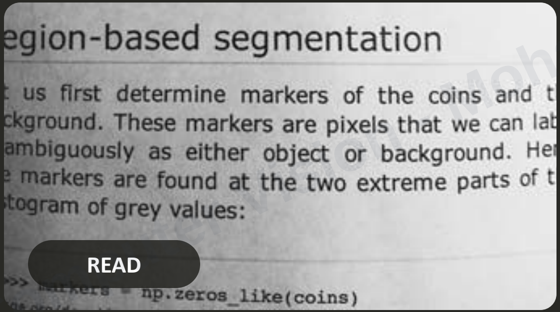
Computer Vision
for CS & AI

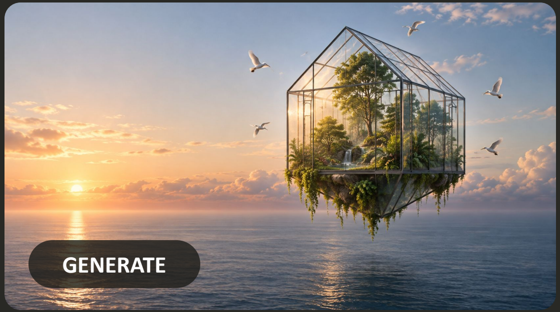
Session 1 — Images, Vision, and Numbers

A two-hour visual introduction: from real applications to pixels, matrices, and robust representations

OPENING QUESTION · 2–5 MIN

What can machines see today?





APPLICATION 1

Autonomous driving: perception before action

The system must locate objects, estimate depth, follow motion, and understand free space—fast enough to influence a decision.

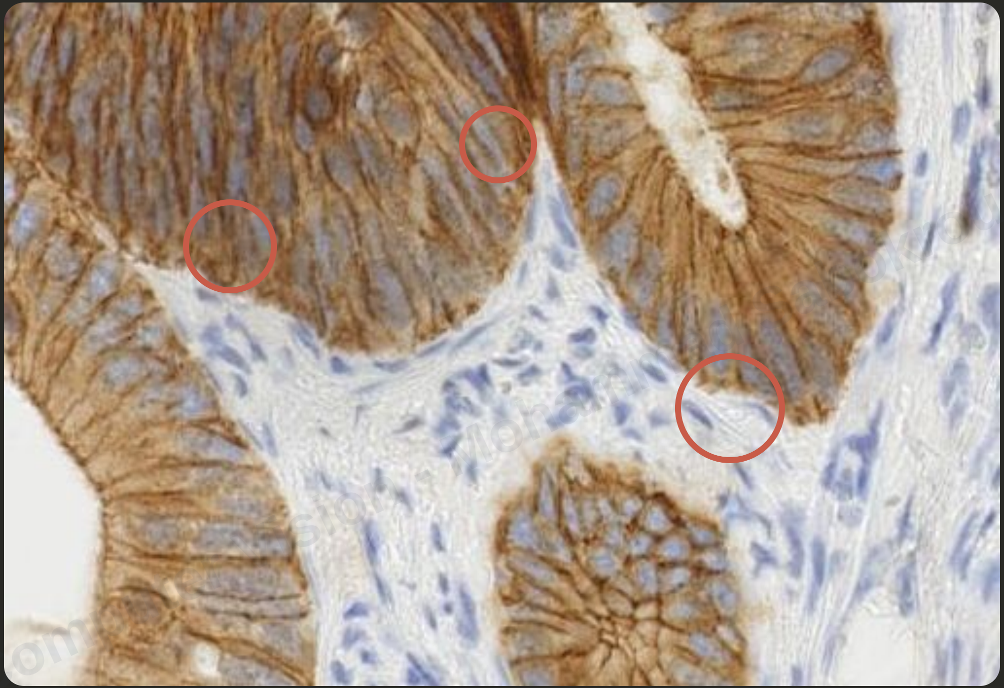
Hard cases

rain · glare · occlusion

unusual roads · rare events

APPLICATION 2

Medical images: highlight evidence, preserve judgement

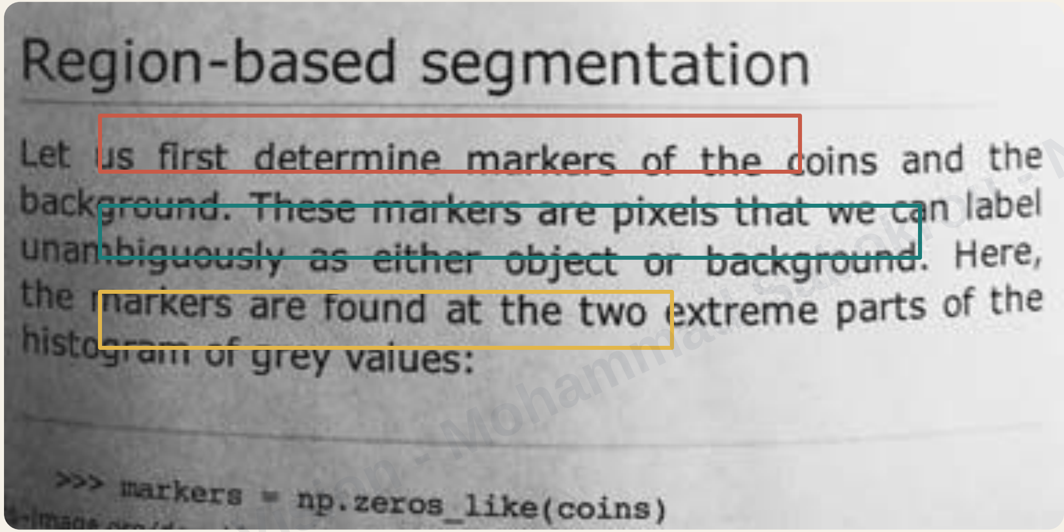


Possible output

A heatmap or region proposal that directs attention—not an unexplained verdict.

Validation must account for scanner changes, rare findings, false reassurance, and demographic shift.

Document understanding: pixels → structure → meaning



Region-based segmentation

Let us first determine markers of the coins and the background. These markers are pixels that we can label unambiguously as either object or background. Here, the markers are found at the two extreme parts of the histogram of grey values:

```
>>> markers = np.zeros_like(coins)
```

1 · Find

page, lines, tables

↓

2 · Read

characters and words

↓

3 · Organize

entities and relationships

Same page, different shadows and perspective: a nuisance-removal problem before OCR even begins.

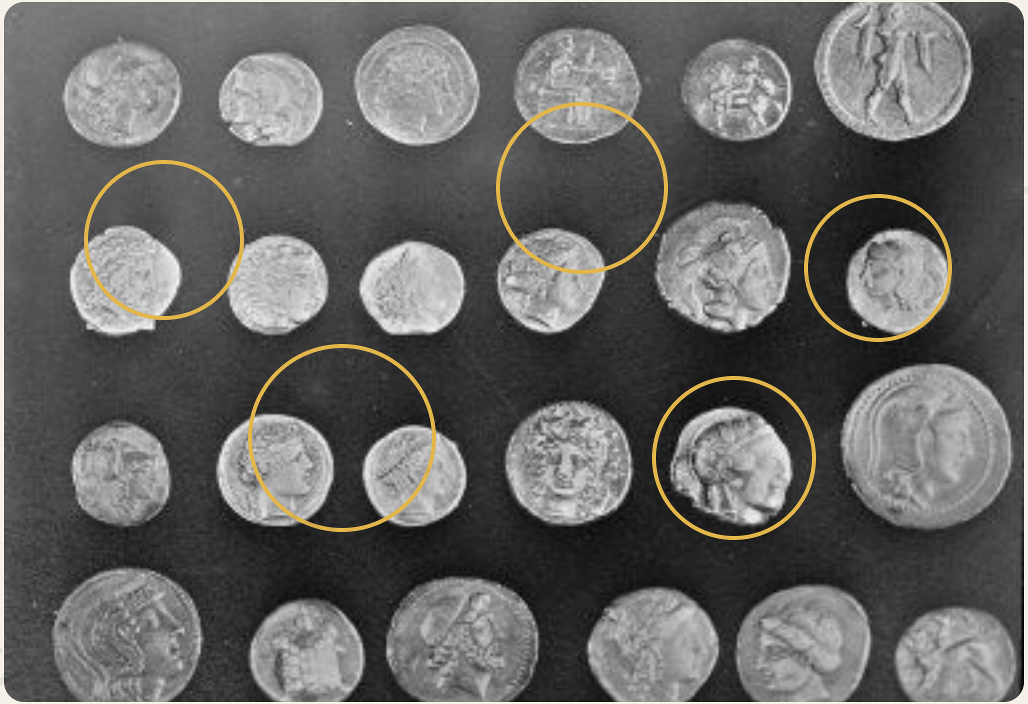
APPLICATION 4

Scientific vision turns faint light into discoveries

Detect · classify · measure shape · estimate redshift proxies · find the unusual

Question: what counts as ground truth when the object is too distant to inspect directly?

Counting is geometry, not only recognition

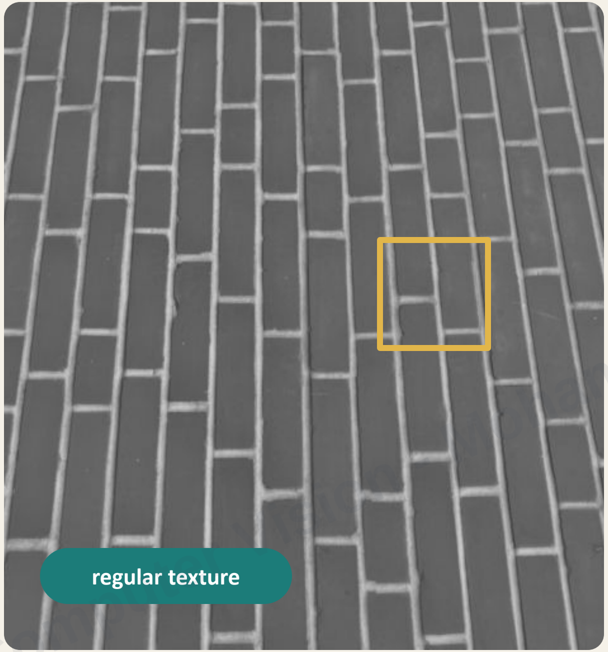


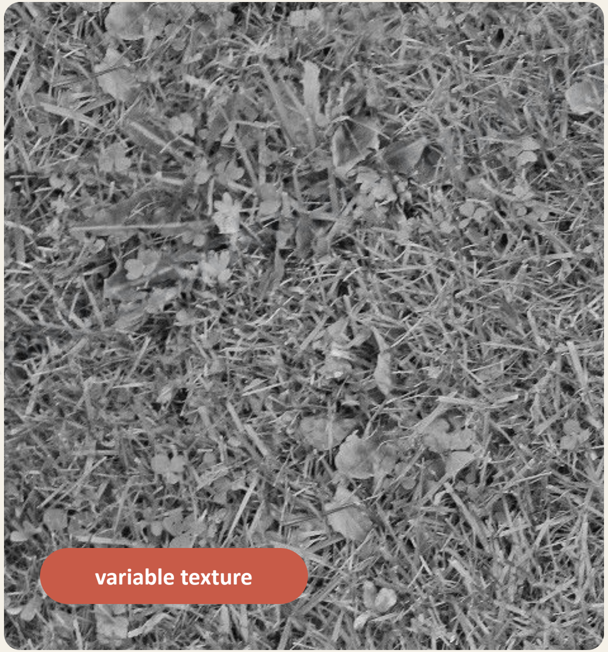
A useful answer may be:

How many?
Where?
How large?
Do objects overlap?

The output is a set of measurements with uncertainty—not just “coins.”

Industrial inspection: detect the rare defect





A defect may be a tiny deviation from a repeated pattern.

The challenge is learning normal variation without hiding the anomaly.

AI-GENERATED EXAMPLE

Vision can also synthesize

A model learns regularities from image collections and produces a new sample conditioned on a prompt or another image.

Key question
Does plausible-looking mean physically correct, original, or safe to use?

CORE IDEA

What is computer vision?

Computer vision builds systems that infer useful structure about the world from visual measurements.

IMAGE / VIDEO
measured light
a tensor of values

→

REPRESENTATION
features, geometry,
learned embeddings

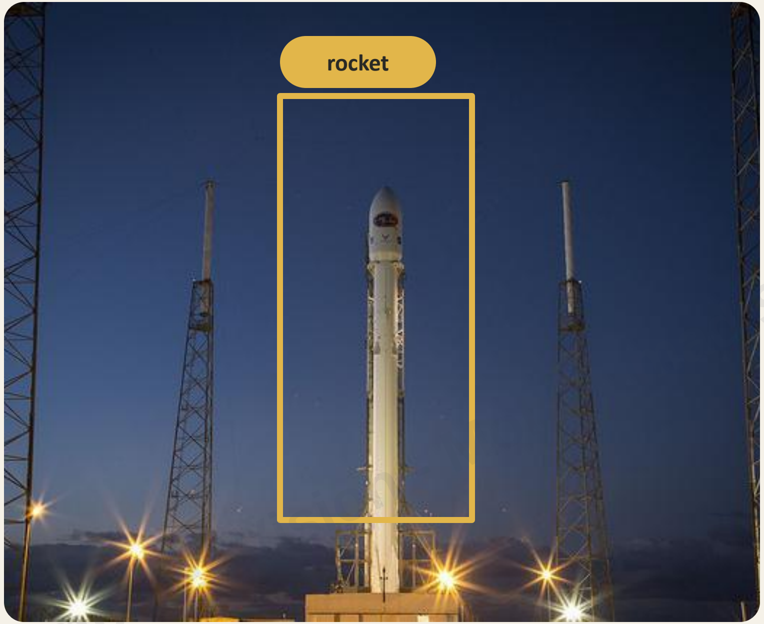
→

OUTPUT
labels, locations,
shape, motion, pixels

$$I \rightarrow z = f_\theta(I) \rightarrow \hat{y} = g(z)$$

ONE INPUT, MANY QUESTIONS

The output determines the vision task



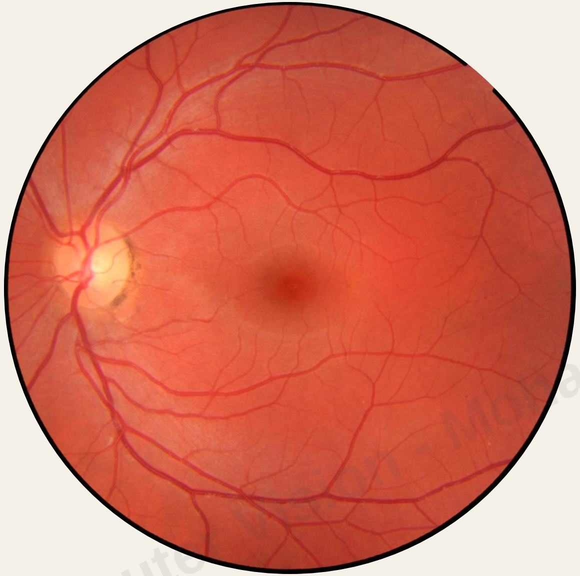
Classification	What is present?
Detection	Where are the objects?
Segmentation	Which pixel belongs to what?
Depth	How far is each surface?
Tracking	Where does it move over time?
Generation	What new image satisfies a condition?

COMPARISON

Human vision and computer vision solve related—not identical—problems

	HUMAN	MACHINE
SENSOR	retina + eye movements	camera / lidar / microscope
EXPERIENCE	embodied, lifelong, contextual	training data + objectives
STRENGTH	common sense and flexible context	speed, scale, repeatability
FAILURE	illusions, attention, fatigue	shift, shortcuts, adversarial patterns

Seeing begins with light; meaning emerges through processing



LIGHT
reflected wavelengths

→

OPTICS
cornea + lens focus

→

RETINA
photoreceptors transduce

NEURAL SIGNALS
contrast, edges, motion

→

BRAIN
integrates context + goals

The optic nerve carries retinal signals to the brain; perception is constructed from those signals, not copied pixel-for-pixel.

A camera records measurements; an algorithm must infer the scene

SCENE

light + surfaces

LENS

projection + blur

SENSOR

sample + quantise

$$\begin{bmatrix} 42 & 45 & 51 \\ 39 & 82 & 96 \\ 33 & 88 & 141 \end{bmatrix}$$

ALGORITHM

array ⟶ representation
⟶ prediction + uncertainty

Context changes what identical measurements appear to mean

A B

An image is a sampled, quantised measurement—not the scene itself

scene radiance

OPTICS + SENSOR

blur · exposure
noise · sampling

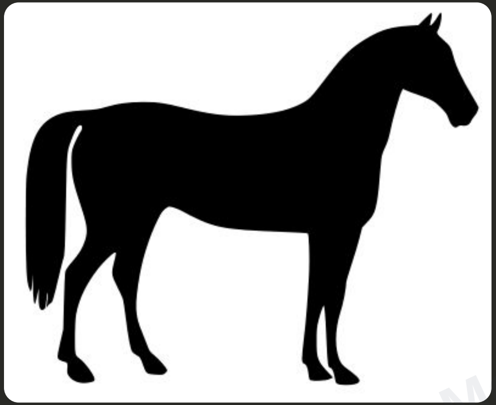
→

$$\begin{bmatrix} 18 & 22 & 29 & 31 \\ 20 & 26 & 94 & 88 \\ 23 & 31 & 121 & 104 \\ 21 & 29 & 76 & 69 \end{bmatrix}$$

$$I[r,c,k] = Q(g\square(E(r,c)))$$

location (r,c) · channel k · irradiance E · sensor response g · quantisation Q

Binary, grayscale, and colour images



BINARY

`H × W · {0,1}`

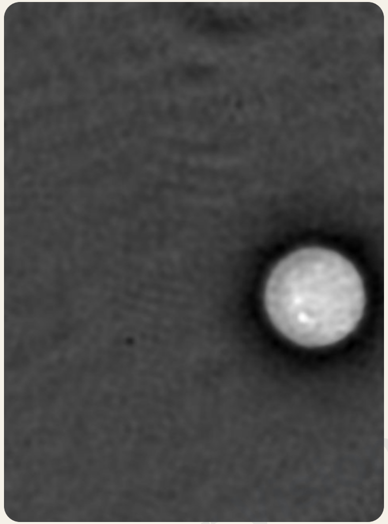
GRAYSCALE

`H × W · intensity`

RGB COLOUR

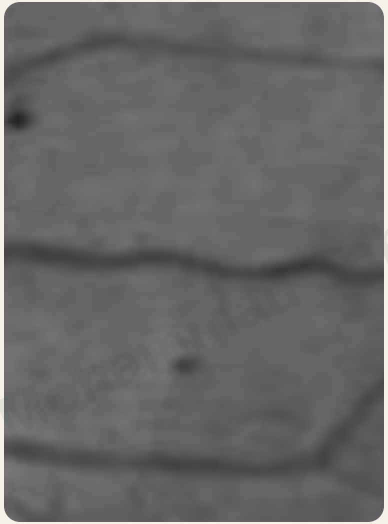
`H × W × 3`

Beyond ordinary photographs



microscopy

sensor-specific intensity

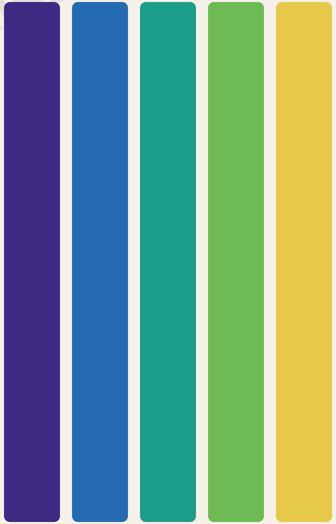


tiny diagnostic crop

small structures matter

depth / range

value means distance



multispectral

many wavelength bands

Shape tells you what the axes mean

GRAYSCALE	$H \times W$	`I[r,c]`
RGB	$H \times W \times 3$	`I[r,c,k]`
BATCH	$N \times H \times W \times C$	`X[n,r,c,k]`
VIDEO	$T \times H \times W \times C$	`V[t,r,c,k]`

Same numbers + different axis convention = different meaning

From visual intuition to evidence

APPS

0–25 min
What vision systems do

VISION

25–50 min
Human + machine pathways

IMAGES

50–65 min
Formation, types, tensors

BREAK

65–70 min
Questions + reset

NUMBERS

70–120 min
Pixels, failure, representations, Lab 1

Prediction → worked example → failure → diagnosis → repair

Foundations: from arrays to convolutional networks

01 Images as numbers; the recognition problem

02 Image formation, colour, intensity and histograms

03 Filtering, convolution and noise

04 Gradients, edges and corners

05 Local features and image matching

06 Geometry, transformations and alignment

07 Machine-learning foundations for vision

08 Neural networks and convolutional networks

Systems: train, evaluate, detect, segment, and track

09 Training CNNs, transfer learning and augmentation

10 Robust evaluation and distribution shift

11 Object detection

12 Semantic and instance segmentation

13 Video, motion and tracking

14 Project development and experimental design

15 Trustworthiness: robustness, fairness and explainability

16 Project presentations, synthesis and review

COURSE STRUCTURE

How the course works

2 hours

Lecture

Concepts, worked examples, questions, and short prediction tasks

2 hours

Laboratory

Run an experiment, break a method, repair it, and record evidence

16 weeks

Weekly record

An executed notebook with outputs, measurements, and a short reflection

ASSESSMENT

How your laboratory work contributes

10%

Lab portfolio

Labs 1–15. Each lab receives a mark out of 20. Lab 0 is compulsory setup work.

15%

Project work

The project develops through Labs 11–14 and measures application building and evaluation.

The official module plan defines the remaining 75% of the course grade.

What earns the 20 marks in each lab

- A working implementation that runs from the first cell to the last
- Measured results, including plots, tables, or printed values requested in the lab
- A deliberate failure case and a diagnosis supported by your own output
- A short interpretation that connects the evidence to the week's concept
- Reproducibility: fixed seeds where needed, saved outputs, and clearly identified assisted code

Marks reward evidence. Polished prose without executed results earns little credit.

What to bring and how to work

Bring

- A laptop with a modern browser
- A Google account with Drive access
- Your saved notebook from the previous lab
- Your project data when the weekly notice asks for it

Course policies

- You may discuss ideas and debug in pairs
- Submit your own executed notebook
- Mark AI-assisted code with # ASSISTED and be ready to explain it
- Never upload private, sensitive, or unlicensed data

Prepare the notebook and short report

Before the lab

Open the released notebook and choose File > Save a copy in Drive.

During the lab

Run every cell in order. Keep outputs visible. Save figures and measurements.

By 23:59

Email both files by 23:59 on the day of your scheduled lab.

Before sending: Runtime > Restart and run all. Keep outputs visible, then attach the notebook and PDF report.

Send the notebook and short report by email

TWO FILES

StudentID_WeekXX.ipynb
StudentID_WeekXX_Report.pdf

EMAIL

m.sabokrou@newuu.uz
Subject: CV Lab XX — Student ID — Full name

Deadline 23:59 on your scheduled laboratory day, unless announced otherwise

Before sending Runtime ⟶ Restart and run all · keep outputs visible · agree late arrangements before the deadline

5-minute break

When we return: one photograph → pixels → matrices → a classifier that fails

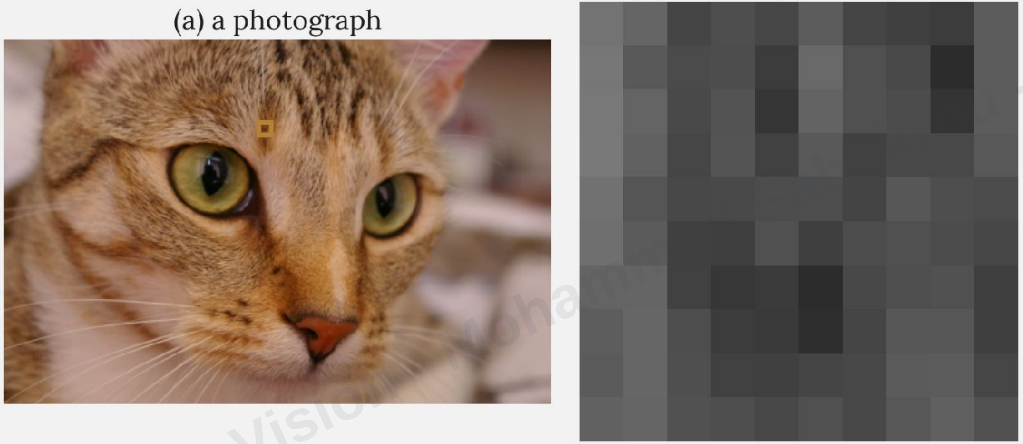
Questions about grading, policy, or submission? Ask now.

THE QUESTION

You knew before you decided anything

(a) a photograph

(b) 10x10 region, magnified



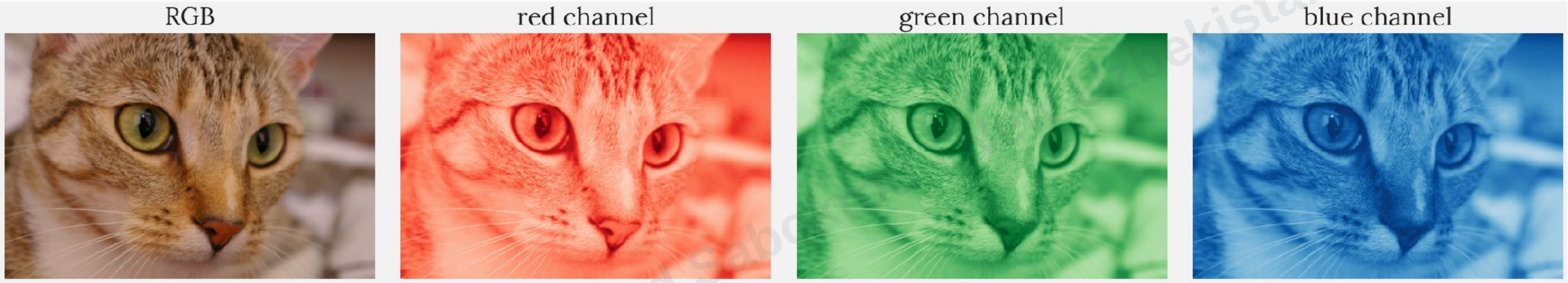
(c) the same region, as numbers

115	101	69	78	71	92	69	64	60	92
119	89	75	77	60	105	80	74	45	92
120	100	74	83	54	99	80	76	53	92
122	98	70	84	66	89	65	73	73	89
114	85	72	74	77	78	68	87	78	75
111	91	64	63	81	63	77	80	76	71
100	101	65	55	59	45	72	79	82	63
93	102	78	61	58	46	69	87	87	63
91	102	79	65	64	69	72	91	92	70
97	101	81	78	72	80	72	88	96	81

A photograph, a 10×10 patch near the eye, and that patch printed as numbers. Panel (c) is the entire input a computer receives.

WHAT AN IMAGE IS

Colour is three grids, not one



Grayscale	a function I(r,c) on a grid — shape (H, W)
Colour	three stacked grids — shape (H, W, 3)
8-bit depth	every value an integer in [0, 255]

405,900

numbers in this one small photograph

WORKED EXAMPLE

A pixel has a location and a value

Red channel

[[240, 220],
 [35, 20]]

Green channel

[[80, 70],
 [190, 170]]

A colour image stores one value per channel at each location. A 300 × 451 RGB image therefore has shape (300, 451, 3).

Always inspect shape, dtype, minimum, and maximum before processing an image.

A PROPOSAL

The idea every student has first

> *"To recognise an object, store a picture of it and compare new images pixel by pixel."*

It is a good proposal, because its failure is instructive.

1. **Store**

   one template image per object

2. **Compare**

   mean absolute difference, pixel by pixel

3. **Decide**

   nearest template wins

WORKED EXAMPLE

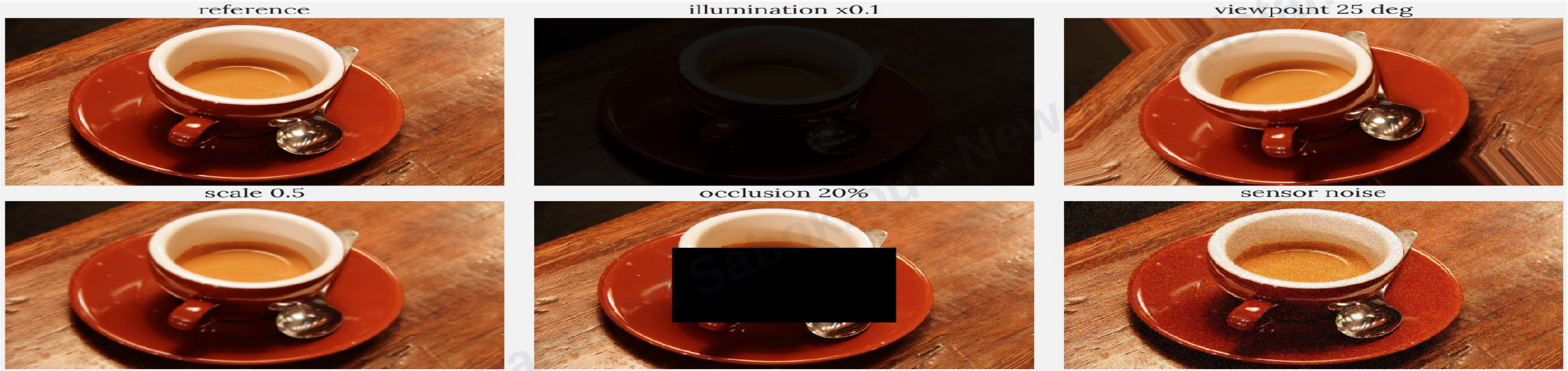
Mean absolute difference, step by step

Template T	Query Q	|Q − T|
[[10, 20], [30, 40]]	[[12, 18], [33, 37]]	[[2, 2], [3, 3]]

MAD = (2 + 2 + 3 + 3) / 4 = 2.5

The calculation is valid. The assumption is fragile: corresponding pixels must describe corresponding parts of the scene.

Same cup of coffee, six times



Illumination · Viewpoint · Scale · Occlusion · Sensor noise · Deformation · Background clutter · Intra-class variation

You classified all six correctly without effort. Learn these eight by name — you will use them all semester to diagnose why a model is failing.

A one-pixel shift changes many comparisons

Original row

[0, 0, 255, 255, 0, 0]

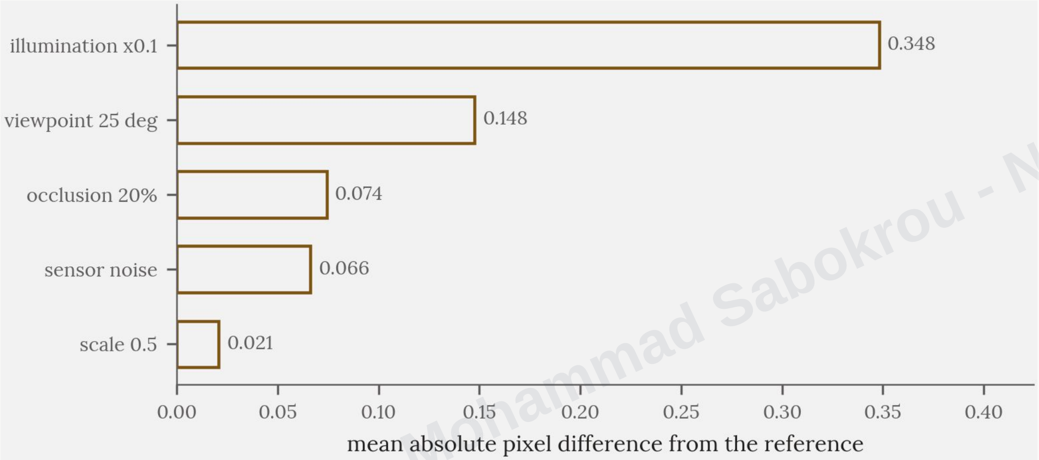
Shifted one pixel

[0, 0, 0, 255, 255, 0]

Four positions agree, two disagree completely. The object did not change, but pixel correspondence did.

THE RESULT

Now measure what the machine measures



Dimming the lights

changes the data more than

covering a fifth of the cup

A rule based on pixel distance calls a dark photo of the cup less like the cup than a heavily occluded one.

In Lab 1 this same effect takes a working classifier from 100% to 10% accuracy — on images a human still reads perfectly.

READING EVIDENCE

What the failure experiment establishes

Observation

A change in illumination moves the same cup farther from its template than covering part of the cup.

Conclusion

Raw pixel distance responds strongly to nuisance changes. It does not reliably measure object identity.

Next question: which transformation keeps the useful signal while reducing sensitivity to lighting?

SO WHY NOT JUST STORE EVERY IMAGE?

$10^{362,000}$

possible 224×224 colour images

10^{80}

atoms in the observable universe

Memorising images is not inefficient. It is arithmetically impossible — so a working system must compute a representation instead.

Useful features keep signal and discard nuisance

Image

150,528 raw values
for 224 × 224 RGB

Representation

A smaller description chosen to preserve information needed for a task

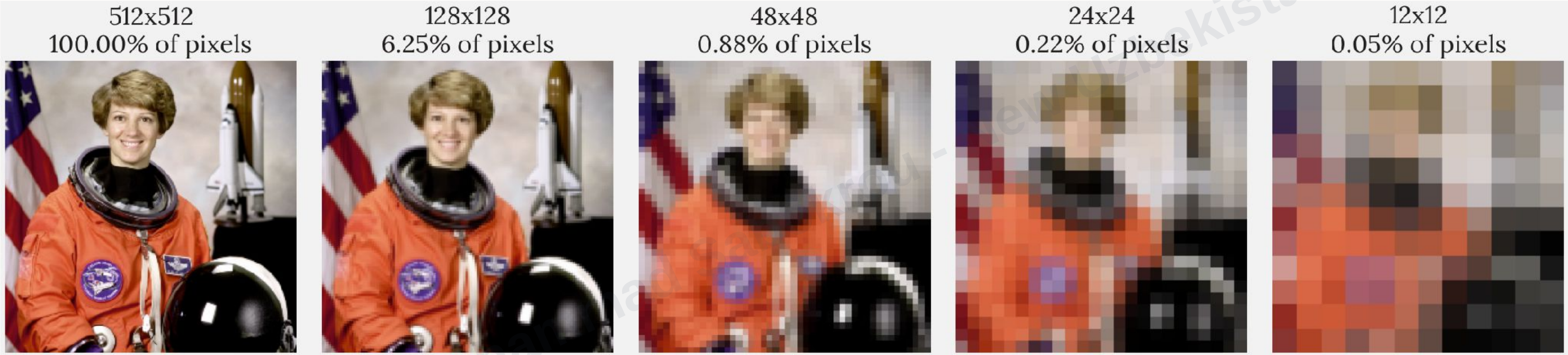
Examples

Histogram
Edge map
Local descriptor
CNN feature vector

The choice depends on what must remain unchanged and what the task needs to distinguish.

ENCOURAGING NEWS

How little is actually enough?



576 pixels — 0.22% of the original — and most people still see a person.
The information that identifies an object survives massive compression, so a compact representation is at least possible.

Compression can preserve recognition cues

224 × 224 × 3

150,528 pixel values

16 × 16 × 3

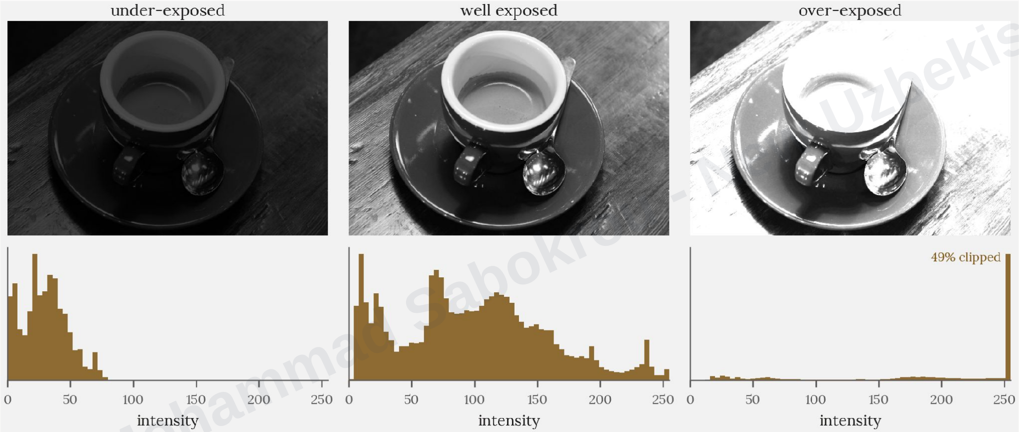
768 pixel values

The smaller image uses about 196 times fewer values.

Recognition can survive because neighbouring pixels repeat structure. Fine detail disappears first, followed by shape and identity as resolution falls further.

THE IDEA STUDENTS SKIP

The data is made, not found



One scene, three exposures. The scene is identical; the data is not — the histograms barely overlap.

A model trained only on well-exposed images has never seen the other two, and will behave unpredictably when it meets them.

Separate the signal from nuisance variation

Question	Good design decision
What should determine the label?	The object or condition that the system must recognise
What should vary within each label?	Lighting, viewpoint, background, device, and other realistic nuisances
What must stay separate?	Near-duplicate images and images from the same capture sequence across train and test

A model learns shortcuts when the dataset makes a nuisance easier to predict than the intended signal.

WHEN THE HISTOGRAM IS A PERSON

In Lab 1 you will measure this yourself

20%

reported overall

40%

group A

0%

group B

Nobody wrote a rule excluding group B. The exclusion is a consequence of reporting an average.

Buolamwini & Gebru (2018) found under 1% error for lighter-skinned men and up to ~35% for darker-skinned women, in systems whose reported overall accuracy looked good.

An overall score can hide complete failure

Group A

2 / 5 correct

40%

Group B

0 / 5 correct

0%

Overall: 2 / 10 = 20%. Reporting only 20% conceals that one group receives no correct predictions.

THE ARC

Where this course goes

CH 1–2	CH 3–4	CH 5–9	CH 10–15
Images, light and histograms	**Filtering, edges, features**	**Learning, CNNs, depth**	**Detection, segmentation, tracking, evaluation**
What the data is	Hand-designed representations	Learned representations	Systems, and judging them

Every week: two hours of lecture, two hours of lab. The lab is where the ideas become yours.

SESSION SUMMARY

Five ideas to keep

- An image is an array whose shape, data type, and value range affect every operation
- Pixel distance assumes correspondence and breaks under common nuisance changes
- A representation should preserve task-relevant signal while reducing nuisance sensitivity
- Dataset choices determine which shortcuts a model can learn
- Evaluation must include meaningful groups and failure cases, not only an overall average

WHAT TO DO NOW

Before we meet again

0

Lab 0 — Getting Set Up

Two hours. Do this BEFORE Lab 1.

- Colab running, saved to your Drive
- A dataset loaded and inspected
- One image read, one image written
- Results saved and read back
- A green environment report

Read

Chapter 1, including the worked examples. They are short and they are the lecture in slow motion.

Bring

A laptop that can open a browser. Nothing is installed locally — all work happens in Colab.

If Lab 0 does not end with a green report, come to the first ten minutes of Lab 1 and we will fix it there.

SELF-STUDY CHECK

Questions to answer before Lab 1

- What four properties should you print immediately after loading an image?
- Why can a one-pixel shift create a large pixel distance?
- What assumption makes template matching fragile?
- Give one example of signal and one example of nuisance for face recognition
- Why should evaluation report group results as well as overall accuracy?

Write complete answers. Definitions alone do not show that you can apply the idea.

A concise answer key

- Print shape, dtype, minimum, and maximum
- A shift changes which locations are compared, even though the object remains the same
- Template matching assumes aligned pixels correspond to the same scene parts
- Face identity is signal. Lighting or camera viewpoint can be nuisance
- An average can conceal severe or complete failure for a subgroup

If any answer still feels vague, return to the worked example rather than memorising the sentence.

Next laboratory

You will build today's simple pixel classifier, watch it score 100%, and then break it with a light dimmer.

Then you will repair it with one line — and that line is the first representation you will ever write.

Build it. Break it. Repair it. Measure it.

1 · BUILD
A raw-pixel classifier from a small labelled set

2 · BREAK
Change brightness and watch accuracy collapse

3 · REPAIR
Normalise intensity before comparison

4 · EVALUATE
Report results, failure evidence, and groups

Deliverable: executed notebook + short report → m.sabokrou@newuu.uz

Next lecture: light, colour, and histograms

IMAGE FORMATION

How illumination, surface reflectance, optics, and exposure create pixel values

→

COLOUR SPACES

RGB, grayscale, HSV, channels, and value-range conventions

→

HISTOGRAMS

Describe intensity distributions and reason about contrast

→

EQUALISATION

When contrast improves—and when noise becomes stronger

Lab 2 · apply histogram equalisation, measure the change, and identify a case where it amplifies noise