أكاديمية كاوست
KAUST ACADEMY

Day 1: Computer Vision Summary

Note: Do Not depend entirely on this and study from the official slides.

Contributed by: Hassan Mohammed Nasr

- Computer Vision (CV) is the field of building artificial systems that process, perceive, and reason about visual data

- CV Applications:
  - Image Classification → Assign single label to the entire image (e.g. main theme is a cat picture)
  - Image Retrieval → Find and return images from database that is similar to the input image
  - Object Detection → Locating specific objects within an image (draw bounding box around them)
  - Image Segmentation → A pixel level classification where every pixel is assigned to classes creating precise outlines of objects
  - Video Classification → Input video and apply classification
  - Activity Recognition → Analyze video input to identify specific human actions (e.g eating)
  - Pose Detection → Estimating the position and orientation of human joints and posture
  - Image Captioning → Describing the image (generate text based on the image)
  - Image Generation → Create entirely new image
  - Style Transfer → Apply an artistic style on the content of another image
  - 3D vision → Reconstruct 3d shapes and depth from 2d images

Images and Neural Networks

- images are represented as Matrices with elements having values between [0, 255] representing pixel value (color intensity)
  - Grayscale images: represented by one matrix (called channel)
  - Colored Image: usually represented using RGB colors, represented as 3 channels one for each color (three matrices) each matrix will contain values from 0 to 255

Neural networks accepts input as one tall vector; solution: flatten the image (matrix) into a vector

- Problem 1: Extremely large number of parameters; small image 300x300 will have 900000 as just input (without the rest of the layers). If we added only one hidden layer with 10 neurons we will have 9000000 parameter
- Problem 2: little or no invariance to shifting or any transformation ; flattening an image ill loose the spatial information, vector of image of 'A' will be very different from that of ∀

Solution: CNNs → Use Filters/Kernels (small matrices) instead of flattening the image

- Solves Problem 1 by allowing Parameter sharing; each group of pixels will share same parameters
- Solves Problem 2 by looking on groups of pixels together (neighbors) and recognize small features like edges and shapes

Convolutional Neural Networks (CNNs)

- CNNs Type of Neural Networks designed for image processing that works by extracting features like edges, combining them to form shapes then high level objects. CNNs consist of two main parts:

1 - Feature Extractor

Learns patterns (features) from the images

- Convolution Layer (main feature extractor)
  - Aim: Detect local patterns (e.g edges)
  - Operation: Slides a kernel (filter) over the input to calculate dot products (sum of products)
- Activation Functions
  - Aim: Adds non-linearity
- Controlling Output Size & Pooling Layers
  - Stride (s) : Step size, larger stride = smaller output (Downsampling)
  - Dilated Convolution: Kernel is spread out, expands the receptive field
  - Padding (p) : Adding Zeros around the boarder to maintain size (gives the kernel the ability to slide over the boundaries of the matrix)
  - Pooling Layer: Downsample to reduce resolution and computation while keeping important features
    - Has no Learnable Parameters

$$n_{out} = \lfloor \frac{n_{in} + 2p - k}{s} \rfloor + 1$$

2 - Classifier

The extracted features are flattened into a vector and then passed to standard NN

CNNs Architectures

Architecture	Key Innovation	Why it matters
• AlexNet	First pioneer, first cnn to achieve good performance on ImageNet	Proved that CNNs work for large-scale vision
• VGGNet	Small filters; used 3x3 filters instead of 7x7	Uses a deeper network with small filters (19 layers)
• InceptionNet	Inception module: used parallel filters (1x1,3x3,5x5)	Went Deep (22 layers) with less parameters
• ResNet	Skip connections (residuals)	Going very deep introduced vanishing gradient, residuals allowed information to flow more easily allowing ultra deep networks (152+ layers)

CNNs Architectures cont.

- EfficientNet (compound scaling)
  - Problem: Adjusting Depth, Width, and Resolution separately is hard
  - Solution : Scale all three dimensions uniformly using single coefficient
  - Compound Scaling: a formula that find the optimal balance among depth(# of layers), width(# of neurons), and resolution (image size) based on the desired model complexity

    Achieved state of the art accuracy with significantly fewer parameters

- MobileNet (Depthwise Separable Conv)
  - Problem: Small sized models are crucial for mobile and embedded devices
  - Solution : mobilenet reduce computational cost and memory usage while maintaining good accuracy
  - Concept: Splits standard convolution into two cheaper steps to save computation

    1 - Depthwise: 1 filter per channel

    2 - Pointwise: 1x1 convolution (combines channels / mixes information)

Standard CNN cost Vs. MobileNet cost

- Standard CNN

  You have a kernel that operates on ALL input channels at ONCE to create one output channel

  Cost = Filter Params x Filter Positions x Number of filters

  Example: Input RGB image and output should be 4x4x5 (kernel size 3x3)

  Cost = (3x3x3) x (4x4) x (5) = 2160

- MobileNet

  Splits operation into two cheaper steps, calculate depth wise (kernel for each channel independent) then point wise operation to mix information

  Cost = Depthwise + Pointwise

  Depthwise = Parameters x positions
  Pointwise = (1x1x input channels) x positions x number of filters(output channels)

  Example: Input RGB image and output should be 4x4x5 (kernel size 3x3)

  Depthwise = (3x3x3) x (4x4) = 432
  Pointwise = (1x1x 3) x (4x4) x 5 =240

  Cost = 432+240 =672