African Institute for Mathematical Sciences — Cameroon
Structured Master's, Data Science Stream · Computer Vision

Machines That See

Week 1 · Day 1: Welcome, Course Overview, and the World of Computer Vision

Yusuf Brima, Ph.D.

Introduction to Computer Vision · AIMS Cameroon, 2026–2027

Today

1. **Welcome**: how this course runs, how I teach, where to find help
2. **What computer vision is**, and how it relates to image processing, graphics, AI, psychology and neuroscience
3. **Why seeing is hard** for a machine, shown on real photographs
4. **What vision systems can do today**: real model outputs on real images
5. **Where vision is used**: health, agriculture, industry, cities, the planet
6. **How we got here**: from hand-crafted features to foundation models
7. **Human vision** as a template for machines

By the end of today you should be able to

- say what computer vision is, and what it is *not*;
- explain, with examples, why vision is hard;
- name the core tasks and what each one outputs;
- place today's methods on a 60-year timeline;
- describe how the eye and brain process images, and which ideas machines borrowed.

Part 1

Welcome: how this course works

Your instructor

Yusuf Brima, Ph.D.

- Ph.D. in Cognitive Science, Osnabrück University (Germany): how machines and brains learn representations
- **AIMS alumnus**: M.Sc. Mathematical Sciences, AIMS Rwanda; M.Sc. Computer Science, University of Dhaka; B.Sc., University of Makeni (Sierra Leone)
- Current work: biomedical AI, medical image analysis (MRI, CT), explainable and fair AI for low-resource health settings
- Email: `ybrima@uos.de`

Why this matters for you

I sit between *cognitive science* and *machine learning*. That is also how this course is built: we start from how *we* see, and use it to understand how machines can.

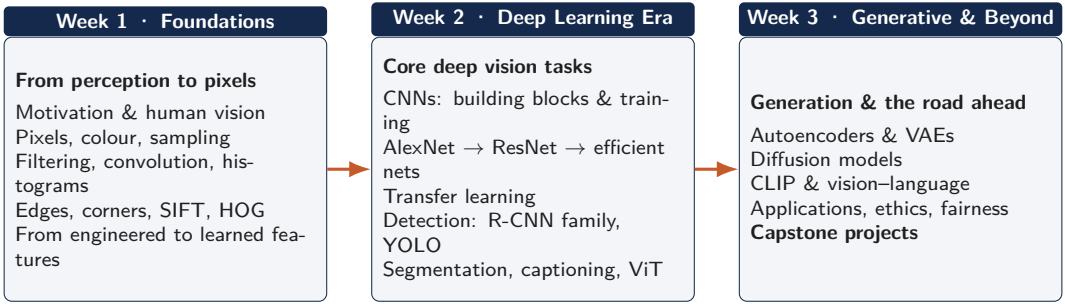
The course at a glance

- **Three-week block**, one theme per week
- **15 lectures** of two hours each, one per weekday
- Lectures are for **ideas**: concepts, intuition, derivations, discussion
- **Tutorials and labs** run separately with the tutors (TAs): this is where you code
- All materials (slides, notes, notebooks, references) are on the **course homepage**

Tools we use

- Python 3.10+, Jupyter, Git, VS Code
- NumPy, Matplotlib, scikit-image, scikit-learn
- OpenCV, Pillow
- PyTorch, torchvision, timm, Ultralytics (YOLO), albumentations
- No GPU needed: free **Google Colab** or **Kaggle** notebooks for heavy labs

The course in one picture



Each week builds on the last: you cannot understand a CNN without convolution, or diffusion without CNNs.

Week 1 lecture by lecture

	Lecture	The question it answers
Day 1	Welcome & the world of computer vision	What is vision, why is it hard, what can it do, and how do *we* do it?
Day 2	Visual data representation	How does light become a grid of numbers, and what is lost on the way?
Day 3	Image processing fundamentals	How do we clean, sharpen and enhance images with filters and histograms?
Day 4	Classical feature extraction	How do we find edges, corners and distinctive patches that survive change?
Day 5	From engineered to learned features	Why did hand-designed features give way to features learned from data?

Intuition

Week 1 is the "physics and craft" of images. Every idea reappears inside deep networks in Week 2.

Weeks 2 and 3

Week 2 · The deep learning era

- Convolutional neural networks: layers, training, regularisation
- Architectures: AlexNet, VGG, ResNet, efficient networks
- Classification with pre-trained backbones and transfer learning
- Object detection: R-CNN family and YOLO
- Semantic and instance segmentation
- Localisation and image captioning
- Vision Transformers (ViT)

Week 3 · Generative vision and beyond

- Autoencoders and variational autoencoders
- Diffusion models and conditional generation
- Vision–language models: CLIP, joint embedding spaces
- Applications: biomedicine, agriculture, robotics, remote sensing, inspection
- Open problems and research directions
- Ethics, fairness, bias and responsible deployment
- Capstone work and final presentations

What you will be able to do after three weeks

1. **Explain** how images are formed, stored and represented, and what that implies for algorithms.
2. **Apply** classical tools (filters, histograms, edges, descriptors) and know when they are enough.
3. **Build and train** CNNs, and adapt pre-trained models to new problems with little data.
4. **Choose** the right task formulation: classification, detection or segmentation.
5. **Understand** generative and vision–language models well enough to use and critique them.
6. **Deliver** a working vision project, evaluated honestly, and think about who it might harm.

Practitioner's Note

The goal is not to memorise architectures (they change every year), but to understand the *principles* well enough to read any new paper and build on it.

What we assume you know, and where to refresh it

Linear algebra

- vectors, matrices, dot products
- matrix–vector products
- eigenvalues (used on Day 4)

Calculus

- derivatives, partial derivatives
- gradients, chain rule

Probability & statistics

- distributions, mean, variance
- Gaussian distribution
- Bayes' rule (intuitively)

Machine learning

- train / validation / test
- loss, gradient descent
- overfitting

Programming

- Python and NumPy arrays
- Jupyter notebooks
- plotting with Matplotlib

Rusty? That is fine.

- I re-derive what we need, when we need it
- tutors run refresher sessions
- ask early, not the night before a deadline

Assessment

Weekly quiz

- One per week
- At most one hour
- Covers that week's lectures
- Tests understanding, not memorisation

Assignments (Weeks 1–2)

- Individual
- Typed in LaTeX
- Theory *and* application
- Short coding mini-projects

Capstone (Week 3)

- Teams of 2–3
- A real-world vision problem
- Presented on the last day
- A short video demo is welcome

How to do well

Come to lectures, attend tutorials, and do the notebooks *yourself*. The quiz and assignments reward people who have actually run and broken the code.

The capstone project

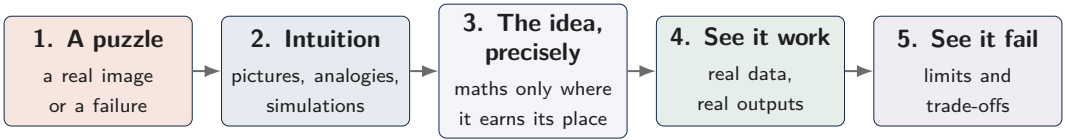
- Apply course ideas to a **real computer vision problem**, in teams of 2–3.
- **Topics** are proposed by me at the end of Week 2.
- **Your own idea?** Submit a one-page synopsis by the *last lecture day of Week 2*:
  - problem statement and motivation
  - proposed method
  - how you will get the data
- **Final presentations** on the last day of Week 3.

Start thinking now

The best projects start from a problem someone actually has: a crop, a clinic, a market, a road, a document archive. Keep a list this week of problems you see where a camera could help.

A good project is *small, real and honestly evaluated*, not big and vague.

How I teach



- **Intuition first.** If you can draw it, you understand it. Equations come after the picture, and I will tell you what each symbol *means*.
- **Real examples.** Real photographs, real models, real numbers, including from Cameroon and the continent.
- **Interactive.** I will ask you questions throughout. There are no wrong guesses, only interesting ones.
- **Failures are features.** Seeing *why* a method breaks teaches more than seeing it work.

What you can expect from me, and what I expect from you

From me

- Slides, notes and code released on the course homepage
- Clear assessment instructions, in advance
- Honest answers, including "I don't know, let me check"
- Feedback on your work you can act on

From you

- Questions beat note-taking: **interrupt me**
- Do the tutorial notebooks yourself, then compare
- Collaborate on ideas; write your own solutions
- Be able to explain every line you submit

On AI assistants

Use them like a tutor to understand, not as a ghost-writer. If you cannot explain your submission without the tool, it is not yet your work.

Where to find help

In the lecture
Ask. If you are confused, others are too.

Tutors (TAs)
Tutorials, lab support and debugging outside lecture hours.

Me
After class, by email, and in project consultations (Week 3).

Course homepage
Slides, notes, notebooks, solutions, announcements.

Your classmates
Study groups. Explaining is the fastest way to learn.

Books & online
Szeliski; Gonzalez & Woods; library docs; CS231n notes.

A rule of thumb

Stuck for more than 30 minutes on a bug? Ask a tutor or a classmate. Stuck for more than a day on a concept? Ask me.

Reading and resources

Core texts

- R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed., 2022. Free PDF from the author's website.
- R. C. Gonzalez & R. E. Woods, *Digital Image Processing*, 4th ed., 2018.
- Course notes (on the homepage), written for this course.

Going deeper

- Goodfellow, Bengio & Courville, *Deep Learning*, 2016
- Zhang et al., *Dive into Deep Learning* (d2l.ai)

Landmark papers we will read together

- Krizhevsky et al. 2012, *AlexNet*
- He et al. 2016, *ResNet*
- Ren et al. 2015, *Faster R-CNN*
- Ronneberger et al. 2015, *U-Net*
- He et al. 2017, *Mask R-CNN*
- Goodfellow et al. 2014, *GANs*
- Dosovitskiy et al. 2020, *ViT*

Practitioner's Note

Library documentation (OpenCV, scikit-image, PyTorch) is a resource, not a last resort.

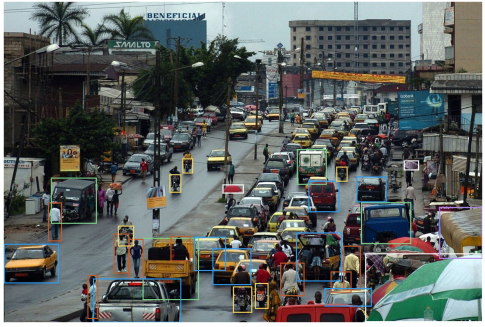
Part 2

What is computer vision?

Computer vision: a working definition

Definition

Computer vision builds algorithms that turn images or video into *useful descriptions* of the world: what is there, where it is, what it is doing, and what will happen next.



Happiraphael, CC BY-SA 4.0

A real detector (YOLOv12) on a real street in Douala found **30 objects**: 8 cars, 8 people, 6 motorcycles, 4 trucks, 3 umbrellas and a bus.

- In: about 1.2 million numbers
- Out: a short list of *things* with positions

Intuition

Vision is **compression into meaning**: millions of numbers in, a few facts out.

What you see, and what the computer sees

What you see



What the computer sees

128	130	128	115	103	96	93	100	107	118
124	122	119	110	100	95	90	86	88	95
80	82	83	83	82	79	73	64	58	54
42	45	48	46	51	50	47	40	32	28
20	25	25	24	26	26	32	31	29	26
23	23	21	23	19	20	24	22	21	19
28	30	26	22	22	22	19	17	21	21
60	50	37	34	33	28	27	27	24	27
94	77	66	55	54	53	48	44	40	52
113	101	100	92	89	71	65	74	72	74

- You see *a cat*, instantly, without effort.
- The computer receives a 300×451 grid of integers between 0 and 255. The highlighted 10×10 patch is part of the cat's eye.

The semantic gap

Nothing in those numbers says "cat". Every bit of meaning has to be *computed* from patterns of numbers. Bridging this gap is the whole subject.

Seeing feels effortless. That is exactly the trap.

For you

- You recognise a familiar object in roughly **100–150 ms**, faster than a blink.
- A large share of the primate cortex, with dozens of distinct areas, is devoted to vision.
- You were trained by years of continuous video, with a body that moves and touches.

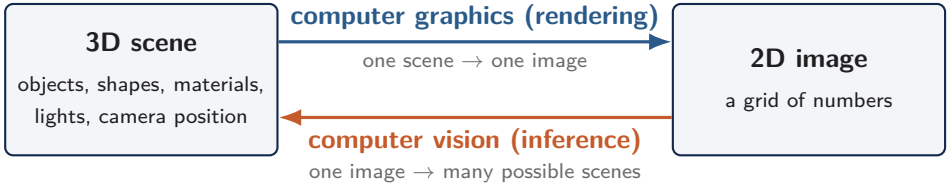
For a machine

- In 1966 an MIT "summer project" aimed to largely solve vision in a few months.
- Sixty years later, it is still an active research field.

Moravec's paradox (1988)

What is hard for humans (chess, algebra) is often easy for computers, and what is easy for humans (seeing, walking) is hard. Skills that evolution polished over millions of years are invisible to us *because* they work so well.

Vision is graphics run backwards

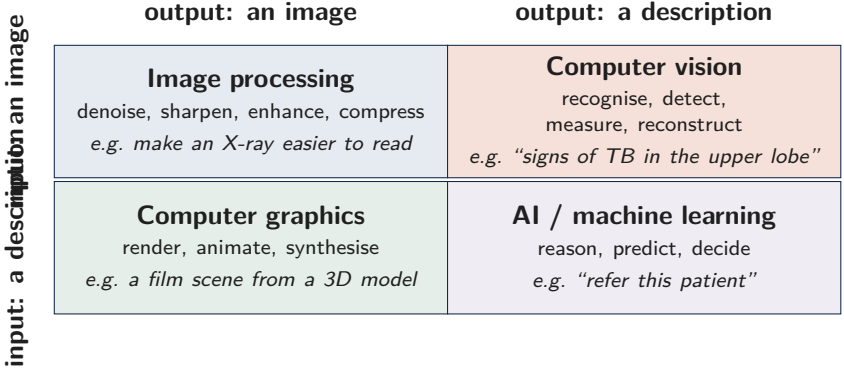


- **Graphics** is a *forward* problem: given the scene, physics tells you exactly what the image is.
- **Vision** is the *inverse* problem: given the image, find the scene. Depth, hidden surfaces and lighting have been thrown away.

Intuition

Inverse problems are *ill-posed*: many answers fit the data. To pick the right one you need **prior knowledge** about how the world usually is: designed by hand (Week 1) or learned from data (Weeks 2–3).

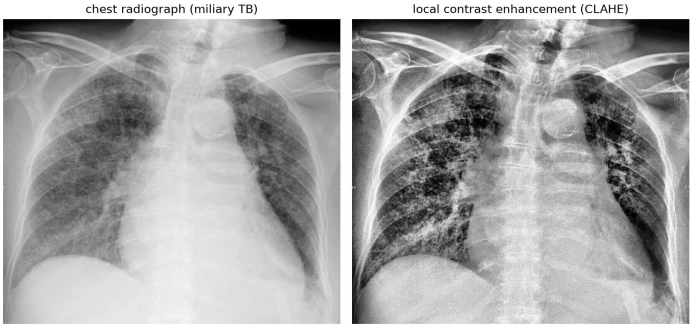
Four neighbouring fields, sorted by input and output



Practitioner's Note

Real systems chain them: processing cleans the image, vision describes it, an AI component decides. Generative models (Week 3) blur the bottom-left box: text in, image out.

Image processing vs. computer vision on the same X-ray



B. Herreros et al., CC BY 4.0

Image processing: image in, image out. Local contrast enhancement (CLAHE, Day 3) makes the fine nodules of miliary TB easier *for a radiologist* to see. The computer has not "understood" anything.

Computer vision: image in, description out.
A system that outputs "abnormal: diffuse micronodular pattern, likely TB" has to *interpret* the image. Image processing is often its first step.

Computer graphics and vision: two sides of one coin

Graphics helps vision

- **Synthetic training data**: render millions of labelled images (cars, robots, factory parts) for free.
- **Analysis by synthesis**: guess the scene, render it, compare with the photo, adjust.
- **Camera models** (Day 2) are shared by both fields.

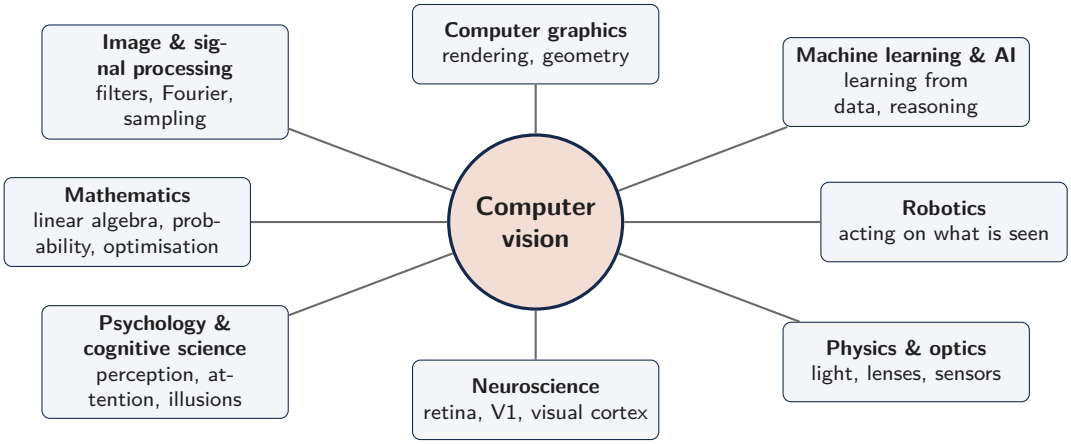
Vision helps graphics

- **3D capture**: reconstruct buildings or people from photographs.
- **Neural rendering** (NeRF 2020, Gaussian splatting 2023): learn a 3D scene from photos, then render new views.
- **Generative models** (Week 3): learn to "render" from text.

Intuition

A system that can *both* render a scene and recognise it has, in a sense, understood it. Much of modern research lives exactly on this boundary.

Computer vision sits at a crossroads



Tools from mathematics, physics and signal processing; ideas from psychology and neuroscience; its engine from machine learning.

Psychology and neuroscience: why they belong in a vision course

Psychology studies *what* we perceive

- Gestalt laws: how we group parts into wholes
- Illusions: where our priors show through
- Attention: why we miss what we are not looking for
- Benchmarks: human error rates tell us what "good" means

Neuroscience studies *how* the brain does it

- Receptive fields $\rightarrow$ convolution
- Simple and complex cells $\rightarrow$ filters and pooling
- Hierarchy of visual areas $\rightarrow$ deep networks
- Today, deep networks are also used as *models of the brain*

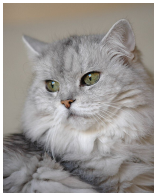
Intuition

The influence runs both ways. Neuroscience inspired the CNN; CNNs now predict neural responses in the visual cortex better than earlier models. We return to this at the end of today.

Part 3

Why is seeing hard? Evidence from real photographs

One cat, many images



Gourdo 1 at en.wikipedia, Public domain



pelican from Tokyo, Japan, CC BY-SA 2.0



Matthew Paul Argall, CC0



Jakub Hałun, CC BY-SA 4.0

All four are "cat". Yet pixel by pixel these images share almost nothing: different poses, colours, fur, backgrounds, lighting and sizes.

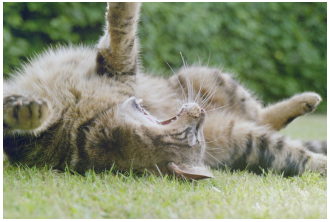
The central tension

A recogniser must be **invariant** to huge changes that do not matter, yet **selective** to small changes that do.

Challenge 1: deformation and pose



pelican from Tokyo, Japan, CC BY-SA 2.0



Clavecin, Public domain



Wasianpower, CC BY-SA 4.0

- Cats, people and animals are **articulated and flexible**: the same body produces endlessly different silhouettes.
- A rigid template ("a cat looks like this") fails immediately.

Intuition

A good representation should describe the *parts* (ears, eyes, paws, fur texture) and be tolerant to how they are arranged. Keep this in mind for the bag-of-words model (Day 5) and CNN pooling (Week 2).

Challenge 2: occlusion



RyanDannyChacko, CC BY-SA 4.0



Famartin, CC BY-SA 4.0



Wikimedia user Rohit Sav, CC BY-SA 4.0



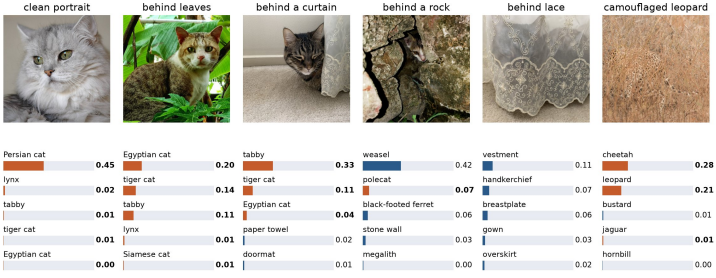
Famartin, CC BY-SA 4.0

- Leaves hide the body; a curtain hides half the face; a rock leaves an eye and an ear; lace covers everything.
- You still see a whole cat: **amodal completion**, perceiving hidden parts as present.

Intuition

Occlusion removes evidence *and* adds misleading evidence: the occluder's own edges and texture.

What does a real network make of these occluded cats?



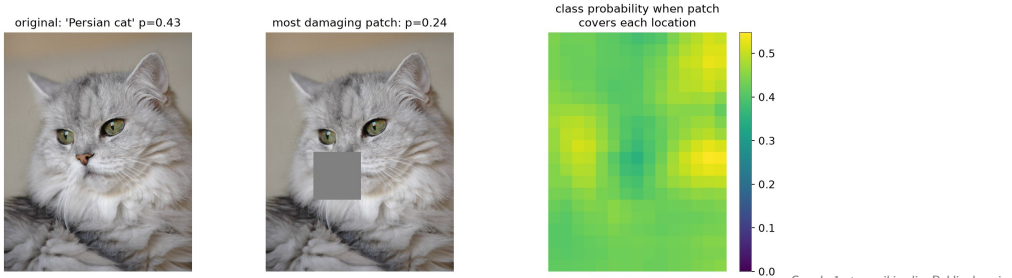
ResNet-50 (ImageNet-1k, torchvision weights): top-5 classes. Orange: a feline class.

A ResNet-50 trained on 1.28 million ImageNet photos: confident on the clean portrait, still right behind leaves and the curtain, but "weasel" behind the rock and "vestment" behind lace.

Intuition

The network uses *local* cues: a small head peering out of rock looks like a weasel; lace texture swamps everything. You reason about the occluder; it does not.

Where is the network looking? An occlusion experiment



Method (Zeiler & Fergus, 2014): slide a grey square over the image; at each position, record the probability of "Persian cat". Dark regions in the map are where hiding the image *hurts* the prediction most.

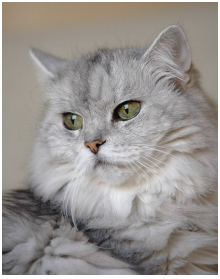
Practitioner's Note

This is the simplest *explainability* tool there is: it needs no access to the network's internals. You use occlusion to break the model, then use it again to understand it.

Challenge 3: scale



Hisashi from Japan, CC BY-SA 2.0



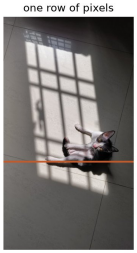
Gourdo 1 at en.wikipedia, Public domain

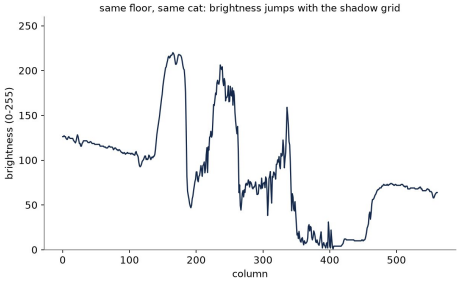
- Left: the cat in the car park covers well under 1% of the image, a few thousand pixels. Right: a cat fills the frame.
- Size in the image depends on **distance**, not on the object.

Intuition

Small objects carry little detail; large ones have detail we must learn to ignore. Day 4 builds **scale space** (look at every scale) and Week 2 builds **feature pyramids**, the same idea inside a network.

Challenge 4: illumination





UnknownCopy, CC BY-SA 4.0

- A window casts a grid of light on the floor and the cat.
- Along one row, brightness swings from dark to almost white: *bigger* than the difference between cat and floor.

Intuition

Ignore changes caused by *light*; keep changes caused by the *object*. Days 3–4 give the first tools.

Image formation, roughly

$$\text{pixel} \approx \underbrace{\text{illumination}}_{\text{the light}} \times \underbrace{\text{reflectance}}_{\text{the object}}$$

One measurement, two unknowns. Separating them is ill-posed.

Challenge 5: background clutter and camouflage



Davidvraju, CC BY-SA 4.0



Senthiaathavan, CC BY-SA 4.0

- Both photos contain a leopard. Evolution has optimised the leopard's coat to *defeat* visual systems.
- The ResNet-50 from earlier calls the grass leopard a "cheetah" (0.28), with "leopard" second (0.21).

Intuition

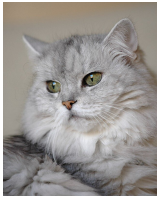
Recognition needs **figure–ground separation**: which pixels are object and which are background. When object and background share colour and texture, local cues fail and *shape* and *context* must take over.

Challenge 6: variation within a class, similarity between classes

Same class, very different



Jakub Hałun, CC BY-SA 4.0



Gourdo 1 at en.wikipedia, Public domain

A hairless Sphynx and a long-haired Persian are both cats. A juvenile husky and a grey wolf are different species with similar faces, colours and fur.

Different classes, very similar



MrPanyGoff, CC BY-SA 3.0

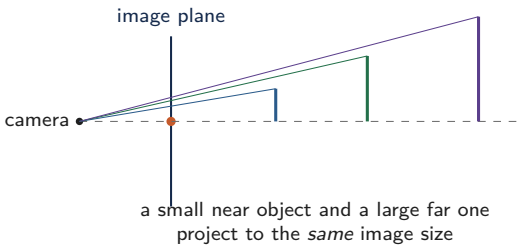


USFWS/Gary Kramer, Public domain

Practitioner's Note

This is common in practice: cassava mosaic vs. brown streak disease, benign vs. malignant lesions, two parasite species in a smear. Fine distinctions need careful data and expert labels.

Challenge 7: a 2D image loses the third dimension

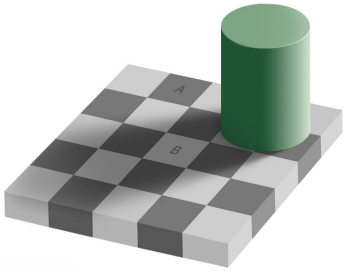


- Every pixel collects light along a whole **ray**. Where along the ray the surface sits is lost.
- So size and distance cannot be separated from one image alone.
- Yet you judge depth easily, using cues: occlusion, perspective, texture gradients, known object sizes, shading.

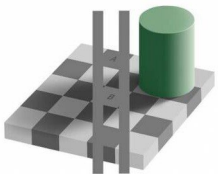
Intuition

Each of these depth cues is a **prior** about the world. Monocular depth networks (later today) learn the same cues from millions of examples.

Challenge 8: the same grey can look different



Edward H. Adelson, CC BY 3.0



Edward H. Adelson, 1995, CC0

Squares **A** and **B** are *exactly the same* grey (right: joined by a grey bar). You see B as lighter because your visual system "knows" B is in shadow and corrects for it.

Intuition

This is not a flaw. Your brain estimates the *reflectance of the surface*, not the light reaching your eye. That is what you need to recognise objects under changing light.

Practitioner's Note

A camera reports the raw pixel value. Any algorithm that compares raw intensities will make the opposite mistake to your eye. Normalisation (Days 2–3) is our first, crude fix.

Why vision is hard: a summary

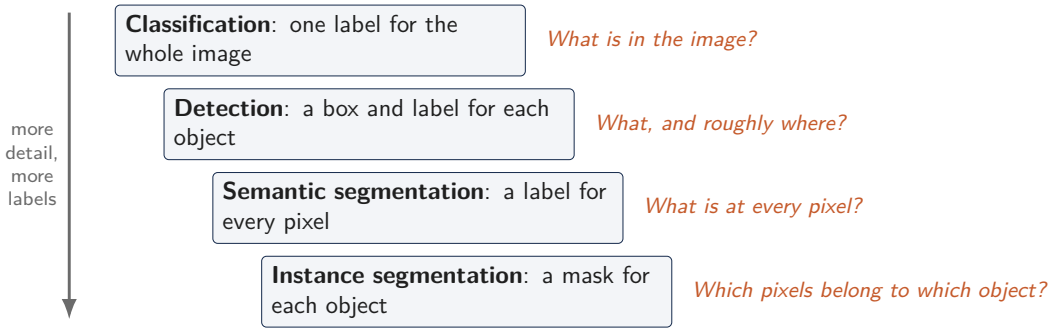
Challenge	What changes in the pixels	What we want
Deformation, pose	shape of the silhouette	invariance to articulation
Occlusion	parts missing, foreign parts added	reason about hidden parts
Scale	number of pixels on the object	scale invariance
Illumination	brightness and shadows	invariance to lighting
Clutter, camouflage	object blends into background	figure–ground separation
Intra-class variation	appearance within a class	generalisation
Inter-class similarity	tiny differences between classes	fine discrimination
3D $\rightarrow$ 2D projection	depth discarded	recover 3D from cues

Every method in this course answers one or more rows. For each new method, ask: *which row is it fixing?*

Part 4

What vision systems can do today

A ladder of tasks: from "what" to "which pixels"



Practitioner's Note

More detailed outputs need more expensive labels. A class label takes a second to annotate; a precise mask can take minutes. Choose the *least* detailed task that solves your problem.

Classification: one label per image

- **Input**: an image. **Output**: a probability for each of K classes.
- The ResNet-50 we used earlier has $K = 1000$ ImageNet classes, including *five* kinds of domestic cat.
- Typical uses: is this leaf diseased? Is this X-ray abnormal? Which species is on this camera-trap image?

What the network outputs

$$p_k = p(\text{class } k \mid \text{image}), \quad k = 1, \dots, K$$

The K probabilities add up to 1. The prediction is the class with the largest probability. The probability itself is a confidence that we must learn to *distrust* (the weasel had 0.42).

Intuition

Classification assumes there is *one* main thing in the image. A street in Douala breaks that assumption, which brings us to detection.

Object detection: what, and where



Bdx, CC BY-SA 4.0

- **Output**: a variable-length list of (box, class, confidence).
- YOLOv12 on a Yaoundé street: 6 cars, 6 people, a motorcycle, 2 handbags.
- Note the *partially visible* taxis and people: occlusion again.

Practitioner's Note

The detector only knows the 80 COCO classes. It cannot say "taxi" or "moto-taxi" unless we train it to: this is *transfer learning* (Week 2).

Detection in a crowded scene



Photokadaffi, CC BY-SA 4.0

- A moto-taxi in Mbouda carrying two passengers and a large load. The detector finds 14 people and the motorcycle.
- The load and the bananas are not detected: they are not COCO classes.
- The motorcycle box is mostly hidden by riders and cargo, yet it is found.

Intuition

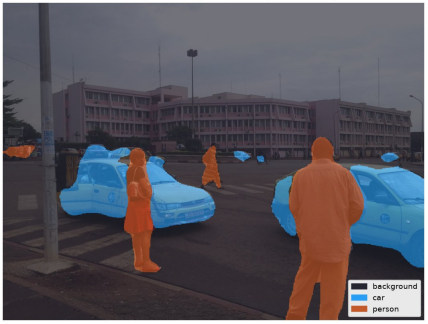
Detection = classification + localisation, repeated for every object. How a network does this in one pass is the YOLO story of Week 2.

Semantic segmentation: a label for every pixel

input



one label per pixel (DeepLabV3, 21 classes)



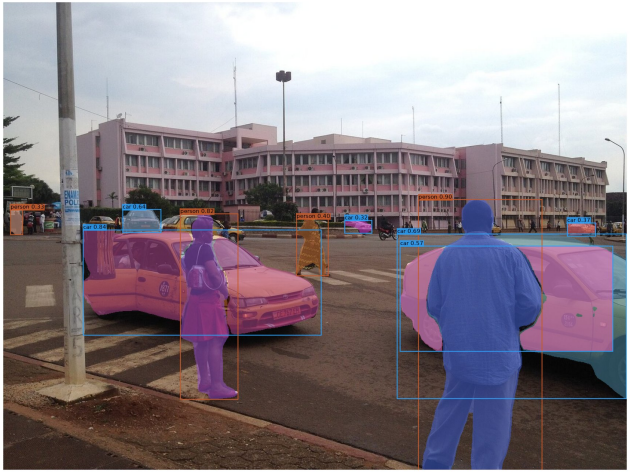
Bdx, CC BY-SA 4.0

DeepLabV3 assigns each pixel to one of 21 classes. Every car pixel is "car" and every person pixel is "person", but it does *not* say how many cars there are.

Practitioner's Note

Used where area and shape matter more than count: tumour outlines in MRI, flooded land in satellite images, drivable road for a self-driving car.

Instance segmentation: a mask for each object



Bdx, CC BY-SA 4.0

- Each car and each person gets its *own* mask and colour.
- Instance = detection's "which one" + segmentation's "which pixels".

Compare

semantic:	"these pixels are car"
instance:	"these pixels are car #3"

Needed when objects touch: counting cells in microscopy, fruit on a tree, people in a queue.

Pose estimation: where are the joints?



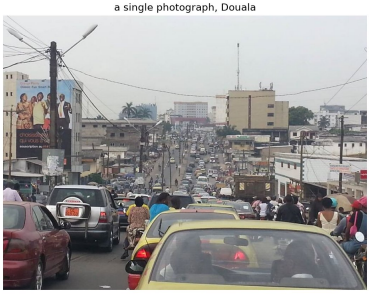
Photokadaffi, CC BY-SA 4.0

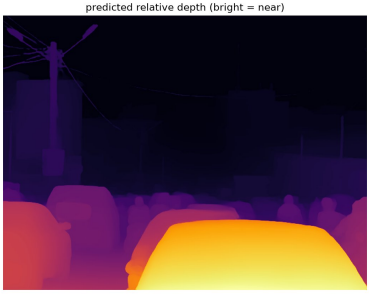
- **Output**: 17 keypoints per person (eyes, shoulders, elbows, wrists, hips, knees, ankles), joined into a skeleton.
- Works through crowds and partial occlusion, but loses joints hidden behind cargo or other people.

Practitioner's Note

Uses: physiotherapy and rehabilitation, sports analysis, ergonomics at work, sign-language recognition, animation.

Depth from a single photograph





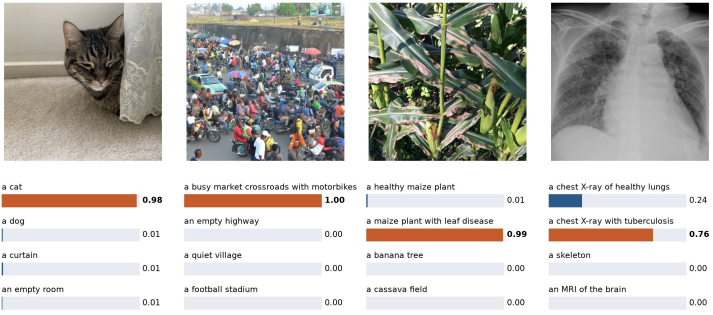
Guaka, CC BY-SA 4.0

Depth Anything V2 predicts how far away each pixel is from **one** ordinary photo: the near taxi is bright, the street fades into the distance, the sky is far.

Intuition

We just said one image cannot determine depth. The network uses learned *priors*: perspective lines, object sizes, occlusion order, haze. It gives *relative* depth, not metres.

Vision meets language: recognition without task-specific training



CLIP ViT-B/32, zero-shot: the candidate sentences were written at test time; no task-specific training.

Photos: Famartin; Minette Lontsie; Chroanch; B. Herreros et al. (CC BY-SA 4.0 / CC BY 4.0)

CLIP was trained on 400 million image–caption pairs from the web. We type candidate sentences at test time; it picks the one that best matches the image. No disease classifier was ever trained here.

Practitioner's Note

Impressive, but be careful: 0.76 for "tuberculosis" on an X-ray is *not* a diagnosis. CLIP has seen few medical images, and its confidence is not calibrated for clinical use.

And many more tasks

Geometry & motion

- Image matching, panoramas
- Stereo and 3D reconstruction
- Optical flow and tracking
- Visual localisation (SLAM)

Low-level vision

- Denoising, deblurring
- Super-resolution
- Colourisation, inpainting

Understanding

- Action recognition in video
- Scene graphs: who does what to whom
- Anomaly detection
- Text in images (OCR)

Faces & people

- Face detection and verification
- Gesture, gait, sign language

Vision + language

- Image captioning
- Visual question answering
- Multimodal assistants

Generation

- Text-to-image, image editing
- Video generation
- Novel-view synthesis

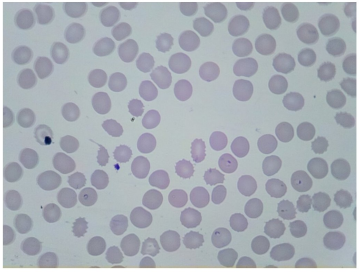
Intuition

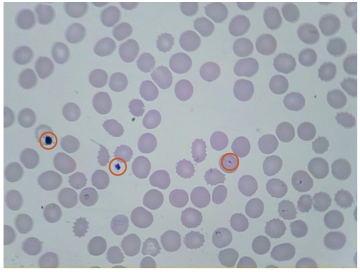
All of these reuse the same small toolbox: convolution, features, losses, optimisation. Learn the toolbox well and new tasks become variations.

Part 5

Computer vision in the world

Health: finding malaria parasites in a blood smear





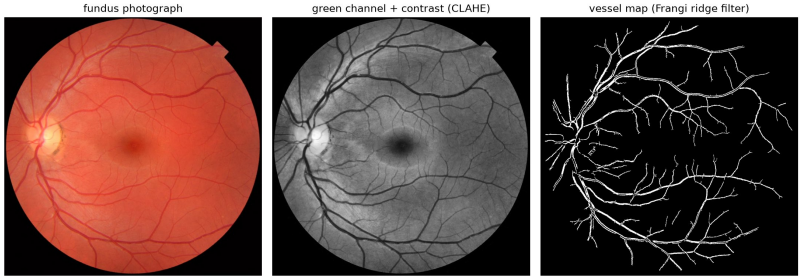
OpenFlexure Project, CC BY-SA 4.0

- Microscopy of Giemsa-stained blood remains the reference test for malaria, but counting parasites by eye is slow and tiring.
- A simple *colour rule* (dark, saturated purple) flags three parasites and misses the faint ring forms.

Practitioner's Note

This image comes from the OpenFlexure project: a 3D-printed, open-source microscope used in Tanzania and elsewhere. Cheap hardware plus good software can bring screening closer to patients.

Health: blood vessels in the retina



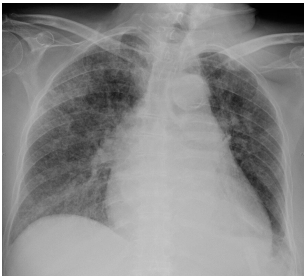
scikit-image sample image (M. Häggström, CC0)

Three classical steps from Week 1: take the green channel (Day 2), enhance local contrast (Day 3), and apply a ridge filter that responds to thin tubular structures (Day 4).

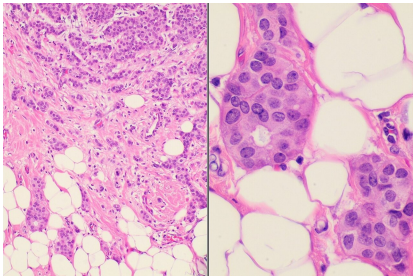
Intuition

Diabetic retinopathy and hypertension change vessel width and shape. Retinal photographs are cheap to take, so automated screening can reach many more people than there are eye specialists.

Health: radiology and pathology



B. Herreros et al., CC BY 4.0



Mikael Häggström, CC0

Left: chest X-ray, miliary tuberculosis.

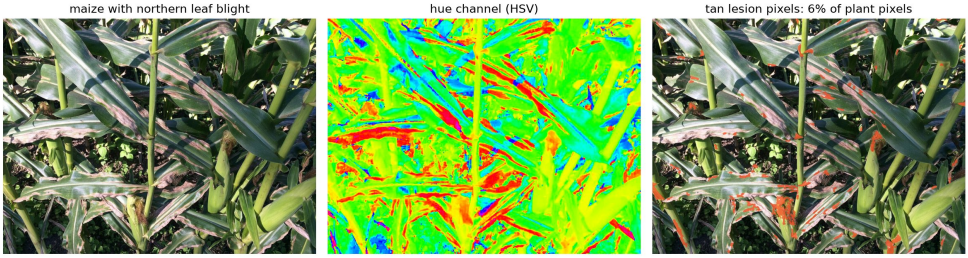
Right: breast tissue under the microscope (H&E stain), invasive ductal carcinoma.

- **Radiology**: detect TB, pneumonia or fractures; triage when radiologists are scarce.
- **Pathology**: a whole slide is gigapixels; models find and count tumour cells.

Practitioner's Note

Medical vision is *high-stakes*: a model must be validated on patients like the ones it will serve. A model trained in one country's hospitals can fail in another's.

Agriculture: measuring crop disease



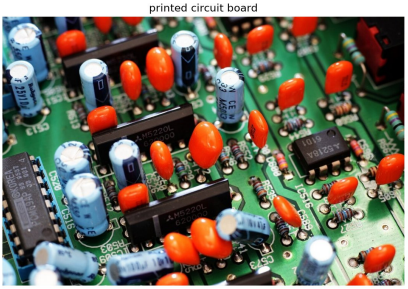
Chroanch, CC BY-SA 4.0

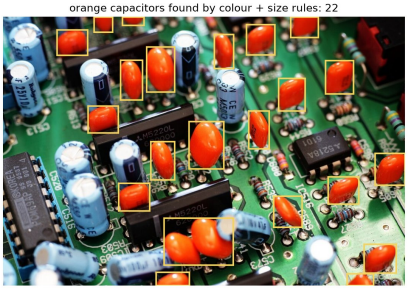
- Northern leaf blight leaves long tan lesions on maize. In the **hue** channel (Day 2), tan and green separate cleanly.
- A colour rule marks about **6%** of plant pixels as lesion: a rough severity score.

Practitioner's Note

Phone-based disease diagnosis for cassava, maize and cocoa is one of the most active applied-vision areas in Africa. The hard part is rarely the model; it is varied field photos, lighting, and expert labels.

Industry: visual inspection on a circuit board





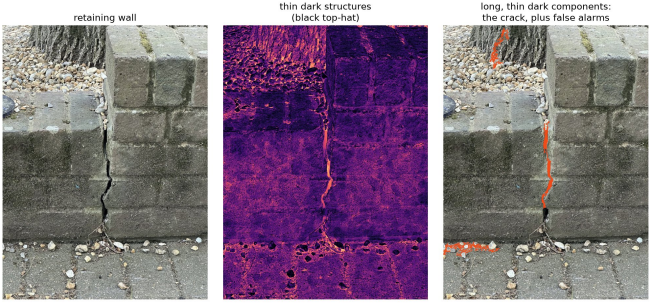
Ashley Pomeroy, CC BY 4.0

- Factories check that every component is *present*, *in place* and *undamaged*, thousands of times per hour.
- Colour and size rules find 22 orange capacitors. Look closely: two touching capacitors become one box, and edge cases are missed.

Intuition

Inspection is the home of **anomaly detection**: defects are rare and varied, so we often learn what "normal" looks like and flag anything different.

Infrastructure: cracks and potholes



Sunolafjagtenben-hur, CC0

A morphological filter highlights thin dark structures and keeps the long thin ones. It finds the crack, plus false alarms in tree bark and gravel. Classical rules are fast and transparent but brittle.



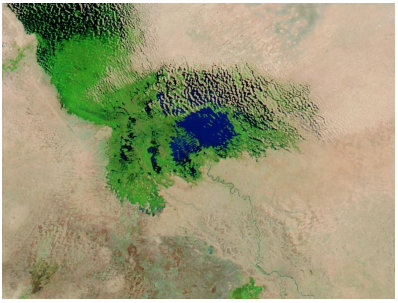
Brihaspati, CC BY-SA 4.0

Right: a pothole on a Nairobi road (2025). Phone or dashcam images can map road damage at city scale.

Earth observation: watching a lake shrink



NASA, Public domain



NASA MODIS Rapid Response, Public domain

Lake Chad, shared by Cameroon, Chad, Niger and Nigeria. **Left**: seen from Apollo 7 in 1968. **Right**: NASA MODIS, March 2024. Open water has largely given way to marsh and vegetation.

Practitioner's Note

Satellites such as Sentinel-2 image the whole planet every few days, for free. Vision models map land use, deforestation, floods, crops and informal settlements. The data are multispectral: more than three colours (Day 2).

Conservation and cities



Dodro, CC BY 4.0

Camera traps fire on motion, day and night (left: infrared image of a fox).

- One survey can produce millions of images, most of them empty.
- Models filter out the empty frames, identify species, and count animals.

Cities and transport

- Counting vehicles and pedestrians for planning
- Reading number plates, monitoring junctions
- Driver assistance and autonomous driving

Intuition

Night-time infrared, motion blur and animals half-hidden by grass are exactly the challenges from our earlier table, now in a real setting.

Applications: the pattern behind them

Domain	Example	Task	The hard part
Health	malaria, TB, retina, pathology	detect, classify, segment	stakes, label cost, shift between hospitals
Agriculture	leaf disease, yield, weeds	classify, segment	field conditions, lighting, varieties
Industry	PCB, welds, textiles	detect, anomaly	defects are rare and varied
Infrastructure	cracks, potholes	segment	thin structures, false alarms
Earth observation	land use, floods, lakes	segment, change detection	huge images, many bands, clouds
Conservation	camera traps	detect, classify	night, blur, empty frames
Cities	traffic, plates	detect, track	crowds, occlusion, privacy

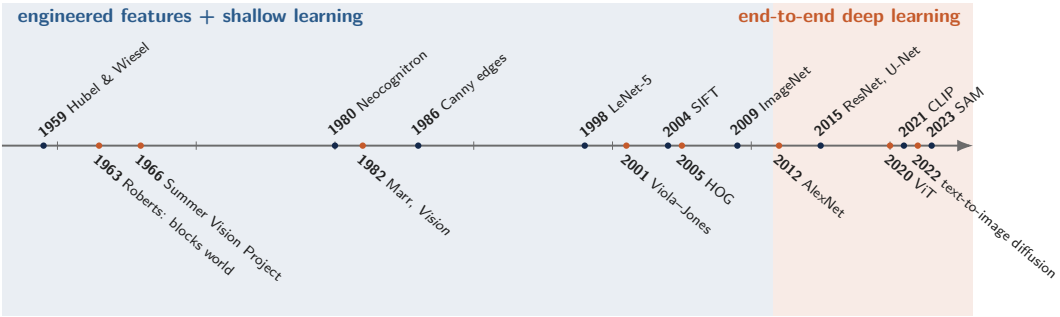
Practitioner's Note

In nearly every row the bottleneck is *data and deployment*, not the newest architecture. Keep this in mind when you choose a capstone.

Part 6

How we got here: sixty years in fifteen minutes

A timeline of computer vision



Intuition

Two eras. For fifty years people *designed* the features and let a simple classifier learn the rest. Since 2012, the features themselves are *learned* from data.

The early years: geometry, blocks and Marr

- **1963, Larry Roberts**: recover 3D polyhedra ("blocks") from line drawings of photographs. Vision as *geometry*.
- **1966, MIT Summer Vision Project**: a plan to connect a camera to a computer and describe what it sees, in one summer.
- **1970s**: edges, shape from shading, stereo, motion. Clean lab scenes, hand-written rules.

David Marr, *Vision* (1982)

Understand any vision system at three levels:

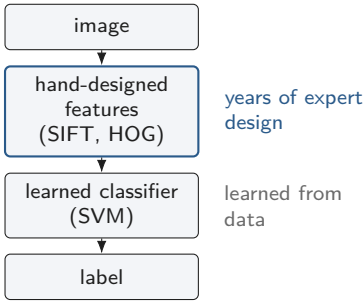
1. **Computational**: what problem is solved, and why?
2. **Algorithmic**: which representation and procedure?
3. **Implementation**: neurons or silicon?

Intuition

Marr's levels are still the best way to read a paper: what is the task, what is the algorithm, and what hardware does it need?

The feature era: hand-designed descriptors (1986–2011)

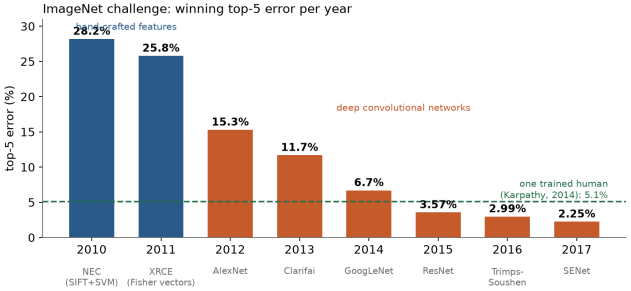
- **Canny (1986)**: optimal edge detection (Day 4)
- **Harris (1988)**: corners (Day 4)
- **LeNet (1989–98)**: CNNs read cheques and postcodes, but did not scale to natural images yet
- **SIFT (1999/2004)**: distinctive local features that survive scale and rotation (Day 4)
- **Viola–Jones (2001)**: real-time face detection, in every camera within a few years (Day 5)
- **HOG (2005)**: pedestrian detection (Day 4)



Practitioner's Note

SIFT still powers 3D mapping today, and CNN first layers learn detectors that resemble these.

ImageNet and the 2012 breakthrough

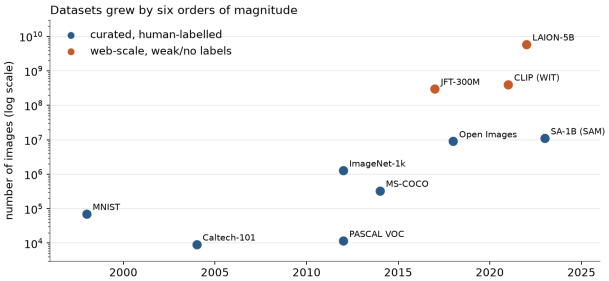


- **ImageNet** (2009): 14 million labelled images; the challenge used 1.28 million in 1000 classes.
- **2012, AlexNet**: a deep CNN trained on two gaming GPUs cut the error from 25.8% to 15.3%. Nobody had seen a jump like that.

Intuition

Within three years, networks passed the 5.1% error of a trained human annotator *on this benchmark*. That does not mean machines "see better than humans": remember the weasel.

Why 2012? Data, compute and a few good ideas



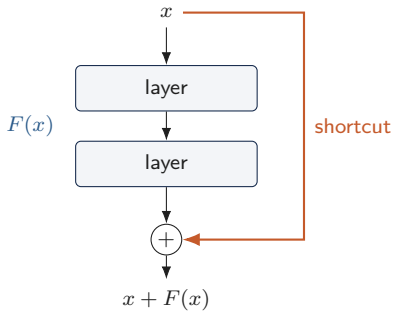
1. **Data**: ImageNet was about a thousand times bigger than earlier labelled datasets.
2. **Compute**: GPUs, built for video games, made training about ten times faster.
3. **Algorithms**: ReLU activations, dropout, data augmentation.

Intuition

The convolution itself was *not* new: LeNet had it in 1998. What changed was having enough data and compute for learning to beat human design. Since then, datasets grew another thousandfold, mostly from the web.

Going deeper: VGG, GoogLeNet, ResNet (2014–2015)

- **VGG (2014)**: stack many small 3×3 filters: 16–19 layers.
- **GoogLeNet (2014)**: process several filter sizes in parallel.
- **The problem**: beyond about 20 layers, plain networks got *worse*, even on training data.
- **ResNet (2015)**: add a shortcut so each block only learns a *correction*. Trained 152 layers; won ImageNet 2015.



Intuition

It is easier to learn "make a small change to the input" than "rebuild the input from scratch". If a block has nothing useful to add, it can learn $F(x) \approx 0$ and pass x through untouched.

Beyond "what": detection and segmentation go deep (2014–2017)

Detection

- **R-CNN (2014)**: propose regions, classify each with a CNN. Accurate but slow.
- **Faster R-CNN (2015)**: the network proposes its own regions.
- **YOLO (2016)**: "you only look once": one pass, real-time. Now in its 12th generation, as you saw on the Douala street.

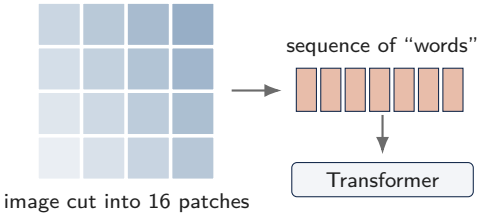
Segmentation

- **FCN (2015)**: a CNN that outputs an image, not a label.
- **U-Net (2015)**: designed for biomedical images with few training examples; still the default in medical imaging.
- **Mask R-CNN (2017)**: detection plus a mask per object: instance segmentation.

Practitioner's Note

All of these are Week 2 material. What matters now: one backbone network, many "heads" for different outputs.

Transformers enter vision (2020)



- The **Vision Transformer (ViT)**: “an image is worth 16×16 words”.
- Each patch can *attend* to every other patch from the first layer: global context at once.
- Needs more data than a CNN, but scales very well.

Intuition

A CNN builds up context slowly, layer by layer, from local neighbourhoods. A transformer asks, at every layer, “which other parts of the image are relevant to this one?”. Week 2 compares the two.

Learning without labels: self-supervised learning

- Labels are expensive; unlabelled images are nearly free.
- **Idea**: invent a task whose answer is in the image itself.
- **Contrastive** (SimCLR, MoCo, 2020): two crops of the *same* photo should get similar features; crops of different photos should not.
- **Masked** (MAE, 2021): hide 75% of the patches and predict them.
- **DINO / DINOv2 (2021–23)**: features so good that a simple classifier on top is enough.

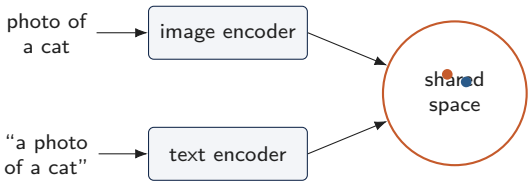
Intuition

Remember amodal completion with the occluded cats? Masked training asks a network to do exactly that: imagine the hidden parts. To do it well, it has to learn what cats look like.

Practitioner's Note

For African use cases with little labelled data, self-supervised pre-training plus a few hundred labels is often the winning recipe.

Vision meets language: CLIP (2021)



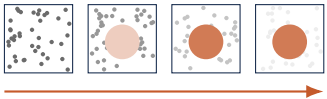
- Trained on **400 million** (image, caption) pairs from the web.
- Learns to place a photo and its caption *close together* in one shared space.
- At test time: write the class names as sentences and pick the closest. That is how we classified maize disease earlier without training.

Intuition

Language becomes the interface: you no longer need a fixed list of 1000 classes. This idea underlies text-to-image generation and today's multimodal assistants (Week 3).

Generative vision: from GANs to diffusion

- **GANs (2014)**: a generator and a critic compete; the generator learns to fool the critic. First photorealistic faces.
- **Diffusion (2020)**: learn to *remove noise*, step by step. Start from pure noise and denoise into an image.
- **Text-to-image (2022)**: DALL·E 2, Stable Diffusion: diffusion guided by CLIP-like text features.
- **Video generation (2024–)**: consistent motion over seconds to minutes.



Intuition

Denoising is image processing (Day 3). A diffusion model is a denoiser so good that it can invent the signal from nothing.

Foundation models: one model, many tasks (2023–today)

- **Segment Anything (SAM, 2023)**: trained on 1.1 billion masks in 11 million images. Click on any object and get its mask, even for objects never seen in training. SAM 2 (2024) extends this to video.
- **Depth Anything (2024)**: monocular depth for any photo (our Douala example).
- **DINOv2 (2023)**: general-purpose visual features without labels.
- **Multimodal assistants** (since 2023): you show them an image and ask questions in words.

The shift

Before: collect data, train a model for *one* task. Now: start from a huge pre-trained model and *adapt* it with a little data, or just a prompt.

Practitioner's Note

These models are large, hungry for compute, and trained mostly on images from the Global North. Knowing *when not to use them* is part of your skill set.

What is still unsolved

- **Robustness**: models break under new conditions: a new microscope, a new country, a dusty lens, the weasel.
- **Fairness**: in 2018, commercial face-analysis systems had error rates up to **34.7%** for darker-skinned women vs. **0.8%** for lighter-skinned men (Buolamwini & Gebru).
- **Data gaps**: African crops, diseases, scenes and skin tones are under-represented in public datasets.
- **Efficiency**: running on a phone or a cheap microscope, offline, on battery.
- **Understanding**: physics, causality, counting, reasoning about hidden things.
- **Trust**: explaining decisions, knowing when the model does not know.
- **Misuse**: surveillance, deepfakes, privacy.

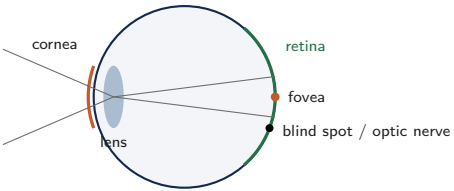
Intuition

Several of these are best tackled by people who understand both the methods *and* the local context. That is you.

Part 7

Human vision: a template for machines

The eye is a camera . . . with a very unusual sensor

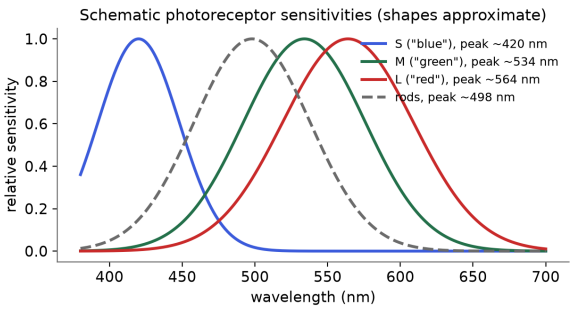


- Cornea and lens focus an *inverted* image on the retina, like a camera.
- **Rods** (about 10^8): very sensitive, no colour, night vision, periphery.
- **Cones** (about 6×10^6): colour and detail, packed into the fovea.
- Sharp vision covers only about 2° (a thumbnail at arm's length); the eye *jumps* 3–4 times a second to sample the scene.
- About 10^6 nerve fibres serve about 10^8 receptors: the retina compresses.

Intuition

A camera spends equal effort on every pixel; the eye samples at high resolution only where it looks. Attention in transformers recovers some of this efficiency.

Three cones, therefore three colour channels

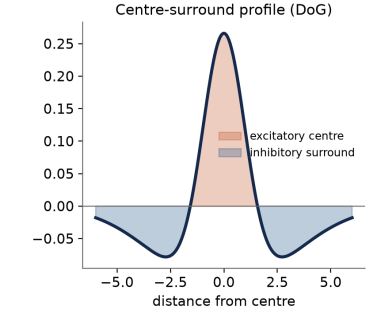


- **Trichromacy**: a colour percept is fixed by three numbers, the responses of the S, M and L cones.
- That is *why* cameras and screens use three channels (RGB): they only need to reproduce three cone responses.
- M and L overlap heavily and dominate perceived brightness. That is why the standard grayscale formula weights green the most (Day 2).

Practitioner's Note

Satellites and hyperspectral cameras record *many* more bands; near-infrared reveals plant health invisible to us. Machines need not inherit the limits of human vision.

The retina already filters: centre–surround receptive fields



Input



Response: edges are emphasised

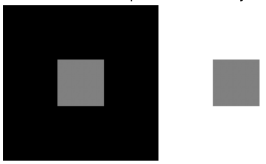


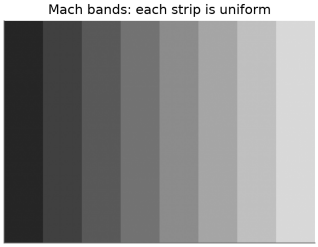
- Each retinal ganglion cell responds to a small patch of the visual field: its **receptive field**.
- ON-centre / OFF-surround: light in the centre excites, light in the ring around it inhibits. Mathematically, a *difference of Gaussians*: a narrow bump minus a wide bump.

Intuition

On a uniform region, centre and surround cancel and the cell is silent. At an edge they do not cancel. So the eye sends *changes and contrast*, not raw brightness. We meet the same filter again on Day 4, inside SIFT.

Seeing is inference, not measurement



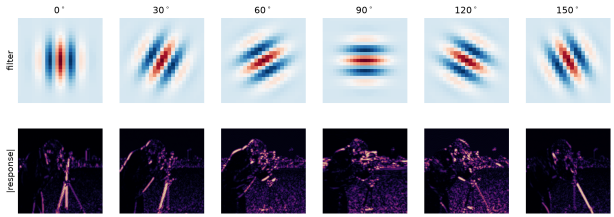


- **Simultaneous contrast**: identical grey looks lighter on black and darker on white.
- **Mach bands**: each strip is uniform, yet you see a light line and a dark line at every step.

Why Mach bands appear

Run the centre–surround filter across a step. Just on the bright side, the surround still sees some dark, so the response overshoots; just on the dark side it undershoots. Your perception follows the filter output, not the light.

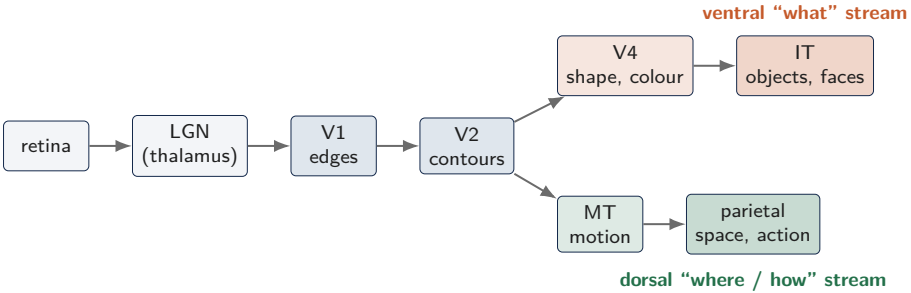
Hubel & Wiesel: orientation-selective cells in V1



Top: Gabor filters, a good model of V1 simple cells. Bottom: their response on an image. Each one "lights up" on edges at its preferred orientation.

- 1959–62, recordings from single neurons in cat visual cortex (Nobel Prize 1981).
- **Simple cells**: respond to a bar or edge at one orientation *and* one position: like a convolution filter.
- **Complex cells**: same orientation, but tolerant to small shifts: like *pooling*.
- Fukushima's Neocognitron (1980), and later CNNs, copy this simple $\rightarrow$ complex stacking.

A hierarchy with two streams



- Moving up the hierarchy, **receptive fields grow**, and features become **more abstract** and **more invariant**.
- A familiar object is recognised in about 100–150 ms: mostly a fast *feed-forward* sweep, with feedback refining it afterwards.

The machine analogue

In a deep network, early layers detect edges, middle layers detect textures and parts, late layers detect objects. Receptive fields grow with depth, just as in cortex (Week 2).

Gestalt: we see wholes, not parts



proximity
two groups, not 16 dots



similarity
columns, by colour



closure
a circle, despite gaps



continuity
two lines crossing



figure–ground
square or two bites?

Gestalt psychologists (1920s) found that we automatically **group** elements by proximity, similarity, closure and good continuation, and separate **figure from ground**.

Intuition

Grouping is exactly what *segmentation* must do. Amodal completion of the occluded cats is the same process. Classical segmentation (Week 1) and attention (Week 2) are both attempts to formalise it.

What should machine vision borrow, and what not?

Borrow

- Local receptive fields $\rightarrow$ convolution
- The same detector everywhere $\rightarrow$ weight sharing
- Simple $\rightarrow$ complex cells $\rightarrow$ filters + pooling
- A hierarchy of increasingly abstract features
- Normalising contrast and brightness
- Attention: spend computation where it matters

Don't copy blindly

- Brains learn from few labels, lots of feedback and an active body; machines (mostly) do not.
- Machines can see beyond RGB: infrared, X-ray, radar, depth.
- Brains are fooled by illusions; networks are fooled by *different* things (texture, the weasel, adversarial noise).
- Aeroplanes fly without flapping: borrow *principles*, not mechanisms.

Intuition

Every item in the left column will reappear this week or next as a concrete operation you can write in a few lines of Python.

Part 8

Wrapping up

Key takeaways

1. **Computer vision** turns images into descriptions. It is the inverse of graphics, uses image processing as a tool, and relies on machine learning for its priors.
2. **Seeing is hard** because the pixels change enormously for reasons that do not matter (pose, occlusion, scale, light, clutter) and only slightly for reasons that do.
3. **Tasks form a ladder**: classification $\rightarrow$ detection $\rightarrow$ semantic $\rightarrow$ instance segmentation, with pose, depth and language alongside.
4. **Progress came in two eras**: designed features until 2012, learned features since. Data and compute were the catalysts; foundation models are the current frontier.
5. **Biology inspired the core ideas**: receptive fields, filters, pooling and hierarchy. We borrow principles, not mechanisms.

Before the next lecture

Set up

- Install Python 3.10+, Jupyter, NumPy, Matplotlib, scikit-image, OpenCV, or open a Colab notebook.
- Run the first tutorial notebook, `W1D1_Images_as_Arrays`, with the tutors.
- Load any photo you took yourself and print its `shape` and `dtype`.

Read

- Course notes, Chapter 1
- Szeliski (2022), Chapter 1 and §2.3, *The digital camera*

Next lecture: how light becomes numbers

- The pinhole camera and the sensor
- Sampling: why too few pixels creates false patterns
- Quantisation: why 256 levels
- Colour spaces: RGB, grayscale, HSV
- The bugs that bite everyone: `uint8` overflow, BGR vs. RGB

Credits

All photographs are from Wikimedia Commons under the licences shown; model outputs were produced for this course with publicly released weights (ResNet-50, DeepLabV3, YOLOv12 and YOLO11-pose, Depth Anything V2, CLIP ViT-B/32).

`cat_leaves`: RyanDannyChacko, CC BY-SA 4.0
`cat_curtain`: Famartin, CC BY-SA 4.0
`cat_lace`: Famartin, CC BY-SA 4.0
`cat_rock`: Wikimedia user Rohit Sav, CC BY-SA 4.0
`cat_stretch`: pelican from Tokyo, Japan, CC BY-SA 2.0
`cat_roll`: Clavecin, Public domain
`cat_curled`: Matthew Paul Argall, CC0
`cat_back`: Wasianpower, CC BY-SA 4.0
`cat_small`: Hisashi from Japan, CC BY-SA 2.0
`cat_shadow`: UnknownCopy, CC BY-SA 4.0
`cat_sphynx`: Jakub Hałun, CC BY-SA 4.0
`cat_persian`: Gourdo 1 at en.wikipedia, Public domain
`leopard_tree`: Davidvraju, CC BY-SA 4.0
`leopard_grass`: Senthiaathavan, CC BY-SA 4.0
`husky`: MrPanyGoff, CC BY-SA 3.0
`wolf`: USFWS/Gary Kramer, Public domain
`checker_shadow`: Edward H. Adelson, CC BY 3.0
`checker_proof`: Edward H. Adelson, 1995, CC0
`xray_tb`: B. Herreros et al., CC BY 4.0
`malaria_smear`: OpenFlexure Project, CC BY-SA 4.0
`histo_breast`: Mikael Häggström, CC0
`pcb`: Ashley Pomeroy, CC BY 4.0
`crack_wall`: Sunolafjagtenben-hur, CC0
`pothole_nairobi`: Brihaspati, CC BY-SA 4.0
`maize_blight`: Chroanch, CC BY-SA 4.0
`douala_deido`: Happiraphael, CC BY-SA 4.0
`douala_ndokoti`: Minette Lontsie, CC BY-SA 4.0
`douala_2014`: Guaka, CC BY-SA 4.0
`yaounde_taxis`: Bdx, CC BY-SA 4.0
`moto_mbouda`: Photokadaffi, CC BY-SA 4.0
`lake_chad_apollo7`: NASA, Public domain
`lake_chad_2024`: NASA MODIS Rapid Response, Public domain
`camera_trap_fox`: Dodro, CC BY 4.0

Retina: scikit-image sample data. Grey cat (pixel grid): scikit-image `chelsea`.