Computer Vision. Lessons from the Past, Signals for the Future

3 Waves of AI

Deep Blue (1997) - Chess

- Showed that AI can beat humans at Chess
- Used brute-force search

AlexNet (2012) - Computer vision

- Won the ImageNet competition by a large margin
- Showed that Large neural networks plus GPUs out-perform hand crafted features
- Marked the raise of supervised learning with large labeled datasets
- Models are still task specific

Image of the computer that was used to train AlexNet. Used 2 GTX 580s

ChatGPT (2022) - NLP

Wave 3 (2022): The ChatGPT Moment

What happened

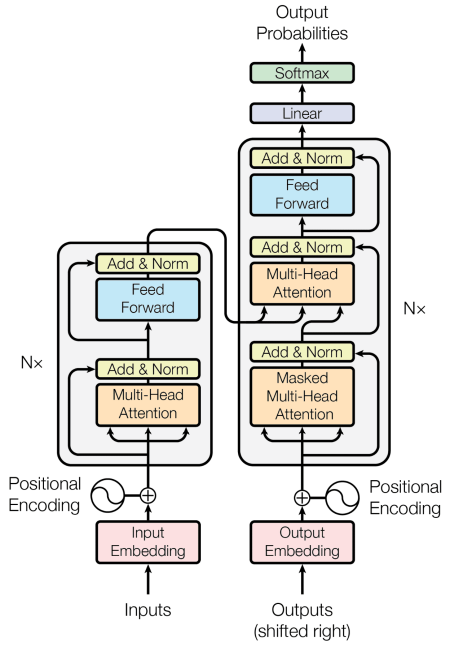
- ChatGPT by OpenAI – Made large language models mainstream.
- Record Launch – Fastest-growing app ever.

What makes it different

- Built on the transformer architecture (2017)
- Large scale pretraining on diverse text data

Why it matters

- One general model can perform thousands of tasks



Wave 4?

What we know

- Transformers and large-scale pretraining transfer well to vision
- Vision models are being trained as foundational models

Current limitation

- Text is primary, everything else is clunky
- Vision and other modalities are not really end to end

What is missing?

- Next major wave will incorporate computer vision

Wave 4 IS NOW KINDOF

What we know

- Transformers and large-scale pretraining transfer well to vision
- Vision models are being trained as foundational models

Current limitation

- Text is primary, everything else is clunky
- Vision and other modalities are not really end to end

What is missing?

- Next major wave will incorporate computer vision

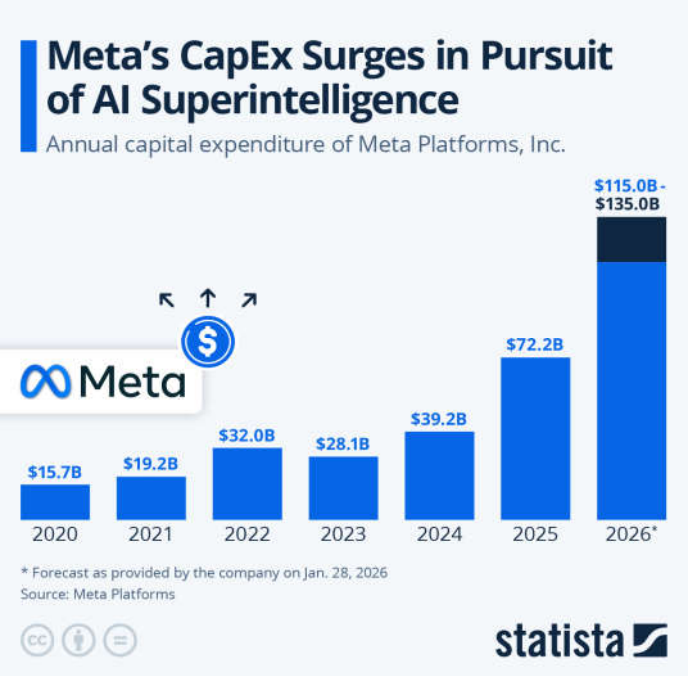
Current Meta in ML for State-of-the-art performance

Best way to do ML today

- Pretraining is key
- Leverage AI labs' investments

Why is this better?

- Improved performance
- Requires less data
- Lower barrier to entry



Computer Vision is still largely in Wave 2

Why wave 2 still dominates in CV

- Tooling is mature and easy to use
- For narrow tasks performance is "good enough"

Why not use the SOTA

- Don't fix what works
- Pretrained models are complex
- Many tasks are bespoke and resource-limited

How do I solve computer vision tasks in 2026?

Stop thinking

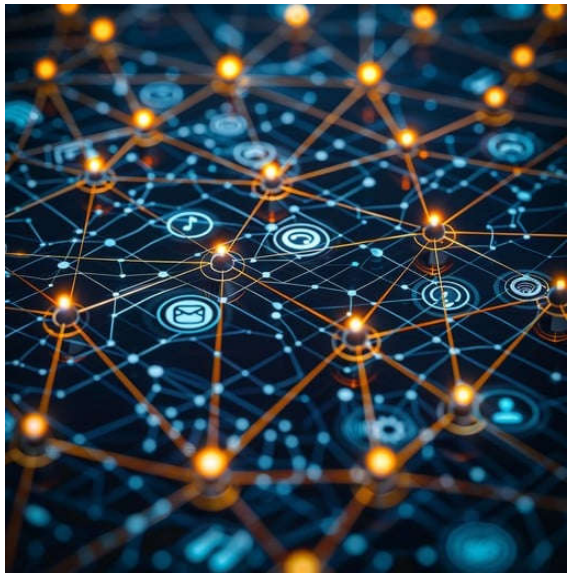
- "I need to train a model for my task"

Start thinking

- "What pretrained visual intelligence can I leverage for my task?"

2026 flow

1. Foundational model
2. Adaption
3. Training from scratch



Practical Workflow

Define the Problem

- What decision are you trying to automate?
- What metric defines success?

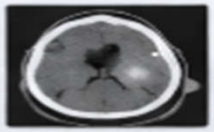
Detect Lung Nodule

Early Detection of Nodules

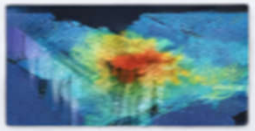
Identify Modalities

- Image, video, depth, multimodal?
- Inputs & outputs of the model?

X-Ray Image



CT Scan



3D Depth Map

Multimodal Data

Map to Tasks & Baseline

- What computer vision task are you trying to solve?
- What pretrained intelligence can you leverage?

Segmentation

Lung Nodule

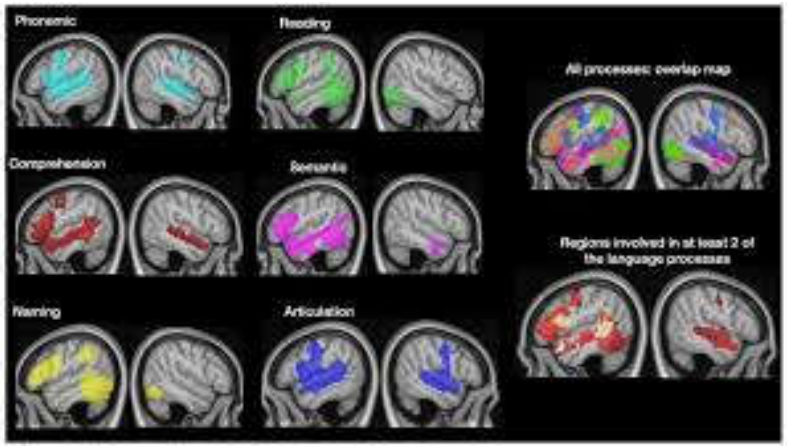
Pretrained Model

Baseline Training

Example Problem 1: Semantic Search

Define the problem

- Radiologists want to search a large medical imaging archive using natural language instead of fixed labels



Identify modalities

- Text in → relevant images out

Map this to task types and foundational baseline

- Perception Encoder (CLIP Based model)

Example Problem 2: Fine-Grained Recognition

Define the problem

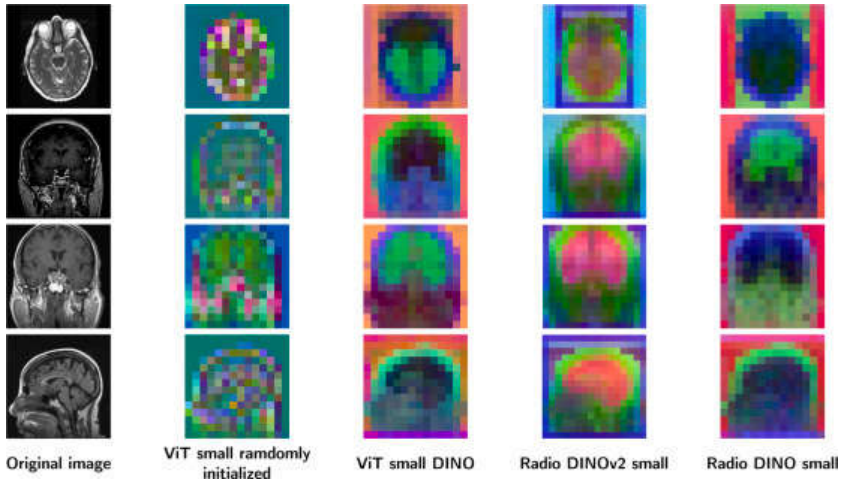
- Doctors need to tell apart very similar-looking medical images
- Example: spotting early signs of cancer in tissue slides where differences are subtle

Identify modalities

- Image in → diagnosis label out

Map this to task types and foundational baseline

- DINOv3 (use dino output features, then train a small classifier on top)



Example Problem 3: Promptable Segmentation

Define the problem

- Doctors need to quickly outline tumors or organs in scans without manually drawing boundaries

Identify modalities

- image + prompt in → Segmentation mask out

Map this to task types and foundational baseline

- Segment Anything (SAM)

Task types and foundational baselines!

Classification
- Simple (Edge / Fixed Classes): CNN or ViT classifier fine-tuned on labeled classes
- Foundation: CLIP

Object detection
- Simple (Edge / Real-time): YOLO
- Foundational: DINOv3 backbone + detection head

Segmentation
- Simple (Edge / Fixed Task): U-Net / DeepLab style supervised segmentation
- Foundation: SAM

Instance Segmentation
- Simple (Edge / Fixed Classes): Mask R-CNN style instance segmentation
- Foundation: SAM + CLIP (SAM for masks, CLIP for labeling)

OCR
- Simple (Clean Docs / Edge): Tesseract or classical OCR pipeline
- Foundation: Transformer / ViT-based OCR models

Pose Estimation
- Simple (Edge / Real-time): MediaPipe / OpenPose / YOLO-Pose
- Foundation: ViT-based pose models with pretrained backbones

Semantic Search / Retrieval
- Simple: haha no
- Foundation (Semantic / Open-Ended): CLIP embeddings (text-to-image, image-to-image)

Anomaly detection
- Simple: Rules, thresholds, or supervised "defect vs ok"
- Foundation: DINOv3 embeddings + outlier detection + supervised finetune

Captioning / understanding
- Simple (Edge / Lightweight): Moondream
- Foundation (High Quality / Zero shot): Large Vision-Language Model fine-tune