VISION IN CODE

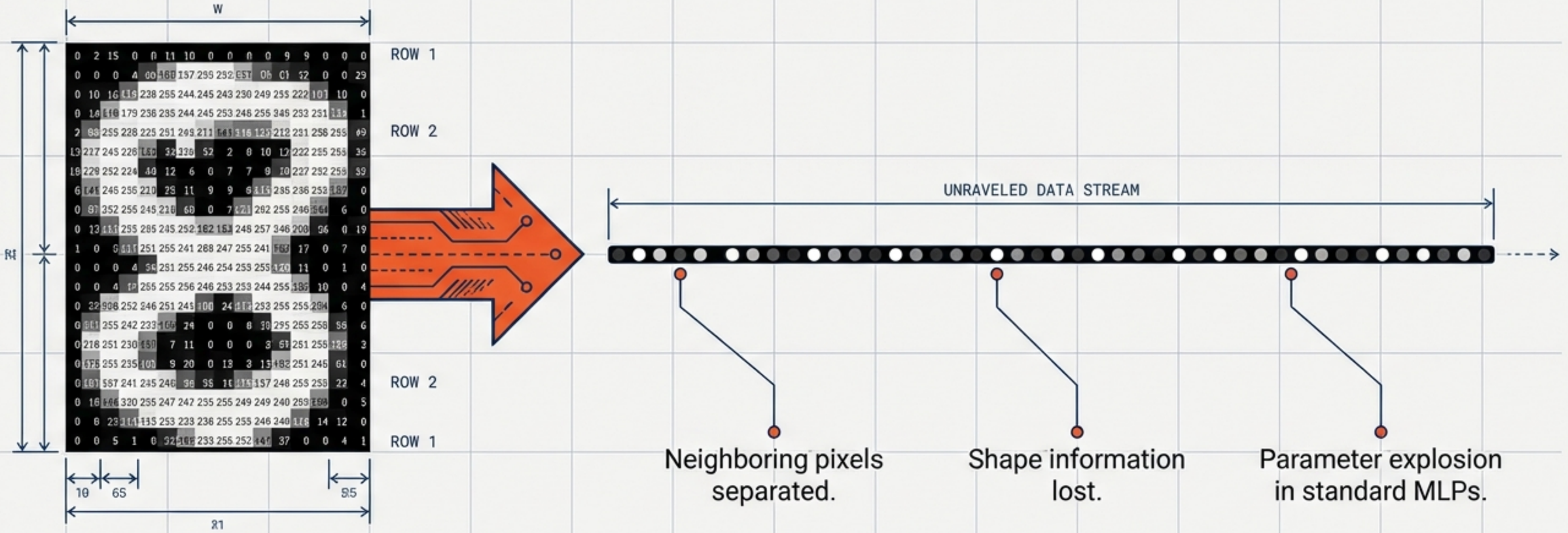
The Mechanics of Convolutional Neural Networks

THE GOAL:
Translate raw pixels into understanding.

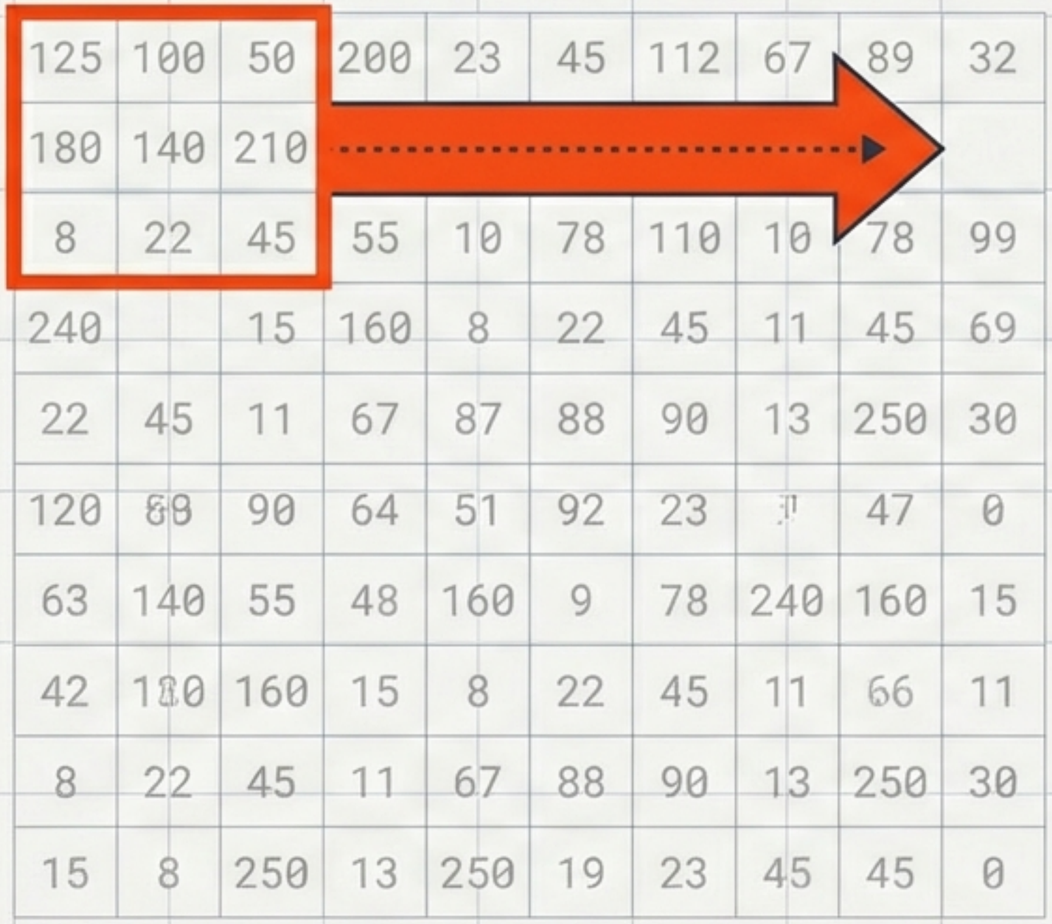
THE METHOD:
Btewiwetic architecture prioritiaing spatial relationships.

THE AUDIENCE:
A comprehensive guide to Deep Learning's vision system.

THE PROBLEM WITH FLATTENING.



THE SOLUTION: LOCAL RECEPTIVE FIELDS.



01. SPATIAL AWARENESS

Pixels are only related to their immediate neighbors. Distant pixels are irrelevant.

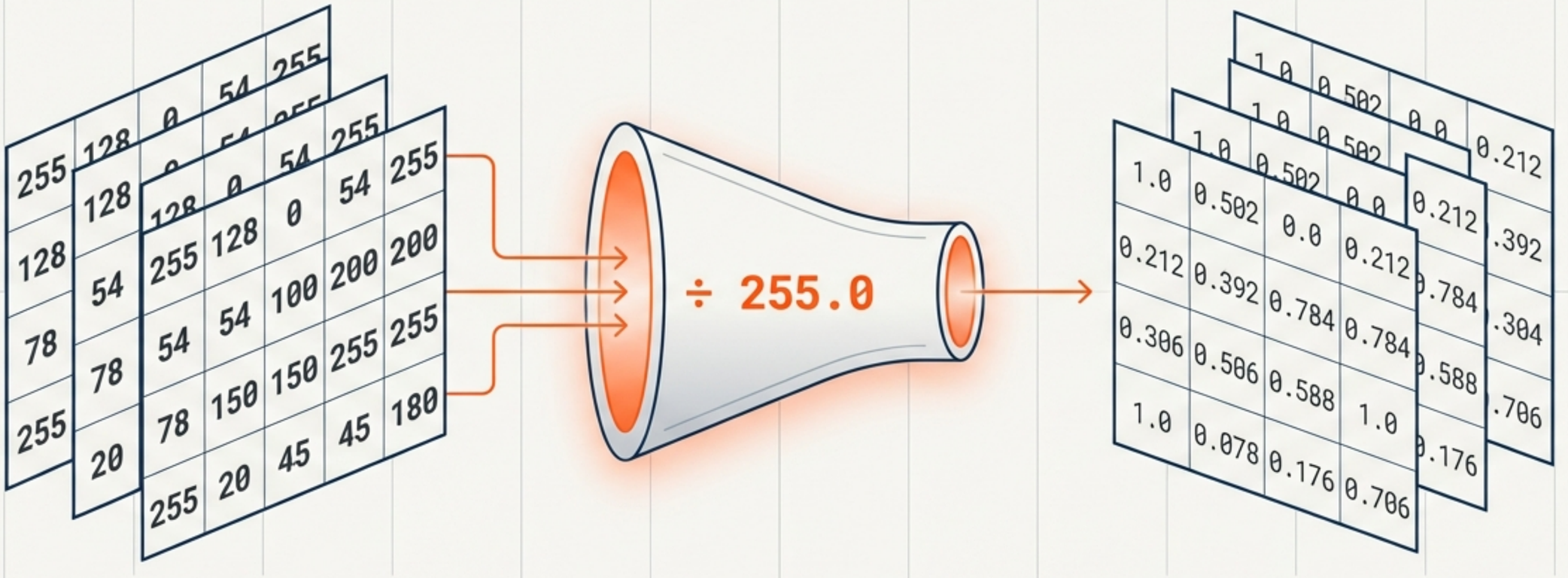
02. PARAMETER SHARING

The same filter (weights) scans the entire image. A vertical edge in the top-left is the same feature as a vertical edge in the bottom-right.

03. THE FLASHLIGHT METAPHOR

We scan the dark room with a beam, processing one patch at a time.

THE RAW INPUT.

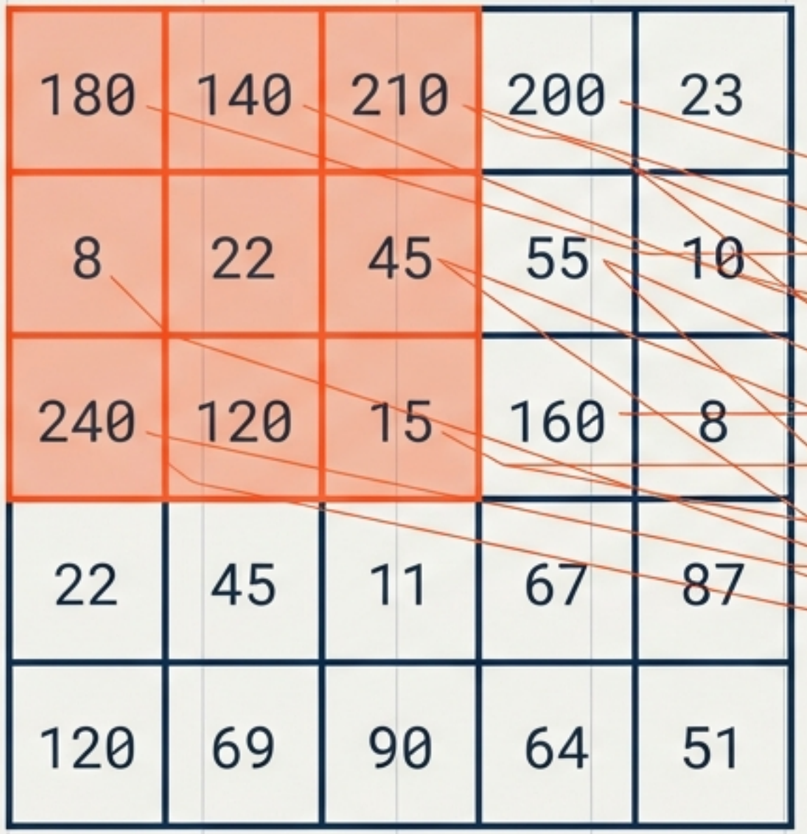


Raw Pixel Data [0 - 255]

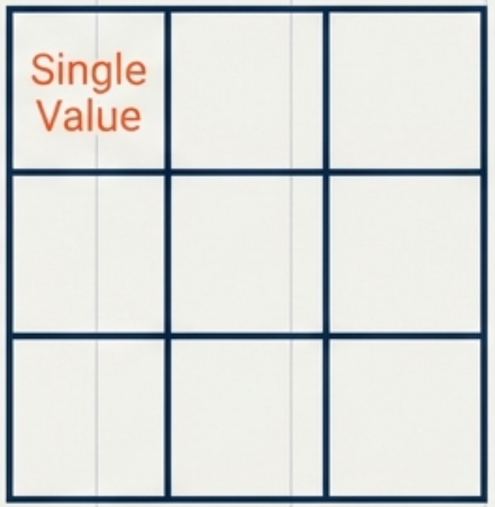
Normalized Tensor [0.0 - 1.0]

WHY NORMALIZE? Large integers destabilize gradients. Small floats (0-1) ensure efficient convergence.

THE ENGINE: CONVOLUTION.



(180 × w1)
+ (140 × w2)
+ (210 × w3)
+ (8 × w4)
+ (22 × w5)
+ (45 × w6)
+ (240 × w7)
+ (120 × w8)
+ (15 × w9)
= Single Value



THE KERNEL (FILTER)

A tiny matrix of learnable weights that slides over the input.

THE DOT PRODUCT

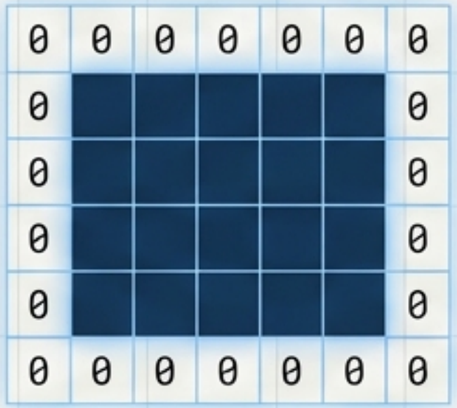
Element-wise multiplication followed by a sum. Measures how much the input patch matches the filter pattern.

CONTROLLING THE SCAN

Stride (s)

Larger stride = Smaller output.

Padding (p)



Zero-padding preserves spatial dimensions.

$$n_{out} = \text{floor}\left(\frac{n_{in} + 2p - k}{s}\right) + 1$$

n_in: Input size
p: Padding
k: Kernel size
s: Stride

THE OUTPUT: FEATURE MAPS & ReLU.

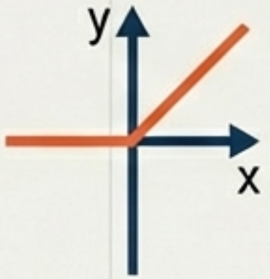
Input

w1	w2	w3
w4	w5	w6
w7	w8	w9

Convolution

120	-15	-22	-45	-12
-15	150	180	-11	-33
-45	-11	200	180	-8
-33	-8	-67	150	-90
-10	-67	-90	-18	120

Feature Map (Pre-Activation)



ReLU Function

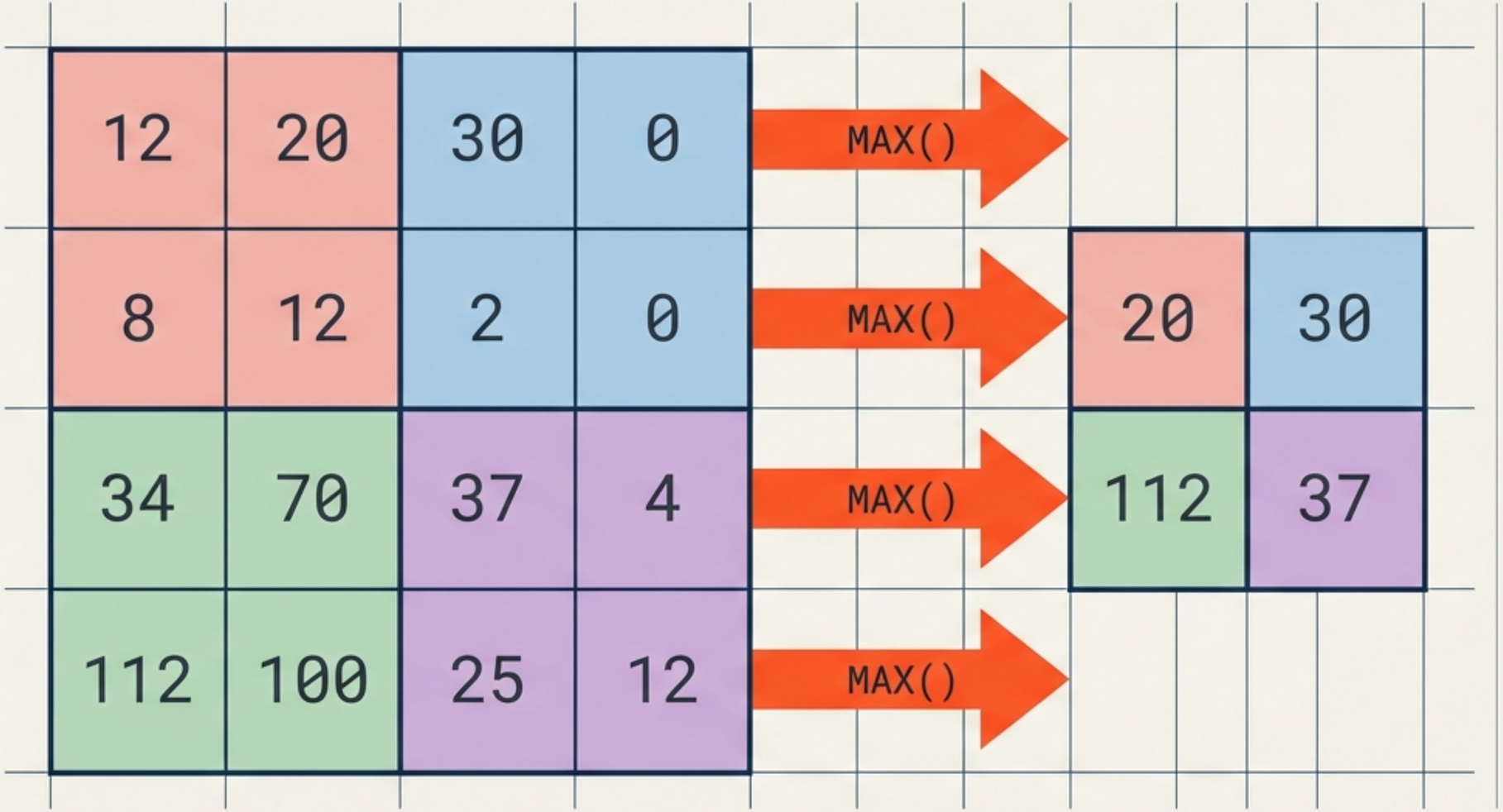
120	0	0	0	0
0	150	180	0	0
0	0	200	180	0
0	0	0	150	0
0	0	0	0	120

Feature Map (Post-Activation)

ACTIVATION (ReLU)

Rectified Linear Unit. It silences negative values (noise) and keeps positive matches. This non-linearity allows the network to learn complex shapes.

DOWNSAMPLING: MAX POOLING



THE PURPOSE

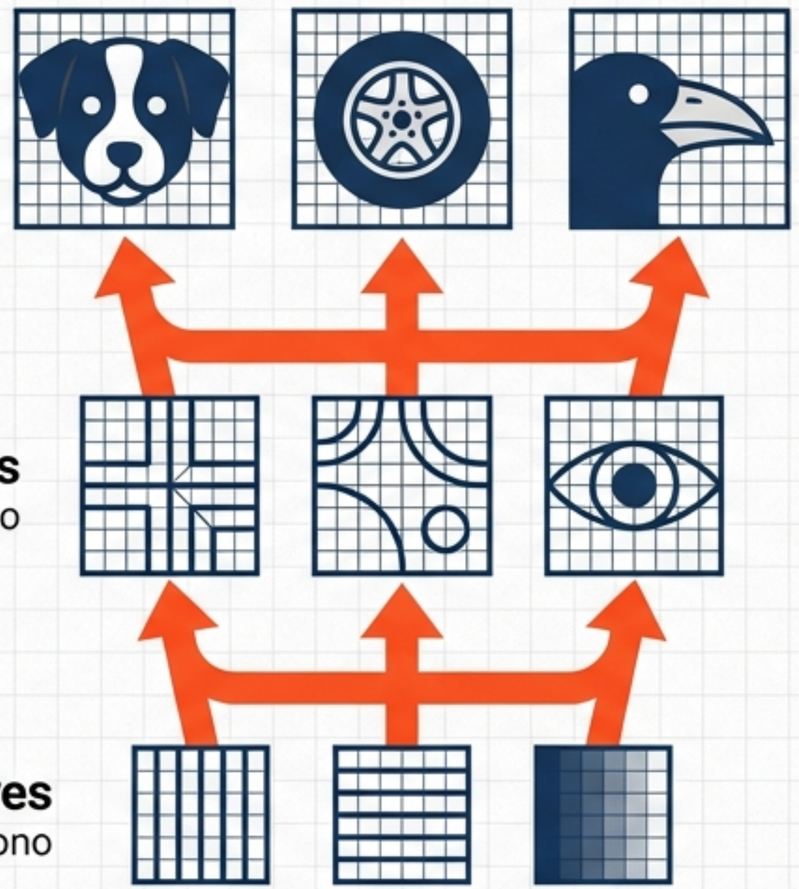
1. **REDUCE COMPUTATION:** Smaller spatial dimensions = fewer calculations later.
2. **INVARIANCE:** We care *that* a feature exists, not precisely *where* it is.
3. **DENOISING:** Only the strongest signals survive.

HIERARCHY OF ABSTRACTION

High-Level Features
Roboto Mono

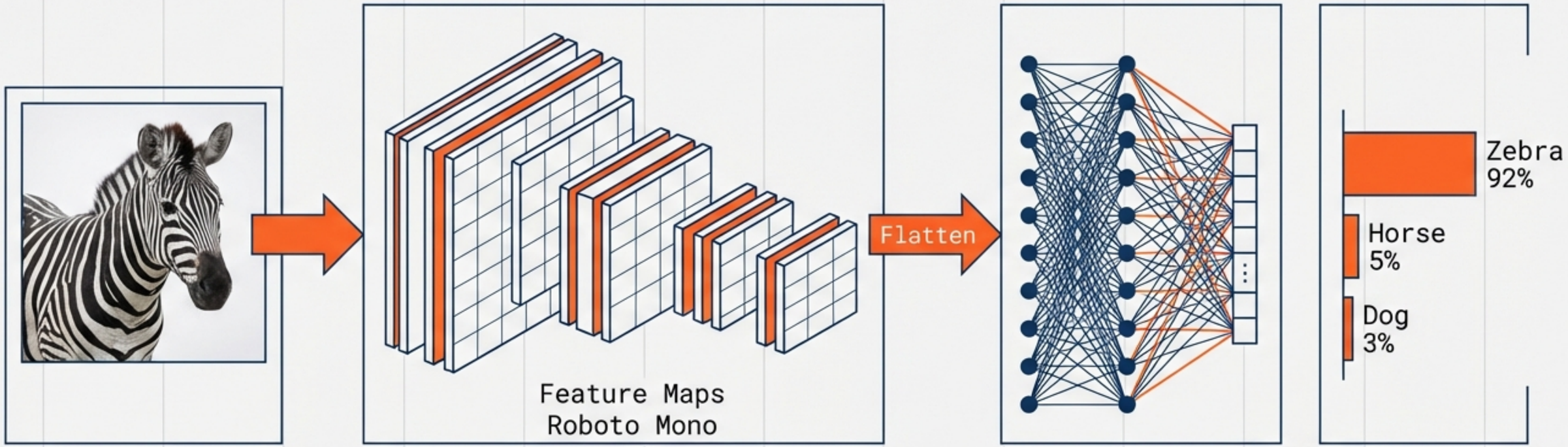
Mid-Level Features
Roboto Mono

Low-Level Features
Roboto Mono



“Deep Learning is the process of building complex understanding from simple beginnings.”
Graphite Roboto Mono

THE COMPLETE IPELINE



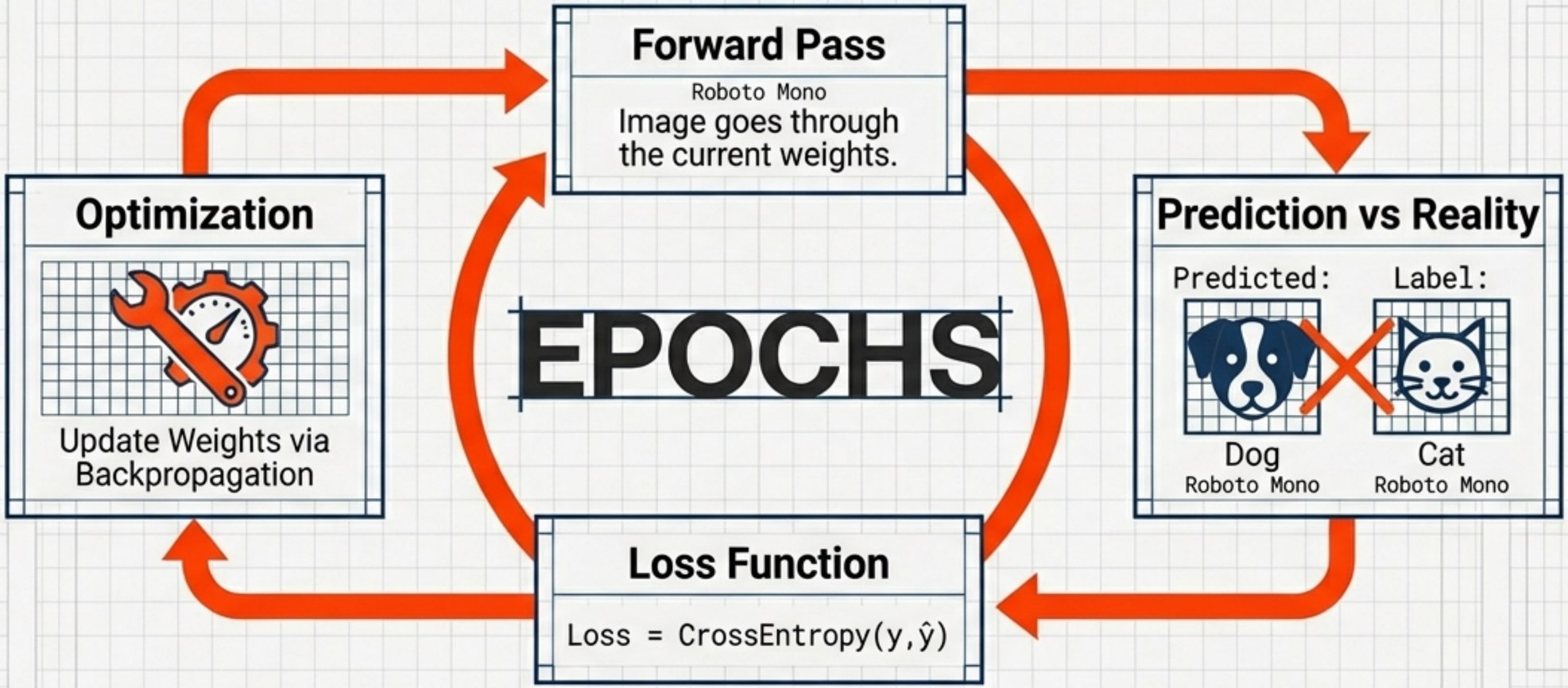
INPUT IMAGE
Roboto Mono

FEATURE EXTRACTION
(Conv + ReLU + Pool)

CLASSIFICATION
(Fully Connected)
Roboto Mono

OUTPUT
Roboto Mono

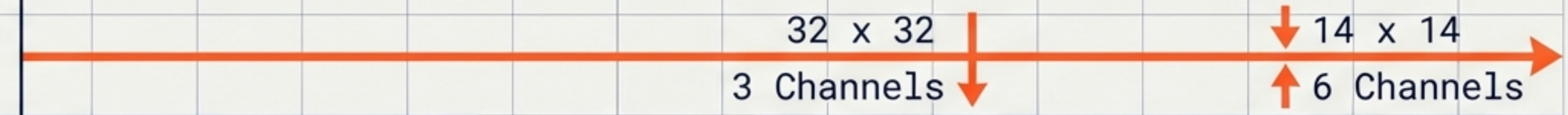
HOW IT LEARNS: TRAINING LOOP.



The kernels are not hard-coded. They start as random noise and are sculpted over thousands of iterations to minimize the error (Loss).

CALCULATING DIMENSIONS: AN EXAMPLE.

Input Image:
32 x 32 x 3



Layer 1: Convolution

(6 Kernels, Size 5x5, Stride 1)

Calculation:
(32 - 5 + 2(0)) / 1 + 1
= 28

Resulting Shape:
28 x 28 x 6

Layer 2: Max Pooling

(Size 2x2, Stride 2)

Calculation:
(28 - 2) / 2 + 1
= 14

Resulting Shape:
14 x 14 x 6

FROM THEORY TO CODE (PyTorch)

```
class CNN_Model(nn.Module):
    def __init__(self):
        super().__init__()
        # Input: 3 channels (RGB), Output: 6 feature maps
        self.conv1 = nn.Conv2d(3, 6, 5)
        self.pool = nn.MaxPool2d(2, 2)
        self.conv2 = nn.Conv2d(6, 16, 5)

    def forward(self, x):
        # Convolution -> ReLU -> Pooling
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
        return x
```

Defines Kernels

Downsampling

CRITICAL ANALYSIS.

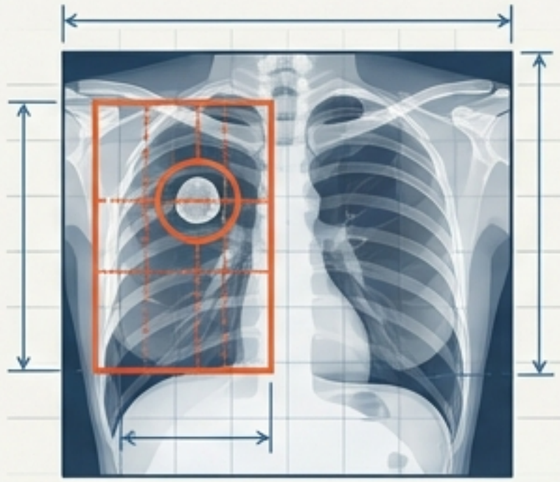
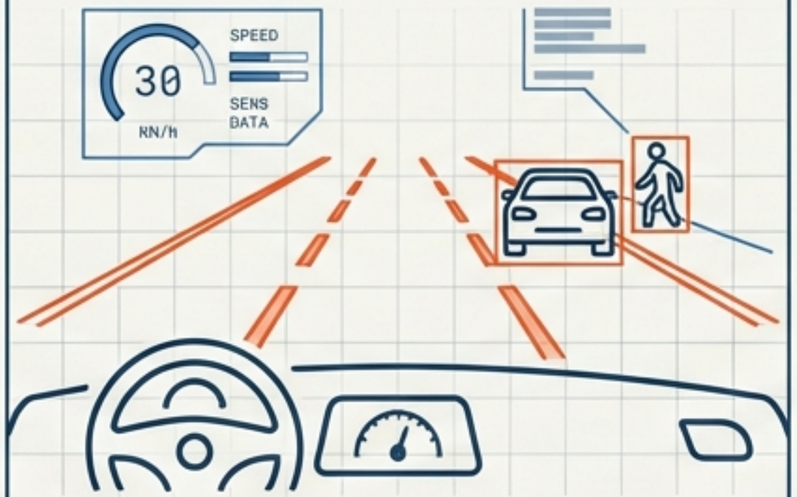
PROS

- **Spatial Invariance:** Recognizes objects regardless of position in the frame.
- **Parameter Efficiency:** Shared weights reduce model size compared to dense networks.
- **State of the Art:** The gold standard for image and video tasks.

CONS

- **Compute Cost:** Training requires significant GPU acceleration.
- **Data Hungry:** Needs massive labeled datasets (e.g., ImageNet) to generalize.
- **Sequence Struggle:** Less effective than Transformers for pure text/NLP tasks.

REAL WORLD IMPACT.

Medical Imaging	Autonomous Systems	Security
		
Tumor Detection & Diagnostics	Self-Driving Perception	Biometric Authentication

While architecture evolves, the Convolutional Neural Network remains the bedrock of Computer Vision.