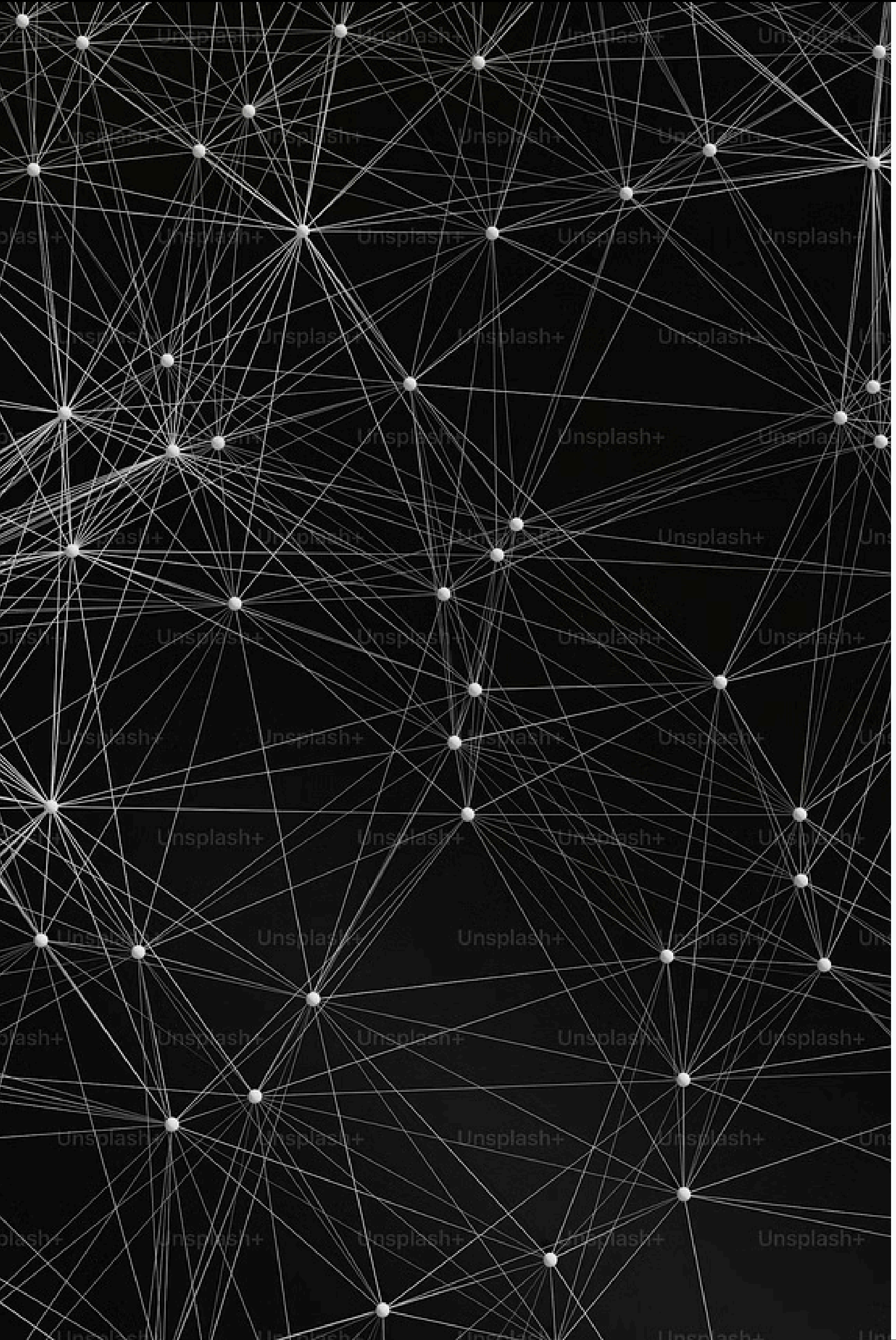
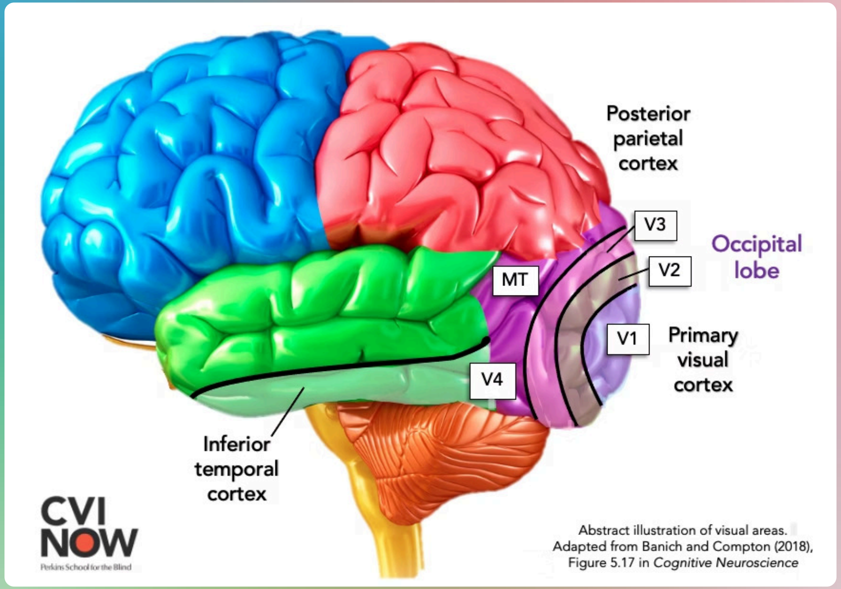


# Convolutional Neural Networks: Learning to See

## Inspired by the Human Visual Cortex

Convolutional Neural Networks represent a paradigm shift in computer vision, mirroring the hierarchical processing mechanisms of the human visual system. These architectures transform raw pixel data into meaningful semantic understanding through learned feature hierarchies.

**References:**
Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*.
LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.
Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *NIPS*.


Posterior
parietal
cortex
V3
Occipital
lobe
MT
V2
V1
Primary
visual
cortex
V4
Inferior
temporal
cortex
CVI NOW
Perkins School for the Blind
Abstract illustration of visual areas.
Adapted from Banich and Compton (2018),
Figure 5.17 in Cognitive Neuroscience

# Biological Inspiration: Visual Cortex to CNN Architecture

1

## Human Visual Pathway

**V1:** Oriented edges and contrast detection

**V2:** Texture and contour integration

**V4:** Shape and form recognition

**IT:** Object identification and categorisation

The visual cortex processes information through increasingly abstract representations, from simple edge detectors to complex object recognition.

2

## CNN Computational Hierarchy

**Conv1:** Edge and gradient filters

**Conv2:** Texture and pattern composition

**Conv3-5:** Part-based representations

**FC Layers:** Semantic classification

Convolutional layers mirror biological feature extraction, building progressively complex representations through learned transformations.

## The Convolution Operation: Local Pattern Detection

The convolution operation applies a learnable kernel across spatial dimensions, computing element-wise products and summations. This sliding window approach preserves spatial relationships whilst detecting local patterns.

### Mathematical Formulation

$$(I * K)(x, y) = \sum_{i,j} I(x+i, y+j) \cdot K(i, j)$$

**Key Properties:**

- Local connectivity reduces parameters
- Weight sharing enforces translation equivariance
- Spatial hierarchy emerges naturally
- Receptive fields expand with depth

Each kernel learns to detect specific visual patterns, from simple edges in early layers to complex textures in deeper layers.

### References

- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE.
- Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. Journal of Physiology.

# Padding and Stride: Spatial Dimension Control

1

### Padding Strategies

**Valid (no padding):** Feature maps shrink with each convolution, potentially losing border information.

**Same (zero padding):** Maintains spatial dimensions, preserving boundary features and enabling deeper architectures.

Padding prevents information loss at image boundaries, crucial for detecting features near edges.

2

### Stride Configuration

**Stride = 1:** Dense sampling preserves fine spatial detail but increases computational cost.

**Stride > 1:** Downsamples feature maps, reducing resolution whilst increasing receptive field coverage.

Stride controls the spatial overlap of convolutional operations, balancing resolution against efficiency.

## Output Dimension Formula

$$O = \frac{N - F + 2P}{S} + 1$$

Where $N$ = input size, $F$ = filter size, $P$ = padding, $S$ = stride. This equation governs spatial transformation through the network, ensuring dimensional compatibility across layers.

# Pooling Layers: Downsampling and Invariance

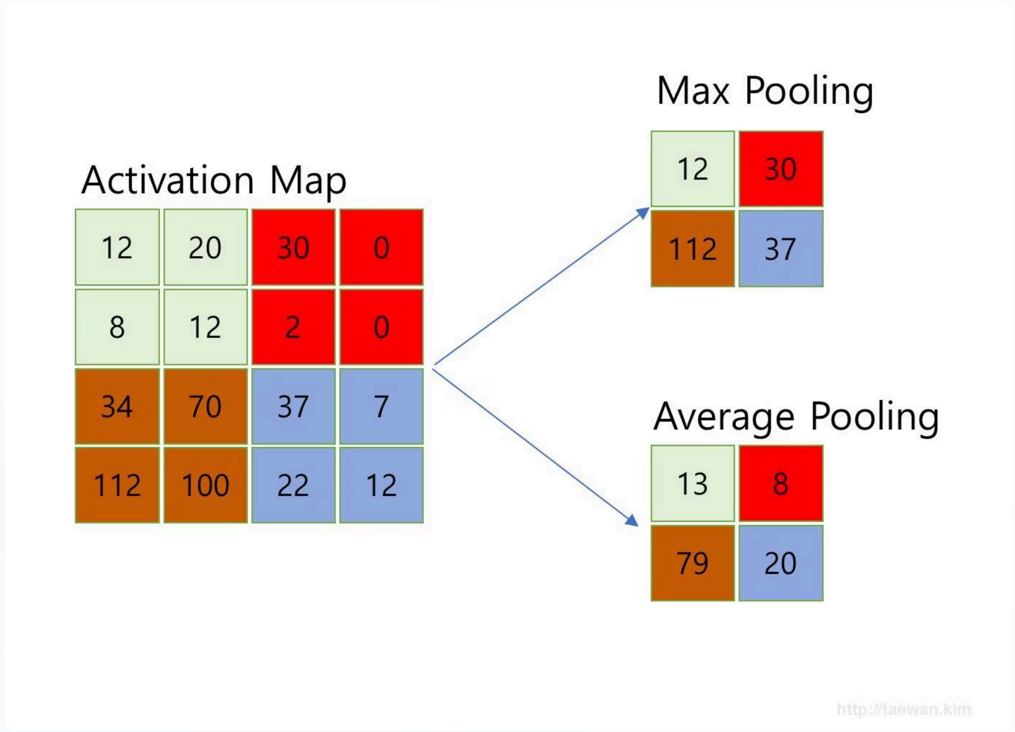
## Pooling Operations

**Max Pooling:** Selects maximum activation within each window, providing translation invariance and highlighting dominant features.

$$y_{ij} = \max_{(p,q)\in \text{window}} x_{i+p,j+q}$$

**Average Pooling:** Computes mean activation, smoothing feature representations and reducing variance.

**Global Average Pooling:** Collapses entire feature map to single value, commonly used before classification layers.



### Dimensionality Reduction

Pooling reduces spatial dimensions by 50-75%, decreasing computational burden whilst maintaining semantic content.

### Translation Invariance

Small spatial shifts in input produce identical pooled outputs, conferring robustness to minor transformations.

### Receptive Field Expansion

Pooling increases the effective receptive field of subsequent layers, enabling detection of larger patterns.

# End-to-End CNN Architecture: From Pixels to Predictions

**Input Image**

Raw pixel values (H×W×C)

**Convolution**

Feature extraction via learned kernels

**ReLU Activation**

Non-linear transformation

**Pooling**

Spatial downsampling

**Flatten**

Convert to 1D vector

**Fully Connected**

Dense classification layers

**Softmax**

Probability distribution

## ReLU Activation Function

$$f(x) = \max(0, x)$$

Introduces non-linearity whilst maintaining gradient flow, enabling learning of complex decision boundaries. Computationally efficient and biologically plausible.

## Softmax Classification

$$P(y = i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Transforms logits into normalised probability distribution over classes, enabling multi-class classification with probabilistic interpretation.

# Backpropagation: The Learning Mechanism

### Forward Pass

Input propagates through network layers, applying learned transformations to generate predictions. Each layer computes activations based on previous layer outputs and current parameters.

### Loss Computation

Prediction error quantified via loss function, measuring discrepancy between network output and ground truth labels. Common choice: categorical cross-entropy.

### Backward Pass

Gradients propagate from loss to parameters via chain rule, computing partial derivatives with respect to each weight. Enables credit assignment across depth.

### Parameter Update

Weights adjusted in direction of negative gradient, minimising loss through iterative optimisation. Learning rate controls step size.

## Cross-Entropy Loss

$$L = -\sum_i y_i \log(\hat{y}_i)$$

Measures divergence between true distribution $y$ and predicted distribution $\hat{y}$. Penalises confident incorrect predictions more heavily than uncertain ones.

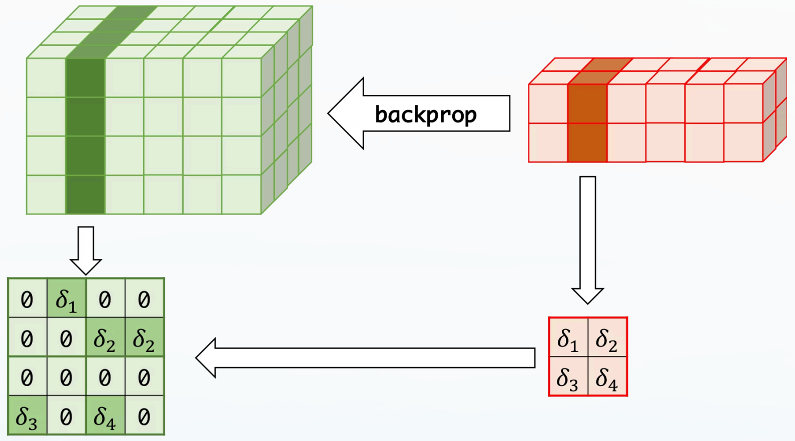
## Gradient Descent Update Rule

$$w_{\text{new}} = w_{\text{old}} - \eta \frac{\partial L}{\partial w}$$

Parameters move opposite to gradient direction, with learning rate $\eta$ controlling update magnitude. Enables convergence to local minima through iterative refinement.

# Gradient Flow in Pooling and Flatten Layers

## Max Pooling Backpropagation



$$\frac{\partial y}{\partial x_i} = \begin{cases} 1, & x_i = \max(x_{\text{window}}) \\ 0, & \text{otherwise} \end{cases}$$

Gradients route exclusively through maximum activation positions. Non-maximum elements receive zero gradient, as they did not influence forward pass output. This sparse gradient flow maintains learning efficiency whilst implementing the max operation's derivative.

## Flatten Layer Gradient Reshaping

The flatten operation performs tensor reshaping without learnable parameters. During backpropagation, gradients from 1D fully connected layers must be reshaped back to 3D spatial dimensions.

This reshaping preserves gradient magnitudes whilst restoring spatial structure, enabling proper gradient flow to convolutional layers. The operation is mathematically equivalent to a transpose of the forward flatten operation.

**Critical Insight:** Pooling and flatten layers contain no learnable parameters, yet play crucial roles in gradient routing. Max pooling implements selective attention through sparse gradients, whilst flatten bridges spatial and fully connected representations.

# Evolution of CNN Architectures: From LeNet to Transformers

1

**LeNet-5 (1998)**

Pioneering architecture for digit recognition. Introduced convolutional layers and hierarchical feature learning. Demonstrated viability of gradient-based learning for visual tasks.

2

**AlexNet (2012)**

ImageNet breakthrough with ReLU activations and GPU training. Introduced dropout regularisation and overlapping pooling. Catalysed deep learning revolution in computer vision.

3

**VGG (2014)**

Emphasised depth through small 3×3 filters. Demonstrated that network depth is critical for performance. Established modular design principles for CNNs.

4

**ResNet (2015)**

Skip connections enable training of 100+ layer networks. Addresses vanishing gradient problem through residual learning. Formula: *y = F(x) + x*

5

**Vision Transformers (2020+)**

Self-attention mechanisms replace convolution. Treats images as sequences of patches. Achieves state-of-the-art with sufficient data and compute.

## Residual Learning: The Skip Connection Innovation

$$y = F(x) + x$$

Residual connections add input directly to output, enabling gradients to flow unimpeded through network depth. This identity mapping allows learning of residual functions rather than complete transformations, dramatically improving trainability of deep networks. The innovation enabled networks exceeding 1000 layers whilst maintaining stable gradient flow.

# Key Takeaways: Learning to See

### Biological Inspiration

CNNs mirror the hierarchical processing of the visual cortex, learning progressively abstract feature representations from edges to objects through layered transformations.

### Architectural Components

Convolution extracts local patterns through weight sharing, whilst pooling provides spatial invariance and dimensionality reduction. Together they build robust visual representations.

### Learning Through Backpropagation

Gradient descent refines filter weights iteratively, transforming random patterns into learned feature detectors. Loss functions guide optimisation toward task-specific representations.

## From Pixels to Understanding

The journey from raw pixel intensities to semantic comprehension exemplifies the power of hierarchical representation learning. Each layer abstracts away irrelevant details whilst preserving task-relevant information.

## Future Directions

Whilst transformers challenge CNN dominance, the core principles—local connectivity, hierarchical abstraction, and learned representations—remain foundational to visual intelligence, biological and artificial alike.