Lecture Narrative: Deep Convolutional Neural Networks

Recording Notes

Each subsection corresponds to one slide in the deck. Slide numbers match the order from the title slide through references.

Slide 1: Deep Convolutional Neural Networks

This lecture introduces Deep Convolutional Neural Networks, or CNNs. These models are the backbone of modern computer vision and have transformed tasks such as image classification, detection, segmentation, and generation.

Slide 2: Credits

These slides are adapted from material by Yong Jae Lee, Rob Fergus, Svetlana Lazebnik, and Jia-Bin Huang. Many of the examples and historical references come from their influential teaching and research.

Slide 3: Overview

We will begin with background on traditional visual recognition pipelines, move into CNN architectures, discuss how CNNs can be understood and visualized, and end with applications and tools.

Slide 4: Traditional Recognition Approach

Historically, recognition systems were built as pipelines. Raw image or video pixels were converted into hand-designed features, which were then fed into a trainable classifier.

Slide 5: Traditional Pipeline Diagram

In this setup, the classifier is trainable, but the feature extractor is fixed. This places a heavy burden on feature engineering rather than learning.

Slide 6: Limitations of Hand-Designed Features

The performance of traditional systems depends strongly on the quality of manually designed features. Improving results often meant inventing better descriptors rather than improving learning algorithms.

Slide 7: Features Drive Progress

Much of the progress in object recognition came from feature innovations rather than classifiers. This raised the question: should we keep designing features by hand?

Slide 8: Examples of Hand-Designed Features

Classic examples include SIFT, HOG, spatial pyramid matching, deformable part models, and color descriptors. Each required deep domain insight and careful tuning.

Slide 9: What About Learning Features?

Instead of designing features manually, deep learning proposes learning a hierarchy of features directly from data, all the way from pixels to classifier.

Slide 10: Feature Hierarchy

Each layer extracts features from the previous layer. Lower layers capture edges and textures, while higher layers represent more abstract concepts.

Slide 11: Shallow vs Deep Architectures

Traditional pipelines are shallow, with only one or two learned stages. Deep architectures stack many learned transformations before classification.

Slide 12: Architecture Comparison

This slide contrasts shallow pipelines with deep neural networks, highlighting how representation learning is integrated throughout the system.

Slide 13: Biological Inspiration

The idea of hierarchical feature extraction is inspired by the human visual system, where neurons respond to increasingly complex patterns.

Slide 14: Biological Neuron

A biological neuron integrates inputs and fires based on activation, motivating computational models of neurons.

Slide 15: Artificial Neuron (Perceptron)

The perceptron computes a weighted sum of inputs followed by a non-linearity. Alone, it is a linear classifier.

Slide 16: Simple and Complex Cells

Hubel and Wiesel proposed simple cells responding to edges and complex cells providing invariance, forming a hierarchy in the visual cortex.

Slide 17: Hubel–Wiesel Architecture

Their work suggested that higher-level features build upon lower-level ones, a concept directly mirrored in CNNs.

Slide 18: Multi-layer Neural Networks

Stacking perceptrons creates multi-layer networks capable of learning non-linear decision boundaries.

Slide 19: Training Neural Networks

Training involves minimizing classification error using gradient descent. This requires differentiable activation functions.

Slide 20: Backpropagation

Backpropagation efficiently computes gradients through the network, enabling end-to-end learning.

Slide 21: Neocognitron

The Neocognitron introduced ideas like local receptive fields and invariance, foreshadowing CNNs.

Slide 22: CNN Definition

CNNs are neural networks with local connectivity, shared weights, and stacked feature extractors, followed by classification layers.

Slide 23: CNN Characteristics

Key properties include convolution, weight sharing, pooling, and hierarchical representations.

Slide 24: LeNet

LeNet is an early CNN architecture successfully applied to handwritten digit recognition.

Slide 25: CNN Forward Pass

An input image is convolved with learned filters, passed through non-linearities, pooled, normalized, and repeated across layers.

Slide 26: Feature Maps

Each convolution produces feature maps that represent learned responses to visual patterns.

Slide 27: Convolution

Convolution enforces locality and translation invariance while reducing parameter count.

Slide 28: Non-Linearity

Non-linear activations such as ReLU allow networks to model complex functions and train efficiently.

Slide 29: Spatial Pooling

Pooling aggregates information spatially, increasing invariance and receptive field size.

Slide 30: Pooling Types

Max and sum pooling are common, each with different trade-offs.

Slide 31: Normalization

Normalization stabilizes training and improves generalization, similar to classical feature pipelines.

Slide 32: Comparison to SIFT

CNNs replicate the filtering, pooling, and normalization steps of SIFT, but learn them from data.

Slide 33: Spatial Pyramid Matching

CNNs can be viewed as learned, hierarchical versions of spatial pyramid representations.

Slide 34: Early CNN Successes

CNNs achieved strong results on MNIST, CIFAR-10, and traffic sign recognition.

Slide 35: Limits of Early CNNs

They initially struggled with complex datasets due to limited data and computation.

Slide 36: ImageNet Dataset

ImageNet introduced millions of labeled images, enabling large-scale training.

Slide 37: ImageNet Challenge

The challenge standardized evaluation and accelerated progress in vision.

Slide 38: AlexNet

AlexNet combined deeper models, GPUs, and large datasets to achieve breakthrough performance.

Slide 39: Why AlexNet Worked

Key factors included scale, GPU acceleration, ReLU activations, and dropout.

Slide 40: Impact of AlexNet

AlexNet dramatically reduced error rates and shifted the field toward deep learning.

Slide 41: Understanding CNNs

As CNNs grew complex, understanding what they learn became critical.

Slide 42: Visualization via Deconvnets

Deconvolution maps activations back to image space to reveal learned patterns.

Slide 43: Layer-wise Visualizations

Lower layers capture edges, while deeper layers represent object parts and semantics.

Slide 44: High-Level Feature Visualization

Deep layers respond to object-level concepts rather than simple textures.

Slide 45: Occlusion Experiments

Occluding parts of images reveals which regions influence predictions most.

Slide 46: Individual Neuron Activations

Some neurons specialize in detecting meaningful visual concepts.

Slide 47: R-CNN Neuron Analysis

Neurons trained for classification can implicitly learn detectors.

Slide 48: Fooling CNNs

CNNs can be confidently wrong when presented with adversarial examples.

Slide 49: Adversarial Examples

Small, targeted perturbations can drastically change predictions.

Slide 50: Why Adversarial Examples Exist

They exploit linearities and high-dimensional geometry in neural networks.

Slide 51: Beyond Classification

CNNs are used for detection, segmentation, regression, and synthesis.

Slide 52: R-CNN

Region-based CNNs extend classification to object detection.

Slide 53: Semantic Segmentation

Fully convolutional networks predict labels for every pixel.

Slide 54: Edge Detection

CNNs can detect contours and boundaries using multi-scale features.

Slide 55: CNNs for Regression

CNNs can predict continuous outputs such as pose or keypoints.

Slide 56: CNNs for Matching

CNNs act as learned similarity measures for faces and image patches.

Slide 57: Image Restoration

CNNs are used for super-resolution, deblurring, and optical flow.

Slide 58: Image Generation

CNNs can generate new images, such as synthetic chairs.

Slide 59: Transfer Learning

Pretrained CNNs can be adapted to new tasks efficiently.

Slide 60: CNN Packages

Popular frameworks include Caffe, Torch, MatConvNet, and others.

Slide 61: Resources and References

These references and online resources provide further reading and practical guidance for deep learning in computer vision.