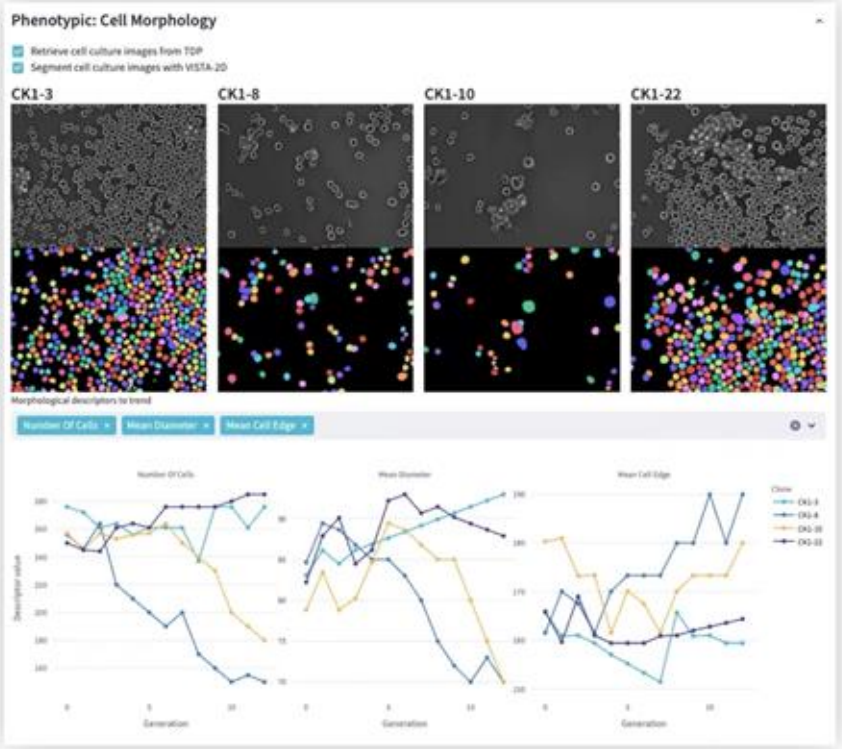


Computer vision



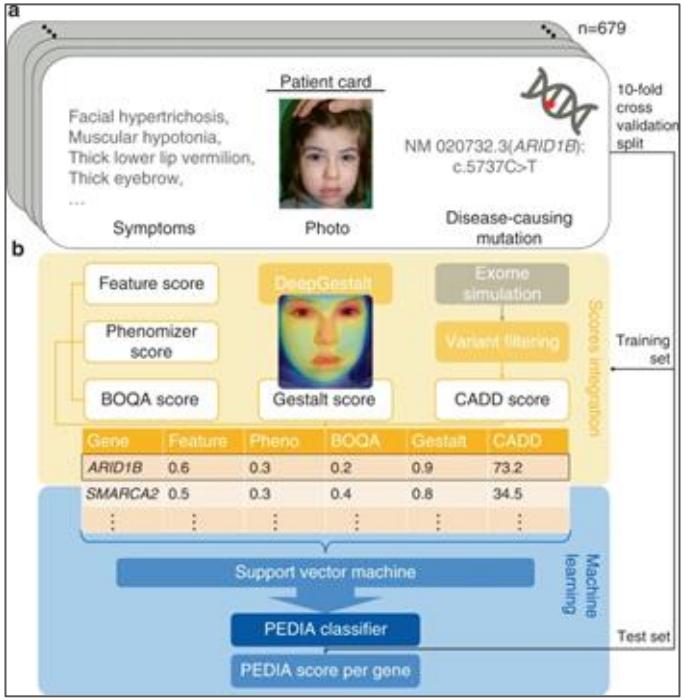


Intro; Who am I?

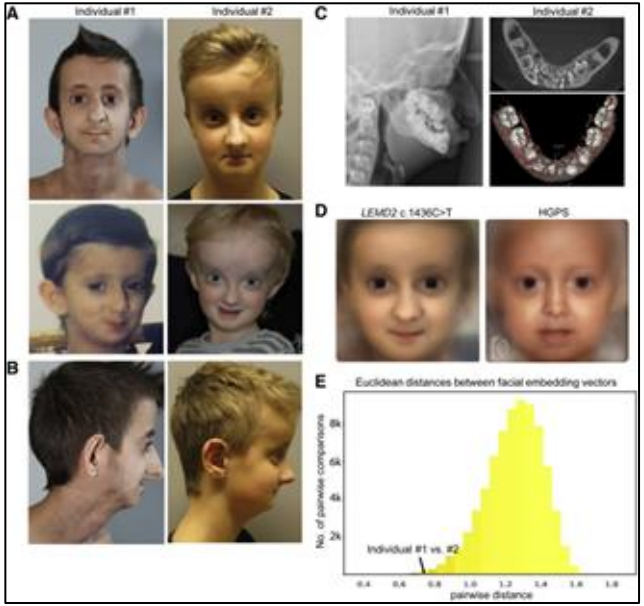


- Senior applied AI/ML at TetraScience
- Applying AI across the pharma pipeline;
  - drug discovery / drug manufacturing / clinical trials support
- Example: cellular morphology analysis as a proxy of cellular viability in biologics optimization experiments
- Linkedin

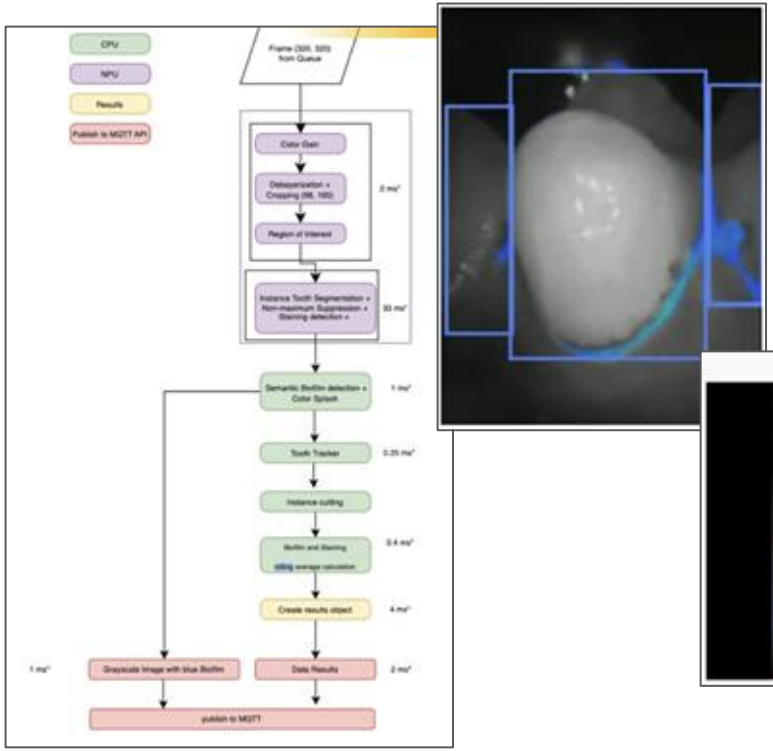
Intro; Who am I?

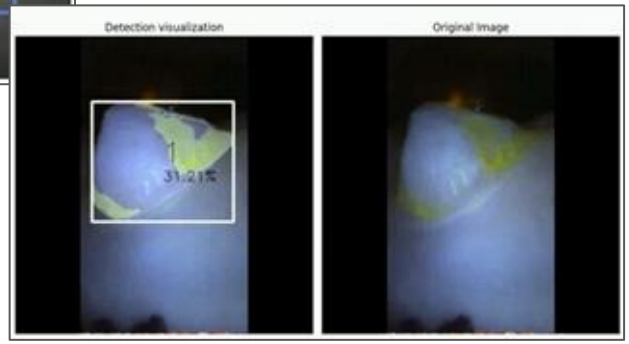


- Image & gene mutation signal fusion in medical diagnostics of rare genetic diseases



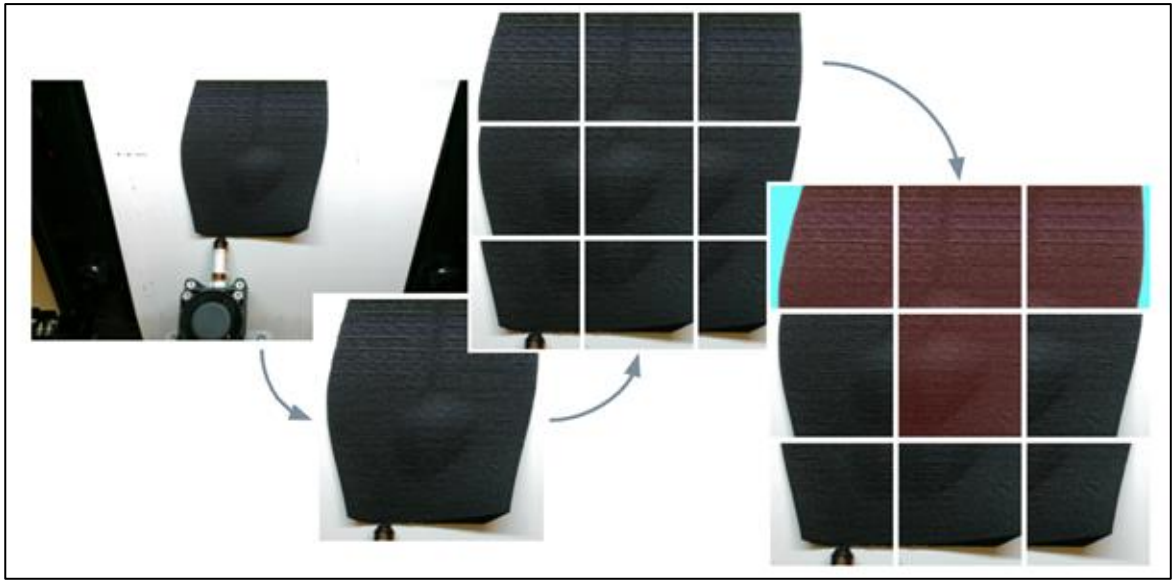
Intro; Who am I?





- Medical device edge stream processing

Intro; Who am I?



- Anomaly detection on high resolution images

Intro; Who am I?

- Anomaly detection on high resolution images

Intro; Who am I?

- Anomaly detection on high resolution images

Conceptual Foundations

The Cold Start Problem

Four concepts make it disappear

Five years ago:

- No data → No labels → No model → No project

Today:

- No data → Foundation model? → prototype

Four pillars towards foundation models in:

- 1. CLIP - the shared language
- 2. Vision Transformers - the better eyes
- 3. Open Vocabulary Detection - from "what" to "where"
- 4. Distillation - from cloud to edge (and beyond)

The Cold Start Problem

Four concepts make it disappear

Five years ago:

- No data → No labels → No model → No project

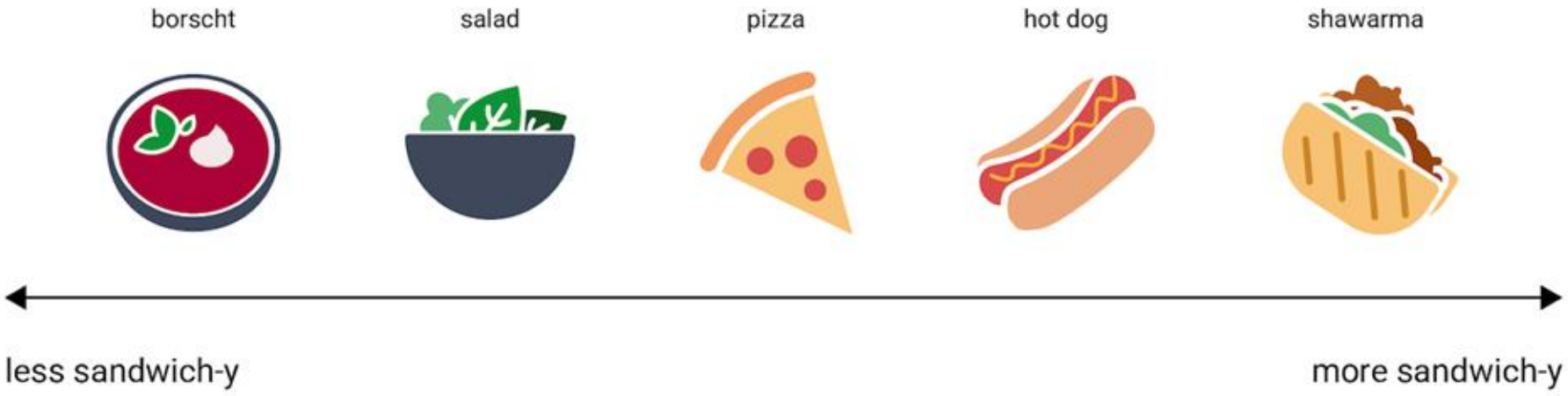
Today:

- No data → Foundation model? → prototype

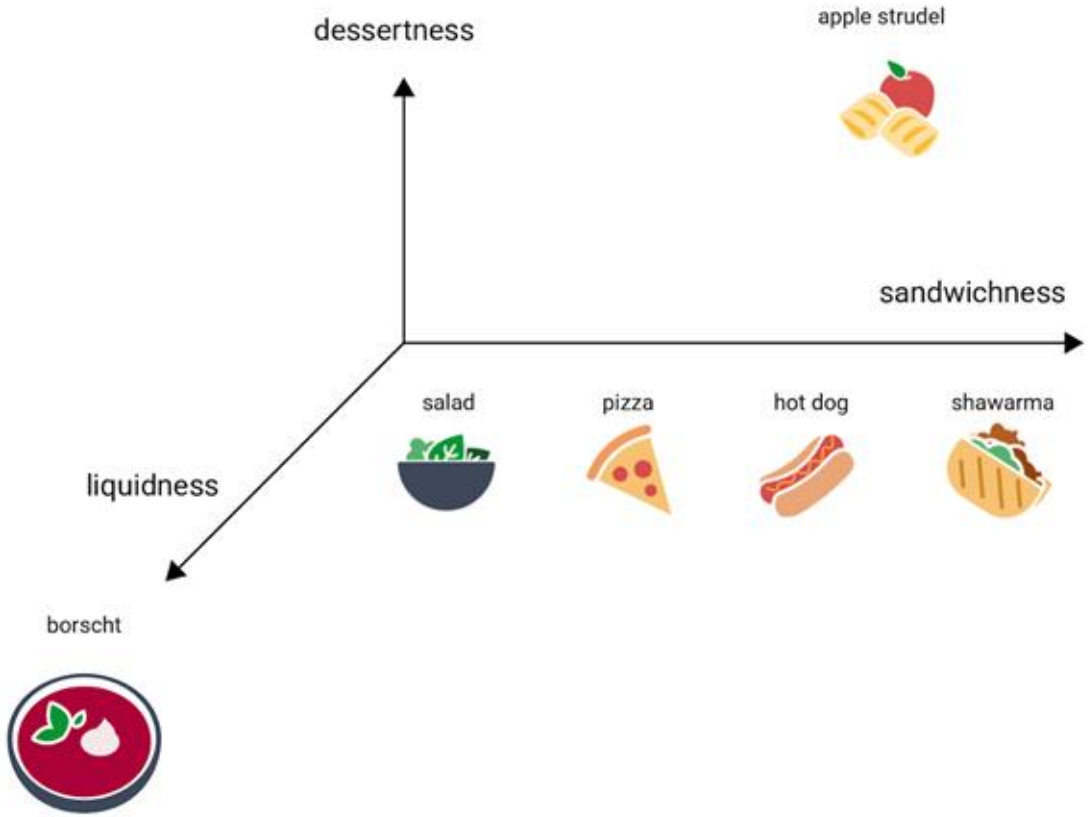
Four pillars towards foundation models in:

- 1. CLIP - the shared language
- 2. Vision Transformers - the better eyes
- 3. Open Vocabulary Detection - from "what" to "where"
- 4. Distillation - from cloud to edge (and beyond)
- (5. Diffusion - literally invent your data)

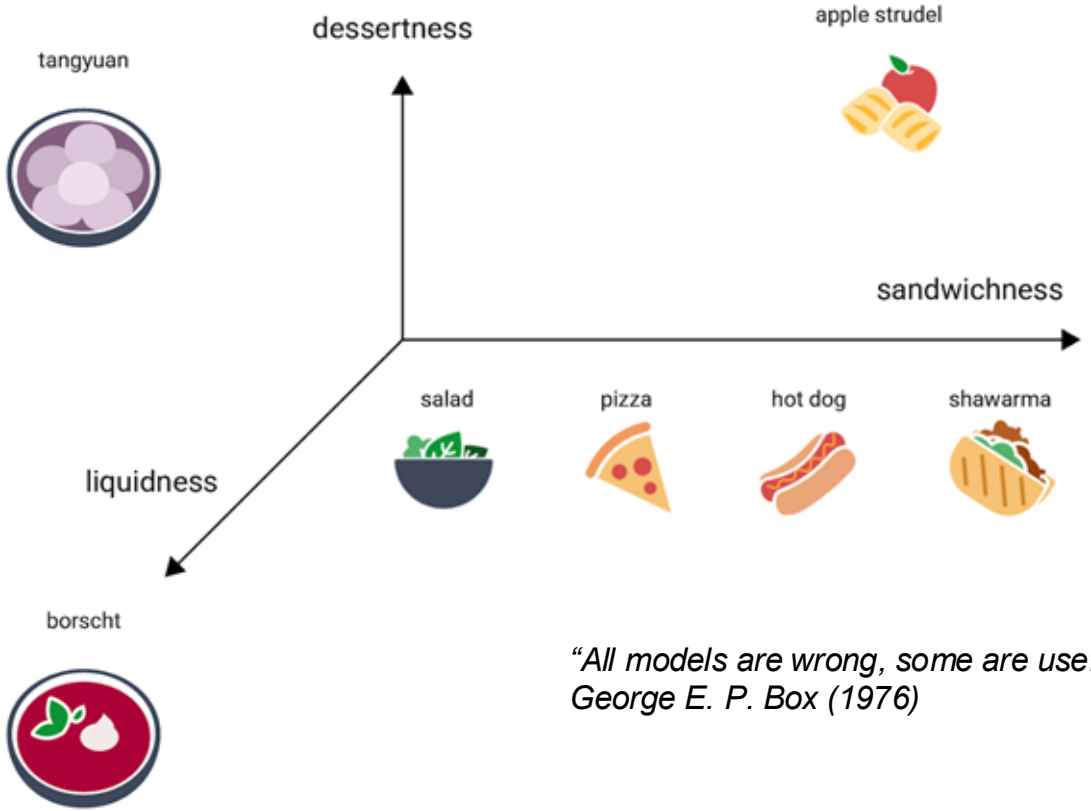
Conceptual foundations; CLIP (embeddings)



Conceptual foundations; CLIP (embeddings)

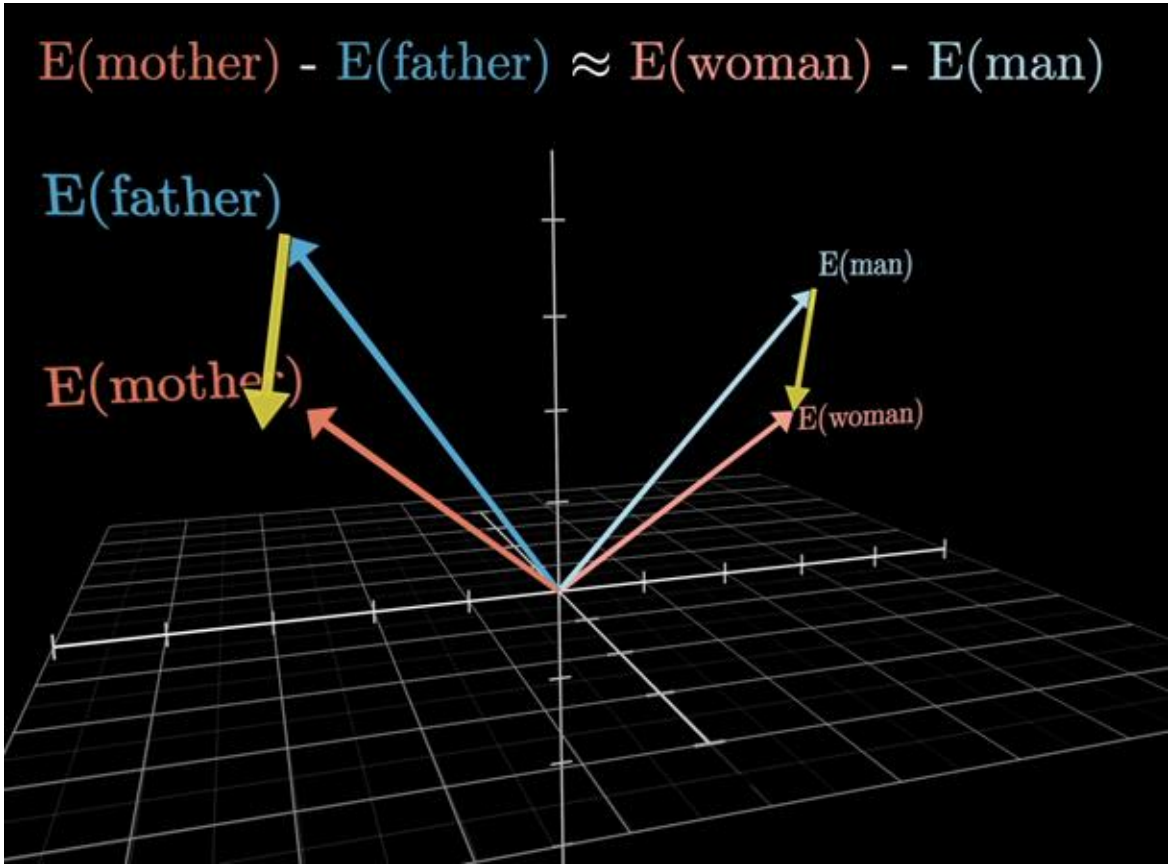


Conceptual foundations; CLIP (embeddings)

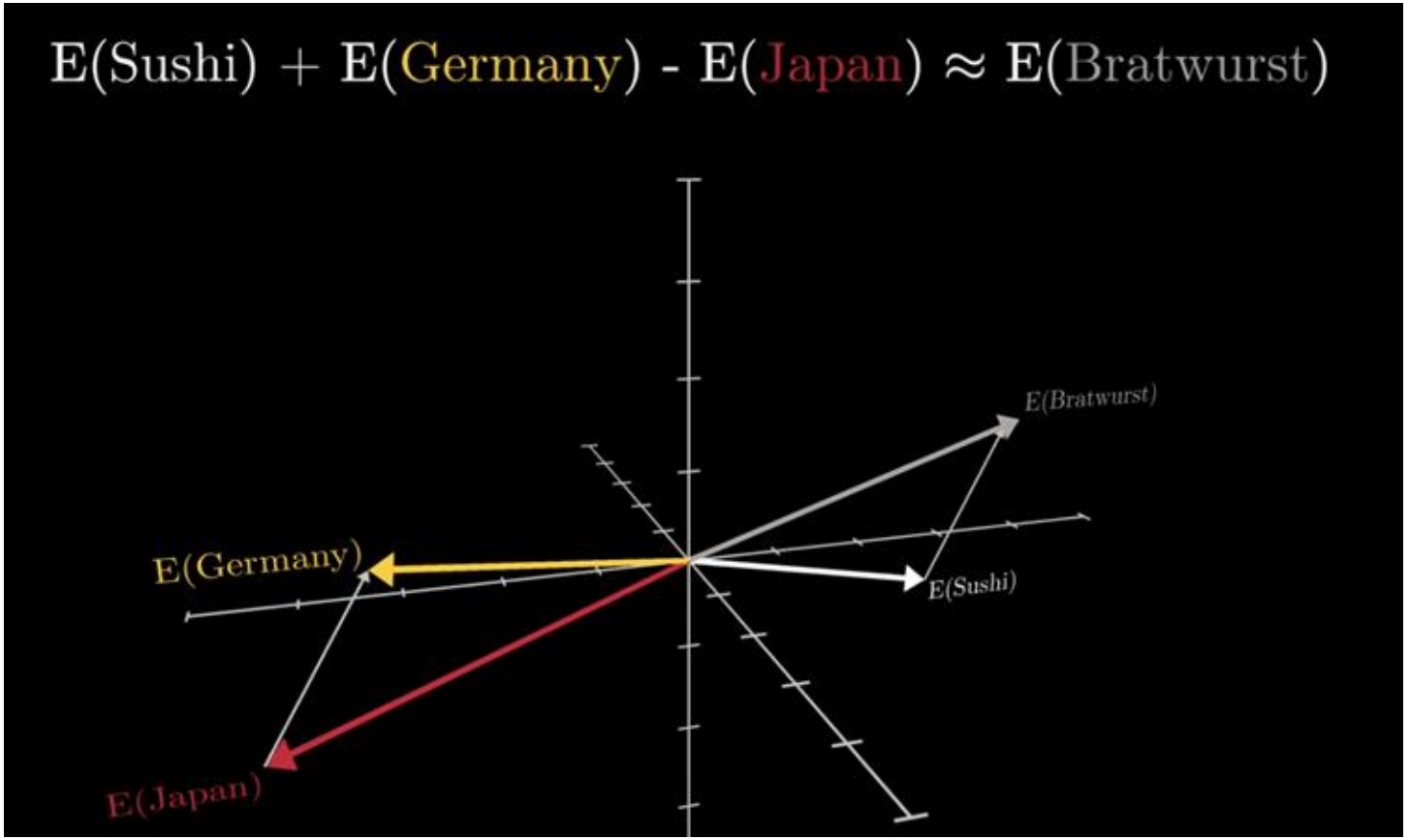


“All models are wrong, some are useful”
George E. P. Box (1976)

Conceptual foundations; CLIP (embeddings; word2vec)



Conceptual foundations; CLIP (embeddings; word2vec)

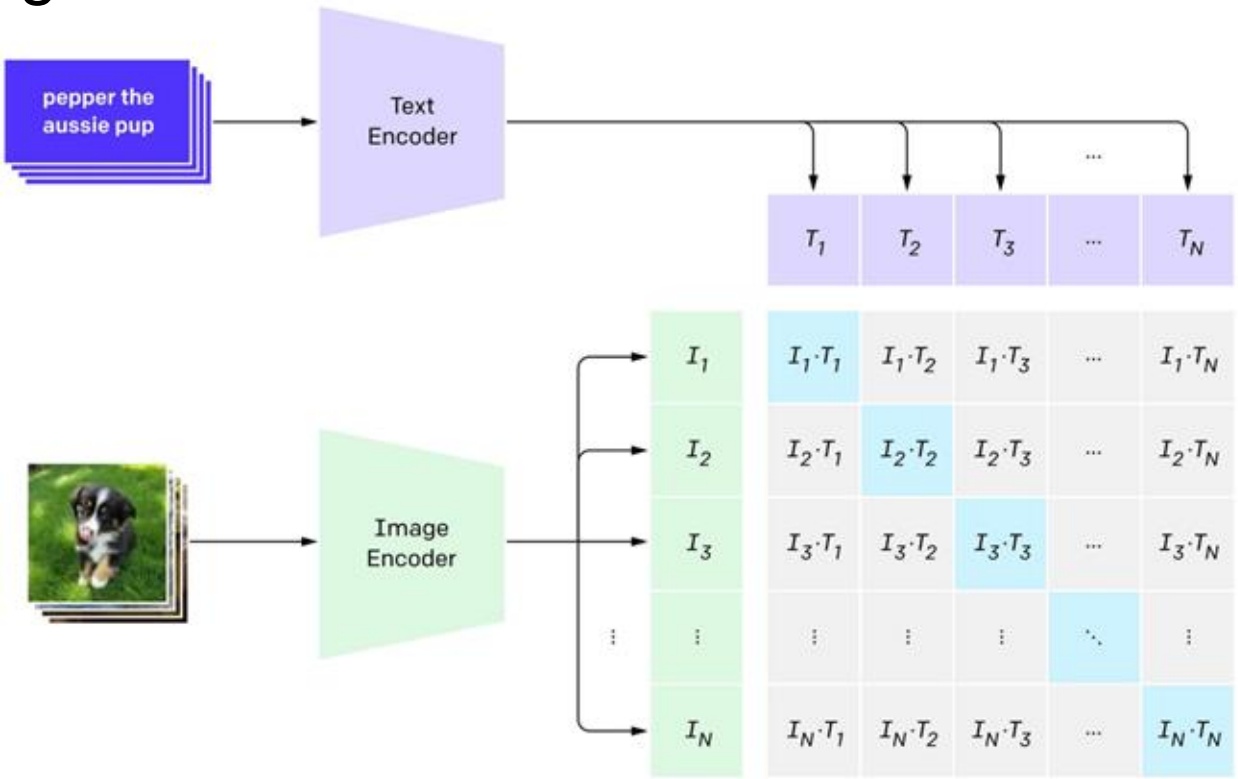


Conceptual foundations; CLIP

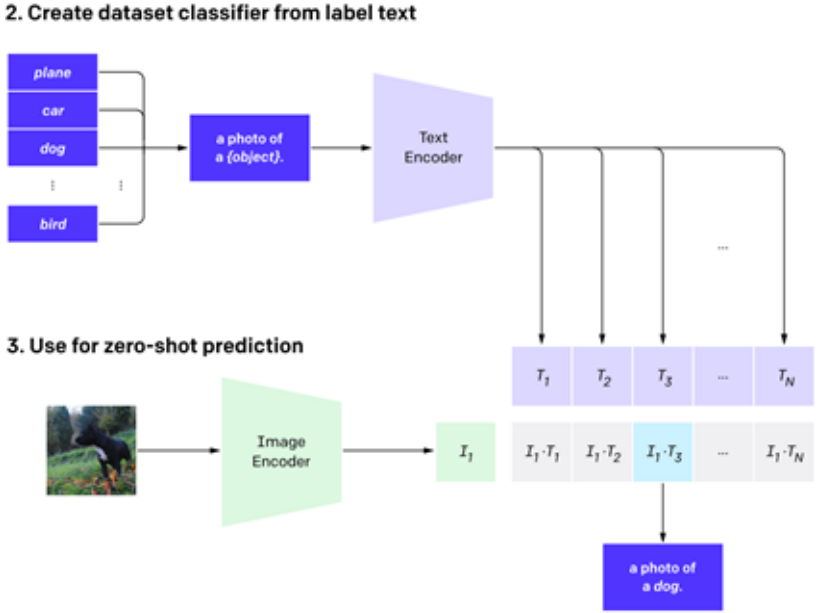
- Standard CV:
  - 1. Make dataset → 2. Annotate your classes → 3. Train a model to detect your classes

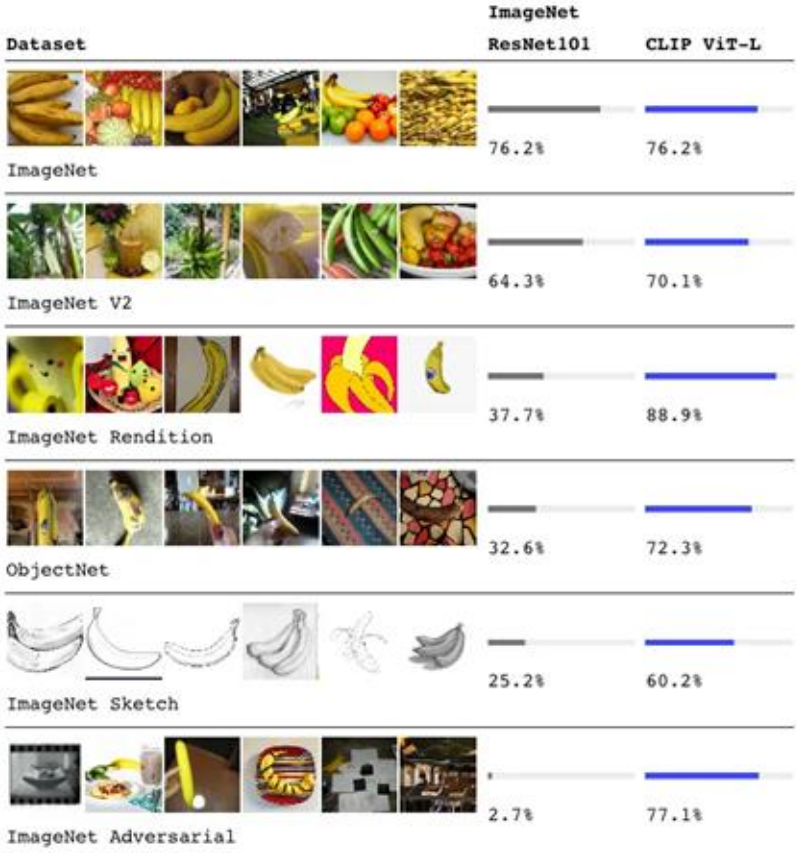
- Major limitations:
  - Constantly need specific datasets for specific things (and making them is expensive)
  - Constrained to exactly the classes you chose → “closed set”
  - Brittle, primitive pattern matching
    - Constant transfer learning experiments

Conceptual foundations; CLIP - Contrastive Language–Image Pre-training



Conceptual foundations; CLIP - Contrastive Language–Image Pre-training

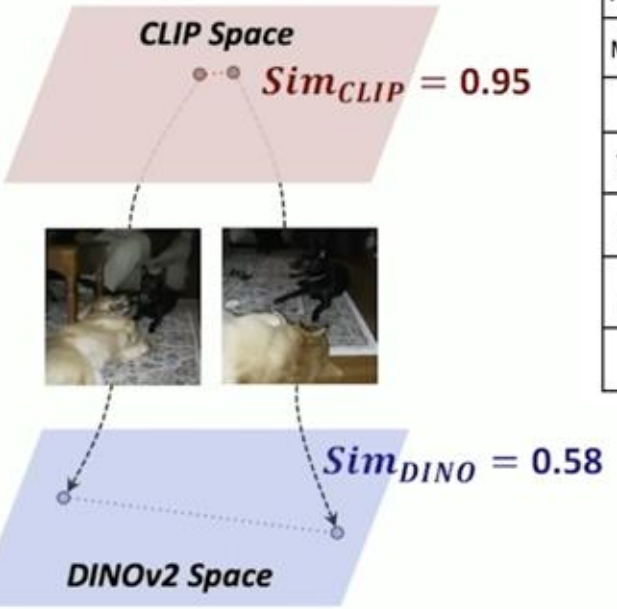


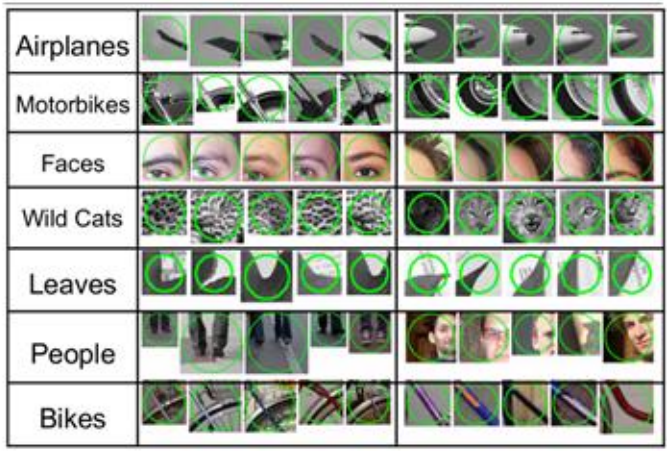


Conceptual foundations; CLIP - Contrastive Language–Image Pre-training

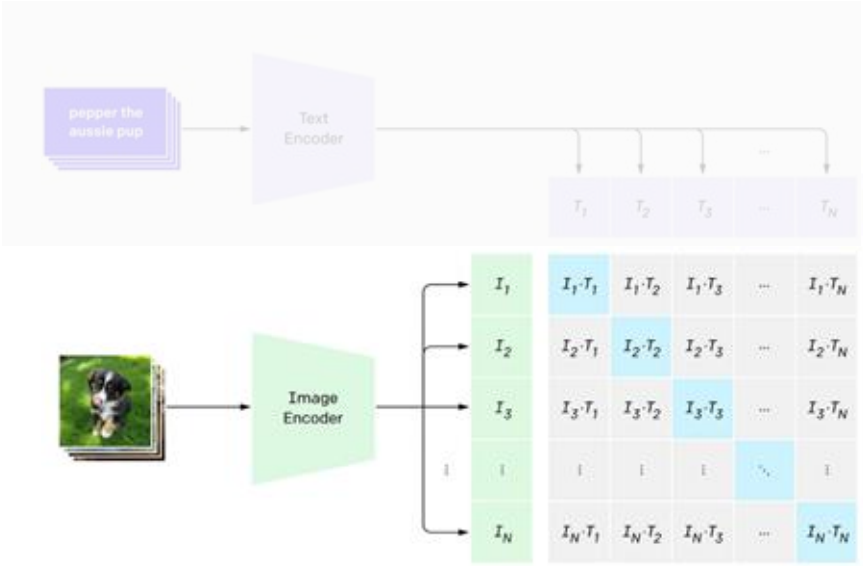
Finding CLIP-blind pairs.

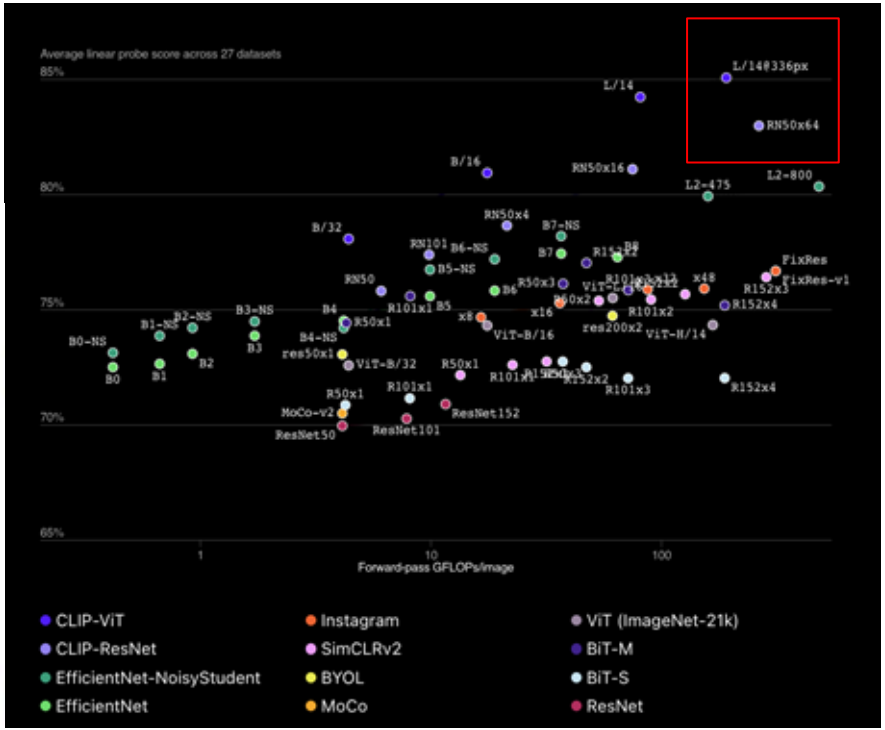
Discover image pairs that are proximate in CLIP feature space but distant in DINOv2 feature space.





Conceptual foundations; CLIP - Contrastive Language–Image Pre-training





CNN → Vision Transformer

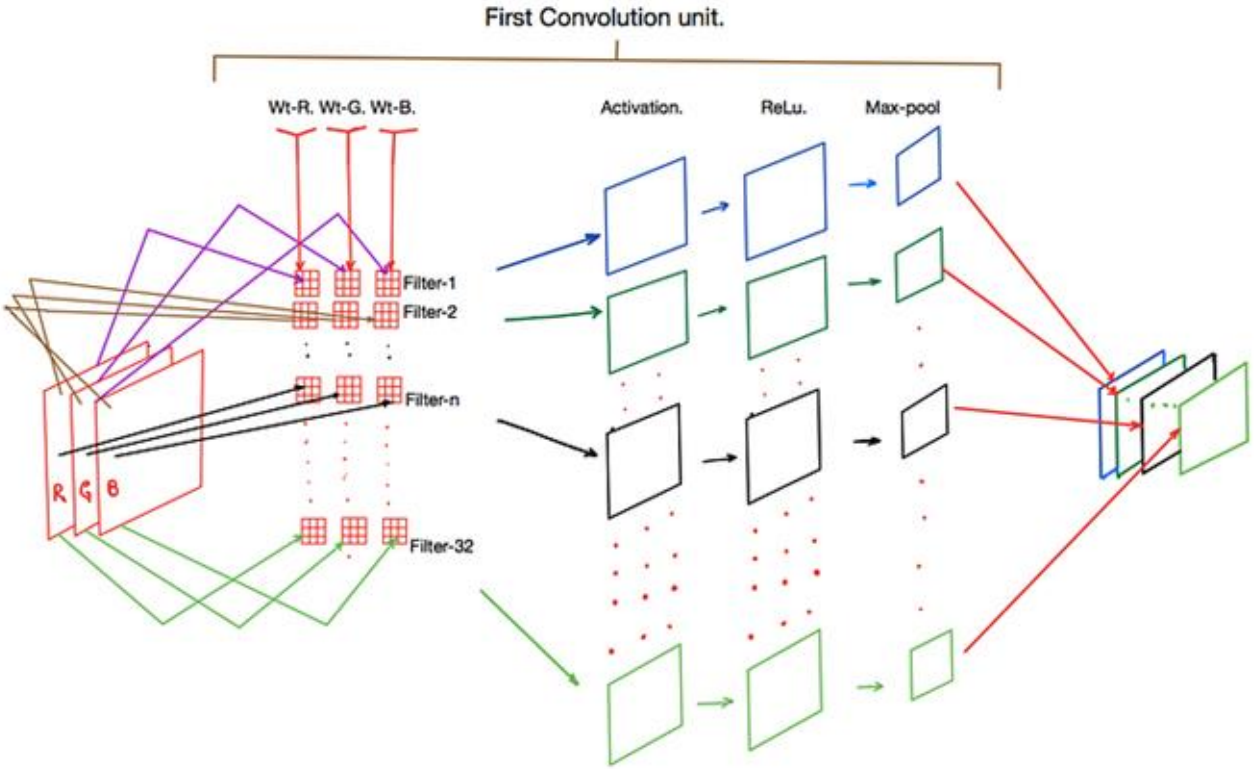
Pillar 2: The Better Eyes

- CNN: "Looking through a drinking straw"
- 3×3 kernel → local features only
- Layer 1: edges → Layer 10: textures
- Layer 30: parts → Layer 50: objects (sequential)

- ViT: "Chop the image into puzzle pieces (and figure out how they relate to one another)"
- 224×224 → 196 patches (16×16 each)
- Each patch = one token (same as GPT)
- Top-left talks to bottom-right in FIRST layer. (parallel)

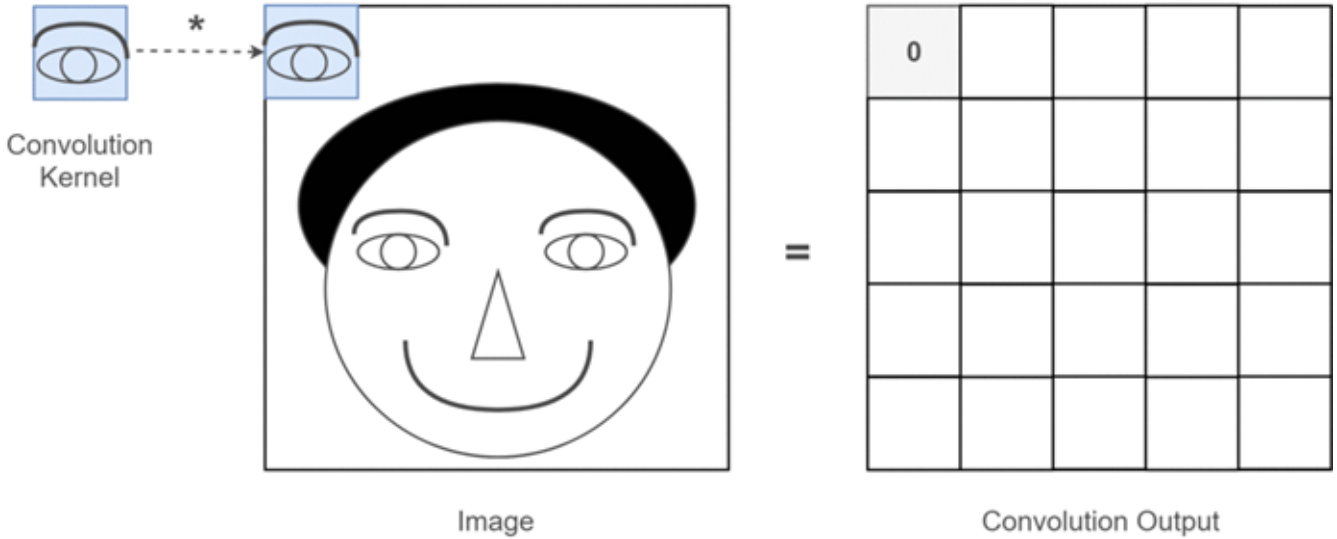
CNN → Vision Transformer

Pillar 2: The Better Eyes



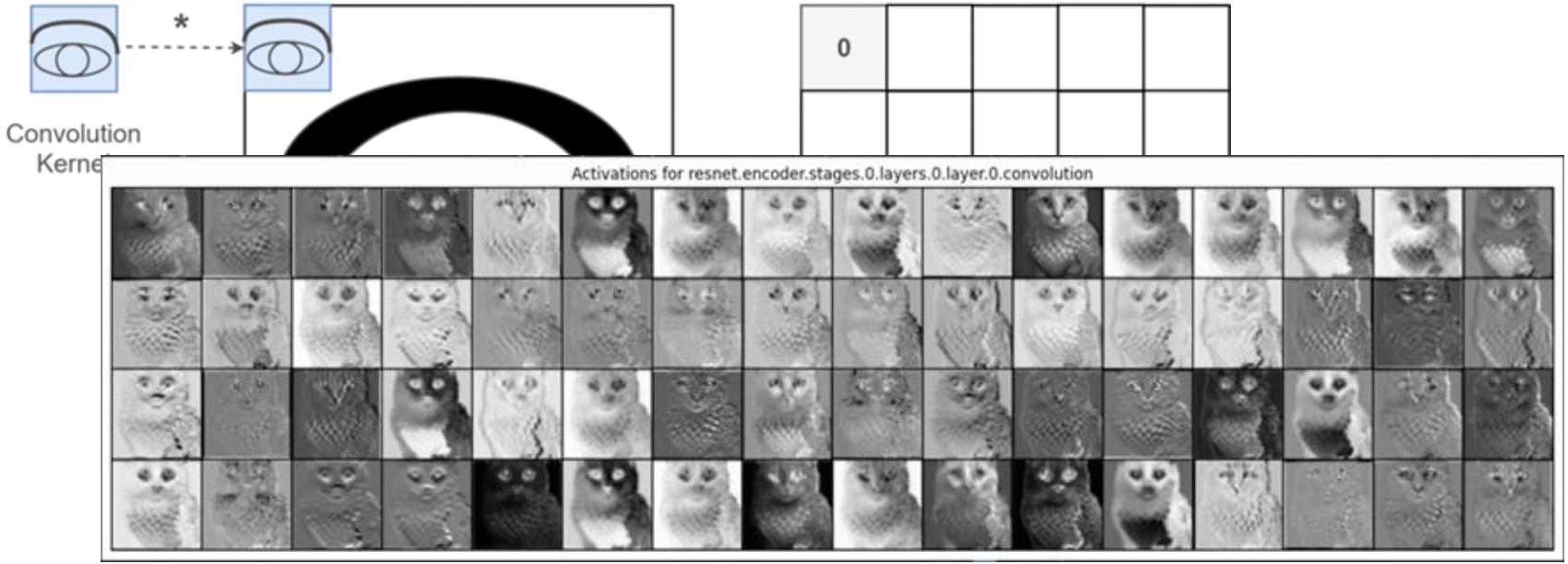
CNN → Vision Transformer

Pillar 2: The Better Eyes



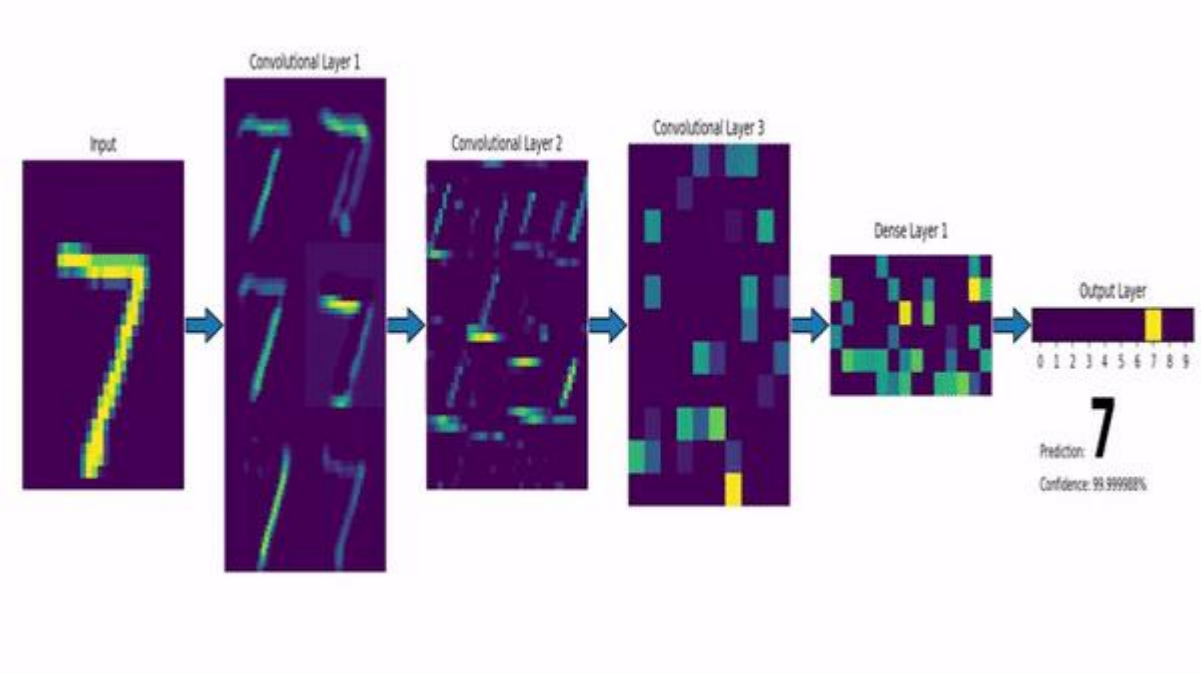
CNN → Vision Transformer

Pillar 2: The Better Eyes



CNN → Vision Transformer

Pillar 2: The Better Eyes

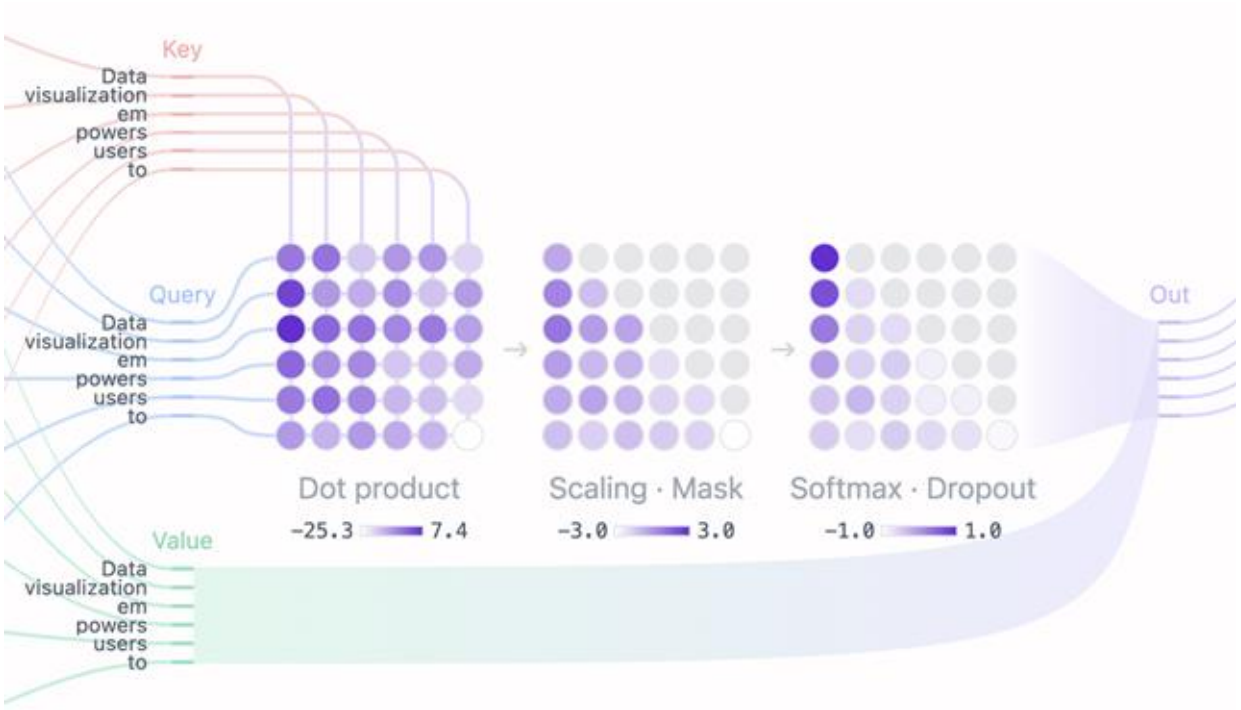


CNN → Vision Transformer

Pillar 2: The Better Eyes

CNN → Vision Transformer

Pillar 2: The Better Eyes



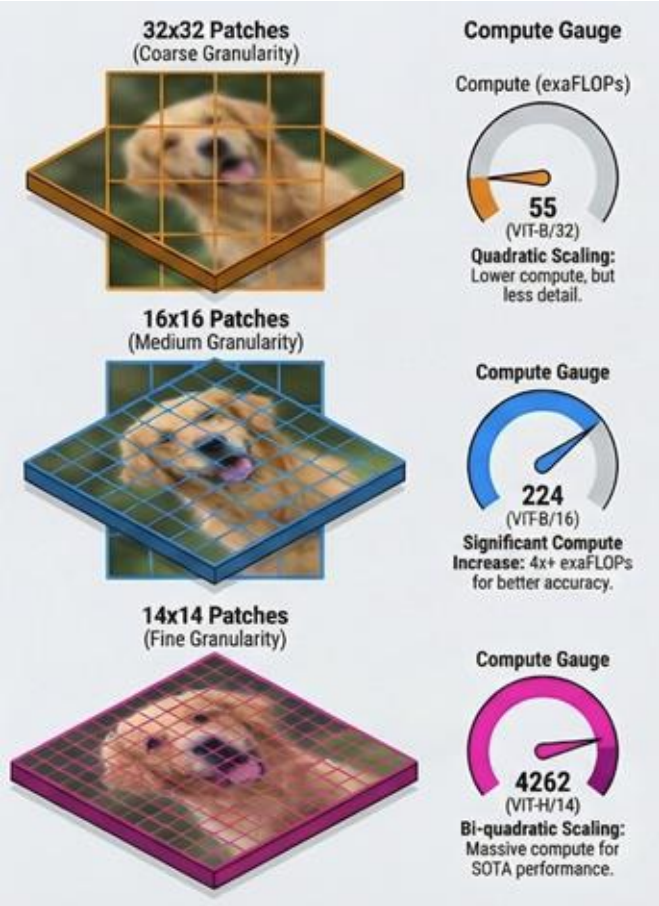
CNN vs ViT in Practice

"Is this a bottle or a cap?"

- CNN approach:
- pixels → edges → shapes → cylinder → bottle → yes
- 50 layers of local processing

- ViT approach:
- [bottle patch] ←— attention —→ [cap patch]
- One attention step. Sees both at once.

- The tradeoff:
- ViTs are data-hungry. No built-in spatial bias.
- CLIP's 400M pairs was the breakthrough.



From "What" to "Where"

Pillar 3: Open Vocabulary Detection; the era of SAM (Segment Anything)

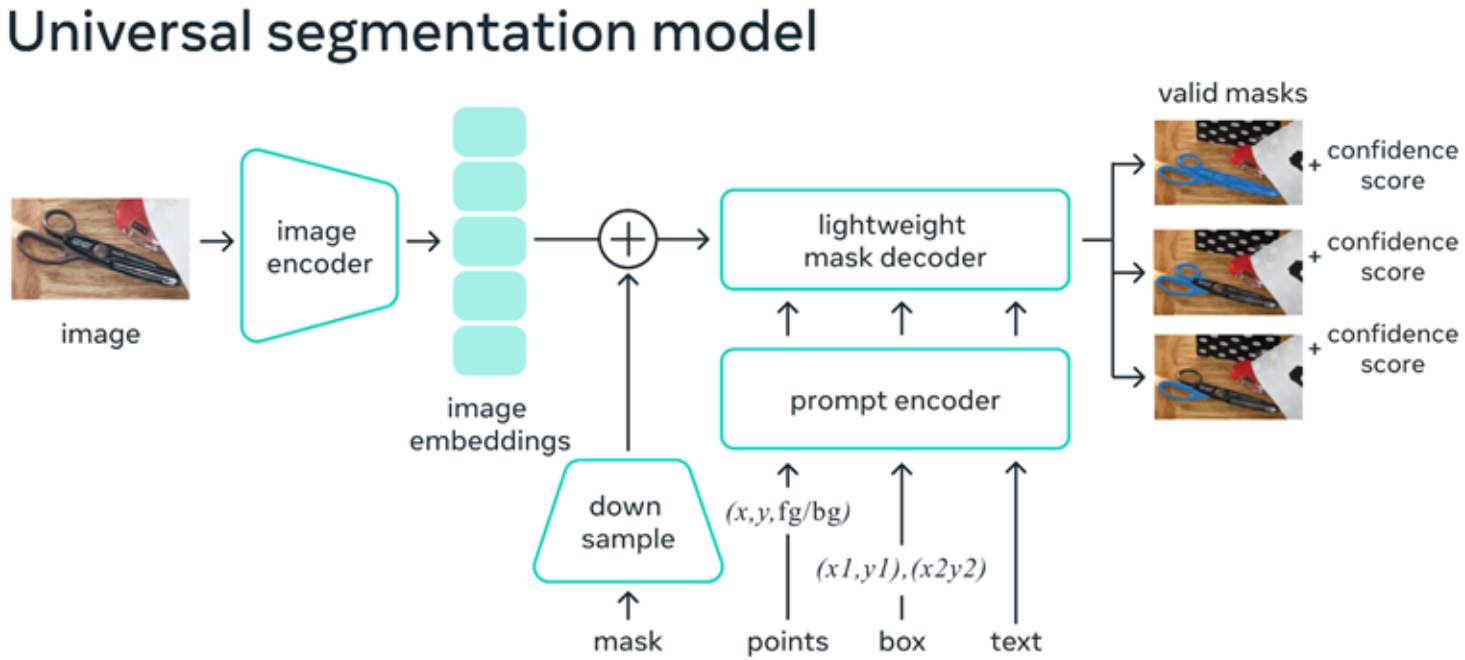
- The problem:
- CLIP tells me WHAT is in the image.
- But I need WHERE. I need bounding boxes.

- The solution:
- Instead of ONE vector per image...
- create a DENSE GRID of vectors. (basically just the ViT approach)
- One vector at every spatial position.

- Like a chessboard - every square has its own fingerprint.

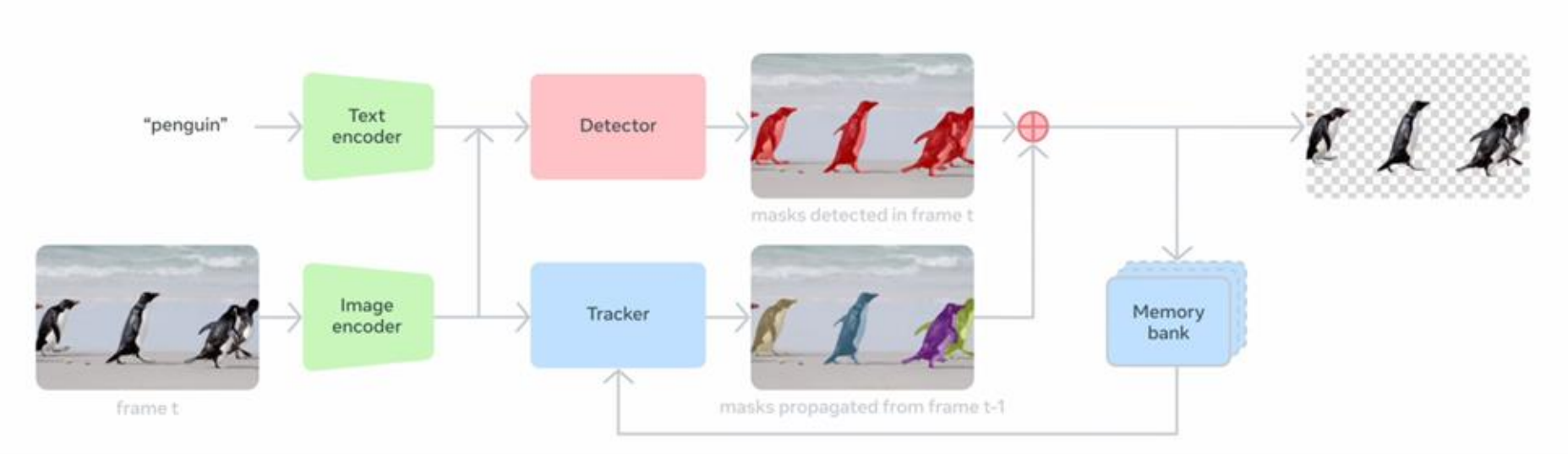
From "What" to "Where"

Pillar 3: Open Vocabulary Detection; the era of SAM (Segment Anything)



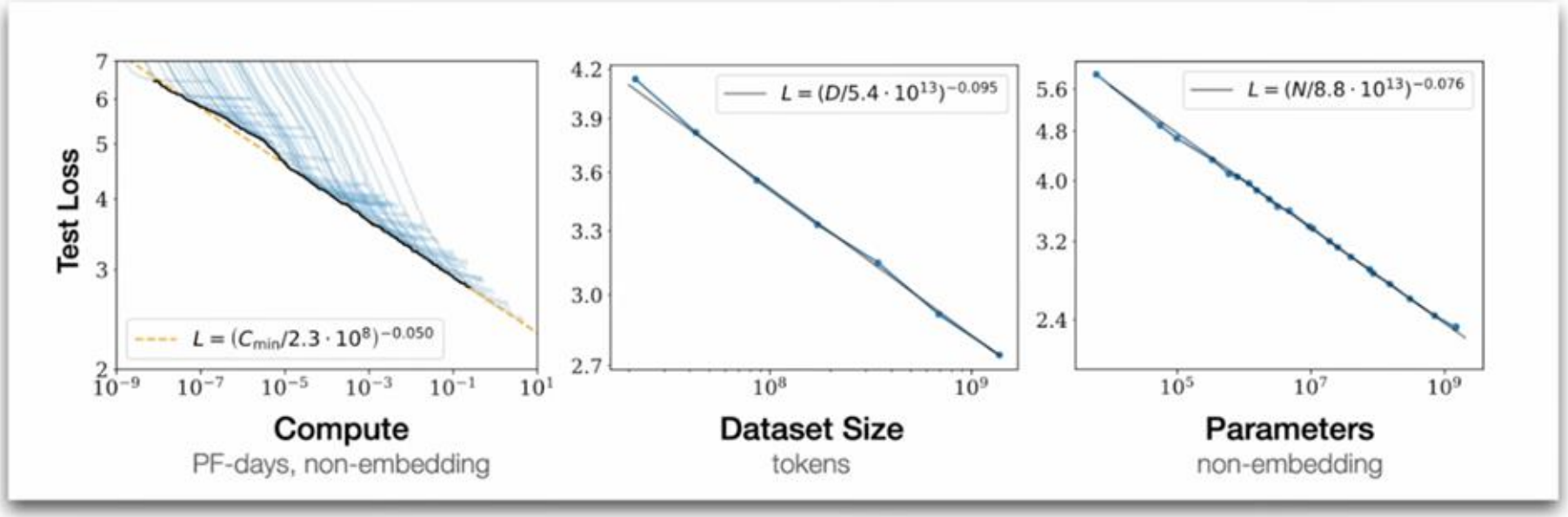
From "What" to "Where"

Pillar 3: Open Vocabulary Detection; the era of SAM (Segment Anything)



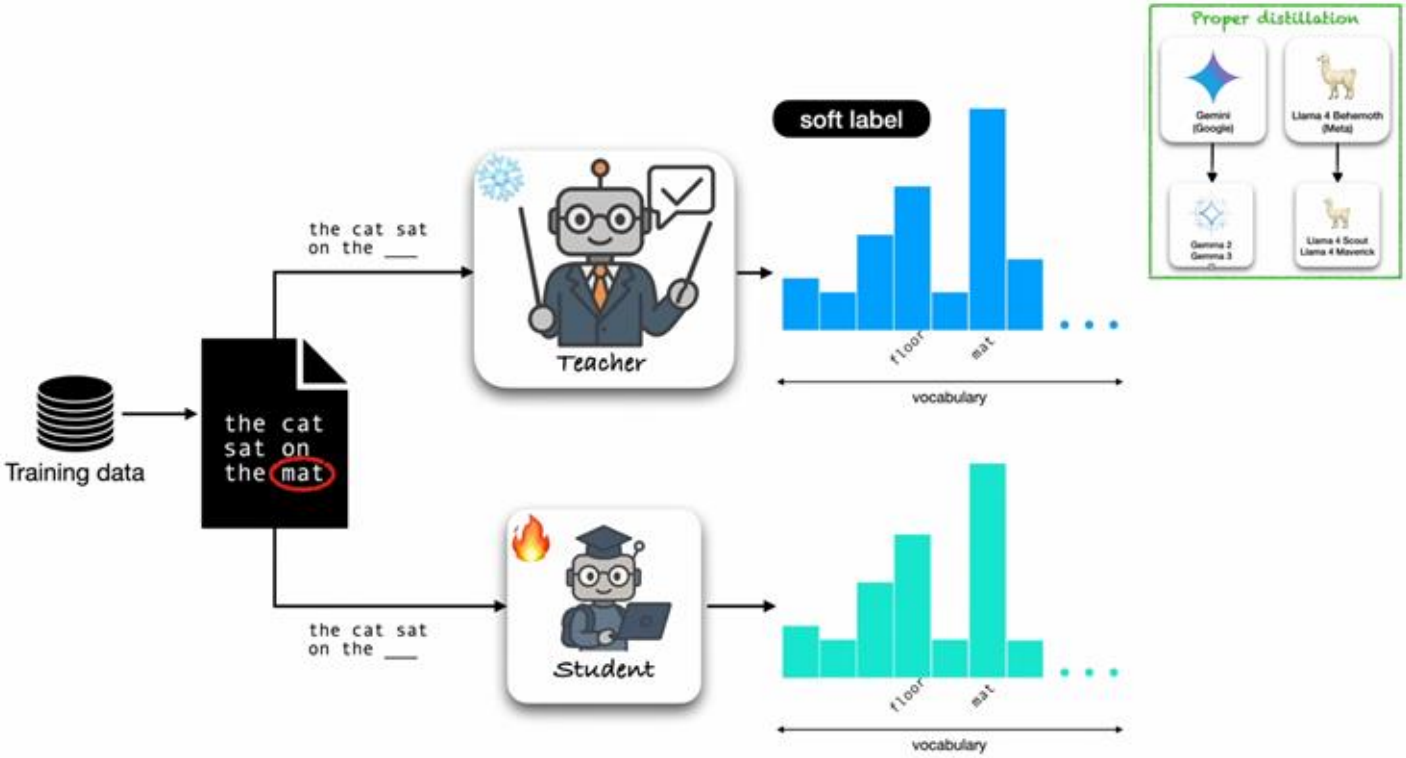
Distillation: The Deployment Answer

Pillar 4: Same output, different teacher



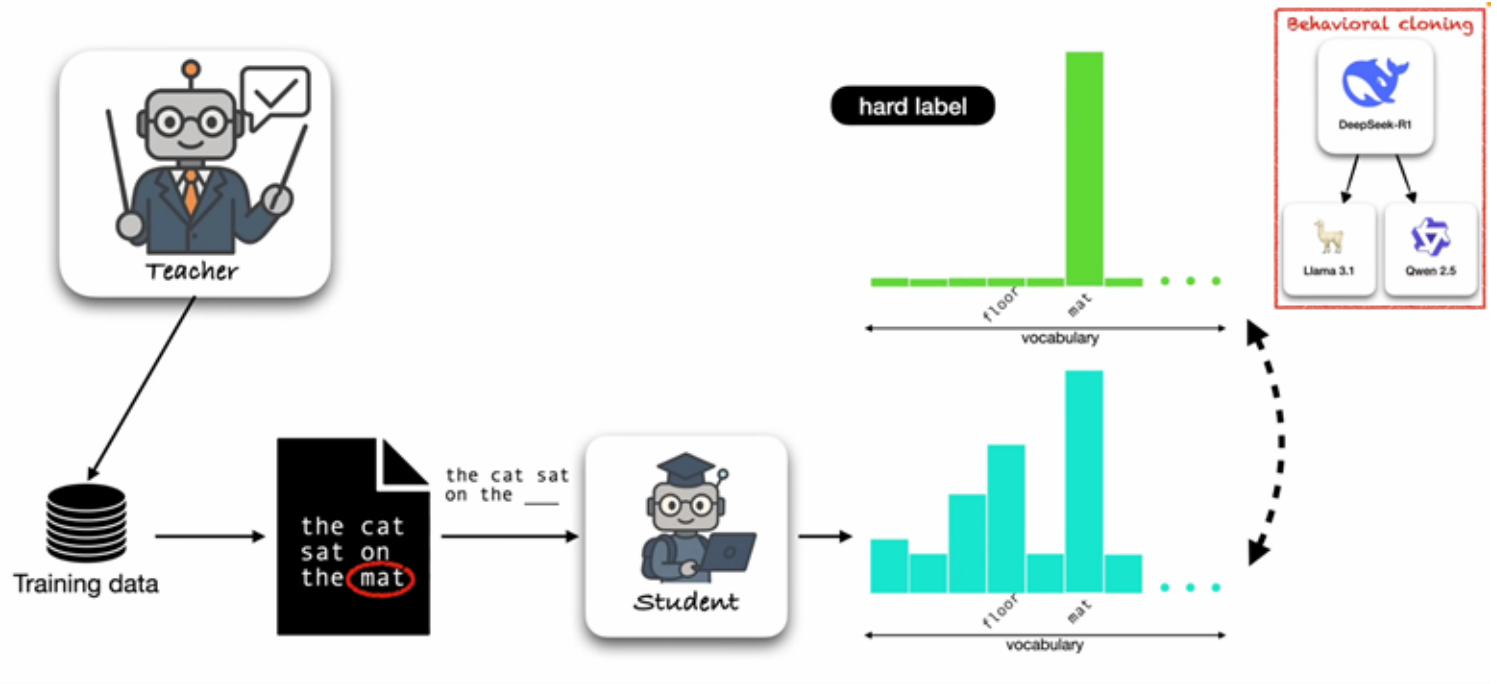
Distillation: The Deployment Answer

Pillar 4: Same output, different teacher



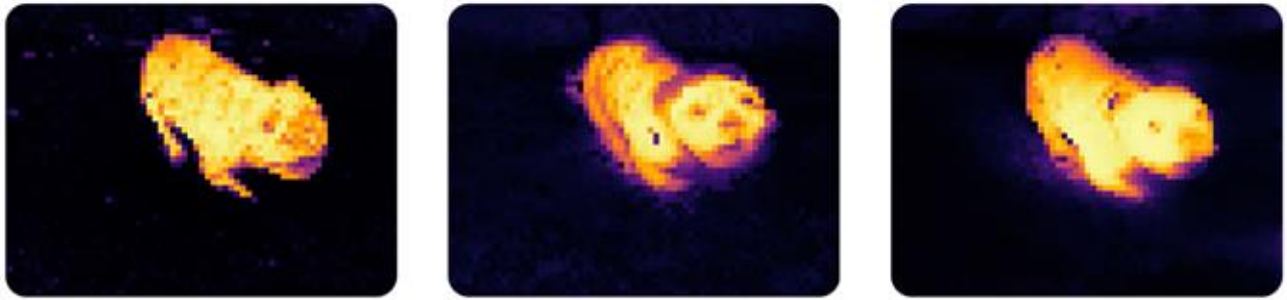
Distillation: The Deployment Answer

Pillar 4: Same output, different teacher



Distillation: The Deployment Answer

Pillar 4: Same output, different teacher



Distillation: The Deployment Answer

Pillar 4: Same output, different teacher

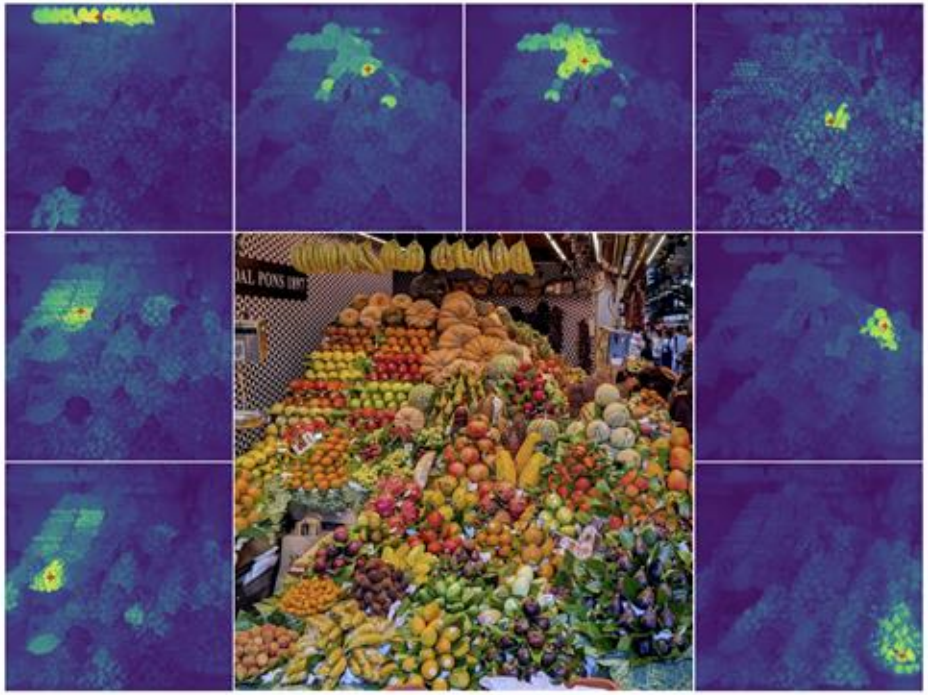
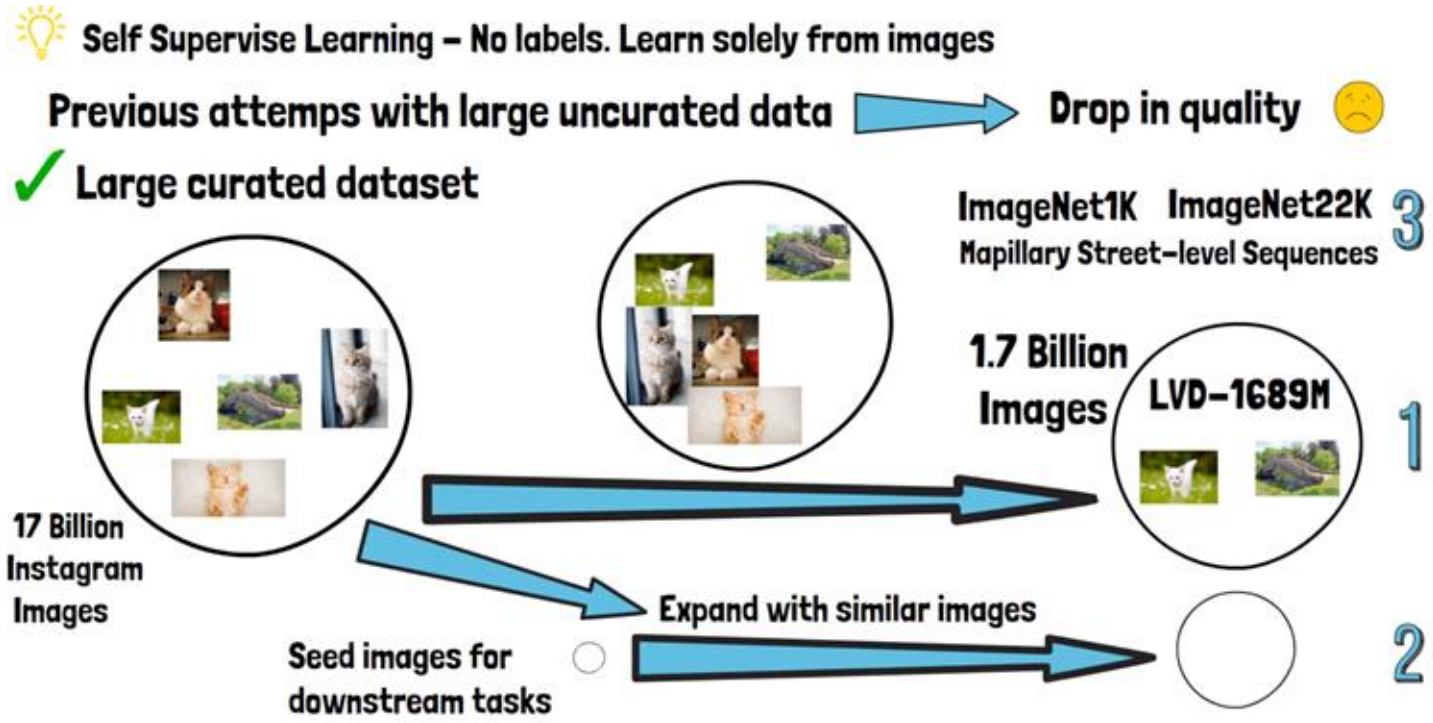


Figure 3: High-resolution dense features. We visualize the cosine similarity maps obtained with DINOv3 output features between the patches marked with a red cross and all other patches. Input image at 4096×4096. *Please zoom in*, do you agree with DINOv3?

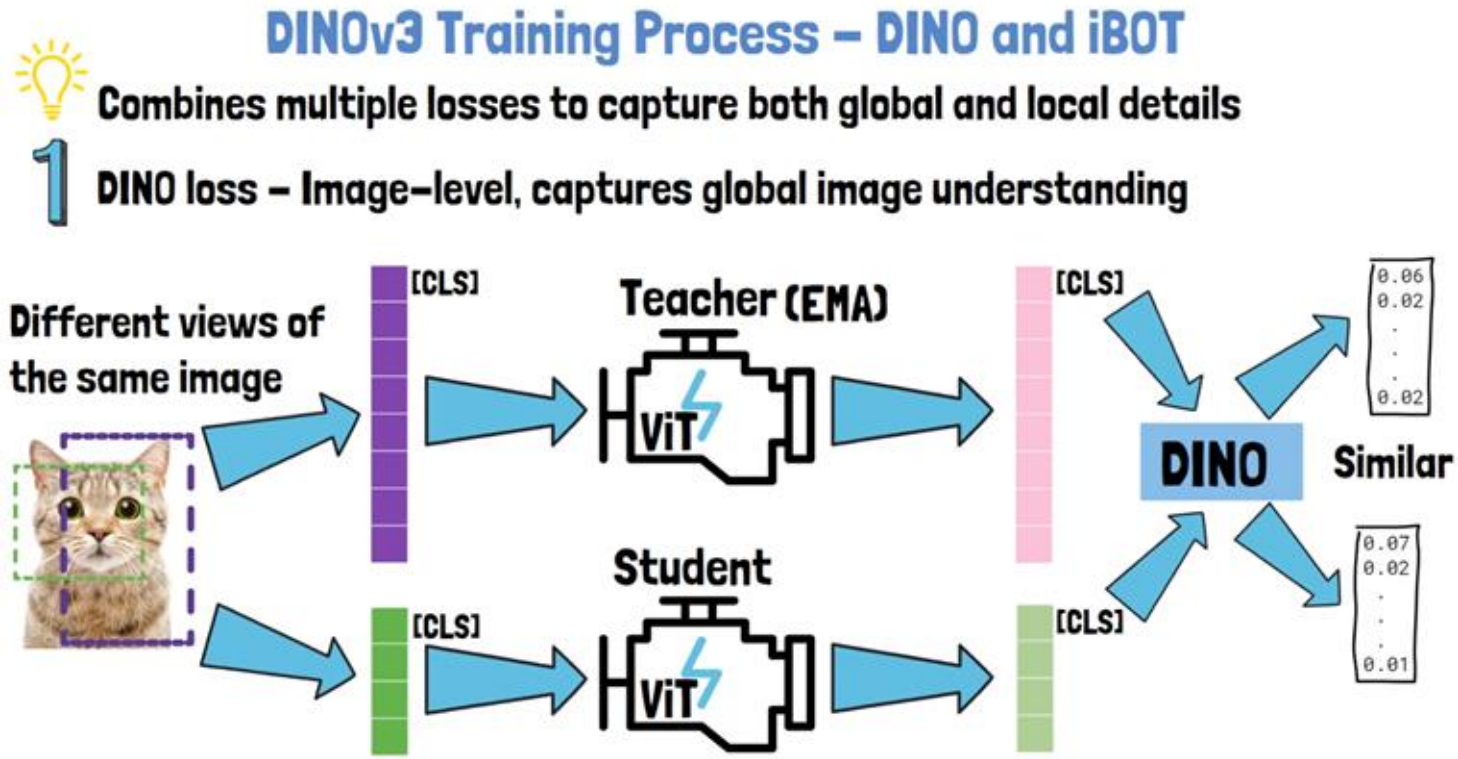
Distillation: The Deployment Answer

Pillar 4: Same output, different teacher



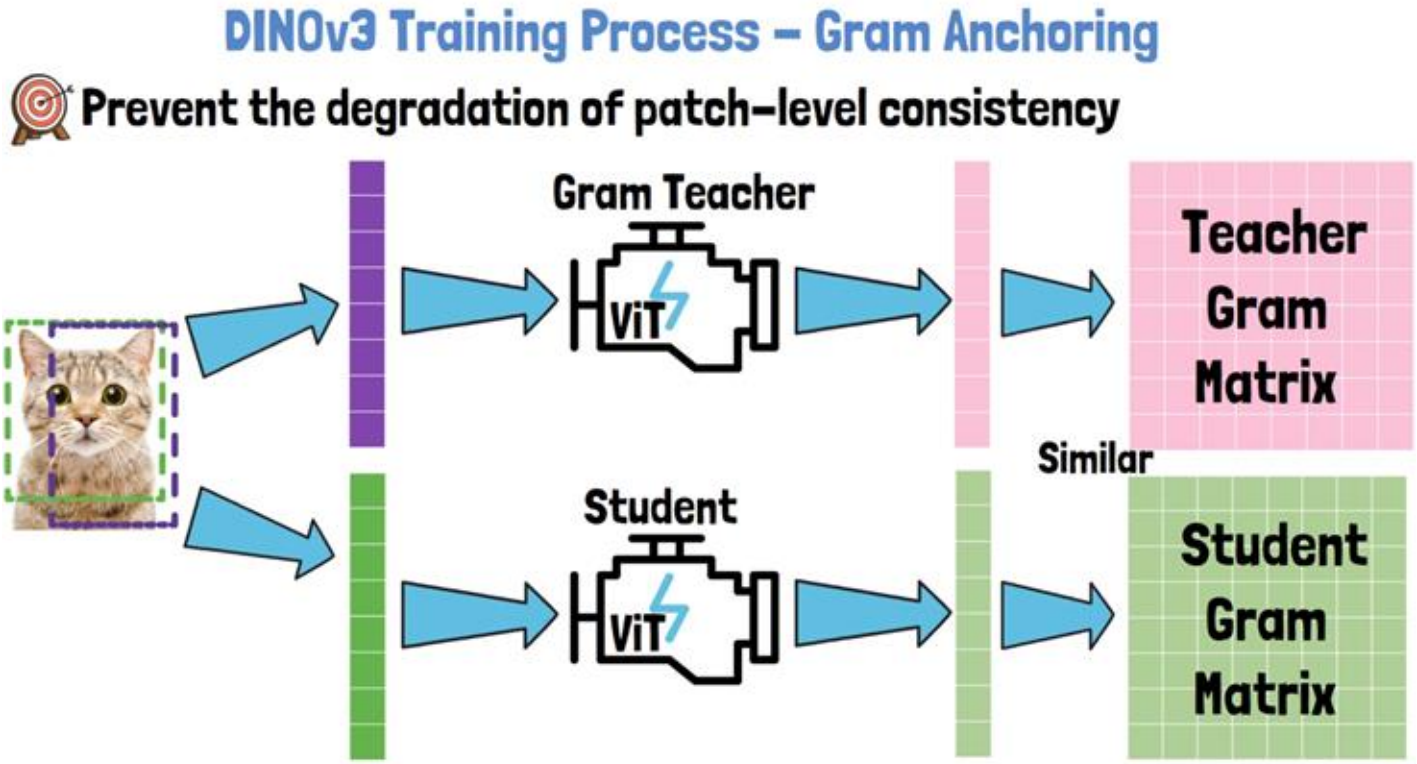
Distillation: The Deployment Answer

Pillar 4: Same output, different teacher



Distillation: The Deployment Answer

Pillar 4: Same output, different teacher



Distillation: The Deployment Answer

Pillar 4: Same output, different teacher

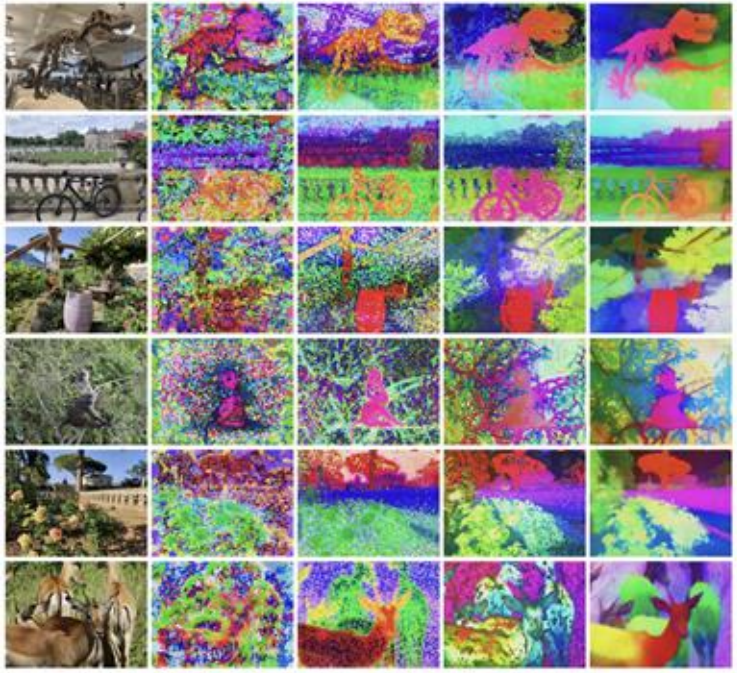


Figure 13: Comparison of dense features. We compare several vision backbones by projecting their dense outputs using PCA and mapping them to RGB. From left to right: SigLIP 2 ViT-g/16, PEspatial ViT-G/14, DINOv2 ViT-g/14 with registers, DINOv3 ViT-7B/16. Images are forwarded at resolution 1280×960 for models using patch 16 and 1120×840 for patch 14, i.e. all feature maps have size 80×60.

The Modern CV Stack

How the pieces fit together

- Demo

What Actually Goes Wrong

The Data Problem Nobody Budgeted For

Pitfall 1: The Annotation Treadmill

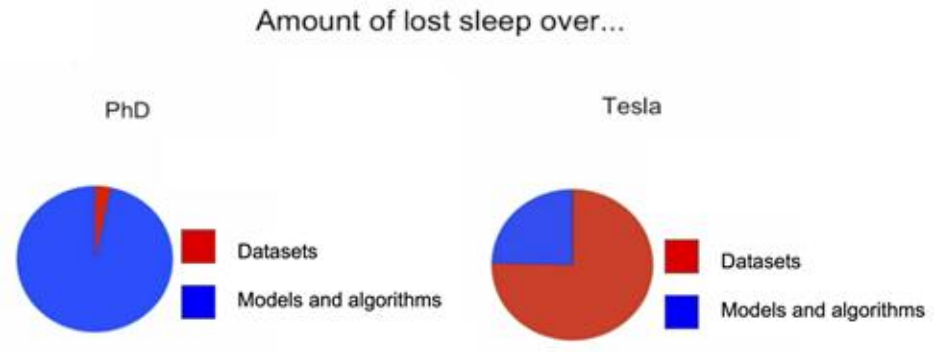
- Hardware gen 1 → Collect → Annotate → Train → Ship ✓
- Hardware gen 2 → New sensor → 40% labels stale → Re-annotate
- Hardware gen 3 → New optics → 60% labels stale → Re-annotate
- Hardware gen 4 → ...

- Each cycle: $50K+ and 3-4 months
- The model was fine.
- The data pipeline was the bottleneck.

"Just Use a Better Model"

Pitfall 2: Model-Centric Thinking

- The instinct when accuracy drops:
- → Try YOLOv26 instead of v11
- → Larger model variant
- → Tune hyperparameters for 2 weeks



- What actually moves the needle:
- → Fix 200 mislabeled images +3 mAP
- → Remove ambiguous objects +2 mAP
- → 500 targeted images for hard class +5 mAP
- → Right prompt for auto-labeling +15 mAP

"Works in the Notebook"

Pitfall 3: The Deployment Gap

- Your dev setup:
- A100 GPU • 80GB VRAM • unlimited power

- Your deployment target:
- Jetson Orin Nano • 8GB • 15W • <50ms latency

- SAM3: 2.9B params | ~200ms on A40
- YOLO26: 2.4M params | ~1.7ms on T4

- That's 1,200× smaller.
- You don't deploy the teacher. You distill it.

Annotation IS Distillation

The Reframe

- Manual annotation:
- Human looks at image → Draws box → Writes label
- The human brain is the teacher model.
- The labeled dataset is the student's curriculum.

- Foundation model annotation:
- SAM3 looks at image → Draws box → Writes label
- 400M image-text pairs is the teacher's education.
- Your labels are the student's curriculum.

- The teacher changed, but the process is identical.

The Modern Answer

Right Tool for the Right Job

- For exploration & labeling: SAM3
- 2.9B params • ~200ms • text prompts • Apache 2.0
- "The teacher."

- For edge deployment: YOLOv26
- 2.4M params • ~1.7ms • fine-tuned • AGPL-3.0*
- "The student."

- For commercial products: RF-DETR
- Apache 2.0 • NMS-free • ~2.3ms nano

Distillation Pipeline

Tomorrow: Your Turn

Day 2 Preview

- You will:
- 1. Get a challenge task
- 2. Generate synthetic training data
- 3. Auto-label with SAM3
- 4. Train YOLO26 on the labels
- 5. See where it fails

- You leave with:
- A working model
- A notebook you can reuse
- The intuition for when it breaks

Let's Debrief

- 1. What was the MOST SURPRISING thing you heard today?
  - (what was the coolest learning / takeaway)
- 2. What's the ONE THING that doesn't make sense yet?
  - (what would you like to see explained in more detail)

Q & A