



## VIDEO ANALYTICS MODULE # 3 : VIDEO ENHANCEMENT AND RESTORATION



**BITS Pilani**  
Pilani | Dubai | Goa | Hyderabad

Seetha Parameswaran  
BITS Pilani

---

The instructor is gratefully acknowledging  
the authors who made their course  
materials freely available online.

This deck is prepared by Seetha Parameswaran.

# TABLE OF CONTENTS

---

- ① MODULE 3 TOPICS
- ② VIDEO ENHANCEMENT AND RESTORATION
- ③ FILTERING PRELIMS
- ④ SPATIOTEMPORAL NOISE FILTERING

# MODULE TOPICS....

---

- Spatio temporal noise filtering
- Coding Artifact reduction
- Blotch reduction and removal
- Vinegar Syndrome removal
- Kinescope moiré removal
- Flicker correction
- Scratch removal
- Application

# TABLE OF CONTENTS

---

- ① MODULE 3 TOPICS
- ② VIDEO ENHANCEMENT AND RESTORATION
- ③ FILTERING PRELIMS
- ④ SPATIOTEMPORAL NOISE FILTERING

# VIDEO ENHANCEMENT – ITS NEED

---

- Video or recorded image sequences suffer from severe degradations.
- Due to
  - ▶ imperfect or uncontrollable recording conditions, such as one encounters in astronomy, forensic sciences, and medical imaging.
  - ▶ visible coding artifacts, such as blocking, ringing, and mosquito noise.
- Improve the visual quality
- Increase the performance of subsequent tasks such as analysis and interpretation.

# VIDEO RESTORATION – ITS NEED

---

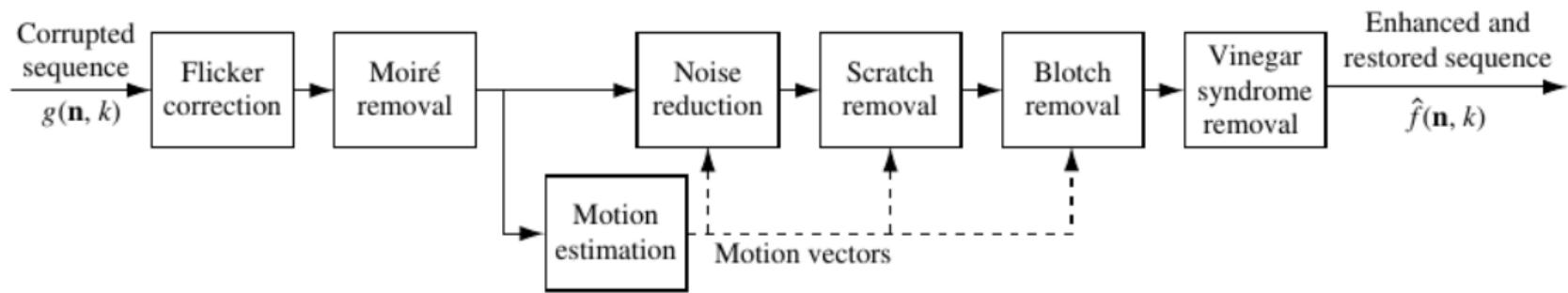
- Restoration is preserving motion pictures and video tapes recorded over the last century.
- The records are deteriorating rapidly due to aging effects of the physical reels of film and magnetic tapes that carry the information.
- Restoration process
  - ① Video is transferred from the original film reels or magnetic tape to digital media.
  - ② Then, all kinds of degradations are removed from the digitized image sequences.
- Objective of restoration
  - ▶ remove irrelevant information such as noise and blotches
  - ▶ restores the original spatial and temporal correlation structure of digital image sequences.

# CHALLENGE

---

- Large amount of data in a video
- Enhancement and Restoration methods for image sequences should have a manageable complexity and should be semiautomatic.
  - ▶ Semiautomatic indicates that professional operators control the visual quality of the restored image sequences by selecting values for some of the critical restoration parameters.

# STEPS IN VIDEO ENHANCEMENT AND RESTORATION



Video enhancement and restoration techniques are sometimes referred to as **spatiotemporal filters or 3D filters**.

# STEPS IN VIDEO ENHANCEMENT AND RESTORATION

**NOISE REMOVAL** spatial and temporal noise to be removed

**CODING ARTIFACT REDUCTION OR BLOCKINESS REDUCTION** due to the lossy Discrete Cosine Transform

**BLOTTCHES** are dark and bright spots that are often visible in damaged film image sequences. The removal of blotches is a temporal detection and interpolation problem.

**VINEGAR SYNDROME** represents a special type of impairment related to film (e.g., partial loss of color, blur).

# STEPS IN VIDEO ENHANCEMENT AND RESTORATION

---

**INTENSITY FLICKER** refers to variations in intensity in time, caused by aging of film, by copying and format conversion (e.g., from film to video), or by variations in shutter time.

**KINESCOPE MOIRÉ** phenomenon appears during film-to-video transfer using telecine devices.

**FILM SCRATCHES** are either bright or dark vertical lines spanning the entire frame. They appear approximately at the same place in consecutive frames.

# TABLE OF CONTENTS

---

- 1 MODULE 3 TOPICS
- 2 VIDEO ENHANCEMENT AND RESTORATION
- 3 FILTERING PRELIMS
- 4 SPATIOTEMPORAL NOISE FILTERING

# FILTERS

---

- Filtering refers to accepting (passing) or rejecting certain components.
- Filters are also called spatial masks, kernels, templates, and windows.
- A **spatial filter** consists of
  - ① a neighborhood, (typically a small rectangle)
  - ② a predefined operation that is performed on the image pixels encompassed by the neighborhood.
- Filtering creates a new pixel with coordinates equal to the coordinates of the center of the neighborhood, and whose value is the result of the filtering operation.
- A processed (filtered) image is generated as the center of the filter visits each pixel in the input image. If the operation performed on the image pixels is linear, then the filter is called a **linear spatial filter**.

# LINEAR SPATIAL FILTER

Pixels of Image section under filter

$f(x - 1, y - 1)$	$f(x - 1, y)$	$f(x - 1, y + 1)$
$f(x, y - 1)$	$f(x, y)$	$f(x, y + 1)$
$f(x + 1, y - 1)$	$f(x + 1, y)$	$f(x + 1, y + 1)$

Filter coefficients

$w(-1, -1)$	$w(-1, 0)$	$w(-1, 1)$
$w(0, -1)$	$w(0, 0)$	$w(0, 1)$
$w(1, -1)$	$w(1, 0)$	$w(1, 1)$

$$\begin{aligned}
 g(x, y) = & w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + w(-1, 1)f(x - 1, y + 1) \\
 & + w(0, -1)f(x, y - 1) + w(0, 0)f(x, y) + w(0, 1)f(x, y + 1) \\
 & + w(1, -1)f(x + 1, y - 1) + w(1, 0)f(x + 1, y) + w(1, 1)f(x + 1, y + 1)
 \end{aligned}$$

Coefficient of the filter,  $w(0, 0)$  aligns with the pixel at location  $(x, y)$ .

# CORRELATION

---

- Correlation of a filter  $w(x, y)$  of size  $m \times n$  with an image  $f(x, y)$
- Denoted as  $w(x, y) \otimes f(x, y)$

$$w(x, y) \otimes f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

$$a = (m - 1)/2, b = (n - 1)/2$$

- This equation is evaluated for all values of the displacement variables  $x$  and  $y$  so that all elements of  $w$  visit every pixel in  $f$ , where we assume that  $f$  has been padded appropriately.

# CORRELATION

		Padded $f$		
		0 0 0 0 0 0 0 0 0 0		
		0 0 0 0 0 0 0 0 0 0		
		0 0 0 0 0 0 0 0 0 0		
Origin $f(x, y)$		0 0 0 0 0 0 0 0 0 0		
0 0 0 0 0		0 0 0 0 0 1 0 0 0 0		
0 0 0 0 0		0 0 0 0 0 0 0 0 0 0		
$w(x, y)$		0 0 0 0 0 0 0 0 0 0		
0 0 1 0 0		0 0 0 0 0 0 0 0 0 0		
1 2 3		0 0 0 0 0 0 0 0 0 0		
0 0 0 0 0		0 0 0 0 0 0 0 0 0 0		
4 5 6		0 0 0 0 0 0 0 0 0 0		
0 0 0 0 0		0 0 0 0 0 0 0 0 0 0		
7 8 9		0 0 0 0 0 0 0 0 0 0		
(a)		(b)		
Initial position for $w$		Full correlation result	Cropped correlation result	
1 2 3 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0	
4 5 6 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0	0 9 8 7 0	
7 8 9 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0	0 6 5 4 0	
0 0 0 0 0 0 0 0 0 0		0 0 0 9 8 7 0 0 0 0	0 3 2 1 0	
0 0 0 0 1 0 0 0 0 0		0 0 0 6 5 4 0 0 0 0	0 0 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0		0 0 0 3 2 1 0 0 0 0		
0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0		
0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0		
0 0 0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0 0 0		
(c)		(d)	(e)	

# CORRELATION

---

- A function that contains a single 1 with the rest being 0s a **discrete unit impulse**.
- Correlation of a function with a discrete unit impulse yields a rotated version of the function at the location of the impulse.
- Correlation can be used to find matches between images.

# CONVOLUTION

- Convolution of a filter  $w(x, y)$  of size  $m \times n$  with an image  $f(x, y)$
- Denoted as  $w(x, y) \oplus f(x, y)$

$$w(x, y) \oplus f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x - s, y - t)$$

$$a = (m - 1)/2, b = (n - 1)/2$$

- This equation is evaluated for all values of the displacement variables  $x$  and  $y$  so that all elements of  $w$  visit every pixel in  $f$ , where we assume that  $f$  has been padded appropriately.

# CONVOLUTION

Rotated $w$	Full convolution result	Cropped convolution result
9 8 7 6 5 4 3 2 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 1 2 3 0 0 0 0 0 0 0 4 5 6 0 0 0 0 0 0 0 7 8 9 0	0 0 0 0 0 0 0 1 2 3 0 0 4 5 6 0 0 7 8 9 0
(f)	(g)	(h)

# CONVOLUTION

---

- Convolution of a function with an impulse copies the function at the location of the impulse.
- If the filter mask is symmetric, correlation and convolution yield the same result.
- **Convolving a mask with an image** often is used to denote the sliding, sum-of-products process, and does not necessarily differentiate between correlation and convolution.

# TABLE OF CONTENTS

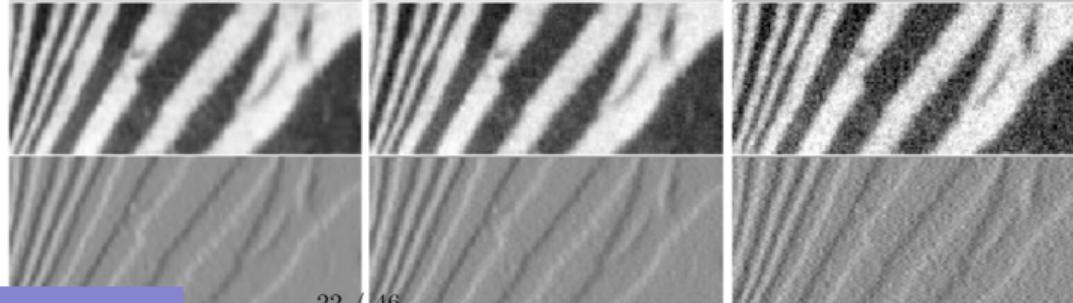
---

- 1 MODULE 3 TOPICS
- 2 VIDEO ENHANCEMENT AND RESTORATION
- 3 FILTERING PRELIMS
- 4 SPATIOTEMPORAL NOISE FILTERING

# NOISE

- Noise is anything in the image that we are not interested in.
  - ▶ Light fluctuations
  - ▶ Sensor noise or Camera noise
  - ▶ Quantization effects
  - ▶ Thermal noise
  - ▶ Granular noise on film
  - ▶ Shot noise originating in electronic hardware and storage on magnetic tape
- Effects of the noise are nonlinear of nature.

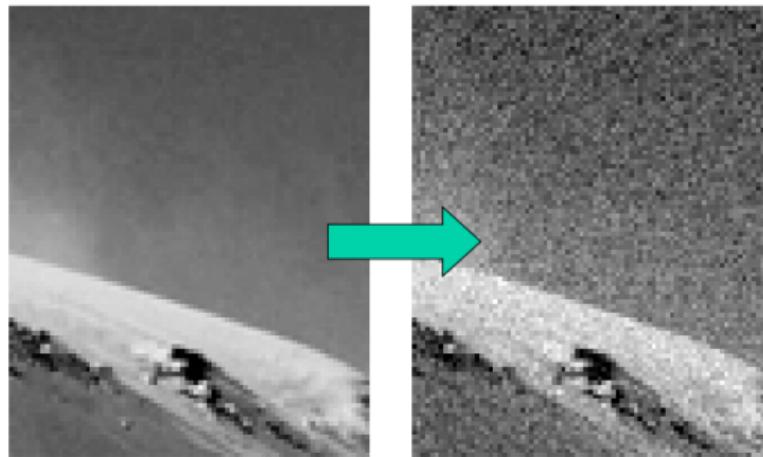
Increasing noise



# GAUSSIAN NOISE

- Also called White noise.
- Generated by the Gaussian curve.

mean 0, sigma = 16



# NOISE REMOVAL

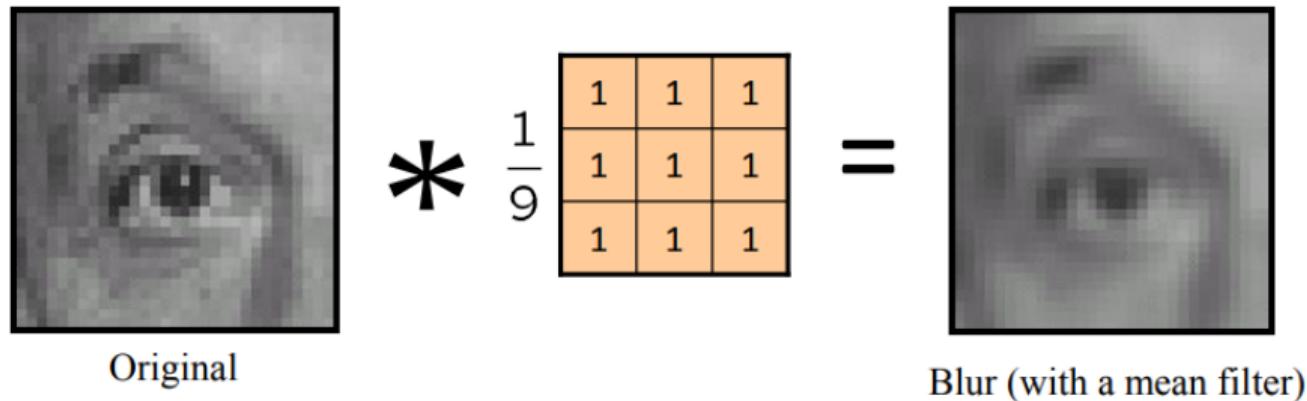
- The aggregated effect of noise is modeled as an additive white (sometimes Gaussian) process with zero mean and variance  $\sigma^2$  that is independent of the ideal uncorrupted image sequence  $f(n_1, n_2, k)$ .
- The recorded image sequence

$$g(n_1, n_2, k) = f(n_1, n_2, k) + w(n_1, n_2, k)$$

- The objective of noise reduction is to make an estimate  $\hat{f}(n, k)$  of the original image sequence given only the observed noisy image sequence  $g(n_1, n_2, k)$ .

# SMOOTHING FILTER

- Smoothing reduces noise.



- The filter is called averaging filter or mean filter or box filter.
- Result is blurred image.

# SMOOTHING + SHARPENING FILTER

- Sharpen the smoothed image to reduce the blur and noise.



$$\text{Original} \quad * \left( \begin{matrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{matrix} - \frac{1}{9} \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \right) =$$



**Sharpening filter**  
(accentuates edges)

# GAUSSIAN SMOOTHING FILTER

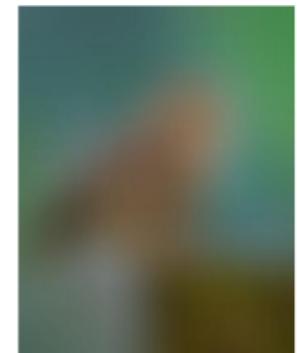
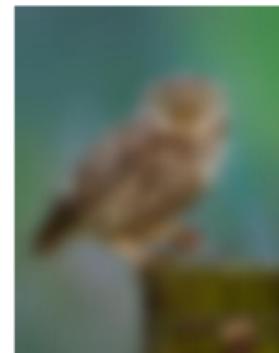
- The coefficients are a 2D Gaussian.
- Gives more weight at the central pixels and less weights to the neighbours.  
The farther away the neighbours, the smaller the weight.

$$G = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

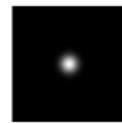
$$w = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$w = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

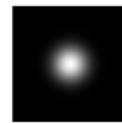
# GAUSSIAN SMOOTHING FILTER



$\sigma = 1$  pixel



$\sigma = 5$  pixels



$\sigma = 10$  pixels



$\sigma = 30$  pixels

# SEPERABLE GAUSSIAN SMOOTHING FILTER

---

$$G = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right) \exp\left(-\frac{y^2}{2\sigma^2}\right) = g(x)g(y)$$

$$w = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} [1 \quad 2 \quad 1]$$

$$w = \frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix} [1 \quad 4 \quad 6 \quad 4 \quad 1]$$

# GAUSSIAN SMOOTHING FILTER

Apply the Gaussian filter to the image:  
 Borders: keep border values as they are

15	20	25	25	15	10
20	15	50	30	20	15
20	50	55	60	30	20
20	15	65	30	15	30
15	20	30	20	25	30
20	25	15	20	10	15

Original image

1	2	1
2	4	2
1	2	1

Or:

$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

$$\frac{1}{4} \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

$$* \frac{1}{16}$$

15	20	24	23	16	10
20	25	36	33	21	15
20	44	55	51	35	20
20	29	44	35	22	30
15	21	25	24	25	30
20	21	19	16	14	15
15	20	24	23	16	10
19	28	38	35	23	15
20	35	48	43	28	21
19	31	42	36	26	28
18	23	28	25	22	21
20	21	19	16	14	15

27



































## VIDEO ANALYTICS MODULE # 4 VIDEO SEGMENTATION

Seetha Parameswaran  
BITS Pilani

**BITS** Pilani  
Pilani | Dubai | Goa | Hyderabad

---

The instructor is gratefully acknowledging  
the authors who made their course  
materials freely available online.

This deck is prepared by Seetha Parameswaran.

# TABLE OF CONTENTS

---

- 1 MODULE 4 TOPICS
- 2 VIDEO SEGMENTATION
- 3 SHOT BOUNDARY DETECTION
- 4 SPATIOTEMPORAL CHANGE DETECTION
- 5 MOTION SEGMENTATION
- 6 SEMANTIC VIDEO OBJECT SEGMENTATION
- 7 SPECIFIC ALGORITHMS/ TECHNIQUES
  - Segmentation

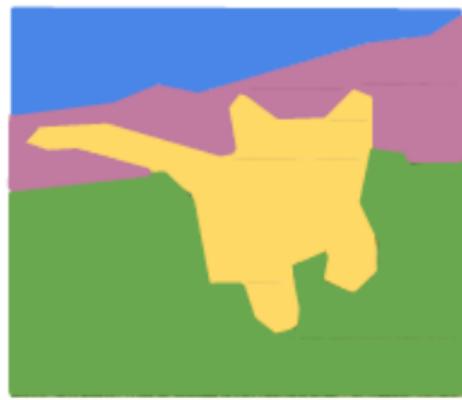
# MODULE TOPICS....

---

- Shot boundary detection
- Spatio temporal Change Detection
- Motion Segmentation
- Semantic video object segmentation
- Application

# IMAGE SEGMENTATION

---



GRASS, CAT,  
TREE, SKY

# TABLE OF CONTENTS

---

- 1 MODULE 4 TOPICS
- 2 VIDEO SEGMENTATION
- 3 SHOT BOUNDARY DETECTION
- 4 SPATIOTEMPORAL CHANGE DETECTION
- 5 MOTION SEGMENTATION
- 6 SEMANTIC VIDEO OBJECT SEGMENTATION
- 7 SPECIFIC ALGORITHMS/ TECHNIQUES
  - Segmentation

# VIDEO SEGMENTATION

**DEFINITION** **Video segmentation refers to the process of partitioning a video sequence into meaningful regions or objects.**

**GOAL** Understand the visual content of the video by automatically identifying and separating different elements like foreground objects, background scenes, and even specific features within objects.

**IMPORTANCE** Enables detailed understanding of content for applications like object tracking and scene analysis.

# NEED FOR VIDEO SEGMENTATION

---

- Fundamental step in understanding and extracting meaningful information from video data.
- Enabling advanced tasks

**OBJECT TRACKING AND RECOGNITION** Segmenting objects allows tracking their movement and identifying their types (e.g., cars, people, animals) within the video.

**SCENE UNDERSTANDING** By segmenting different regions like sky, buildings, and roads, the video's overall scene can be comprehended.

**SELF-DRIVING CARS** Identifying and segmenting objects (cars, pedestrians, lanes) is critical for autonomous vehicles to navigate safely.

**MEDICAL IMAGE ANALYSIS** Segmenting organs and tissues in medical videos aids in diagnosis and treatment decisions.

# APPLICATIONS OF VIDEO SEGMENTATION

**AUTONOMOUS VEHICLES** Segmenting objects like cars, pedestrians, and lanes is critical for self-driving cars to navigate safely and understand their surroundings.

**VIDEO SURVEILLANCE** Identifying and tracking objects in surveillance videos can enhance security and monitoring capabilities.

**MEDICAL IMAGING** Segmentation of organs and tissues is crucial for various medical applications, including diagnosis, treatment planning, and surgical guidance.

**CONTENT CREATION** Video editing and visual effects rely heavily on segmentation for isolating objects, applying special effects, and creating realistic compositions.

**ACTION RECOGNITION** Understanding the actions taking place in a video often requires identifying and segmenting the objects involved in the actions.

# CHALLENGES OF VIDEO SEGMENTATION

---

**COMPLEXITY OF VISUAL CONTENT** Videos can contain diverse objects, varying lighting conditions, and motion blur, making it difficult to distinguish boundaries accurately.

**TEMPORAL CONSISTENCY** Segmentation across frames needs to be consistent to avoid flickering or object disappearance, requiring methods that consider both spatial and temporal information.

**COMPUTATIONAL COST** Real-time applications often demand efficient algorithms that balance accuracy with processing speed.

**LACK OF LABELED DATA** Supervised learning approaches require large amounts of labeled video data, which can be expensive and time-consuming to acquire.

# TYPES OF VIDEO SEGMENTATION

---

Categorize video segmentation by

- Segmentation Level
- Approach
- Application

# VIDEO SEGMENTATION BY SEGMENTATION LEVEL

---

**OBJECT SEGMENTATION** Aims to identify and separate individual objects of

interest within the video frame. This can be further divided into

**SEMANTIC SEGMENTATION** Assigns each pixel in the frame to a specific object category (e.g., car, person, sky).

**INSTANCE SEGMENTATION** Not only identifies the object category but also assigns a unique identifier to each individual instance of that object (e.g., car1, car2).

**SCENE SEGMENTATION** Focuses on dividing the video scene into meaningful semantic regions, not necessarily individual objects (e.g., sky, road, building).

**MOTION SEGMENTATION** Groups pixels that belong to the same moving object based on their motion patterns.

# VIDEO SEGMENTATION BY APPROACH

---

**FRAME-BASED SEGMENTATION** Treats each frame individually, applying segmentation techniques similar to static images.

**MOTION-BASED SEGMENTATION** Leverages information about motion between frames to detect and segment moving objects.

**SUPERVISED LEARNING** Utilizes machine learning models trained on labeled video data to perform segmentation tasks.

**UNSUPERVISED LEARNING** Discovers patterns and features within unlabeled video data to automatically group similar pixels into segments.

# VIDEO SEGMENTATION BY APPLICATION

---

**FOREGROUND/BACKGROUND SEGMENTATION** Separates the main object(s) of interest (foreground) from the surrounding scene (background).

**SALIENCY SEGMENTATION** Identifies the most visually prominent or attention-grabbing regions within the frame.

**OPTICAL FLOW SEGMENTATION** Segments objects based on their motion patterns and direction.

**SEMANTIC VIDEO UNDERSTANDING** Aims to comprehensively understand the scene and objects depicted in the video, including their relationships and activities.

# SPATIAL VIDEO SEGMENTATION

---

Also known as semantic video segmentation or object segmentation.

## **Understanding the "What" in Video**

- Divides each frame of the video into spatial regions based on color, texture, intensity, or other low-level features.
- Identifying and separating individual objects of interest within each frame of a video sequence.
- Distinguish individual objects like cars, people, or animals.

# SEMANTIC VIDEO SEGMENTATION

---

**PIXEL-LEVEL LABELING** Each pixel in the frame is assigned a specific object category label, precisely outlining the boundaries of each object. This level of detail enables finer-grained understanding of the video content.

**MULTIPLE OBJECTS** It can handle scenarios with multiple objects of different types present in the same frame, accurately segmenting each object individually.

**INSTANCE SEGMENTATION** assigns unique identifiers to each individual instance of an object (e.g., car1, car2), allowing for tracking and counting within the video.

# SEMANTIC VIDEO SEGMENTATION - USE CASES

---

**AUTONOMOUS VEHICLES** Identifying and segmenting objects like cars, pedestrians, and lanes in real-time.

**VIDEO SURVEILLANCE** Accurately segmenting people and objects in surveillance videos. For instance, detecting people wearing red shirts in a crowded scene.

**ACTION AND ACTIVITY RECOGNITION** Understanding the actions taking place in a video often requires identifying and segmenting the objects involved in those actions.

**TRAFFIC MONITORING AND ANALYSIS** Segmenting vehicles and their types in traffic videos

**RETAIL ANALYTICS** Segmenting customers

# TEMPORAL VIDEO SEGMENTATION

---

## Understanding Change Over Time

- Analyse how the objects and their relationships change over time.
- Analyse the temporal continuity of video frames to identify changes or transitions over time.
- Identify and track the movement of objects across consecutive frames in a video sequence
- Identifying different scenes, actions, or events in the video sequence.

# TEMPORAL VIDEO SEGMENTATION

---

**MOTION ANALYSIS** It uses motion information between frames to track object movement, analyze trajectories, and understand interactions between objects.

**EVENT DETECTION** By identifying changes in object positions and relationships, temporal segmentation can detect specific events within the video, such as a person entering a room, a car changing lanes, or an object being manipulated.

**OBJECT TRACKING** It allows for efficient and robust tracking of specific objects across the entire video sequence, even when they are partially occluded or change appearance slightly.

# TEMPORAL VIDEO SEGMENTATION - USE CASES

**AUTONOMOUS VEHICLES** Track surrounding objects like cars, pedestrians, and cyclists.

**VIDEO SURVEILLANCE** Identifying and tracking suspicious activities in surveillance videos requires analyzing object movement patterns and interactions, which temporal segmentation facilitates.

**SPORTS ANALYTICS** Tracking players and analyzing their movements during a game can provide valuable insights for strategy development, performance evaluation, and broadcasting purposes.

Identify distinct phases of a game, such as detecting when a goal is scored or when players engage in specific actions.

**ANOMALY DETECTION IN VIDEOS** Identifying unusual or suspicious object movements in public spaces or sensitive areas can be achieved by analyzing deviations from typical movement patterns using temporal segmentation.

# SEMANTIC VIDEO SEGMENTATION

---

- Assigns semantic labels to each segment or region based on the content.
- Understand the meaning or category of different parts of the video, such as identifying objects or scenes.
- Use Case: In video content analysis, semantic segmentation can be applied to identify and label different scenes, such as categorizing outdoor scenes, indoor scenes, or specific environments.

# EVENT-BASED VIDEO SEGMENTATION

---

- Utilizes the concept of events to automatically identify and segment meaningful sections of a video sequence.
- Segments a video based on events or activities occurring within the video.
- **Event** sudden motion, changes in lighting, audio cues, or specific object appearances.

# EVENT-BASED VIDEO SEGMENTATION - USE CASES

---

**ACTION RECOGNITION** Segment videos based on relevant actions (e.g., person entering a room, object manipulation)

**VIDEO SUMMARIZATION** Identifying key events and summarizing the video based on those events can create concise and informative summaries, especially for long videos.

**ANOMALY DETECTION** Detecting deviations from typical events (e.g., unexpected object movement) can be used for anomaly detection in surveillance, healthcare, or industrial applications.

**SCENE UNDERSTANDING** Segmenting based on scene changes (e.g., transition from indoor to outdoor) can aid in understanding the overall context and composition of the video.

# FOREGROUND-BACKGROUND VIDEO SEGMENTATION

---

## Foreground Extraction (alternate name)

- Focuses on separating the foreground (objects of interest) from the background in each frame.
- Separate the main object(s) of interest (foreground) from the surrounding scene (background) in each frame of a video sequence.
- Binary classification: Each pixel in the frame is classified as either belonging to the foreground object(s) or the background.
- Used in applications such as object detection and tracking, object analysis, surveillance, content editing.

# SPATIO-TEMPORAL VIDEO SEGMENTATION

---

## **Joint space-time segmentation** (alternate name)

- Combines spatial and temporal information to create coherent video segments that represent both the spatial and temporal aspects of the video content.
- Combines the strengths of spatial video segmentation and temporal video segmentation to provide a comprehensive understanding of video content.
- It takes into account both the spatial information within each frame (identifying and separating objects) and the temporal information across frames (analyzing movement and interactions) for a richer and more nuanced understanding.

# SPATIO-TEMPORAL VIDEO SEGMENTATION - USE CASES

---

**AUTONOMOUS VEHICLES** Accurately tracking and understanding the movement of surrounding objects like cars, pedestrians, and cyclists

**ACTION RECOGNITION**

**VIDEO SURVEILLANCE** Analyzing object interactions and movements over time

**HUMAN-COMPUTER INTERACTION** Tracking user gestures and movements in real-time

**MEDICAL IMAGING** Analyzing organ movements and interactions throughout scans

# CLASSICAL TECHNIQUES FOR VIDEO SEGMENTATION

---

- Background Subtraction
- Optical Flow
- Graph-Based Segmentation
- Region Growing
- Clustering-Based Techniques
- MRF (Markov Random Fields) Models

# CLASSICAL TECHNIQUES FOR VIDEO SEGMENTATION

## ● BACKGROUND SUBTRACTION

**CONCEPT** Identifies foreground objects by comparing each frame to a pre-defined background model, assuming static background.

**ILLUMINATION-BASED METHODS** model the background as a statistical distribution and detect foreground objects based on deviations from this model.

**FRAME DIFFERENCING** Compares current frame to previous frame, highlighting pixels with significant differences (moving objects).

**RUNNING AVERAGE** Updates background model continuously with a temporal average, adapting to slow background changes.

**GAUSSIAN MIXTURE MODELS (GMMs)** Models background with multiple Gaussian distributions, handling dynamic backgrounds better.

# CLASSICAL TECHNIQUES FOR VIDEO SEGMENTATION

## ● OPTICAL FLOW

- ▶ Concept: Estimates pixel motion between consecutive frames, revealing object movement and boundaries. Segmentation is achieved by grouping pixels with similar motion vectors.
- ▶ Use Case: In action recognition, optical flow can be applied to capture the motion patterns of human activities, such as recognizing a person waving or running.

## ● THRESHOLDING

- ▶ Concept: Segments based on pixel intensity or color values exceeding a set threshold.
- ▶ Global thresholding: Uses a single threshold for the entire image.
- ▶ Adaptive thresholding: Employs local thresholds based on image region statistics.

# CLASSICAL TECHNIQUES FOR VIDEO SEGMENTATION

## ● GRAPH-BASED SEGMENTATION

- ▶ Concept: Model image pixels as nodes and connections between similar pixels as edges, segmenting based on graph properties. Graph-cut algorithms are then employed to partition the graph into segments.
- ▶ Normalized cuts: Minimizes differences between segments while considering their similarity within.
- ▶ Graph cuts: Finds the minimum cost cut to partition the graph into segments.
- ▶ Use Case: In medical imaging, graph-based segmentation can be applied to identify and delineate different anatomical structures in a sequence of medical images.

## ● REGION-BASED TECNIQUES

- ▶ Concept: Group pixels based on similarity in features like intensity, color, or texture.
- ▶ Region growing: Starts with seed points and expands regions based on similarity

# CLASSICAL TECHNIQUES FOR VIDEO SEGMENTATION

---

- CLUSTERING-BASED TECHNIQUES

- ▶ Clustering algorithms, such as k-means or mean-shift, group pixels with similar features into clusters, which can represent different segments in the video.

- MRF (MARKOV RANDOM FIELDS) MODELS

- ▶ MRF models represent the spatial dependencies between pixels. They incorporate information from neighboring pixels to achieve coherent segmentation.

# DEEP LEARNING FOR VIDEO SEGMENTATION

---

- ① CNNs
- ② U-Net
- ③ Temporal 3D CNN
- ④ DeepLab
- ⑤ LSTM and GRU Networks
- ⑥ Attention Mechanisms
- ⑦ Video Transformer Network (VTN)
- ⑧ Swin Transformer for Video Instance Segmentation
- ⑨ Convolutional Autoencoders

# CHOICE OF DL FOR VIDEO SEGMENTATION

---

- ① Semantic segmentation: CNNs like FCNs or U-Net
- ② Temporal segmentation: RNNs like LSTMs or Transformers
- ③ Joint spatio-temporal segmentation: Transformers
- ④ Unsupervised learning: Autoencoders

# PERFORMANCE MEASURES FOR VIDEO SEGMENTATION

---

**ACCURACY** Measures how well the segmented regions match the ground truth.

**PRECISION** Ratio of correctly segmented pixels to the total pixels segmented.

**RECALL** Ratio of correctly segmented pixels to the total pixels in the ground truth.

**INTERSECTION OVER UNION (IoU)** Measures the overlap between the segmented region and the ground truth.

# TABLE OF CONTENTS

---

- 1 MODULE 4 TOPICS
- 2 VIDEO SEGMENTATION
- 3 SHOT BOUNDARY DETECTION
- 4 SPATIOTEMPORAL CHANGE DETECTION
- 5 MOTION SEGMENTATION
- 6 SEMANTIC VIDEO OBJECT SEGMENTATION
- 7 SPECIFIC ALGORITHMS/ TECHNIQUES
  - Segmentation

# SHOT BOUNDARY DETECTION (SBD)

---

## Unveiling the Cuts in Video

**DEFINITION** Shot boundary detection process of automatically identifying transitions between shots within a video sequence.

**SHOTS** refers to a continuous sequence of frames captured from a single camera position without any internal cuts or edits.

**SHOT BOUNDARY** marks a significant change in content, camera perspective, or scene.

# TYPES OF SHOT BOUNDARIES

---

**CUT** A straightforward transition where one shot is immediately replaced by another. The most common and straightforward type of shot boundary.

**FADE** A gradual change in intensity, typically from or to black. This can include fade-in (gradual appearance) or fade-out (gradual disappearance).

**DISSOLVE** A gradual transition where one shot is replaced by another by blending the two shots together over a certain duration.

**WIPE** A spatial transition where the new shot appears to push or wipe the previous shot off the screen.

**ZOOM** A change in the focal length of the lens, resulting in a change in the apparent size of objects in the frame.

# APPLICATIONS OF SHOT BOUNDARY DETECTION

---

**VIDEO INDEXING AND RETRIEVAL** Enables efficient indexing of video content, allowing users to search for specific scenes or segments.

**CONTENT ANALYSIS** Facilitates content-based analysis by breaking down the video into meaningful units, aiding in tasks such as object detection, action recognition, and scene understanding.

**VIDEO SUMMARIZATION** Helps in generating concise video summaries by identifying key moments or scenes.

**VIDEO EDITING** Assists in the automatic or semi-automatic editing of videos by detecting suitable points for transitions.

**VIDEO COMPRESSION** Can be used in video compression algorithms to optimize encoding based on shot boundaries.

# TECHNIQUES FOR SHOT BOUNDARY DETECTION

**PIXEL-BASED METHODS** Analyze changes in pixel values to detect abrupt transitions.

**HISTOGRAM-BASED METHODS** Analyze changes in color or intensity histograms between consecutive frames to detect significant differences.

**EDGE-BASED METHODS** Analyze changes in edge detection results between frames, assuming cuts cause significant edge variations.

**FEATURE-BASED METHODS** Extract various features like brightness, motion vectors, or object statistics, identify sudden changes indicating cuts.

**MOTION-BASED METHODS** Analyze motion vectors between frames to detect changes in camera perspective or object movement.

**DOMAIN-SPECIFIC APPROACHES** Utilize domain knowledge about specific video types (e.g., news, sports).

# CHALLENGES IN SHOT BOUNDARY DETECTION

---

**VARIABILITY IN CONTENT** Different genres or styles of videos may exhibit diverse shot transition patterns.

**NOISE AND ARTIFACTS** Presence of noise, compression artifacts, or special effects can complicate shot boundary detection.

**COMPLEX TRANSITIONS** Transitions that involve multiple types (e.g., fade and dissolve) or subtle changes can be challenging to detect.

**COMPLEX CAMERA MOVEMENTS** Zooms, pans, and tilts can blur the distinction between shots.

**SUBJECTIVE PERCEPTION** Human perception of shot boundaries can differ, making evaluation of SBD algorithms challenging.

# CLASSICAL TECHNIQUES FOR SHOT BOUNDARY DETECTION

- **HISTOGRAM DIFFERENCE**

- ▶ Compares color histograms between consecutive frames and identifies a shot boundary when the histogram difference exceeds a certain threshold.

- **PIXEL DIFFERENCE**

- ▶ Computes the pixel-wise difference between frames and detects a shot boundary when the pixel difference exceeds a predefined threshold.

- **EDGE HISTOGRAMS**

- ▶ Focuses on detecting edges and computes histograms of edge orientations to find shot boundaries.

# CLASSICAL TECHNIQUES FOR SHOT BOUNDARY DETECTION

---

- COLOR HISTOGRAMS

- ▶ Analyzes changes in color histograms between frames to detect shot boundaries.

- FRAME DIFFERENCING

- ▶ Computes the absolute difference between pixel values of consecutive frames and identifies shot boundaries based on the resulting difference.

- MOTION VECTOR ANALYSIS

- ▶ Analyzes the motion vectors between frames and detects shot boundaries when significant changes occur.

# DEEP LEARNING TECHNIQUES FOR SHOT BOUNDARY DETECTION

---

- ① Two-stream networks
- ② 3D CNN
- ③ LSTM
- ④ Siamese Networks
- ⑤ Video Transformer Network (VTN)

# TABLE OF CONTENTS

---

- 1 MODULE 4 TOPICS
- 2 VIDEO SEGMENTATION
- 3 SHOT BOUNDARY DETECTION
- 4 SPATIOTEMPORAL CHANGE DETECTION
- 5 MOTION SEGMENTATION
- 6 SEMANTIC VIDEO OBJECT SEGMENTATION
- 7 SPECIFIC ALGORITHMS/ TECHNIQUES
  - Segmentation

# SPATIOTEMPORAL CHANGE DETECTION (STCD)

---

## Unveiling Dynamics Across Space and Time

**DEFINITION** Spatiotemporal change detection analyzes data captured across both space and time, identifying and characterizing modifications that occur within the data over a period. It essentially tracks how things change in different locations over time.

**GOAL** pinpoint the "where" and "when" of modifications within the data, as well as measure the magnitude and nature of those changes. Understand the driving forces behind those changes, providing valuable insights into various dynamic phenomena.

# SPATIOTEMPORAL CHANGE DETECTION (STCD)

---

**EXAMPLE** Monitor of a forest area using satellite imagery over multiple months.

- Spatiotemporal change detection involve identifying changes in vegetation cover, deforestation, or the occurrence of natural events like wildfires.
- The analysis would consider both spatial aspects (alterations in the arrangement of trees and vegetation) and temporal aspects (changes in vegetation cover over time).

# KEY ASPECTS OF SPATIOTEMPORAL CHANGE DETECTION

**Spatial Domain** refers to the analysis of changes in the spatial arrangement or characteristics of objects, features, or scenes.

Examples] Changes in land cover, urban expansion, object appearance or disappearance.

**Temporal Domain** involves analyzing changes that occur over time, focusing on the evolution of patterns, states, or conditions.

Changes in weather patterns, movement of objects, growth of vegetation.

**Combined Analysis** Spatiotemporal change detection integrates both spatial and temporal information to provide a comprehensive understanding of evolving phenomena.

Enables the identification of dynamic patterns and trends, distinguishing between short-term fluctuations and long-term changes.

# APPLICATIONS OF SPATIOTEMPORAL CHANGE DETECTION

---

**ENVIRONMENTAL MONITORING** Tracking deforestation, glacier retreat, land-use changes, and other environmental dynamics.

**URBAN PLANNING** Monitoring urban development, traffic patterns, and population shifts.

**MEDICAL IMAGING** Identifying tumor growth, analyzing organ functions, and assessing treatment effectiveness.

**REMOTE SENSING** Detecting wildfires, floods, and other natural disasters.  
Monitoring changes in Earth's surface, such as urbanization or agricultural developments.

**CLIMATE CHANGE STUDIES** Monitoring variations in sea level, ice cover, and temperature patterns.

# APPLICATIONS OF SPATIOTEMPORAL CHANGE DETECTION

---

**VIDEO SURVEILLANCE** Monitoring suspicious activities and identifying potential threats.

**PRECISION AGRICULTURE** Identifying crop health issues, optimizing irrigation, and managing crop rotation.

**BIODIVERSITY CONSERVATION** Assessing habitat changes and species distributions.

**ARCHAEOLOGY** Mapping ancient settlements and uncovering historical patterns.

**PUBLIC HEALTH SURVEILLANCE** Detecting and tracking disease outbreaks.

# CHALLENGES IN SPATIOTEMPORAL CHANGE DETECTION

---

**DATA COMPLEXITY** Dealing with large, multidimensional datasets from various sources (e.g., satellite imagery, sensor measurements) can be demanding.

**NOISE AND UNCERTAINTY** Distinguishing real changes from noise or measurement errors requires robust algorithms.

**MIXED SPATIAL AND TEMPORAL SCALES** Capturing changes occurring at different spatial and temporal resolutions can be difficult.

**LIMITED GROUND TRUTH DATA** Verifying and validating change detection results can be challenging due to the lack of comprehensive ground truth information.

# CLASSIC TECHNIQUES FOR SPATIOTEMPORAL CHANGE DETECTION

## ● IMAGE DIFFERENCING

- ▶ Concept: Compares consecutive images pixel-wise, highlighting differences exceeding a threshold as potential changes.
- ▶ Simple absolute difference: Subtracts pixel values in corresponding locations between images.
- ▶ Ratio image differencing: Divides corresponding pixel values to handle lighting changes.

## ● THRESHOLDING TECHNIQUES

- ▶ Concept: Applies a threshold to a chosen metric (e.g., difference image, standard deviation) to segment potential change areas.
- ▶ Global thresholding: Uses a single threshold for the entire image.
- ▶ Adaptive thresholding: Employs local thresholds based on image regions.

# CLASSIC TECHNIQUES FOR SPATIOTEMPORAL CHANGE DETECTION

---

- **CHANGE VECTOR ANALYSIS (CVA)**

- ▶ Concept: Analyzes spectral changes between images by subtracting corresponding bands and visualizing the differences.

- **PRINCIPAL COMPONENT ANALYSIS (PCA)**

- ▶ Concept: Projects data onto lower-dimensional subspaces, highlighting changes based on principal components capturing most variance.

- **TEMPORAL AVERAGE DEVIATION**

- ▶ Calculating the average deviation of pixel values over time to detect temporal changes.

# CLASSIC TECHNIQUES FOR SPATIOTEMPORAL CHANGE DETECTION

---

- TEMPORAL RATIO IMAGE

- ▶ Computing the ratio between pixel values at different time points to highlight temporal changes.

- MOVING OBJECT DETECTION TECHNIQUES

- ▶ Utilizing techniques like background subtraction or optical flow to identify moving objects as indicators of change.

- SPATIOTEMPORAL FILTERS

- ▶ Applying filters that consider both spatial and temporal information to enhance change detection.

# DEEP LEARNING FOR SPATIOTEMPORAL CHANGE DETECTION

---

- ① 3D CNN
- ② GRU and LSTM
- ③ Video Transformer Network (VTN)
- ④ Siamese Network
- ⑤ Convolutional Autoencoders

# TABLE OF CONTENTS

---

- 1 MODULE 4 TOPICS
- 2 VIDEO SEGMENTATION
- 3 SHOT BOUNDARY DETECTION
- 4 SPATIOTEMPORAL CHANGE DETECTION
- 5 MOTION SEGMENTATION
- 6 SEMANTIC VIDEO OBJECT SEGMENTATION
- 7 SPECIFIC ALGORITHMS/ TECHNIQUES
  - Segmentation

# MOTION SEGMENTATION

---

## Breaking Down Movement in Visual Data

**DEFINITION** Motion segmentation is the process of partitioning a sequence of frames in a video into different regions based on the motion characteristics of objects within the scene.

**GOAL** distinguish between regions of the video that exhibit motion (dynamic regions) and those that remain static (static regions).

**EXAMPLE** Consider a surveillance video capturing a busy street scene.

- Motion segmentation would identify and track individual moving objects such as vehicles, pedestrians, or other dynamic elements, effectively isolating them from the static background.
- Used in tasks like traffic monitoring or anomaly detection.

# APPLICATIONS OF MOTION SEGMENTATION

---

**SURVEILLANCE AND SECURITY** Identifying intruders, abandoned objects, or unusual activities.

**VIDEO EDITING AND POST-PRODUCTION** Isolating specific objects or motions for creative effects.

**ROBOT VISION AND NAVIGATION** Segmenting dynamic obstacles and landmarks for safe movement.

**SIGN LANGUAGE RECOGNITION** Understanding hand gestures by segmenting individual signs.

**TRAFFIC MONITORING AND ANALYSIS** Tracking vehicles and estimating traffic flow.

# CHALLENGES OF MOTION SEGMENTATION

---

**COMPLEX MOTION PATTERNS** Handling diverse motions like overlapping objects, camera motion, and background clutter is demanding.

**ILLUMINATION CHANGES AND NOISE** Variations in lighting and sensor noise can hinder accurate segmentation.

**OCCLUSIONS AND PARTIAL VISIBILITY** Segments may be incomplete due to objects hiding or overlapping each other.

**REAL-TIME PROCESSING REQUIREMENTS** Certain applications necessitate efficient algorithms for fast processing.

# CLASSICAL TECHNIQUES FOR MOTION SEGMENTATION

---

- OPTICAL FLOW

- ▶ Concept: Estimates the displacement of pixels between consecutive frames, revealing motion information.
  - ▶ Use Horn-Schunck algorithm or Lucas-Kanade algorithm.

- FRAME DIFFERENCING

- ▶ Involves subtracting consecutive frames to highlight regions with changes.

- POINT FEATURE TRACKING

- ▶ Concept: Tracks distinctive points (e.g., corners, blobs) across frames, inferring motion based on their displacement.
  - ▶ KLT (Kanade-Lucas-Tomasi) tracker or Harris corner detector.

# CLASSICAL TECHNIQUES FOR MOTION SEGMENTATION

---

## ● BACKGROUND SUBTRACTION

- ▶ Concept: Models the background scene and identifies moving objects as deviations from the model.
- ▶ Static background model: Assumes a static background and detects motion based on pixel intensity changes.
- ▶ Gaussian Mixture Models (GMMs): Represent the background with a mixture of Gaussians to handle dynamic backgrounds.

## ● K-MEANS CLUSTERING

- ▶ Groups pixels with similar motion characteristics into clusters, separating moving objects from the background.

# CLASSICAL TECHNIQUES FOR MOTION SEGMENTATION

---

## ● REGION-BASED SEGMENTATION

- ▶ Concept: Groups pixels with similar motion characteristics into coherent regions, representing moving objects.
- ▶ Mean-shift algorithm: Iteratively shifts segments towards higher density regions in the feature space.
- ▶ Graph-based segmentation: Constructs a graph where nodes represent pixels and edges represent similarity based on motion features, then segments based on graph partitioning.

# DL TECHNIQUES FOR MOTION SEGMENTATION

---

- Motion-based CNN
- Recurrent Convolutional Neural Networks (RCNNs)
- Video Transformer Network (VTN)
- Swin Transformer for Motion Segmentation
- Convolutional Autoencoders with Motion-Aware Regularization
- Fusion of multiple CNN architectures trained on different datasets

# TABLE OF CONTENTS

---

- 1 MODULE 4 TOPICS
- 2 VIDEO SEGMENTATION
- 3 SHOT BOUNDARY DETECTION
- 4 SPATIOTEMPORAL CHANGE DETECTION
- 5 MOTION SEGMENTATION
- 6 SEMANTIC VIDEO OBJECT SEGMENTATION
- 7 SPECIFIC ALGORITHMS/ TECHNIQUES
  - Segmentation

# SEMANTIC VIDEO OBJECT SEGMENTATION (SVOS)

## **Understanding the "What" and "Where" of Moving Objects**

**DEFINITION** Semantic video object segmentation refers to the task of segmenting objects in a video and assigning semantic labels to the segmented regions.

SVOS aims to automatically identify and segment moving objects in a video sequence, not only understanding their location ("where") but also their semantic meaning ("what").

**GOAL** understand the content of video frames at a higher level, associating each segmented object with a specific semantic class or category.

**EXAMPLE** Consider a video captured by a surveillance camera monitoring a busy street.

- Identifying and label individual objects such as pedestrians, vehicles, and cyclists in each frame, providing a detailed

understanding of the scene

# KEY ASPECTS OF SEMANTIC VIDEO OBJECT SEGMENTATION

---

**OBJECT-LEVEL SEGMENTATION** delineate and segment individual objects or entities within a video sequence.

**SEMANTIC LABELING** assign semantic labels to each segmented object, specifying the object's class or category.

**TEMPORAL CONSISTENCY** Considers the temporal coherence of object segmentation and semantic labeling across consecutive frames in a video.

# APPLICATIONS OF SEMANTIC VIDEO OBJECT SEGMENTATION

---

**SURVEILLANCE AND SECURITY** Identifying intruders, analyzing suspicious activities, and classifying objects of interest.

**AUTONOMOUS VEHICLES** Tracking surrounding vehicles and pedestrians for collision avoidance and safe navigation.

**TRAFFIC MONITORING AND ANALYSIS** Classifying vehicles, counting traffic flow, and detecting traffic violations.

**SPORTS ANALYSIS** Tracking players, analyzing their movements, and automatically generating statistics.

**VIDEO EDITING AND POST-PRODUCTION** Segmenting specific objects or actions for creating special effects or editing content seamlessly.

# CHALLENGES OF SEMANTIC VIDEO OBJECT SEGMENTATION

---

**COMPLEX VISUAL CONTENT** Handling diverse object appearances, occlusions, background clutter, and fast-moving objects is demanding.

**REAL-TIME PROCESSING REQUIREMENTS** Certain applications necessitate efficient algorithms for fast segmentation in real-time.

**SEMANTIC AMBIGUITY** Differentiating between objects with similar appearances or actions can be challenging.

**LIMITED TRAINING DATA** Obtaining large datasets with accurate semantic labels for every object can be expensive and time-consuming.

# CLASSIC TECHNIQUES OF SVOS

---

- ① Optical Flow: Estimates pixel displacement between consecutive frames, revealing motion information but without semantic labels.
- ② Background Subtraction: Identifies moving objects as deviations from a modeled background, but doesn't provide semantic labels.
- ③ Point Feature Tracking: Tracks distinctive points through frames, inferring motion but not semantics.
- ④ Region-based Segmentation: Groups pixels with similar motion characteristics, useful for segmenting moving regions but not providing object-level labels.

# DEEP LEARNING OF SVOS

---

- ① Fully Convolutional Networks (FCNs)
- ② LSTM
- ③ Video Transformer Network (VTN)
- ④ Deep Video Segmentation Network (DVSN)
- ⑤ STC (Spatiotemporal Convolutional Networks)
- ⑥ Deep Temporal Linear Encoding Networks
- ⑦ Tiramisu

# TABLE OF CONTENTS

---

- 1 MODULE 4 TOPICS
- 2 VIDEO SEGMENTATION
- 3 SHOT BOUNDARY DETECTION
- 4 SPATIOTEMPORAL CHANGE DETECTION
- 5 MOTION SEGMENTATION
- 6 SEMANTIC VIDEO OBJECT SEGMENTATION
- 7 SPECIFIC ALGORITHMS/ TECHNIQUES
  - Segmentation

# SEGMENTATION

---

- Segmentation is a way to partition the image into regions or segments with homogenous behaviour. These segments to be meaningful.
- Segmentation is a way to separate the image frame into simple regions.
- Group pixels with similar visual characteristics.

# SEGMENTATION STRATEGIES

- Image segmentation is highly subjective by humans.  
Segmentation is highly intuitive for humans. It is very hard to translate these intuitions to an algorithm.
- Top-down approach – pixels belong together because they come from same object. (Gestalt's principles)
- Bottom-up approach – pixels belong together because they look similar.

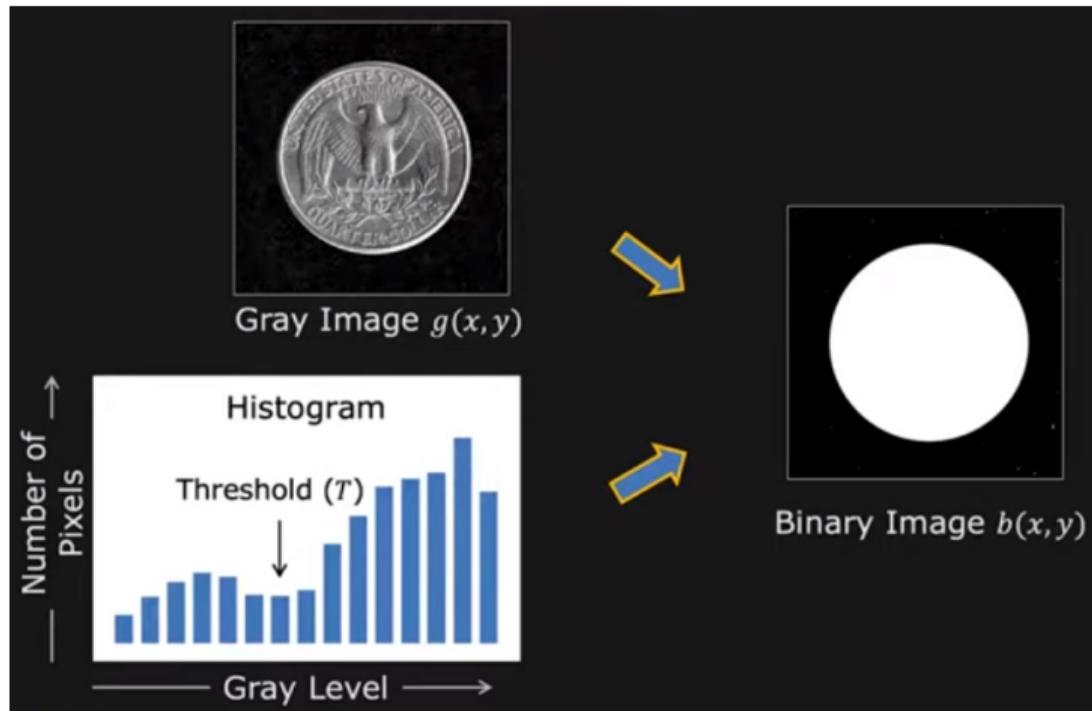


# SEGMENTATION APPROACHES

---

- Segmentation as clustering
  - ▶ Each pixel in the image is high-dimensional feature vector.
  - ▶ Cluster points together, in such a way that the clusters represent meaningful regions or segments.
  - ▶ Use k-means clustering algorithm
- Region growing is a technique that groups pixels into regions which are homogeneous in nature.
- Graph cut segmentation

# SEGMENTATION



# VISUAL CHARACTERISTICS OF PIXELS

---

- Brightness
- Colour
- Position
- Depth
- Motion
- Texture

Pixel can be considered a feature vector. (R,G,B)

# SIMILARITY OF PIXELS

---

- Consider any two pixels  $p_i$  and  $p_j$  whose feature vectors are  $f_i$  and  $f_j$ .
- Similarity as L2 distance between  $f_i$  and  $f_j$

$$S(f_i, f_j) = \sqrt{\sum_k (f_{ik} - f_{jk})^2}$$

- smaller the distance, greater the similarity.

# HOMOGENEITY

- Homogeneous regions are identified by using a similarity measure between pixels.
- Similarity measure can be Euclidean or Manhattan distance
  - ▶ Intensity level or color level
  - ▶ Texture
  - ▶ Shape
- Example: Use Euclidean distance to measure pixel homogeneity based on color.
- Let two image points be  $p_1 = (91, 134, 234)$  and  $p_2 = (231, 105, 100)$  in RGB representation.

$$\begin{aligned}d_{p_1, p_2} &= \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \\&= \sqrt{(91 - 231)^2 + (134 - 105)^2 + (234 - 100)^2} \approx 195\end{aligned}$$

# REGION GROWING ALGORITHM

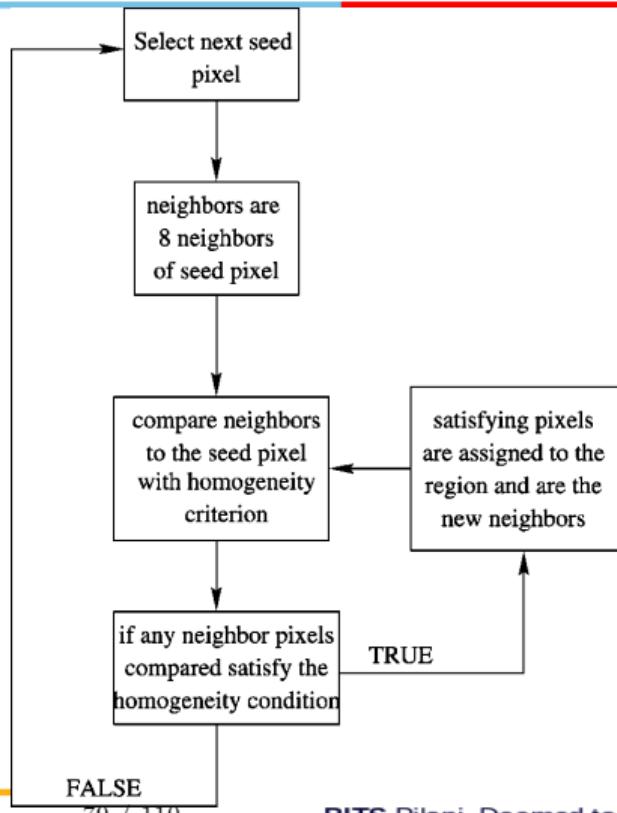
- Select an initial seed.
- Grow the region from the seed by merging neighbouring pixels.
  - ▶ The algorithm defines a homogeneity threshold  $T$  between pixel intensities.

$d_{p_1,p_2} \leq T \rightarrow$  pixels are homogeneous

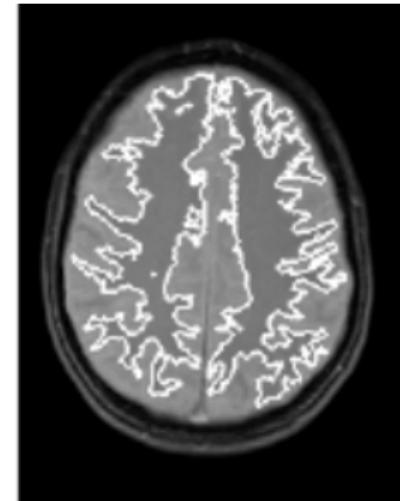
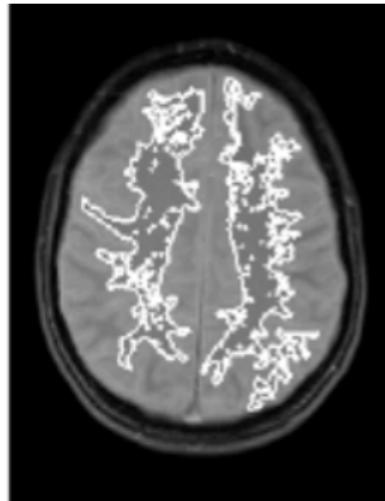
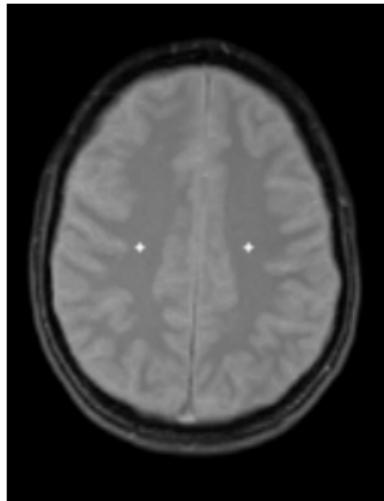
$d_{p_1,p_2} > T \rightarrow$  pixels are **not** homogeneous

- Stop when all pixels have been processed.

# REGION GROWING ALGORITHM



# REGION GROWING SEGMENTATION



# REGION GROWING ALGORITHM

---

## Algorithm 1: REGION GROWING

---

**Data:** seed

**Result:** segmented regions

```
1 each region mean = seed value ;  
2 set error as similarity measure for regions ;  
3 while there is unallocated pixels do  
4   for each pixel in each region do  
5     if unallocated neighbors of (2x2 , 3x3 or 4x4) patches within a specified  
       errors from region mean then  
6       add to region and recalculate region mean ;  
7     else  
8       mark as visited and do not add to region;
```

---

# REGION GROWING EXAMPLE

- Consider an image patch given below. Apply region growing method to identify the segments. Assume a seed value of 7 and similarity threshold of 10. Use 8-connectivity neighbourhood.

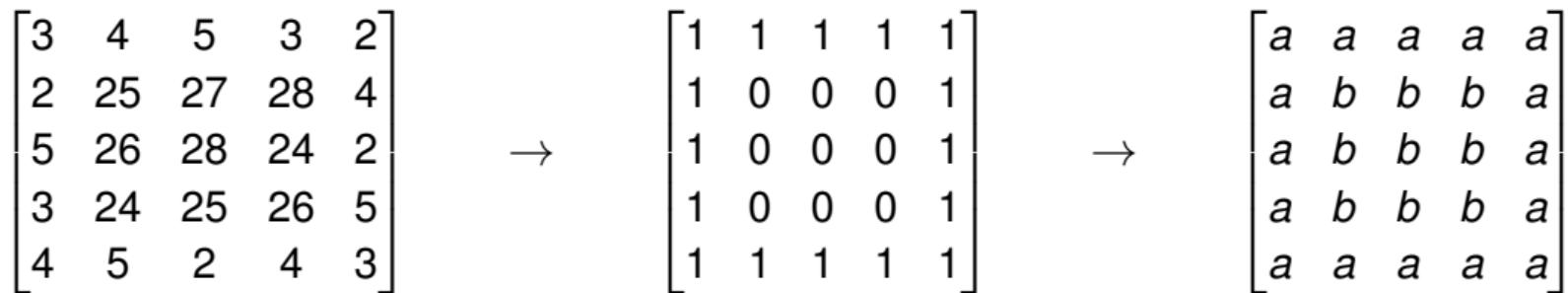
$$\begin{bmatrix} 3 & 4 & 5 & 3 & 2 \\ 2 & 25 & 27 & 28 & 4 \\ 5 & 26 & 28 & 24 & 2 \\ 3 & 24 & 25 & 26 & 5 \\ 4 & 5 & 2 & 4 & 3 \end{bmatrix}$$

# REGION GROWING EXAMPLE

- Condition 1: Homogeneity. Identify pixel homogeneity using Manhattan distance.

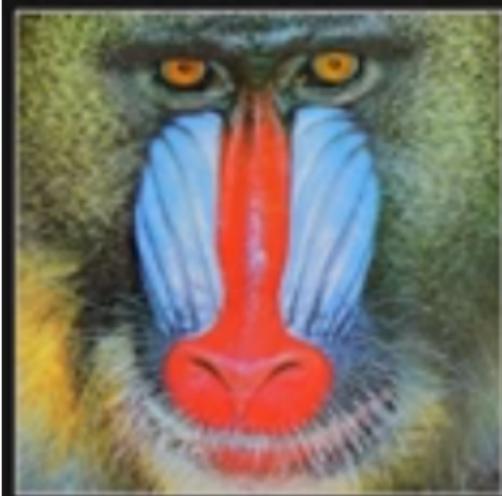
$$| \text{pixel} - \text{seed} | < \text{threshold}$$

- Condition 2: 8-connectivity neighbourhood. Label into different regions.

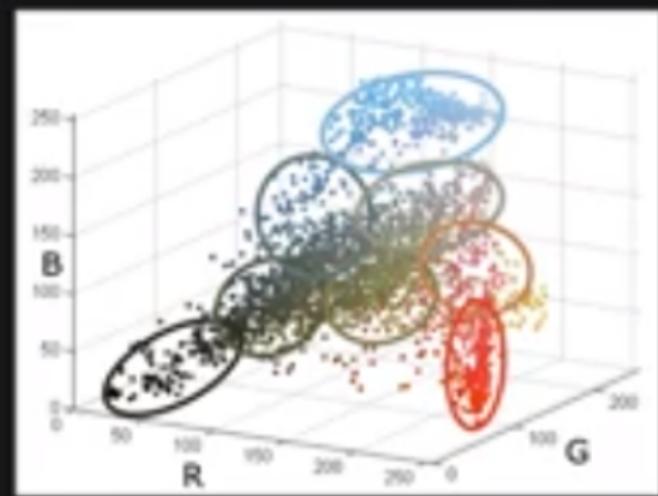


# K-MEANS SEGMENTATION

Form clusters based on color of each pixel.



Input Image



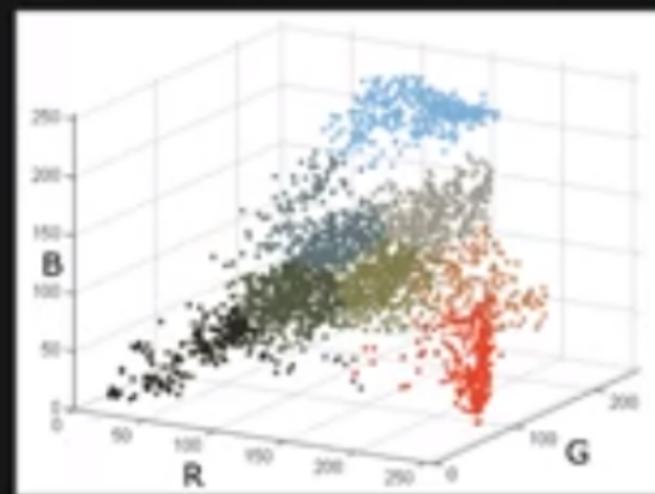
Pixel RGB Color Distribution  
(Color of feature point  $\equiv$  Color of image pixel)

# K-MEANS SEGMENTATION

Label based on the average colour of each cluster.



Segmented Image



Color-Coded Clusters  
(Color of feature point  $\equiv$  Color of image segment)

# K-MEANS SEGMENTATION

Given: Image with  $N$  pixels and number of cluster  $k$

Task: Segment the image into  $k$  clusters.

Clustering:

- ① Select  $k$  random feature points as the initial centroids (means)  $\{m_1, m_2, \dots, m_k\}$  of the  $k$  clusters in the feature space. If two points are very close, resample.
- ② For each pixel  $x_i$  find the nearest cluster mean  $m_i$  and assign pixel to cluster  $i$ .
- ③ Recompute mean for each cluster using its assigned pixel.
- ④ Stop when the mean for each cluster converge, else go to step 2.

# GRAPH BASED SEGMENTATION

---

- Represent image as a graph.
- Use cuts to find clusters in the graph.

# IMAGE AS A GRAPH

---

- A node for each pixel.
- An edge between each pair of pixels.
- Graph notation:  $G = (V, E)$  where  $V$  and  $E$  are sets of nodes and edges respectively.
- Each edge is weighted by the affinity or similarity between its two vertices.
  - ▶ Let  $p_i$  and  $p_j$  be two pixels whose features are  $f_i$  and  $f_j$ .
  - ▶ Pixel Dissimilarity

$$s(f_i, f_j) = \sqrt{\sum_k (f_{ik} - f_{jk})^2}$$

- ▶ Pixel Affinity = weight of the edge

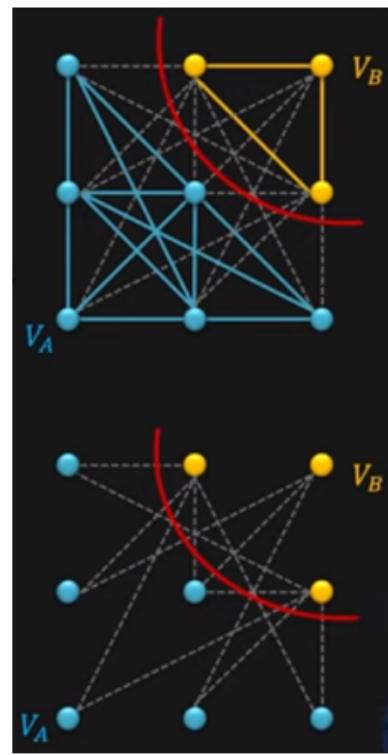
$$w(i, j) = A(f_i, f_j) = \exp\left(\frac{-1}{2\sigma^2} s(f_i, f_j)\right)$$

- **Smaller the dissimilarity, larger the affinity.**

# GRAPH CUT

- Cut  $C = (V_A, V_B)$ 
  - ▶ Cut is a partition of nodes  $V$  of the graph  $G = (V, E)$  into two **disjoint subsets**  $V_A$  and  $V_B$ .
- Cut-set
  - ▶ Set of edges whose nodes are in different subsets of partition.
- Cost of Cut
  - ▶ Sum of weights of cut-set edges.

$$\text{cut}(V_A, V_B) = \sum_{u \in V_A, v \in V_B} w(u, v)$$



# GRAPH CUT SEGMENTATION

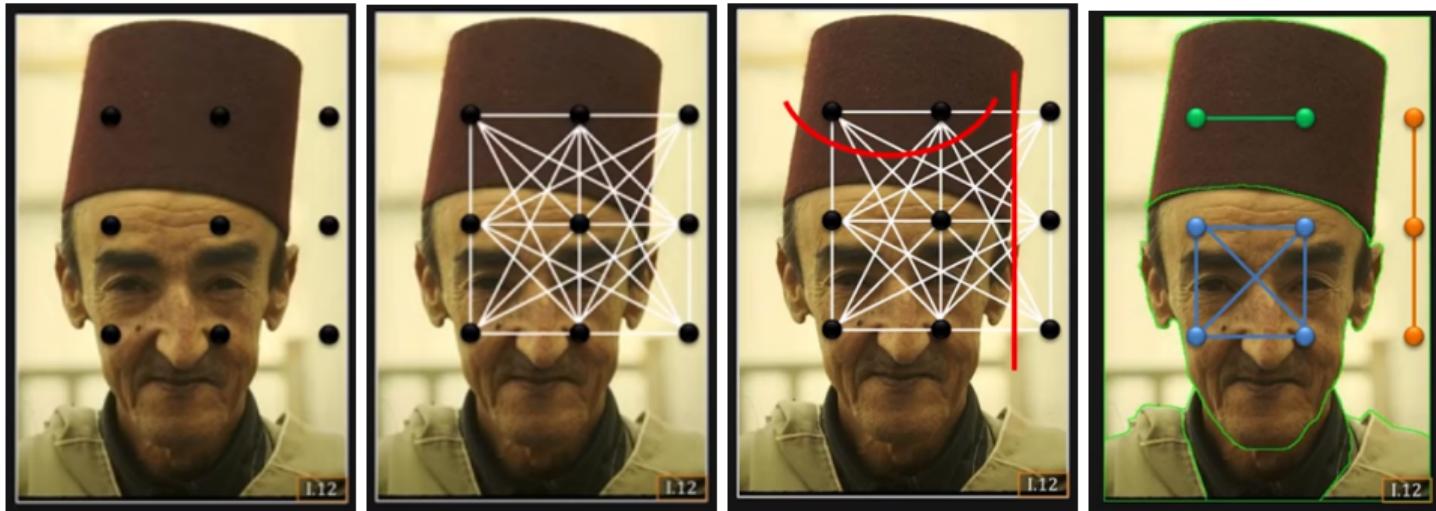
---

Criteria for Graph Cut:

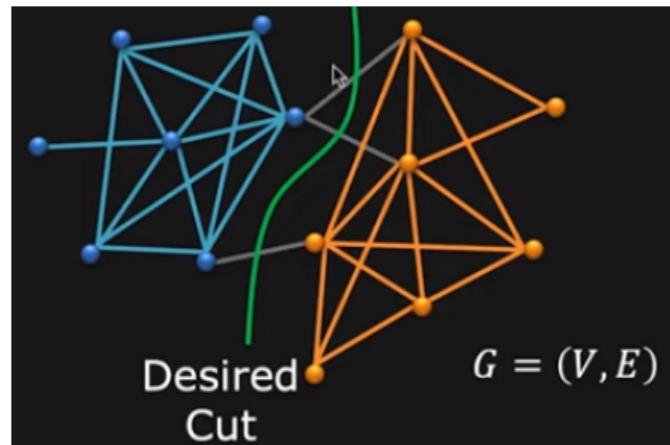
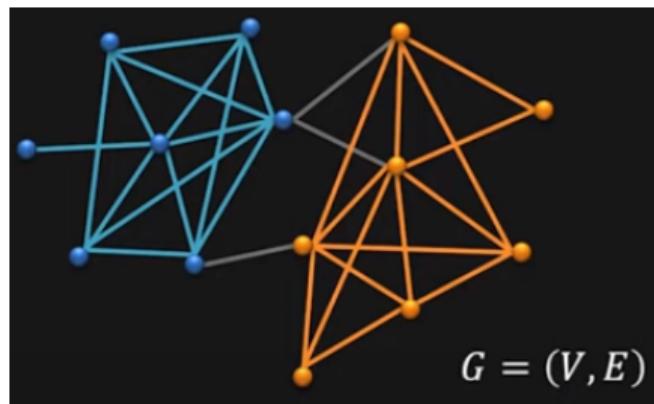
- A pair of nodes within a subgraph have **high affinity**.
- A pair of nodes from two different subgraphs have **low affinity**.

Minimize the cost of cut. Also called as **Min-Cut**

# GRAPH CUT SEGMENTATION EXAMPLE

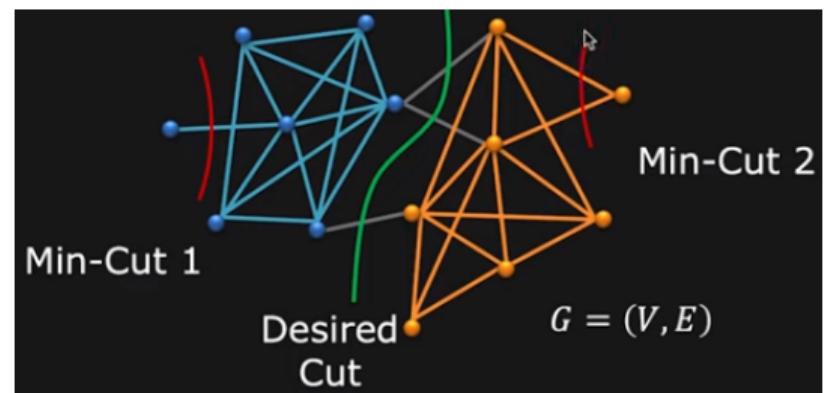
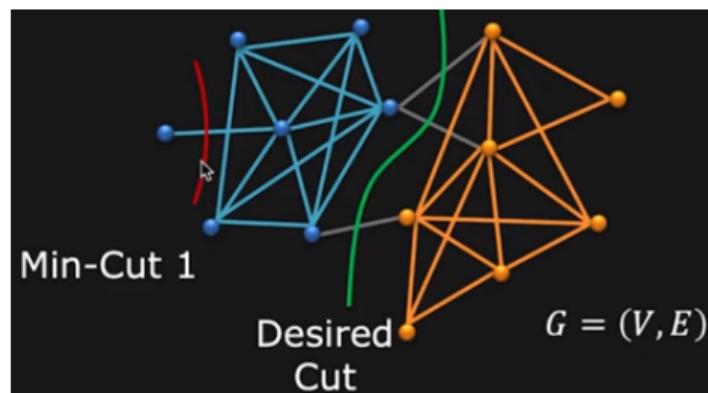


# GRAPH MIN-CUT SEGMENTATION



# GRAPH MIN-CUT SEGMENTATION

Min-cut is biased to small, isolated segments.

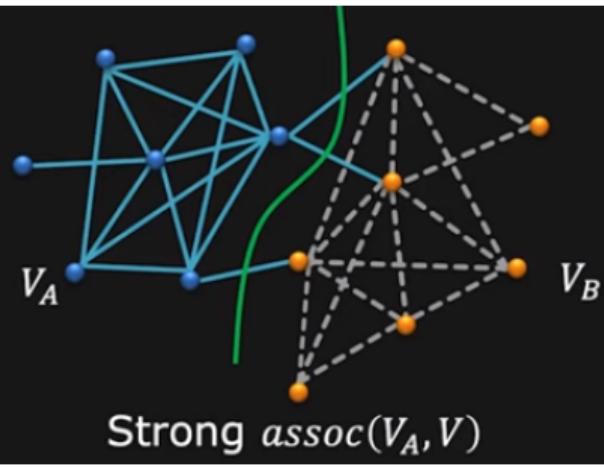
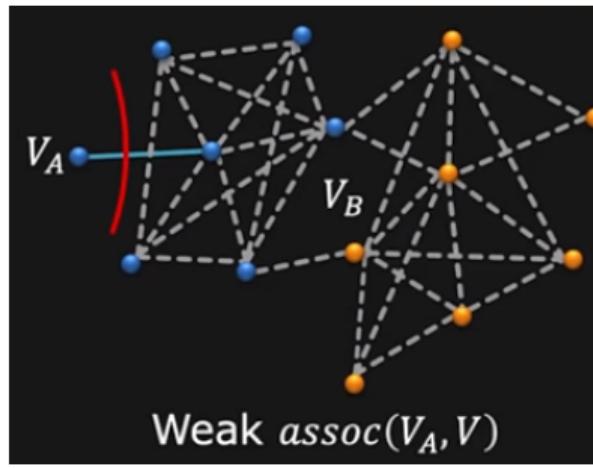


Solution: Normalize cut to favour larger subgraphs

# MEASURE SUBGRAPH SIZE

- Compute how strongly vertices  $V_A$  are associated with nodes  $V$ .
- $\text{assoc}()$  is the sum of weights of the solid edges.

$$\text{assoc}(V_A, V) = \sum_{u \in V_A, v \in V} w(u, v)$$



# NORMALISED CUT (NCUT)

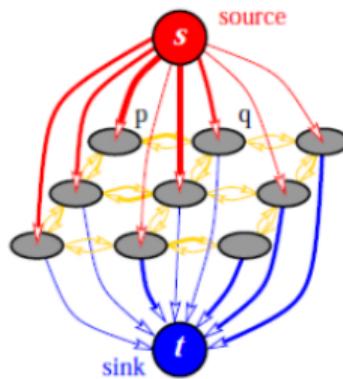
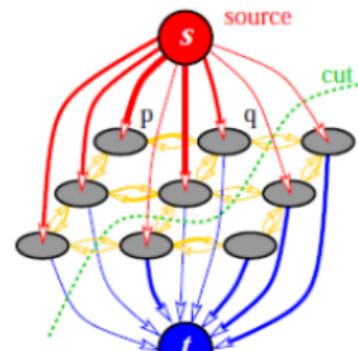
- Minimize cost of normalised cut.

$$NCut(V_A, V_B) = \frac{cut(V_A, V_B)}{assoc(V_A, V)} + \frac{cut(V_A, V_B)}{assoc(V_A, V)}$$

- NP complete problem.
- Approximate solution exists.

# EXAMPLE

- Two terminal graph on a  $3 \times 3$  image with two labels.
- An  $s/t$  cut  $C$  on a graph with two terminals is a partitioning of the nodes in the graph into two disjoint subsets  $S$  and  $T$  such that the source  $s$  is in  $S$  and the sink  $t$  is in  $T$ .

(a) A graph  $\mathcal{G}$ (b) A cut on  $\mathcal{G}$

# BOYKOV-KOLMOGOROV ALGORITHM

---

This algorithm has following three stages:

- **Growth stage:** search trees  $S$  and  $T$  grow until they touch giving an  $s \rightarrow t$  path,
- **Augmentation stage:** the found path is augmented, search tree(s) break into forest(s),
- **Adoption stage:** trees  $S$  and  $T$  are restored.

Worst case complexity is  $O(mn^2|C|)$ . (where  $m$  is number of nodes,  $n$  is number of edges and  $|C|$  is cost of minimum cut)

# BOYKOV-KOLMOGOROV ALGORITHM

- Maintain two non-overlapping search trees  $S$  and  $T$  with roots at the source  $s$  and the sink  $t$ , correspondingly.
- In tree  $S$  all edges from each parent node to its children are non-saturated.
- In tree  $T$  edges from children to their parents are non-saturated.
- The nodes that are not in  $S$  or  $T$  are called **free**.

$$S \subset V, \quad s \in S, \quad T \subset V, \quad t \in T, \quad S \cap T = \emptyset$$

- The nodes in the search trees  $S$  and  $T$  can be either active or passive.
- The **active nodes** represent the outer border in each tree. Active nodes allow trees to "grow" by acquiring new children.
- The **passive nodes** are internal. Passive nodes can not grow as they are completely blocked by other nodes from the same tree.
- An augmenting path is found as soon as an active node in one of the trees detects a neighboring node that belongs to the other tree.

# BOYKOV-KOLMOGOROV ALGORITHM

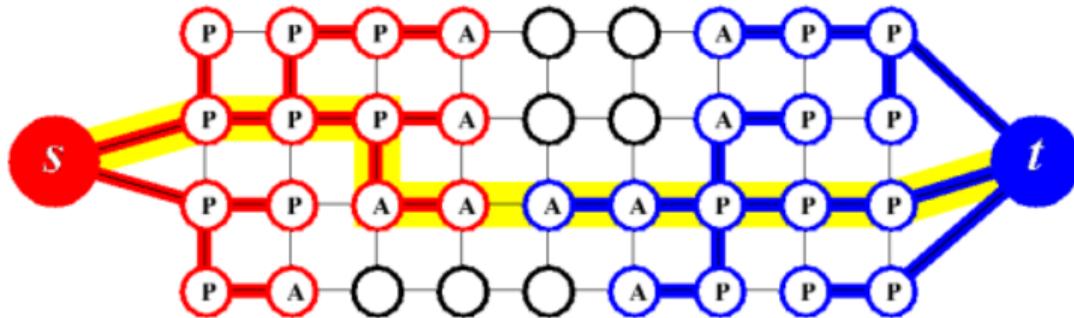


Figure 4: Example of the search trees  $S$  (red nodes) and  $T$  (blue nodes) at the end of the growth stage when a path (yellow line) from the source  $s$  to the sink  $t$  is found. Active and passive nodes are labeled by letters **A** and **P**, correspondingly. Free nodes appear in black.

# BOYKOV-KOLMOGOROV ALGORITHM

---

- $TREE(p)$  indicate affiliation of each node  $p$ .

$$TREE(p) = \begin{cases} S & (\text{if } p \in S) \\ T & (\text{if } p \in T) \\ \emptyset & (\text{if } p \text{ is free}) \end{cases}$$

- If node  $p$  belongs to one of the search trees then the information about its parent will be stored as  $PARENT(p)$ .
- $tree\_cap(p \rightarrow q)$  describe residual capacity of either edge  $(p, q)$  if  $TREE(p) = S$  or edge  $(q, p)$  if  $TREE(p) = T$ .

# BOYKOV-KOLMOGOROV ALGORITHM

---

## Algorithm 2: BOYKOV-KOLMOGOROV ALGORITHM

---

```
1 initialize:  $S = \{s\}$ ,  $T = \{t\}$ ,  $A = \{s, t\}$ ,  $O = \emptyset$ 
2 while true do
3   grow  $S$  or  $T$  to find an augmenting path  $P$  from  $s$  to  $t$ 
4   if  $P = \emptyset$  then
5     | terminate
6   augment on  $P$ 
7   adopt orphans
```

---

# BOYKOV-KOLMOGOROV ALGORITHM - GROWTH STAGE

---

- At the growth stage the search trees expand.
- Active nodes explore adjacent non-saturated edges and acquire new children from set of free nodes.
- Newly acquired nodes will become active members of corresponding search trees.
- When all neighbors of an active node are explored, it becomes passive.
- The growth stage terminates if an active node encounters a neighboring node that belongs to the opposite tree.
- This detects a path from the source to the sink.

# BOYKOV-KOLMOGOROV ALGORITHM

---

## Algorithm 3: BOYKOV-KOLMOGOROV ALGORITHM - GROWTH STAGE

---

```
1 while  $A \neq \phi$  do
2     pick an active node  $p \in A$ 
3     for every neighbor  $q$  such that  $\text{tree\_cap}(p \rightarrow q) > 0$  do
4         if  $\text{TREE}(q) = \phi$  then
5             add  $q$  to search tree as an active node
6              $\text{TREE}(q) = \text{TREE}(p)$ 
7              $\text{PARENT}(q) = p$ 
8              $A = A \cup q$ 
9             if  $\text{TREE}(q) \neq \phi$  and  $\text{TREE}(q) \neq \text{TREE}(p)$  then
10                return  $P = \text{PATH}_{\{s \rightarrow t\}}$ 
11        remove  $p$  from  $A$ 
12    return  $P = \phi$ 
```

# BOYKOV-KOLMOGOROV ALGORITHM - AUGMENTATION STAGE

- The augmentation stage augments path found in growth stage.
- Since we push through the largest flow possible some edge(s) in the path become saturated.
- So, when we remove such edge(s), some of the nodes in the trees  $S$  and  $T$  may become orphans as the edges linking them to their parents are no longer valid (they are saturated).
- The augmentation phase may split the search trees  $S$  and  $T$  into forests.
- The source  $s$  and the sink  $t$  are still roots of two of the trees while orphans form roots of all other trees.

# BOYKOV-KOLMOGOROV ALGORITHM

---

## Algorithm 4: Boykov-Kolmogorov Algorithm - Augmentation Stage

---

- 1 find the bottleneck capacity  $\Delta$  on  $P$
- 2 update the residual graph by pushing flow  $\Delta$  through  $P$
- 3 **for** each edge  $(p, q)$  in  $P$  that becomes saturated **do**
- 4     **if**  $TREE(p) = REE(q) = S$  **then**
- 5         set  $PARENT(q) := \phi$
- 6         set  $O := O \cup \{q\}$
- 7     **if**  $TREE(p) = TREE(q) = T$  **then**
- 8         set  $PARENT(p) := \phi$
- 9         set  $O := O \cup \{p\}$

---

# BOYKOV-KOLMOGOROV ALGORITHM - ADOPTION STAGE

- The goal of the adoption stage is to restore single-tree structure of sets  $S$  and  $T$  with roots in the source and the sink.
- At this stage we try to find a new valid parent for each orphan.
- A new parent should belong to the same set,  $S$  or  $T$ , as the orphan.
- A parent should also be connected through a non-saturated edge.
- If there is no qualifying parent we remove the orphan from  $S$  or  $T$  and make it a free node.
- We also declare all its former children as orphans.
- The stage terminates when no orphans are left and, thus, the search tree structures of  $S$  and  $T$  are restored.

# BOYKOV-KOLMOGOROV ALGORITHM

---

## Algorithm 5: Boykov-Kolmogorov Algorithm - Adoption Stage

---

```
1 while  $O \neq \emptyset$  do
2   pick an orphan node  $p \in O$  and remove it from  $O$ 
3   process  $p$ 
```

---

# BOYKOV-KOLMOGOROV ALGORITHM - TERMINATION

- After the adoption stage is completed the algorithm returns to the growth stage.
- The algorithm terminates when the search trees  $S$  and  $T$  can not grow (no active nodes) and the trees are separated by saturated edges. This implies that a maximum flow is achieved.
- The corresponding minimum cut can be determined by  $S$  and  $T$ .

# FURTHER READING / VIEWING

- ① A Survey on Event-Based Video Segmentation (2020)  
<https://ieeexplore.ieee.org/document/9138762>
- ② A Review of Saliency Detection Models and Applications (2012)
- ③ A Survey on Spatio-Temporal Video Segmentation (2020)  
<https://arxiv.org/abs/2005.07330>
- ④ Background subtraction techniques: a review (2018) <https://www.sciencedirect.com/science/article/pii/S0894177723001371>
- ⑤ Adaptive thresholding techniques: A survey (2013)  
[https://www.researchgate.net/publication/220632416\\_A\\_Survey\\_of\\_Thresholding\\_Techniques](https://www.researchgate.net/publication/220632416_A_Survey_of_Thresholding_Techniques)
- ⑥ A survey of region-based image segmentation algorithms (2015)  
<https://www.sciencedirect.com/science/article/pii/S1877050915028574>

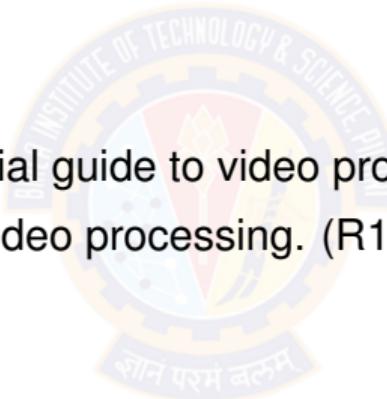
# FURTHER READING / VIEWING

---

- ① A review of real-time scene change detection methods (2012)  
<https://ieeexplore.ieee.org/document/10144064>
- ② A review of multitemporal remote sensing data change detection techniques (2014) <https://core.ac.uk/download/pdf/80147816.pdf>
- ③ Spatiotemporal Change Detection Using Markov Random Fields and a Nonparametric Change Prior (2011)  
[<https://arxiv.org/pdf/2201.11722>
- ④ Fully Convolutional Networks (FCNs) <https://arxiv.org/abs/1411.4038>
- ⑤ Recurrent Convolutional Neural Networks (RCNNs)  
<https://proceedings.mlr.press/v32/pinheiro14.html>
- ⑥ Motion-based CNNs <https://arxiv.org/pdf/1502.04681>

# REFERENCES

- ① Bovik, Alan C. The essential guide to video processing. (T1)
- ② Tekalp, A. Murat. Digital video processing. (R1)



**Thank You!**



## VIDEO ANALYTICS MODULE # 5 MOTION TRACKING IN VIDEO



**BITS Pilani**  
Pilani | Dubai | Goa | Hyderabad

Seetha Parameswaran  
BITS Pilani

---

The instructor is gratefully acknowledging  
the authors who made their course  
materials freely available online.

This deck is prepared by Seetha Parameswaran.

# TABLE OF CONTENTS

---

- 1 MODULE 5 TOPICS
- 2 MOTION TRACKING IN VIDEO
- 3 ALGORITHMS

# MODULE TOPICS....

---

- Motion Tracking in Video
- Region-Based Object Tracking
- Feature-Based Object Tracking
- Template-Based Object Tracking
- Kalman Filters and Extended Kalman Filters

# TABLE OF CONTENTS

---

1 MODULE 5 TOPICS

2 MOTION TRACKING IN VIDEO

3 ALGORITHMS

# MOTION TRACKING IN VIDEO

---

**DEFINITION** Motion tracking in digital video aims at deriving the trajectory over time of moving objects or, the trajectory of the camera.

**OUTPUT OF AN OBJECT TRACKING ALGORITHM** depends on the application and the representation used to describe the object that is being tracked over time. The output can be the contour (silhouette) of the object, the 2D image coordinates of its center of mass, its 3D position in world coordinates, the posture of an articulated object.

# TWO APPROACHES

---

- ① Object detection is performed on each frame of a video sequence and, subsequently, correspondences between objects detected in successive frames are sought. Thus, the trajectory of each object is established.
- ② Combines the detection and correspondence finding steps, the objects positions, and/or orientations in the next frame(s) are predicted, rather than detected, using information derived from the current (or previous) frames.

# APPLICATION DOMAIN

---

**HUMAN–MACHINE INTERFACES** where tracking along with human motion analysis and behavior understanding. Gesture recognition, body and face pose estimation, facial expression analysis and recognition.

**SMART SURVEILLANCE** detect motion and classify the motion (human or nonhuman motion)

**VIRTUAL REALITY AND COMPUTER ANIMATION** Animated avatars, Animating autonomous human-like characters in virtual reality or games (Sophia, Teaching robots)

**MOTION PICTURES INDUSTRY** involving both real and computer-generated actors

# APPLICATION DOMAIN

---

**SPORTS** Training athletes and analyzing their performance.

**MEDICAL DIAGNOSIS** of gait disorders and treatment support can be performed by gait analysis

**TRAFFIC MONITORING** road surveillance regarding traffic flow, automatic path planning and obstacle avoidance.

**CONTENT-BASED QUERYING, INDEXING, AND RETRIEVAL** in multimedia databases. Eg: motion path data obtained by tracking and analyzing the motion of players in sports video footage can be used for content-based indexing and retrieval of such data.

# VIDEO OBJECT TRACKING CATEGORIES

---

Based on devices used

**ACTIVE OBJECT TRACKING TECHNIQUES** involves placing devices (e.g., sensors or transmitters) on the object. Active trackers are **intrusive** and mainly suitable for well-controlled environments.

**PASSIVE OBJECT TRACKING TECHNIQUES** rely on measuring signals naturally emitted by the tracked object, such as light or sound. Passive trackers are preferable, but more difficult to devise to active ones.

# VIDEO OBJECT TRACKING CATEGORIES

Based on Dimensionality of the tracking space

**2D OBJECT TRACKING** aims at recovering the motion in the image plane of the projection of objects.

**3D OBJECT TRACKING** attempts to estimate the actual 3D object movement using the 2D information conveyed by video data captured by one or more cameras.

# VIDEO OBJECT TRACKING CATEGORIES

---

Based on Structure of the object to be tracked

**RIGID AND DEFORMABLE OBJECT TRACKING** refer to estimating the motion of rigid and deformable objects respectively.

**ARTICULATED OBJECT TRACKING** refers to estimating the motion of articulated objects, that is objects composed of rigid parts (links) connected by joints allowing rotational or translational motion in 1, 2, or 3 degrees of freedom. Articulated motion can be defined as piecewise rigid motion, where the rigid parts conform to the rigid motion constraints, but the overall motion is not rigid.

# VIDEO OBJECT TRACKING CATEGORIES

---

Based on Mode of operation - tracking can either be performed online or off-line.

**ONLINE TRACKERS** employ information about the object coming from one or more previous frames to predict its location in the current frame (information from future frames is not available).

**OFFLINE TRACKERS** make use of the entire image sequence, prior and posterior to the frame of interest. This provide better results because more information is available.

# VIDEO OBJECT TRACKING CATEGORIES

---

Based on whether a model (geometric or other) of the object that needs to be tracked (e.g., a human body model when tracking people in video sequences) is used.

**MODEL-FREE** In surveillance applications 3D geometry models of the object to be tracked are hardly necessary, because the parameters of interest involve only the presence and the spatial position of humans.

**MODEL-BASED** In a motion capture application aiming at obtaining data for the animation of a virtual actor, a detailed 3D face and body model is required.

The decision to use a model, its type (2D image template, 3D volumetric or surface geometry model, color distribution model, etc.), as well as its complexity depend on the application.

# VIDEO OBJECT TRACKING CATEGORIES

---

Based on prediction algorithm used

**SINGLE-HYPOTHESIS TRACKERS** Based on the information extracted from previous frames, the state of the object (e.g., its location) is predicted and compared with the state of objects identified in the actual frame in question. To model of the evolution of object in time is required, an excellent framework for prediction is the **Kalman filter**.

**MULTIPLE-HYPOTHESIS TRACKERS** Prediction schemes that are capable of keeping track of multiple hypotheses, such as Condensation algorithm.

# ASSUMPTIONS FOR VIDEO OBJECT TRACKING

---

- Related to the motion of the camera or subjects  
(fixed camera, single-person scenes, occlusion-free scenes, known motion models, e.g., front-to-parallel movement with respect to the camera, etc.)
- Appearance of the environment (constant lighting conditions, uniform or static background, etc.) or the subject(s) (tight clothing, tracking of a specific type of objects, e.g., cars, etc.).

Assumptions can provide constraints that can greatly facilitate the solution of the tracking problem.

# VIDEO OBJECT TRACKING ALGORITHMS

- Tracking algorithms can be characterized on the basis of whether they focus on
  - ▶ tracking specific objects, such as car tracking
  - ▶ tracking of human body parts (face, hand, etc.)
  - ▶ number of views available (single-view, multiple-view, and omnidirectional view tracking techniques)
  - ▶ the state of the camera (moving vs. stationary)
  - ▶ tracking environment (indoors vs. outdoors)
  - ▶ number of tracked objects (single object, multiple objects, groups of objects)

# INITIALIZATION IN VIDEO OBJECT TRACKING

---

- Initialization can be performed off-line or online and aims at recovering information about the camera and/or the scene and/or the object to be tracked.
- The scope of the initialization step differs between different tracking algorithms.
- In a 2D contour-based object tracking method, the initialization could be an object detection step, aiming at finding the contour of the object in the first frame of the video sequence.
- If the method is a feature point-based one, the initialization should provide the 2D coordinates of all the object feature points that are to be tracked.

# OCCLUSIONS IN VIDEO OBJECT TRACKING

- **Occlusion is usually distinguished in partial and total occlusion, where the object of interest is partially or totally occluded by another object (fixed or moving).**
- In **Self-occlusion**, parts of the object are occluded by the object itself. (e.g., limb occlusion when walking humans are being tracked or face occlusion caused by hand gestures during a conversation).
- Ways to handle occlusion
  - ▶ **Reinitialization:** The initialization phase (e.g., object detection) is applied periodically or at certain instances to detect (and then continue to track) objects that might have been occluded.
  - ▶ Object position prediction schemes could be employed for the whole duration of the occlusion.

# TABLE OF CONTENTS

---

1 MODULE 5 TOPICS

2 MOTION TRACKING IN VIDEO

3 ALGORITHMS

# 2D RIGID OBJECT TRACKING

---

- 2D rigid object tracking tries to determine the motion of the projection of one or more rigid objects on the image plane.
- This motion is induced by the relative motion between the camera and the observed scene.
- A basic assumption behind 2D rigid motion tracking is that there is only one, rigid, relative motion between the camera and the observed scene
- Eg: moving car

# METHODS FOR 2D RIGID OBJECT TRACKING

---

Classification based on the tools that are used in tracking

- Region-based methods
- Contour-based methods
- Feature point-based methods
- Template-based methods

# REGION BASED OBJECT TRACKING

---

- An image region can be defined as a set of pixels having homogeneous characteristics.
- Region can be derived by image segmentation, which can be based on distinctive object features (e.g., color, edges) and/or on the motion observed in the frames of a video sequence.
- A region would be the image area covered by the projection of the object of interest onto the image plane.
- A region can be the bounding box or the convex hull of the projected object under examination.

# CLASSIC TECHNIQUES FOR REGION BASED OBJECT TRACKING

---

- Color segmentation and tracking. Illumination invariance can be achieved by converting to HS color space.
- Color-based object tracking using chroma-keying
  - ▶ Background is single-colored (blue) and the object colors are very different from the background.
  - ▶ Simple color thresholding can be employed to separate the object from the background and track it.
- Color-related region-based object tracking using color histograms
- Background subtraction

# FEATURE BASED OBJECT TRACKING

---

- Feature-based object tracking involves tracking distinctive features or keypoints in video frames across time.
- The features serve as reference points, and their motion is analyzed to track the object of interest.
- Classic techniques for feature-based object tracking include various methods that leverage these distinctive features.

# CLASSIC TECHNIQUES FOR FEATURE BASED OBJECT TRACKING

- KLT Tracker (Kanade-Lucas-Tomasi)
- Lucas-Kanade Optical Flow
- SIFT (Scale-Invariant Feature Transform)
- SURF (Speeded Up Robust Features)
- ORB (Oriented FAST and Rotated BRIEF)
- Gabor wavelets based feature selection

cv2.sift

feature extraction  
tech

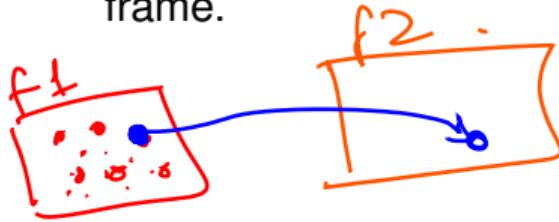
# DEEP LEARNING TECHNIQUES FOR FEATURE BASED OBJECT TRACKING

- DeepSORT (Deep Simple Online and Realtime Tracking)
- Deep Matching and Verification
- Deep Feature Networks for Tracking
- Siamese Networks for Tracking
- Deep Regression Networks for Object Tracking

# TEMPLATE BASED OBJECT TRACKING

$f_1 \rightarrow SIFT$  descriptor  $\rightarrow$  match  $\rightarrow$  trajectory.

- Template-based object tracking involves using a reference template of the target object to track its presence and location in subsequent frames of a video sequence.
- Template is a model of the image region or object to be tracked.
- Classic techniques for template-based object tracking typically rely on measuring the similarity between the template and regions in the current frame.



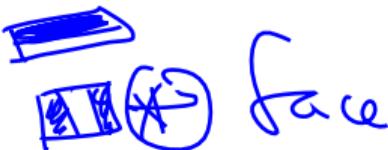
SIFT ( $f_1$ )  
SIFT ( $f_2$ ) align/match

# TEMPLATE BASED OBJECT TRACKING – STEPS

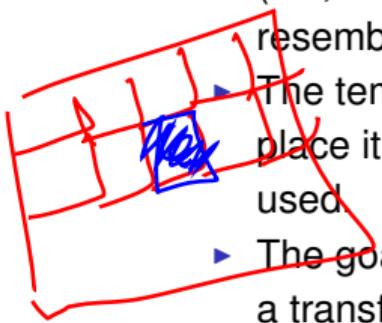
→ set template.

- First / Initialization step is to select the template that will be used.
  - ▶ Templates can be acquired prior to tracking in a number of ways. First, a template that is specific for a particular instance of a class of objects can be created.
  - ▶ A template can be created off-line by employing statistical methods. For example, in a face tracking application, average human face can be used.

video



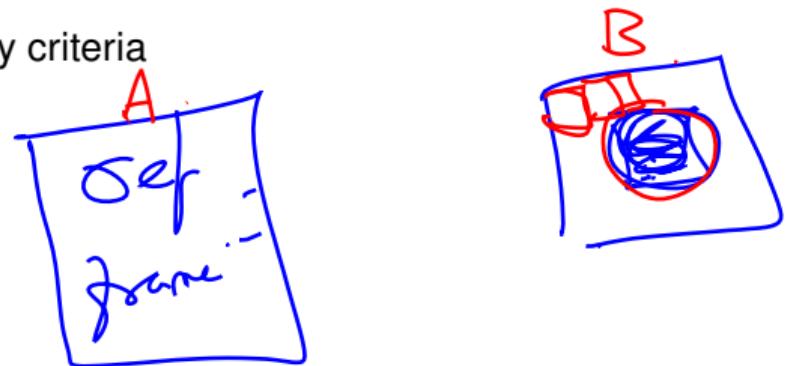
# TEMPLATE BASED OBJECT TRACKING – STEPS

- Template matching
  - ▶ Template matching can be defined as the process of searching the target image (i.e., the current frame of the video sequence) to determine the image region that resembles the template, based on a similarity or distance measure.
  - ▶ The template region should undergo a geometrical transformation that would place it onto the target image in such a way as to minimize the distance measure used.
  - ▶ The goal of a template matching algorithm is to estimate the parameters of such a transformation.
  - ▶ Background subtraction can be employed to determine image regions where motion activity appears and limit the search in these regions.
  - ▶ Prediction schemes like Kalman filters can be used to estimate the location of the object being tracked in the next frame and use it as the center of a limited-size search region.

# TEMPLATE BASED OBJECT TRACKING – STEPS

- Similarity / distance metrics used in the template matching step
  - ▶ Minimise one of the following distance metrics
    - ★ Hamming distance
    - ★ Sum of absolute differences (SAD)
    - ★ Sum of squared differences (SSD)
  - ▶ Maximize one of the following similarity criteria
    - ★ Normalized correlation
    - ★ Joint entropy
    - ★ Mutual information
    - ★ Maximum likelihood

Correlation



## CLASSIC TECHNIQUES FOR TEMPLATE BASED OBJECT TRACKING

---

NORMALIZED CROSS-CORRELATION (NCC) measures the similarity between the template and candidate regions in the current frame using cross-correlation. It is a widely used method for template matching.

SUM OF ABSOLUTE DIFFERENCES (SAD) measures the absolute pixel-wise differences between the template and candidate regions. It provides a measure of dissimilarity.

MEAN-SHIFT TRACKING is an iterative algorithm that shifts the position of a target to maximize a similarity measure, often using color histograms as templates. It is effective for tracking objects with distinct color patterns.

# CLASSIC TECHNIQUES FOR TEMPLATE BASED OBJECT TRACKING

**TEMPLATE MATCHING WITH NORMALIZED GRADIENT CORRELATION (NGC)** measures the normalized gradient correlation between the template and candidate regions. It is robust to changes in illumination and contrast.

**TEMPLATE-BASED CORRELATION FILTERS** learn a filter that represents the appearance of the object in the frequency domain. These filters are convolved with the image to find the target location.

**DISCRIMINATIVE CORRELATION FILTER (DCF) TRACKERS** learn a discriminative correlation filter to track the object in the Fourier domain.

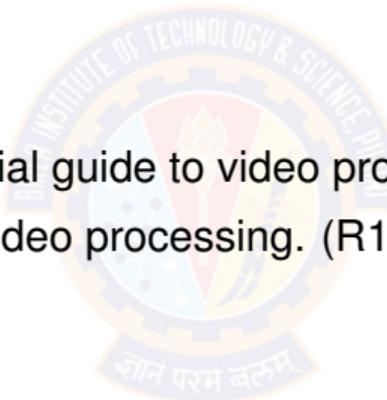
Yolo<sup>v8</sup> → ImageNet / Coco .

# DEEP LEARNING TECHNIQUES FOR TEMPLATE BASED OBJECT TRACKING

- Siamese Networks for Tracking
- Correlation Filter-Based Trackers with Deep Features
- MDNet (Multi-Domain Network)
- SiamFC (Siam Fully Convolutional)
- ATOM (Accurate Tracking by Overlap Maximization)
- CFNet (Correlation Filter Network)

# REFERENCES

- ① Bovik, Alan C. The essential guide to video processing. (T1)
- ② Tekalp, A. Murat. Digital video processing. (R1)



**Thank You!**



**BITS Pilani**

Pilani | Dubai | Goa | Hyderabad

# VIDEO ANALYTICS

## MODULE # 6 VIDEO INDEXING, SUMMARIZATION, BROWSING, AND RETRIEVAL

Seetha Parameswaran

BITS Pilani

---

The instructor is gratefully acknowledging  
the authors who made their course  
materials freely available online.

This deck is prepared by Seetha Parameswaran.

# TABLE OF CONTENTS

---

- 1 MODULE 6 TOPICS
- 2 MULTIMODAL ANALYSIS
- 3 IMAGE AND VIDEO FEATURES
- 4 VIDEO ANALYSIS
- 5 VIDEO REPRESENTATION
- 6 VIDEO BROWSING
- 7 VIDEO RETRIEVAL
- 8 VIDEO INDEXING
- 9 VIDEO SUMMARIZATION
- 10 VIDEO SKIMMING
- 11 VIDEO SYNOPSIS

# MODULE TOPICS....

---

- Image and Video Features
- Video Analysis
- Video Representation
- Video Browsing
- Video Retrieval

# TABLE OF CONTENTS

---

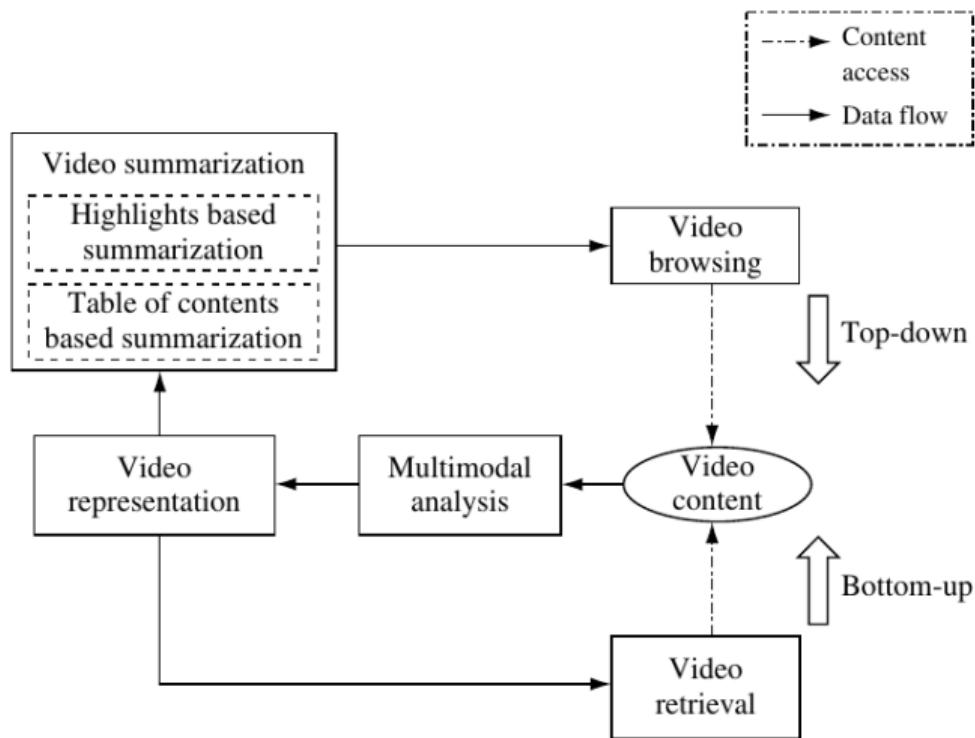
- 1 MODULE 6 TOPICS
- 2 MULTIMODAL ANALYSIS
- 3 IMAGE AND VIDEO FEATURES
- 4 VIDEO ANALYSIS
- 5 VIDEO REPRESENTATION
- 6 VIDEO BROWSING
- 7 VIDEO RETRIEVAL
- 8 VIDEO INDEXING
- 9 VIDEO SUMMARIZATION
- 10 VIDEO SKIMMING
- 11 VIDEO SYNOPSIS

# MULTIMODAL ANALYSIS

---

- Multimodal analysis deals with the signal processing part of the video system, including shot boundary detection, key-frame extraction, key object detection, audio analysis, closed caption analysis, etc.
- Involves
  - ▶ video representation
  - ▶ video summarization
  - ▶ video browsing
  - ▶ video retrieval
- The first three bases focus on metadata generation and organization.
- The last two focus on metadata consumption.

# MULTIMODAL ANALYSIS



# MULTIMODAL ANALYSIS

---

**VIDEO REPRESENTATION** is concerned with the structure of the video.

**VIDEO SUMMARIZATION** deals with how to use the representation structure to provide the viewers top-down access using the summary for video browsing.

**VIDEO RETRIEVAL** is concerned with retrieving specific video objects. It is useful when we know exactly what we are looking for in the content.

Bottom-up approach

**VIDEO BROWSING** is useful when we need to get an essence of the content.

Top-down approach

**VIDEO INDEXING** is the process of creating indexes or metadata for videos to enable efficient and effective retrieval, search, and analysis of video content.

# SCRIPTED VIDEO

---

- **Scripted content is a video that is carefully produced according to a script or plan that is later edited, compiled, and distributed for consumption.**
- Examples: News videos, dramas, and movies
- Carefully structured as a sequence of semantic units.
- The essence can be obtained by enabling a traversal through representative items from these semantic units.
- **Table of Contents (ToC)-based video browsing** caters to summarization of scripted content.
- Example: A news video composed of a sequence of stories can be summarized/browsed using a key-frame representation for each of the shots in a story.

# SCRIPTED VIDEO



10 key moments in this video ^

From 00:27

2 tbsp Butter

From 00:32

1 stick Cinnamon

From 01:15

1 tbsp Garlic,  
chopped

From 01:27

1/3 tsp Turmeric

From 01:

1/2 tb<sup>s</sup> Powder

# UNSCRIPTED VIDEO

---

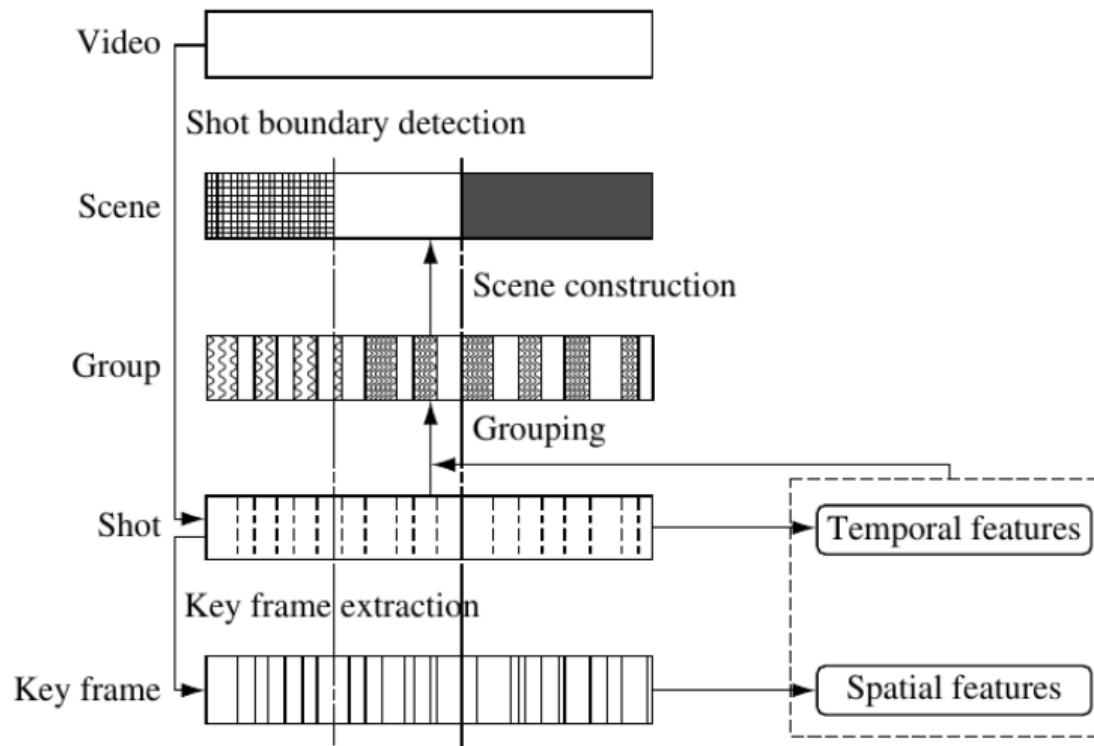
- **Video content that is not scripted is referred to as unscripted content.**
- Examples: Surveillance video and sports video, the events happen spontaneously.
- Summarization requires a **highlights extraction framework** that only captures remarkable events that constitute the summary.

# SCRIPTED VS UNSCRIPTED VIDEO

---

- **Scripted video data** can be structured into a hierarchy consisting of five levels: video, scene, group, shot, and key frame, which increase in granularity from top to bottom.
- **Unscripted video data** can be structured into a hierarchy of four levels: play/break, audiovisual markers, highlight candidates, highlight groups, which increase in semantic level from bottom to top .

# SCRIPTED VIDEO



# TERMINOLOGY FOR SCRIPTED VIDEO

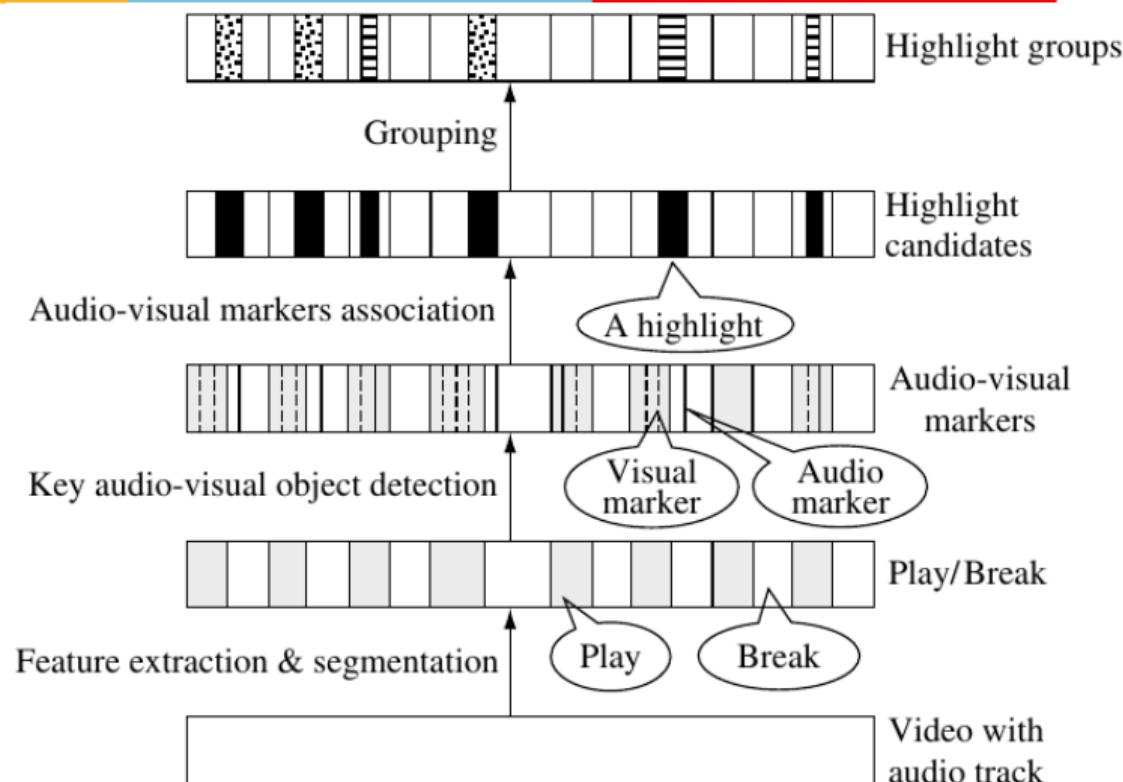
**VIDEO SCENE** is defined as a collection of semantically related and temporally adjacent shots depicting and conveying a high-level concept or story. Shots are marked by physical boundaries, scenes are marked by semantic boundaries.

**VIDEO GROUP** is an intermediate entity between the physical shots and semantic scenes and serves as the bridge between the two. Examples of groups are temporally adjacent shots or visually similar shots.

**VIDEO SHOT** is a consecutive sequence of frames recorded from a single camera. It is the building block of video streams.

**KEY FRAME** is the frame that represents the salient visual content of a shot. Depending on the complexity of the content of the shot, one or more key frames can be extracted.

# UNSCRIPTED VIDEO



# TERMINOLOGY FOR UNSCRIPTED VIDEO

---

**PLAY AND BREAK** is the first level of semantic segmentation in sports video and surveillance video.

- In sports video (e.g., soccer, baseball, golf), a game is in **play** when the ball is in the field and the game is going on.
- **Break**, or out of play, is the complement set. Eg: Whenever the ball has completely crossed the goal line or touch line, whether on the ground or in the air or the game has been halted by the referee.
- In surveillance video, a play is a period in which there is some activity in the scene.

# TERMINOLOGY FOR UNSCRIPTED VIDEO

---

**AUDIO MARKER** is a contiguous sequence of audio frames representing a key audio class that is indicative of the events of interest in the video.

Eg: audio marker for sports video can be the audience reaction sound (cheering and applause) or commentator's excited speech.

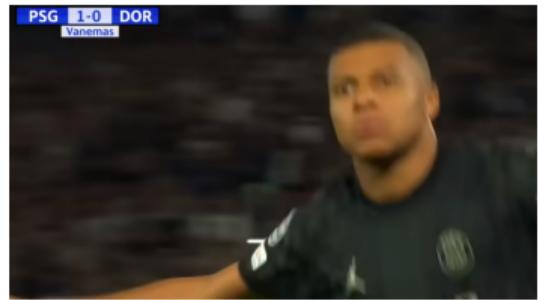
**VIDEO MARKER** is a contiguous sequence of video frames containing a key video object that is indicative of the events of interest in the video.

Eg: video marker for baseball videos is the video segment containing the squatting catcher at the beginning of every pitch.

**HIGHLIGHT CANDIDATE** is a video segment that is likely to be remarkable and can be identified using the video and audio markers.

**HIGHLIGHT GROUP** is a cluster of highlight candidates.

# UNSCRIPTED VIDEO - PLAY AND BREAK



# TABLE OF CONTENTS

---

- 1 MODULE 6 TOPICS
- 2 MULTIMODAL ANALYSIS
- 3 IMAGE AND VIDEO FEATURES
- 4 VIDEO ANALYSIS
- 5 VIDEO REPRESENTATION
- 6 VIDEO BROWSING
- 7 VIDEO RETRIEVAL
- 8 VIDEO INDEXING
- 9 VIDEO SUMMARIZATION
- 10 VIDEO SKIMMING
- 11 VIDEO SYNOPSIS

# FEATURES

---

- A feature is defined as a descriptive parameter that is extracted from an image or video stream.
- Features may be used to interpret visual content or as a measure for similarity in image and video databases.

**STATISTICAL FEATURES** are extracted from an image or video sequence without regard to content.

**COMPRESSED-DOMAIN FEATURES** are extracted from a compressed image or video stream without regard to content.

**CONTENT-BASED FEATURES** are features that are derived for the purpose of describing the actual content in an image or video stream.

# STATISTICAL FEATURES

---

- Features are extracted directly from image pixels without regard to the content.
- Eg: scene changes, Image Difference, motion flow, and video structure in the image domain and sound discrimination in the audio domain.

# STATISTICAL FEATURES

---

- Image Difference
  - ▶ Absolute difference
  - ▶ Histogram difference
- Segmentation
  - ▶ Scene change
- Motion Analysis
  - ▶ Camera motion
  - ▶ Object motion
- Others
  - ▶ Texture
  - ▶ Shape and features
  - ▶ Audio
  - ▶ Hierarchical structure

# IMAGE DIFFERENCE

---

- A difference measure between images serves as a feature to measure similarity.
- Two methods for image difference:
  - ▶ absolute difference
  - ▶ histogram difference.

# ABSOLUTE DIFFERENCE

---

- The image difference of two images is defined as the sum of the absolute difference at each pixel.

$I_t$  : first image

$I_{t-T}$  : second image at temporal distance  $T$

$M$  : resolution or number of pixels in the image

Absolute difference     $D(t) = \sum_{i=0}^M |I_{t-T}(i) - I_t(i)|$

- requires less computation
- noisy and extremely sensitive to camera motion and image degradation.

# ABSOLUTE DIFFERENCE ON SUBREGIONS

---

- the sum of the absolute difference at each pixel applied over a subregion

$I_t$  : first image

$I_{t-T}$  : second image at temporal distance  $T$

$S$  : starting position for a particular region

$n$  : represents the number of subregions

$$\text{Absolute difference } D_d(t) = \sum_{j=S}^{\frac{H}{n}} \sum_{i=S}^{\frac{W}{n}} |I_{t-T}(i, j) - I_t(i, j)|$$

- less noisy and may be used as a more reliable parameter for image difference.

# HISTOGRAM DIFFERENCE

- A histogram difference detects significant changes in the weighted color histogram of two images.

$H_t$  : histogram of first image

$H_{t-1}$  : histogram of second image

$N$  : number of bins in the histogram, typically 256.

Histogram difference     $D_H(t) = \sum_{v=0}^N | H_{t-1}(v) - H_t(v) |$

- $D_H(t)$  will rise during scene changes, image noise, and camera or object motion.
- more robust measure for image correspondence.
- less sensitive to subtle motion and noise.

# HISTOGRAM DIFFERENCE FOR RGB

- If the histogram is actually three separate sets for RGB, the difference may simply be summed.

$$D_{H-R}(t) = \sum_{v=0}^N | H_{R(t-1)}(v) - H_{R(t)}(v) |$$

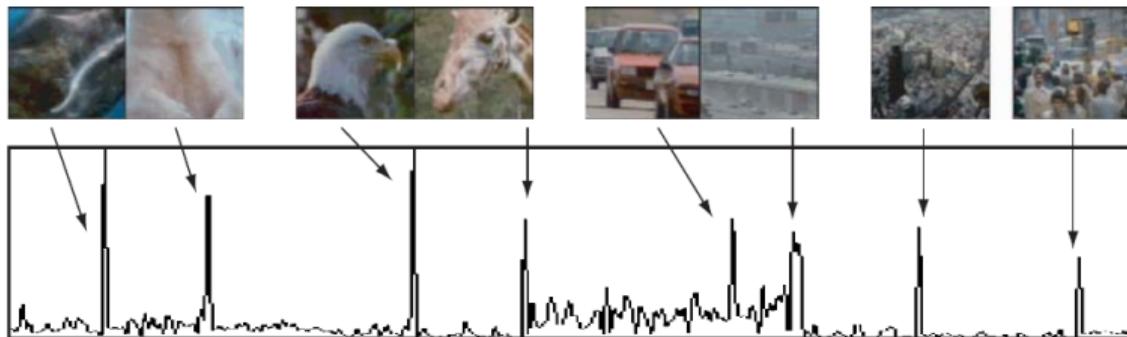
$$D_{H-G}(t) = \sum_{v=0}^N | H_{G(t-1)}(v) - H_{G(t)}(v) |$$

$$D_{H-B}(t) = \sum_{v=0}^N | H_{B(t-1)}(v) - H_{B(t)}(v) |$$

$$D_{H-RGB}(t) = \frac{D_{H-R}(t) + D_{H-G}(t) + D_{H-B}(t)}{3}$$

# SCENE CHANGE

- The difference measures, absolute  $D(t)$  and histogram  $D_H(t)$ , may be used to determine the occurrence of a scene change.
- The most fundamental scene change is the video cut.
- For most cuts, the difference between image frames is very distinct.
- A histogram difference is less sensitive to subtle motion and is an effective measure for detecting scene cuts and gradual transitions. This technique is simple and yet robust enough to maintain high levels of accuracy.



# SCENE CHANGE CATEGORIES

---

**FAST CUT** A sequence of video cuts, each very short in duration, represents a fast cut. This technique heightens the sense of action or excitement. To detect a fast cut, look for a sequence of scene changes that are in close proximity.

**DISTANCE CUT** occurs when the camera cuts from one perspective of a scene to another some distance away. This shift in distance usually appears as a cut from a wide shot to a close-up shot or vice versa.

**INTERCUTTING** When scenes change back and forth from one subject to another, we say the subjects are intercut. This concept is similar to the distance cut, but the images are separate and not inclusive of the same scenes.

# SCENE CHANGE CATEGORIES

---

**DISSOLVES AND FADES** A fade occurs when a scene changes over time from its original color scheme to a black background. This procedure is commonly used as a transition from one topic to another.

Dissolve occurs when a scene changes over time and morphs into a separate scene. This transition is less intrusive and is used when subtle change is needed.

**WIPES AND BLENDS** A wipe usually consists of the last frame of a scene being folded like a page in a book.

A blend may be shown as pieces of two separate scenes combined in some artistic manner. A wipe is often used to convey a change in time or location.

# MOTION ANALYSIS

---

- Statistics from optical flow may also be used to detect scene changes.
- Optical flow is computed from one frame to the next.
- When the motion vectors for a frame are randomly distributed without coherency, this may suggest the presence of a scene change.
- The quality of the camera motion estimate is used to segment video. An analysis of optical flow quality may be used to avoid false detection of scene changes.

# CAMERA MOTION

- Camera motion is characterized by flow throughout the entire image.
- An affine model is used to approximate the flow patterns consistent with all types of camera motion.

$$u(x_i, y_i) = ax_i + by_i + c \quad v(x_i, y_i) = dx_i + ey_i + f$$

- Affine parameters  $a, b, c, d, e$ , and  $f$  are calculated by minimizing the least squares error of the motion vectors.
- Compute average flow  $\bar{v}$  and  $\bar{u}$

$$\bar{u} = \sum_{i=0}^N ax_i + by_i + c \quad \bar{v} = \sum_{i=0}^N dx_i + ey_i + f$$

- Using the affine flow parameters and average flow, we classify the flow pattern.

# CAMERA MOTION

---

- Check if there is the convergence or divergence point  $(x_0, y_0)$  where  $u(x_i, y_i) = 0$  and  $v(x_i, y_i) = 0$ .
- If  $\begin{vmatrix} a & b \\ d & e \end{vmatrix} = 0$ 
  - ▶  $(x_0, y_0)$  is located inside the image, it is focus of expansion.
  - ▶ If  $\bar{v}$  and  $\bar{u}$  are large, then this is the focus of the flow and camera is zooming.
  - ▶ If  $(x_0, y_0)$  is outside the image, and or are large, then the camera is panning in the direction of the dominant vector.
- If  $\begin{vmatrix} a & b \\ d & e \end{vmatrix} \approx 0$ 
  - ▶  $(x_0, y_0)$  does not exist and the camera is panning or static.
  - ▶ If  $\bar{v}$  and  $\bar{u}$  are large, the motion is panning in the direction of the dominant vector.
- Otherwise, there is no significant motion and the flow is static.

# CAMERA MOTION

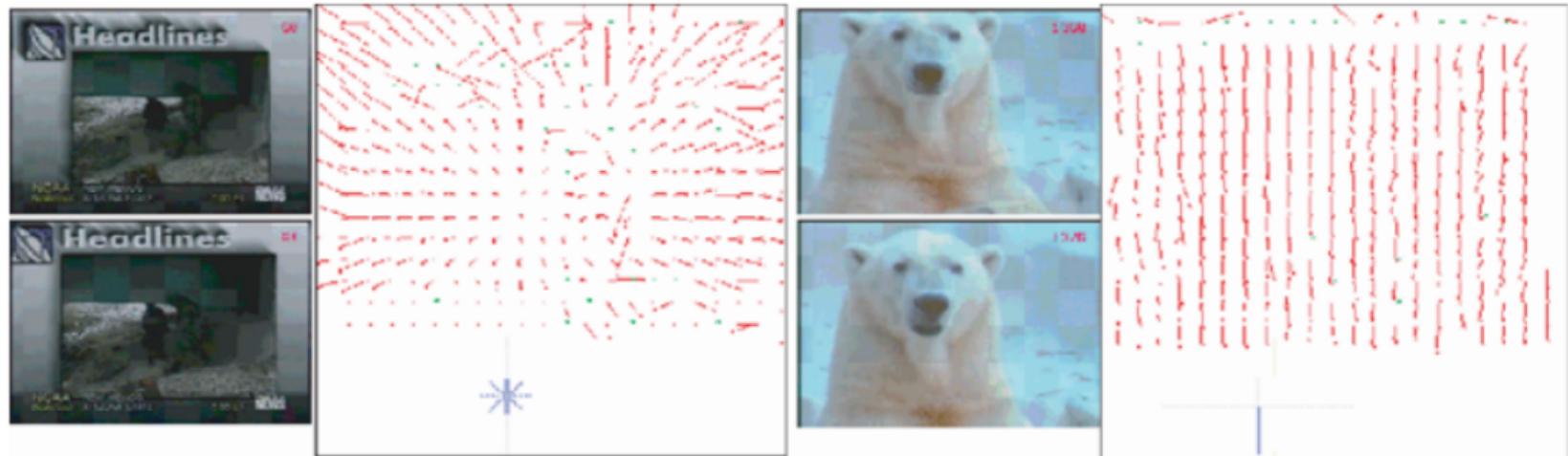
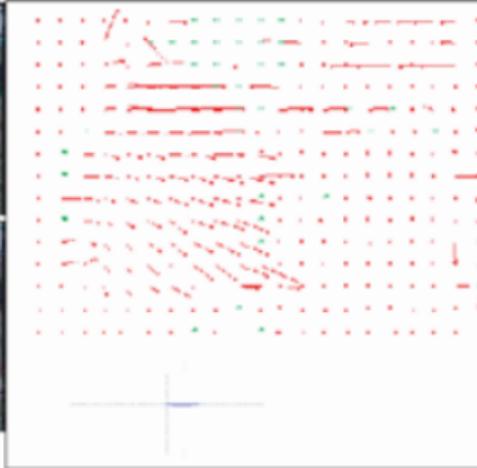
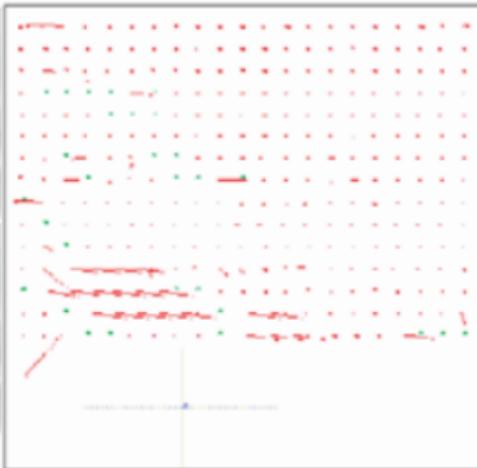


FIGURE: Optical flow fields for a pan and zoom

# OBJECT MOTION



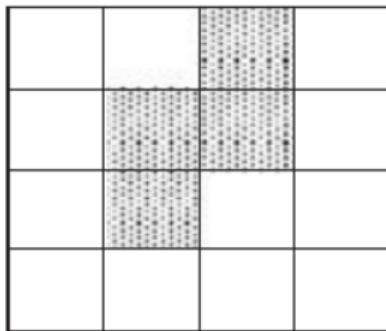
# OBJECT MOTION

- Object motion exhibits flow fields in specific regions of an image.
- The flow field is partitioned into a grid. A motion grid should consist of at least a  $4 \times 4$  array of motion vectors.
- If the average velocity for the vectors in a particular grid is high (typically  $> 2.5$  pixels), then that grid is designated as containing motion.
- $G_m(i)$  represents the status of motion grid at position  $i$  and  $M$  represents the number of neighbors.

$$G_m(i) = \begin{cases} 0 & G_m(i-1) = 0, G_m(i+1) = 0 \dots M \\ 1 & otherwise \end{cases}$$

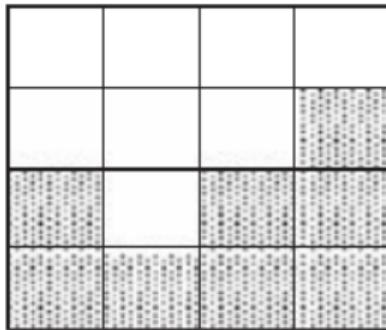
- If  $G_m$  is high (typically  $G_m > 7$ ), the flow is camera motion.
- If  $G_m$  is not high, but greater than some small value (typically two grids), the motion is isolated in a small region of the image and the flow is object motion.

# CAMERA AND OBJECT MOTION



Object motion

-  Static grid
-  Motion grid



Camera motion

# AUDIO FEATURES

---

- Loud sounds, silence, and single frequency sound markers may be detected analytically without actual knowledge of the audio content.
- **Loud sounds** imply a heightened state of emotion in video, and are easily detected by measuring a number of audio attributes, such as signal amplitude or power.
- **Silent video** signify an area of less importance.
- **Single frequency sound markers**, typically a 1000 Hz. tone, can be used to mark a particular point in the beginning of a video. This tone may be detected to determine the exact point in which a video will start.

# HIERARCHICAL VIDEO STRUCTURE

---

- Video is produced with a particular format and structure.
- News segments are typically 30 min in duration and follow a rigid pattern from day to day.
- Commercials are of fixed duration, making detection less difficult.
- In most broadcast video, a **black frame** is shown between a transition of two segments.
- A black frame or any single intensity image may be detected by summing the total number of pixels in a particular color space.
- By detecting the location of black frames in video, a hierarchical structure may be created to determine transitions between segments.
- In news broadcast, black frame usually occurs between a story and a commercial.

# COMPRESSED-DOMAIN FEATURES

---

- Problem: Given large amounts of compressed materials (e.g., Moving Picture Experts Group [MPEG]), how to index and retrieve the content rapidly?

# COMPRESSED-DOMAIN FEATURES

---

## Approach 1

- Decompress all the data and utilize the statistical features.
- Decompression implies extra computation.
- Process of decompression and recompression, often referred to as “reencoding,” results in further loss of image quality.
- The size of decompressed data is much larger than the compressed form.  
Most hardware and CPU cycles are needed to process and store the data.

# COMPRESSED-DOMAIN FEATURES

## Approach 2

- Extract features directly from the compressed data. These features can be used for indexing and retrieval.
- ① Motion vectors that are available in all video data compressed using standards such as H.261/H.263 and MPEG-1/2 can be used to detect scene changes and effects like dissolve, fade in, and fade out.
- ② Segmentation of motion vectors into regions of similar vectors can be used to detect moving objects and track their positions. They can also be used to derive camera motions such as zoom and pan.
- ③ DCT coefficients form a natural representation of texture in the original image. DCT coefficients can also be used to match images and to detect scene changes.

# COMPRESSED-DOMAIN FEATURES

## Approach 3

- For a P-frame, a large percentage of intrablocks implies a lot of new information for the current frame that cannot be predicted from the previous frame. Therefore, such a P-frame indicates the beginning of a new scene right after a scene change.
- For a B-frame, the ratio between the number of forward-predicted blocks and the number of backward-predicted blocks can be used to conclude whether the scene change happens before this B-frame or after this B-frame.
  - If the number of forward-predicted blocks is larger than the number of backward-predicted blocks, then the scene change happens after the B-frame.
  - If the number of forward-predicted blocks is smaller than the number of backward-predicted blocks, then scene change happens before the B-frame.

# CONTENT-BASED FEATURES

---

- Both Statistical and Compression based features provide understanding of the structure of the video.
- But these features in no way estimate the actual image or video content.
- To approximate the actual content of a video, features required are include
  - ▶ Object detection
  - ▶ Audio and language
  - ▶ Rule based features
  - ▶ Embedded video features

# OBJECT DETECTION

---

- Identify significant objects that appear in the video frames
  - ▶ Human subjects
    - ★ Used in analysis of news footage and sports.
    - ★ An anchorperson will often appear at the start and end of a news broadcast, which is useful for detecting segment boundaries.
    - ★ In sports, anchorpersons will often appear between plays or commercials.
- Captions
- Graphics

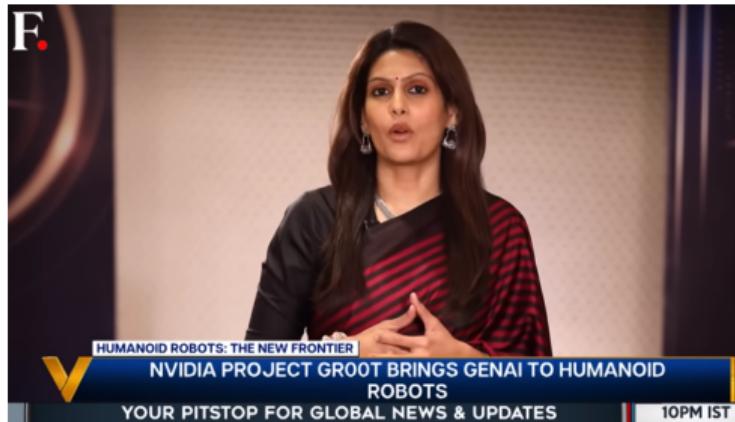
# OBJECT DETECTION

---

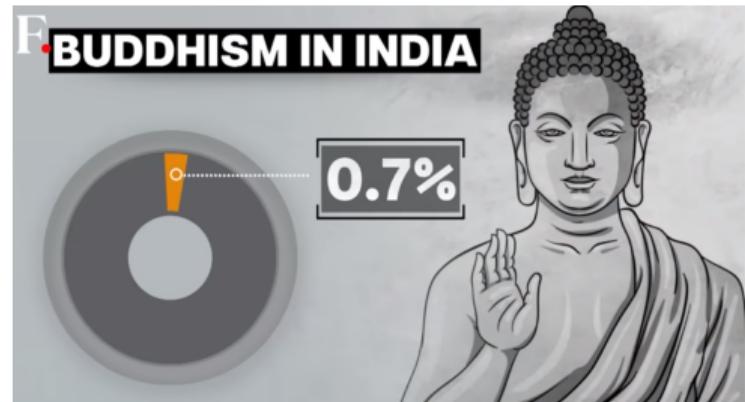
- Captions

- ▶ Text in video and not closed captions
- ▶ A typical text region can be characterized as a horizontal rectangular structure of clustered sharp edges because characters usually form regions of high contrast against the background.
- ▶ Captions are high-contrast text such as the black and white chyron commonly found in news video.
- ▶ Consistent detection of the same text region over a period of time ensures better text extraction.
- ▶ Process of text detection in horizontal titles and captions
  - ★ Use OCR for text capture and extraction

# CAPTIONS AND GRAPHICS



# CAPTIONS AND GRAPHICS



# OBJECT DETECTION

---

- Captions Examples

- ▶ In news, captions of the broadcasting company are shown at low opacity as a watermark in a corner without obstructing the actual video.
- ▶ A ticker tape is widely used in news to display information such as the weather, sports scores, or the stock market.
- ▶ In sports, a score of the play is shown in a corner or border at low opacity.
- ▶ Captions are used in documentaries to describe a location, person of interest, title, or event.
- ▶ Commercials use some form of captions to describe a product or institution, because their time is limited to only a few minutes.
- ▶ In a film, captions may be used to convey a change in time or location.

# OBJECT DETECTION

---

- Graphic
  - ▶ A graphic is usually a recognizable symbol, which may contain text.
  - ▶ Graphic illustrations or symbolic logos are used to represent many institutions, locations, and organizations.
  - ▶ A logo representing the subject is often placed in a corner next to an anchorperson during dialogue.
  - ▶ Detection of graphics may serve as a scene break.
    - ★ Use Histogram difference analysis of isolated image regions instead of the entire image

# AUDIO AND LANGUAGE

---

- Words specific to the actual content or “key words” can be extracted using natural language processing techniques.
- Key words may be used to reduce indexing and provide abstraction for video sequences.
- Audio segmentation can distinguish spoken words from music, noise, and silence.
- Speech recognition is necessary to align and translate the words into text.

# RULE-BASED FEATURES

- Rules describe a particular type of video scene. This creates an additional set of content-based features.
- Introduction Scenes
  - ▶ A scene contains a proper name, and a large human face is detected in the scenes that follow is an introduction scene.
- Adjacent Similar Scenes
  - ▶ The color histogram difference measure gives a simple routine for detecting similarity between scenes.
  - ▶ Scenes between successive shots of a human face usually imply illustration of the subject.
- Short Successive Scenes
  - ▶ Short successive shots often introduce an important topic.
  - ▶ By measuring the duration of each scene these regions can be detected and identified as short successive sequences.

# EMBEDDED VIDEO FEATURES

---

- Video production manuals provide insight into the procedures used during video editing and creation.
- A producer will often create production notes that describe in detail action and scenery of a video, scene by scene.
- Timecode and geospatial (GPS/GIS) data are useful in indexing precise segments in video or a particular location in spatial coordinates.

# EMBEDDED VIDEO FEATURES

---

- Structural information is a useful tool for indexing video.
  - ▶ Type of video being used (documentaries, news footage, movies, and sports) and its duration.
  - ▶ The exact locations of the anchorperson can be used to delineate story breaks.
  - ▶ In documentaries, a person of expertise will appear at various points throughout the story when topical changes take place.
  - ▶ In documentaries, the scenes prior to the introduction of a person usually describe their accomplishments and often precede scenes with large views of the person's face.

# TABLE OF CONTENTS

---

- 1 MODULE 6 TOPICS
- 2 MULTIMODAL ANALYSIS
- 3 IMAGE AND VIDEO FEATURES
- 4 VIDEO ANALYSIS
- 5 VIDEO REPRESENTATION
- 6 VIDEO BROWSING
- 7 VIDEO RETRIEVAL
- 8 VIDEO INDEXING
- 9 VIDEO SUMMARIZATION
- 10 VIDEO SKIMMING
- 11 VIDEO SYNOPSIS

# VIDEO ANALYSIS

---

- Scripted Video
  - ▶ Shot boundary detection
  - ▶ Key-frame extraction
- Unscripted Video
  - ▶ Play/break segmentation
  - ▶ Audio marker detection
  - ▶ Visual marker detection

# SHOT BOUNDARY DETECTION

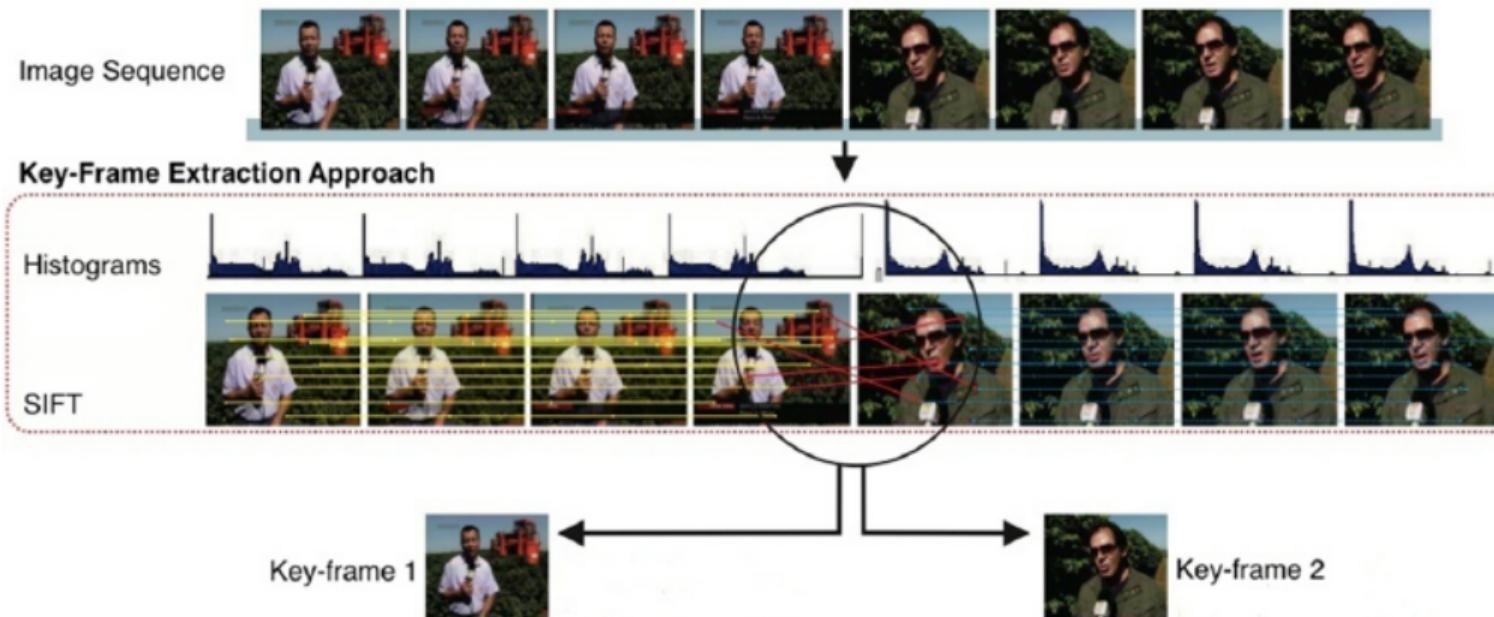
- Decompose the video clip into shots.
- Automatic shot boundary detection techniques include
  - ▶ Pixel-based
    - ★ Use pixel-wise intensity difference to mark shot boundaries.
  - ▶ Statistics-based
    - ★ Use intensity statistics (mean and standard deviation) as shot boundary detection measures.
  - ▶ Transform-based
    - ★ Use the compressed DCT coefficients.
    - ★ Use of motion vectors.
  - ▶ Feature-based
    - ★ Edge features are first extracted from each frame. Shot boundaries are then detected by finding sudden edge changes.
  - ▶ Histogram-based (best)
    - ★ Use histograms of the pixel intensities as the measure.

# KEY-FRAME EXTRACTION

---

- Shot boundary detection
  - ▶ Detect shot boundaries first.
  - ▶ Extract the first and last frames of each shot as the key frames.
- Edge-Based Methods
  - ▶ Keyframes are selected based on edge density, edge magnitude, or edge distribution.
  - ▶ Frames with high edge content or sudden changes in edge structure may be chosen as keyframes.
- Motion-Based Methods
  - ▶ Analyze motion information, such as optical flow or motion vectors, to identify frames with significant motion or scene changes.
  - ▶ Frames with high motion activity or abrupt motion changes may be chosen as keyframes.

# KEY-FRAME EXTRACTION



# KEY-FRAME EXTRACTION USING FRAME DIFFERENCE

(a) Key frame extraction using Consecutive frame difference  
Sample Video Sequence 1 (Solo)



Sample Video Sequence 2 (Darth Vader)



# KEY-FRAME EXTRACTION USING ENTROPY

(c) Key frame extraction using entropy difference  
Sample Video Sequence 1 (Solo)



Sample Video Sequence 2(Darth Vader)

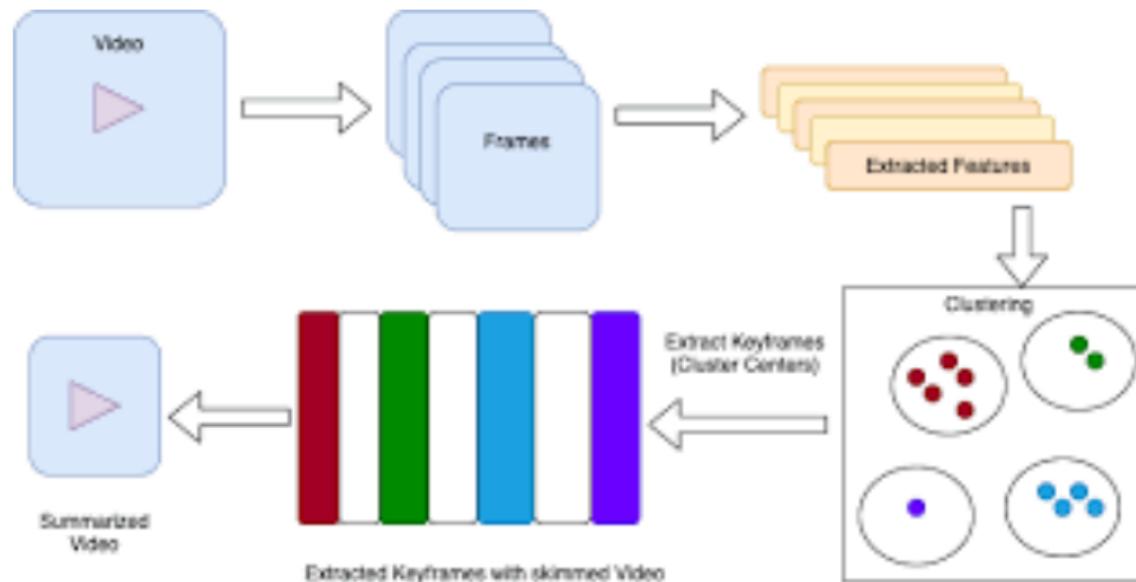


# KEY-FRAME EXTRACTION

---

- Content-Based Methods
  - ▶ Use visual content analysis techniques, such as object detection, scene recognition, or semantic analysis, to identify frames with specific visual content or semantic significance.
  - ▶ Keyframes are selected based on the presence of predefined visual content or semantic concepts.
  - ▶ Frames containing important objects, events, or semantic elements may be chosen as keyframes.
- Clustering Methods
  - ▶ Group similar frames into clusters based on visual similarity and select representative frames (centroids) from each cluster as keyframes.

# KEY-FRAME EXTRACTION USING CLUSTERING



# PLAY/BREAK SEGMENTATION

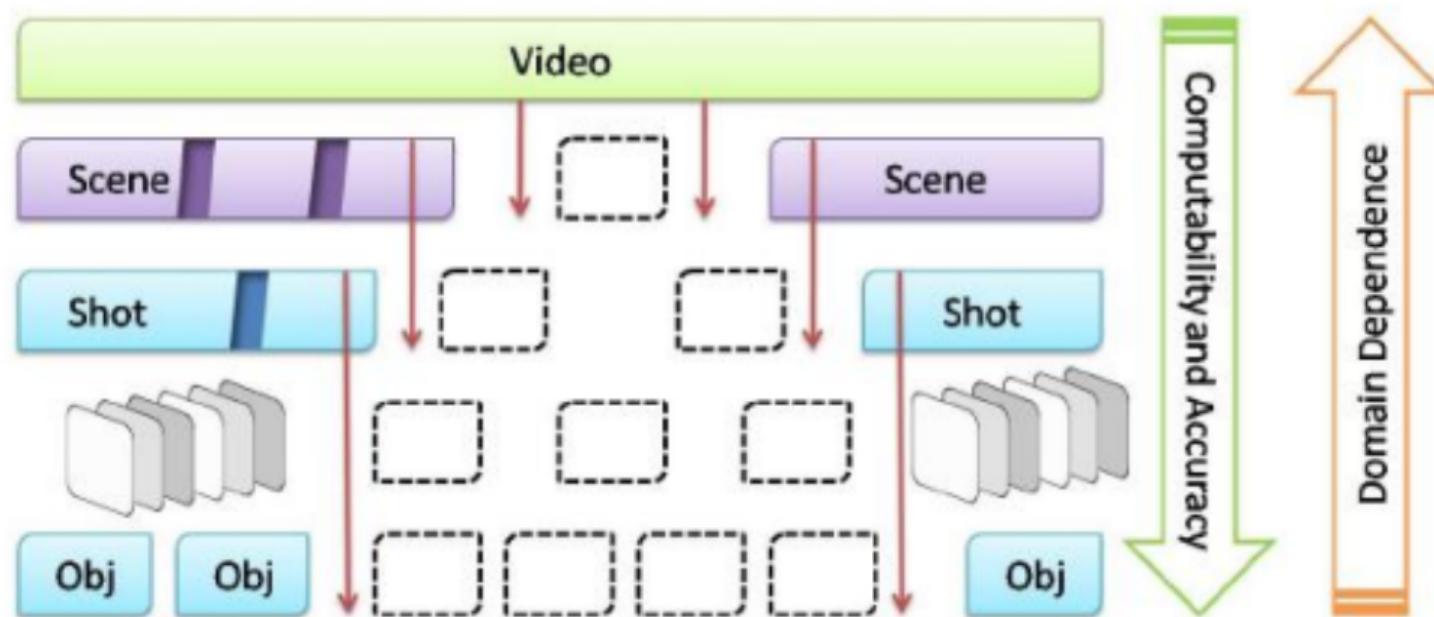
---

- Motion Analysis
  - ▶ Optical flow or motion vectors.
  - ▶ Play segments have more dynamic motion.
  - ▶ Break segments may have little to no motion.
- Temporal Analysis
  - ▶ Play segments exhibit consistent or periodic visual patterns.
  - ▶ Break segments may have irregular or intermittent visual activity.
  - ▶ Temporal analysis techniques, include Fourier analysis, autocorrelation analysis, or wavelet analysis.
- Action Recognition
  - ▶ Action recognition techniques, such as object detection, activity recognition, or gesture recognition.
  - ▶ Classify frames into play and break segments based on the presence or absence of specific actions or activities.

# PLAY/BREAK SEGMENTATION

- Visual Dynamics Analysis
  - ▶ Visual dynamics include changes in visual complexity or entropy.
  - ▶ Play segments have higher visual complexity and entropy due to dynamic action.
  - ▶ Break segments have lower visual complexity and entropy during periods of downtime.
  - ▶ Use entropy-based segmentation.
- Object Tracking
  - ▶ Object tracking techniques, such as Kalman filtering, mean-shift tracking, or deep learning-based tracking, can be used to track objects of interest.
  - ▶ Segment play and break segments based on the presence or absence of tracked objects.
- Scene Change Detection
  - ▶ Play segments often consist of continuous action within a scene.
  - ▶ Break segments may be characterized by scene changes or cuts.

# SEGMENTATION USING OBJECT DETECTION



# SEGMENTATION USING SCENE CHANGE DETECTION



# AUDIO MARKER

- Speech Recognition
  - ▶ Convert audio speech segments into textual representations.
  - ▶ Speech recognition enables the identification of specific words, phrases, or keywords that serve as markers for speech events within the video.
- Audio Segmentation
  - ▶ Audio events or segments, serve as markers.
  - ▶ Divide the audio track into segments based on changes in audio characteristics, such as energy, spectral content, or temporal properties.
  - ▶ Segmentation methods include energy-based segmentation, spectral clustering, or dynamic time warping.
- Keyword Spotting
  - ▶ Detect specific predefined keywords or phrases within the audio track.
  - ▶ Match audio patterns corresponding to target keywords by using keyword spotting algorithms.

# AUDIO MARKER

- Music Detection
  - ▶ Identify segments of music within the audio track.
  - ▶ By analyzing audio features such as tempo, rhythm, melody, or spectral characteristics, music detection algorithms can distinguish between music and non-music segments within the audio stream.
  - ▶ Detected music segments can serve as markers for music-related events in the video.
- Sound Event Detection
  - ▶ Detect and classify various sound events or effects within the audio track.
  - ▶ Identify specific sound events such as applause, cheering, footsteps, or environmental sounds.
  - ▶ Detected sound events can serve as markers.

# AUDIO MARKER

---

- Environmental Sound Recognition
  - ▶ Classify ambient or environmental sounds captured in the audio track.
  - ▶ Environmental sound recognition algorithms identify sounds such as traffic noise, bird chirping, wind blowing, or crowd noise.
  - ▶ Recognized environmental sounds can serve as markers.
- Audio-Based Scene Detection
  - ▶ Analyse audio features to detect transitions between different scenes or segments within the video content.
  - ▶ Scene detection algorithms can detect scene boundaries or transitions, which can serve as markers.

# VIDEO MARKER

---

- Shot Boundary Detection
  - ▶ Shot boundaries can serve as markers for scene changes or transitions between different segments of the video.
- Object Detection and Tracking
  - ▶ Detected objects or tracked regions can serve as markers.
- Event Detection
  - ▶ Identify specific events or actions within the video content and mark their occurrences as video markers.
  - ▶ Events include actions performed by objects or individuals, such as gestures, interactions, or activities.

# VIDEO MARKER

---

- Semantic Analysis
  - ▶ Extract higher-level semantic information from the video content and use it to identify meaningful segments or events.
  - ▶ Semantic analysis can involve object recognition, scene classification, activity recognition, or sentiment analysis.
  - ▶ Detected objects, scenes, activities, or sentiments can serve as markers.
- Content-Based Segmentation
  - ▶ Divide the video into segments based on visual content characteristics.
  - ▶ Segmentation methods analyse video frames to identify boundaries between different segments, such as scenes, actions, or visual patterns.
  - ▶ Segmentation results serve as markers.

# VIDEO MARKER

---

- Visual Dynamics Analysis
  - ▶ Analyse temporal patterns or distributions of visual features over time.
  - ▶ Visual dynamics can include changes in motion, color, texture, or composition.
  - ▶ Segments with high visual dynamics can serve as markers.

# TABLE OF CONTENTS

---

- 1 MODULE 6 TOPICS
- 2 MULTIMODAL ANALYSIS
- 3 IMAGE AND VIDEO FEATURES
- 4 VIDEO ANALYSIS
- 5 VIDEO REPRESENTATION
- 6 VIDEO BROWSING
- 7 VIDEO RETRIEVAL
- 8 VIDEO INDEXING
- 9 VIDEO SUMMARIZATION
- 10 VIDEO SKIMMING
- 11 VIDEO SYNOPSIS

# VIDEO REPRESENTATION

---

- Each video frame is a 2D object and the temporal axis makes up the third dimension, a video stream spans a 3D space.
- **Video representation is the mapping from the 3D space to the 2D view screen.**

# VIDEO REPRESENTATION FOR SCRIPTED CONTENT

---

- Representation Based on Sequential Key Frames
  - ▶ Sequentially lay out the key frames of the video, from top to bottom and from left to right.
- Representation Based on Groups
  - ▶ Divide the entire video stream into multiple video segments, each of which contains an equal number of consecutive shots. Each segment is further divided into subsegments, thus constructing a tree-structured video representation.
  - ▶ Cluster-based video hierarchy, in which the shots are clustered based on their visual content.

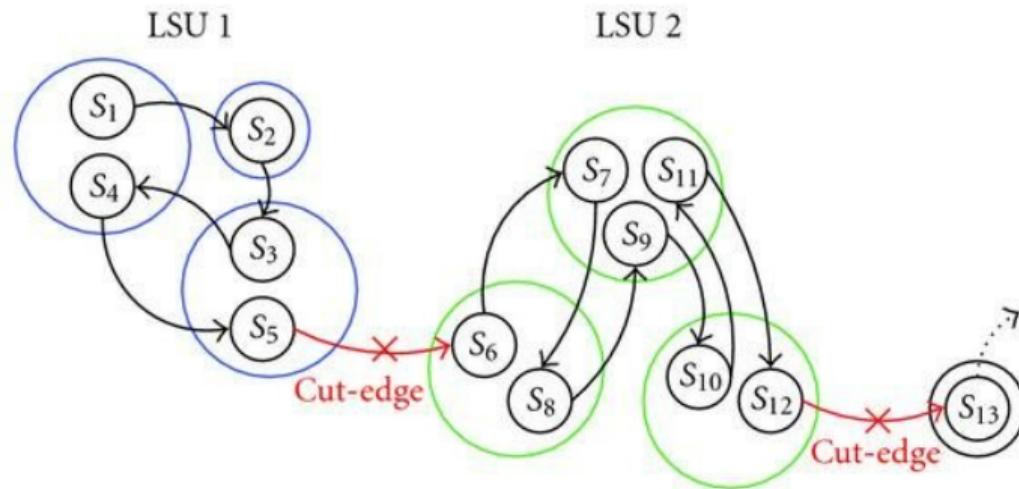
# VIDEO REPRESENTATION FOR SCRIPTED CONTENT

---

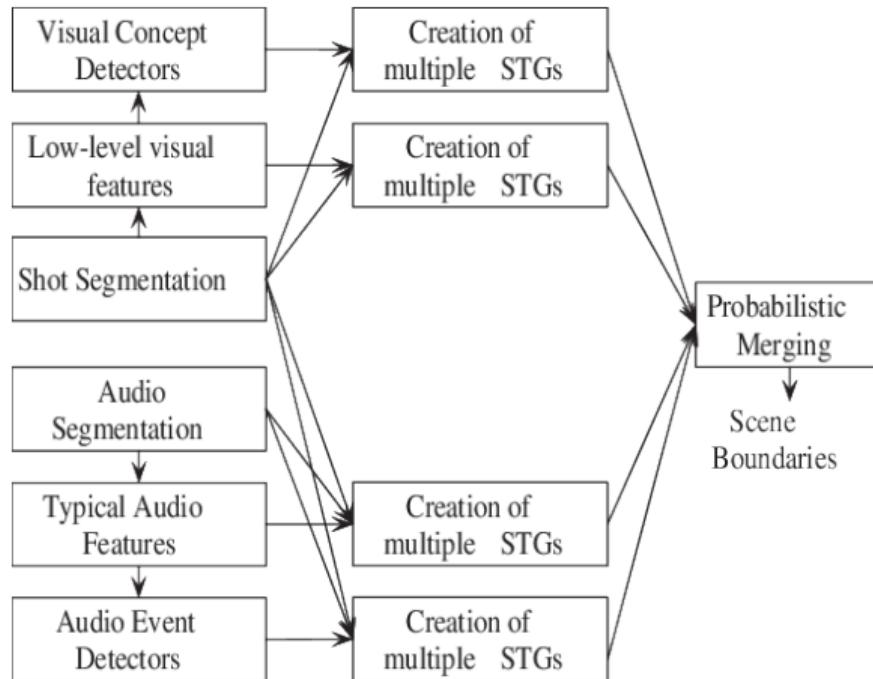
- Representation Based on Scenes

- ▶ People watch the video by its semantic scenes rather than the physical shots or key frames.
- ▶ Scene conveys the semantic meaning of the video to the viewers.
- ▶ Scene transition graph (STG) of video representation
  - ★ Video sequence is first segmented into shots.
  - ★ Shots are then clustered by using time-constrained clustering.
  - ★ The STG is then constructed based on the time flow of the clusters.

# SCENE TRANSITION GRAPH



# SCENE TRANSITION GRAPH



# VIDEO REPRESENTATION FOR SCRIPTED CONTENT

---

- Representation Based on Video Mosaics
  - ▶ information within a shot is decomposed into three components:
    - ★ Extended spatial information captures the appearance of the entire background imaged in the shot and is represented in the form of a few mosaic images.
    - ★ Extended temporal information captures the motion of independently moving objects in the form of their trajectories.
    - ★ Geometric information captures the geometric transformations that are induced by the motion of the camera.

# VIDEO REPRESENTATION FOR UNSCRIPTED CONTENT

- Emphasize detection of remarkable events to support highlights extraction.
- Representation Based on Play/Break Segmentation
  - ▶ Play/break segmentation using low-level features gives a segmentation of the content at the lowest semantic level.
  - ▶ Represent a key frame from each of the detected play segments
- Representation Based on Audiovisual Markers
  - ▶ Higher semantic level than play/break representation
- Representation Based on Highlight Candidates
  - ▶ Association of an audio marker with a video marker enables detection of highlight candidates that are at a higher semantic level.
- Representation Based on Highlight Groups
  - ▶ Grouping of highlight candidates would give a finer resolution representation of the highlight candidates.

# TABLE OF CONTENTS

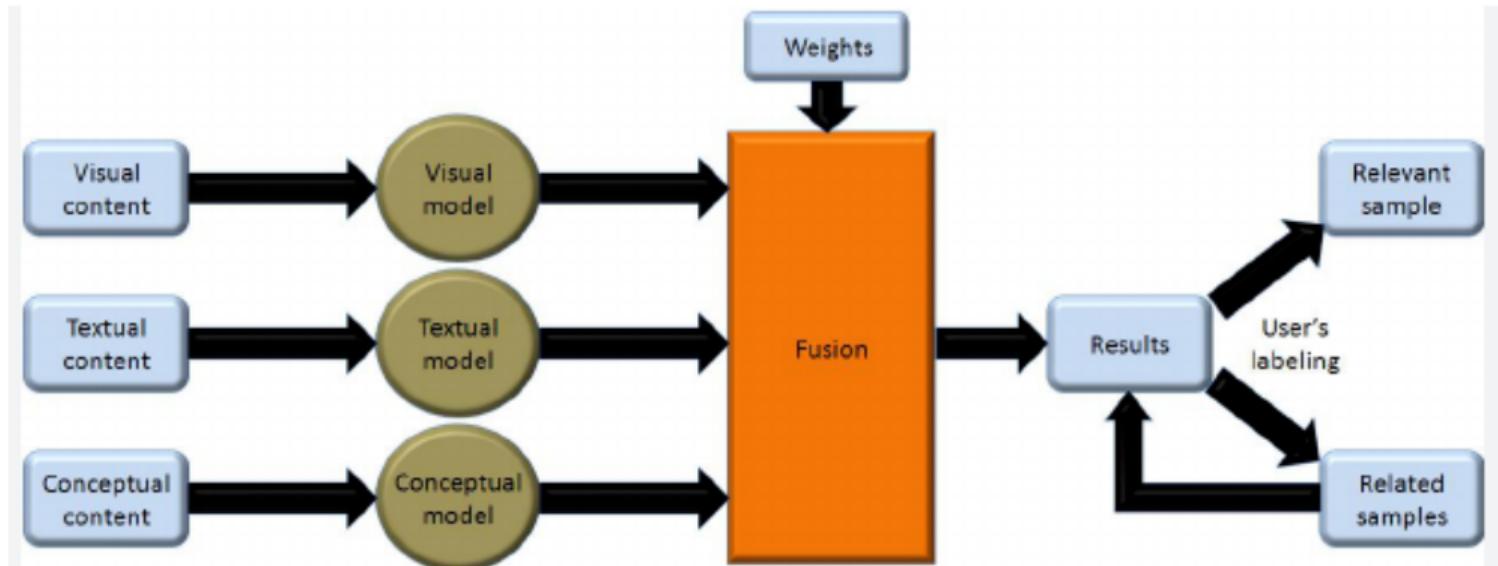
---

- 1 MODULE 6 TOPICS
- 2 MULTIMODAL ANALYSIS
- 3 IMAGE AND VIDEO FEATURES
- 4 VIDEO ANALYSIS
- 5 VIDEO REPRESENTATION
- 6 VIDEO BROWSING
- 7 VIDEO RETRIEVAL
- 8 VIDEO INDEXING
- 9 VIDEO SUMMARIZATION
- 10 VIDEO SKIMMING
- 11 VIDEO SYNOPSIS

# VIDEO BROWSING USING ToC-BASED SUMMARY

- Representation based on sequential key frames
  - ▶ sequential browsing, scanning from the top-left key frame to the bottom-right key frame.
- Representation based on groups
  - ▶ hierarchical browsing
  - ▶ At the coarse level, only the main themes are displayed. Once the user determines which theme he is interested in, he can then go to the finer level of the theme. This refinement process can go on until the leaf level.
- STG representation
  - ▶ indication of time flow embedded within the representation.
  - ▶ By following the time flow, the viewer can browse through the video clip.

# VIDEO BROWSING



# VIDEO BROWSING USING HIGHLIGHTS-BASED SUMMARY

- Representation based on play/break segmentation
  - ▶ Sequential browsing, enabling a scan of all the play segments from the beginning of the video to the end.
- Representation based on audiovisual markers
  - ▶ Queries to find a particular video segment
- Representation based on highlight candidates
  - ▶ Queries to find a particular video segment
- Representation based on highlight groups
  - ▶ Detailed queries to find a particular video segment

# VIDEO BROWSING TECHNIQUES

---

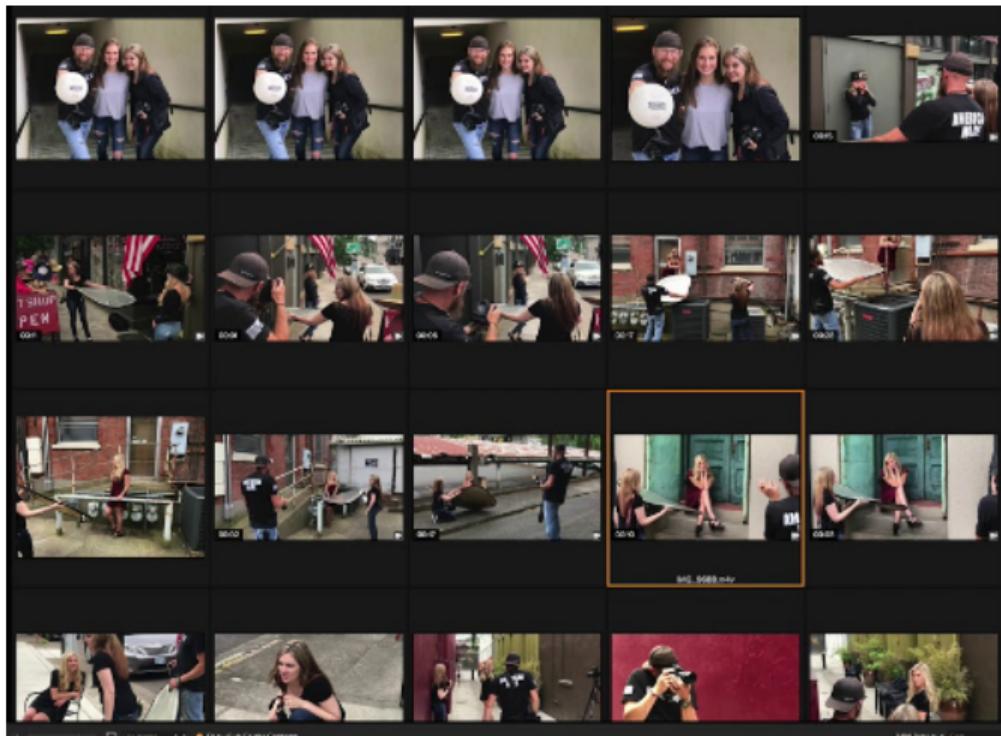
Enable users to navigate and explore video content efficiently.

**KEYFRAME SUMMARIZATION** selects representative frames (keyframes) from the video to create a concise summary. Users can browse through keyframes to get an overview and quickly identify relevant segments.

**THUMBNAIL NAVIGATION** displays a grid of thumbnail images representing different segments of the video. Users can preview the content by hovering over or clicking on thumbnails.

**TIMELINE SCRUBBING** provides a visual timeline representing the duration of the video. Users can drag a slider or cursor along the timeline to scrub through the video and preview different segments.

# THUMBNAIL NAVIGATION



# TIMELINE SCRUBBING



# VIDEO BROWSING TECHNIQUES

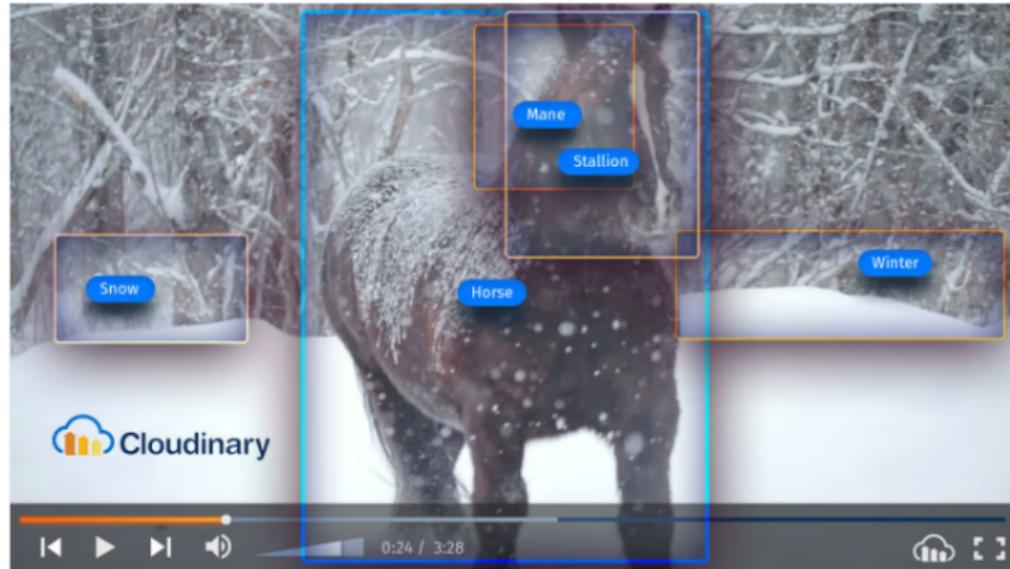
---

**CONTENT-BASED SEARCH** enables users to search for specific video content based on visual features, audio features, or metadata. Users can enter keywords, select visual attributes, or specify audio characteristics to retrieve relevant video segments.

**INTERACTIVE TAGS** are metadata labels associated with specific segments or objects within the video. Users can click on tags to navigate to related content or filter the video based on specific criteria, such as people, objects, or scenes.

**TEMPORAL THUMBNAILS** provide a visual representation of the video content by displaying a sequence of thumbnail images along the timeline. Users can hover over or click on thumbnails to preview video segments.

# INTERACTIVE TAGGING



# VIDEO BROWSING TECHNIQUES

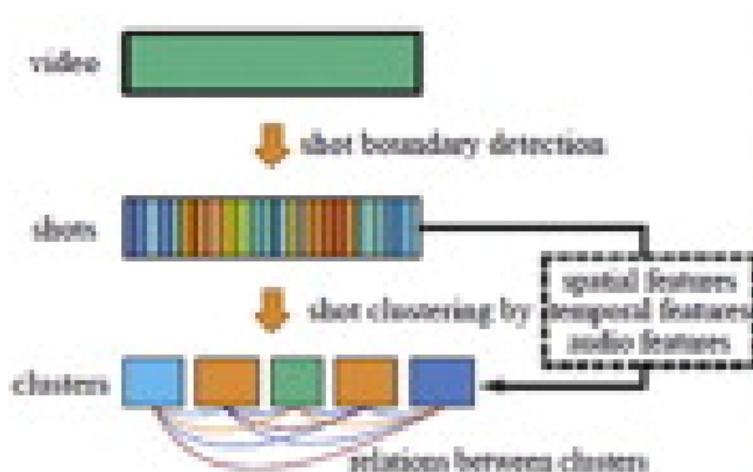
**SCENE DETECTION** automatically identifies transitions between scenes or shots in the video. Users can navigate between scenes using scene markers or thumbnails.

**DYNAMIC SUMMARIZATION** generates on-the-fly summaries of video content based on user interactions or preferences. Users can adjust summarization parameters such as duration, content relevance, or diversity to customize the summary according to their preferences.

**SEMANTIC ZOOMING** provides different levels of detail for video content, allowing users to zoom in or out of specific segments or scenes. Users can navigate between different levels of granularity to explore video content at varying levels of detail.

**PERSONALIZED RECOMMENDATIONS** suggest relevant video content based on user preferences, viewing history, or contextual information.

# MOSAIC



# TABLE OF CONTENTS

---

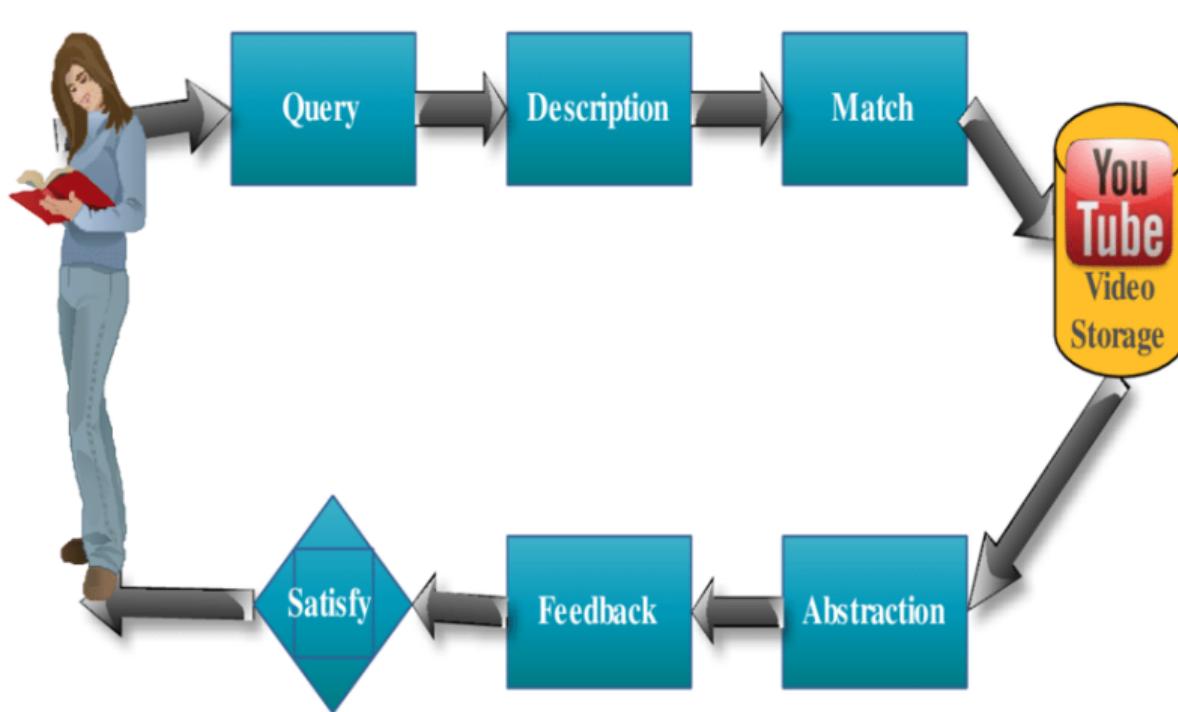
- 1 MODULE 6 TOPICS
- 2 MULTIMODAL ANALYSIS
- 3 IMAGE AND VIDEO FEATURES
- 4 VIDEO ANALYSIS
- 5 VIDEO REPRESENTATION
- 6 VIDEO BROWSING
- 7 VIDEO RETRIEVAL
- 8 VIDEO INDEXING
- 9 VIDEO SUMMARIZATION
- 10 VIDEO SKIMMING
- 11 VIDEO SYNOPSIS

# VIDEO RETRIEVAL

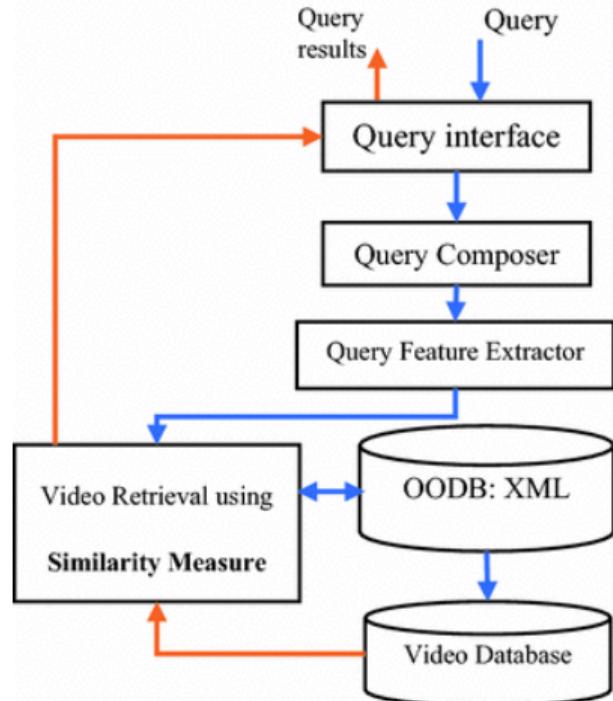
---

- Video retrieval refers to the process of searching for and retrieving relevant videos from large collections based on user queries or predefined criteria.
- It involves analyzing the content, metadata, or context of videos to identify matches with the user's search intent.

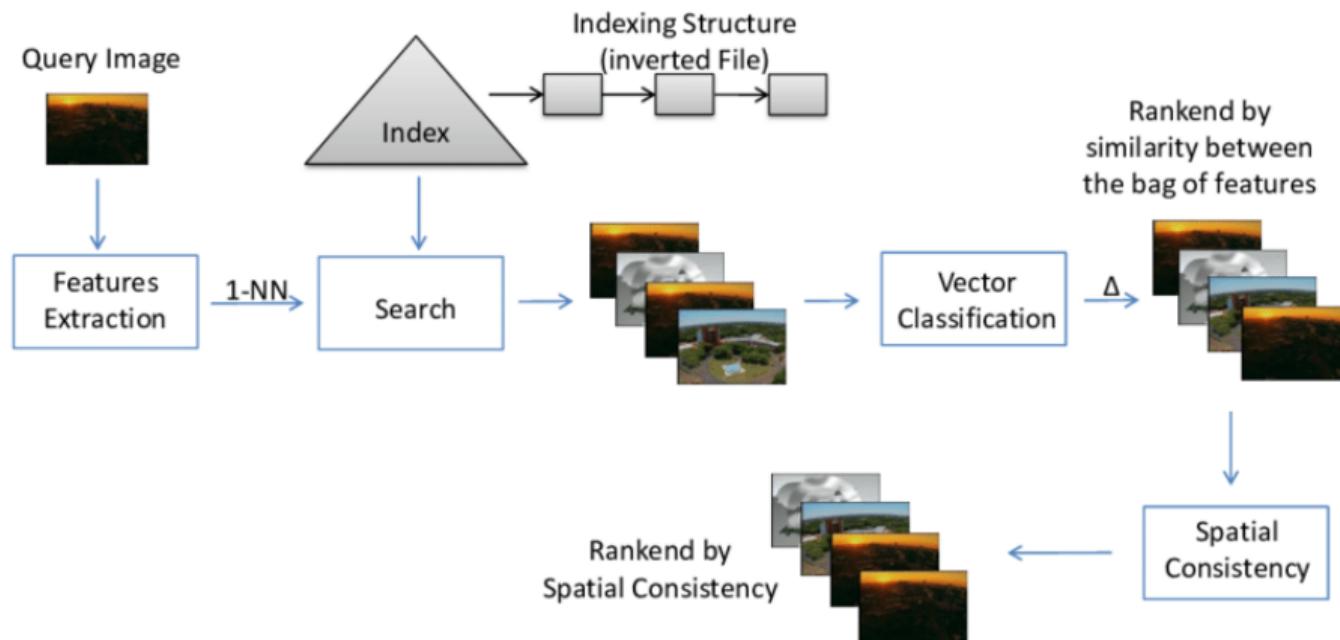
# VIDEO RETRIEVAL FRAMEWORK



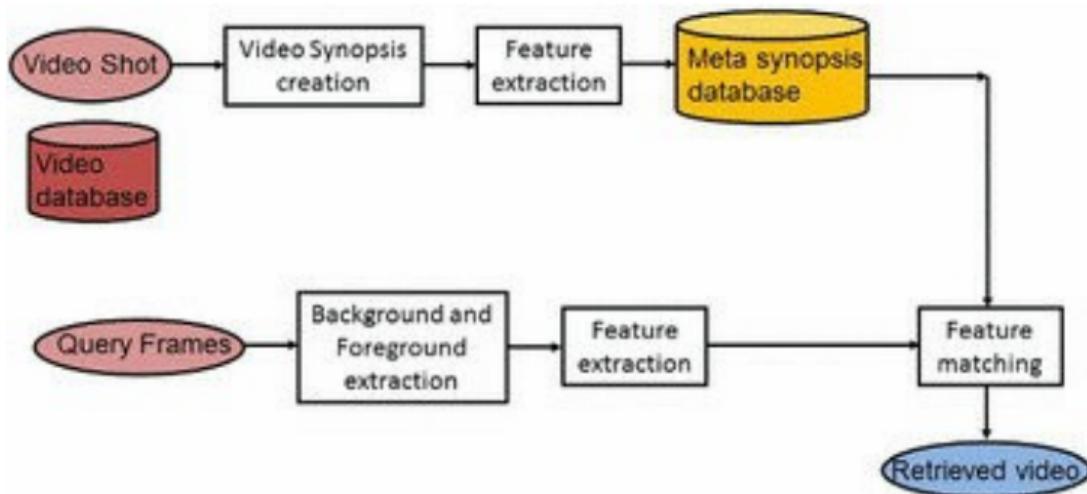
# VIDEO RETRIEVAL FRAMEWORK



# VIDEO RETRIEVAL USING INDEXING



# VIDEO RETRIEVAL USING SYNOPSIS



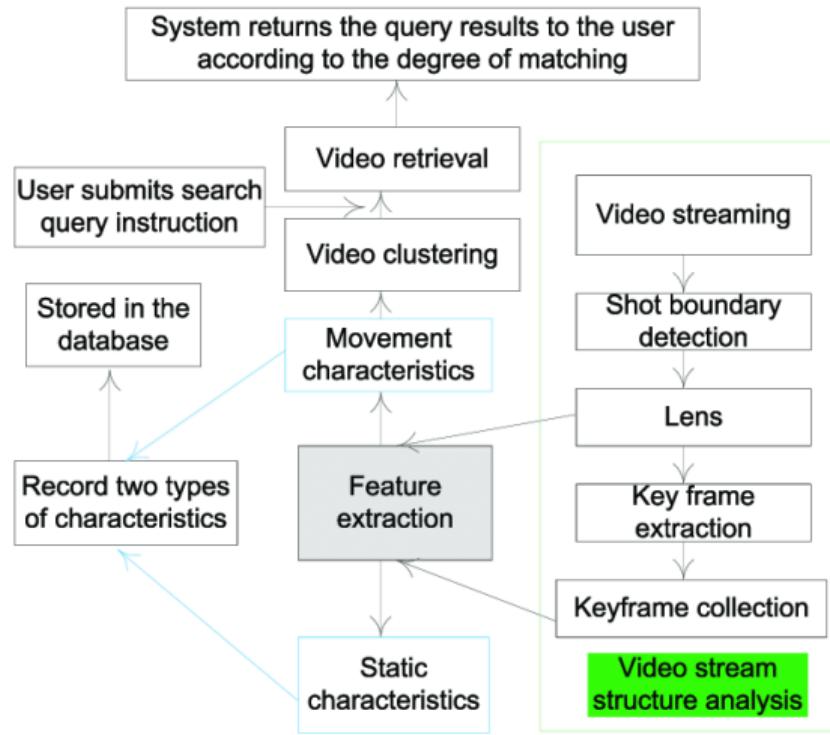
# VIDEO RETRIEVAL TECHNIQUES

---

**KEYWORD SEARCH** allows users to enter text-based queries, such as keywords or phrases, to search for videos containing specific content or topics. Video metadata, including titles, descriptions, and tags, are indexed and searched to retrieve relevant videos.

**CONTENT-BASED RETRIEVAL** analyze the visual and/or audio content of videos to search for similar or related videos. Features such as color histograms, texture descriptors, motion vectors, or audio spectrograms are extracted from video frames and/or audio tracks. Similarity measures, such as distance metrics or similarity scores, are then used to compare the features and retrieve videos with similar content.

# CONTENT-BASED VIDEO RETRIEVAL



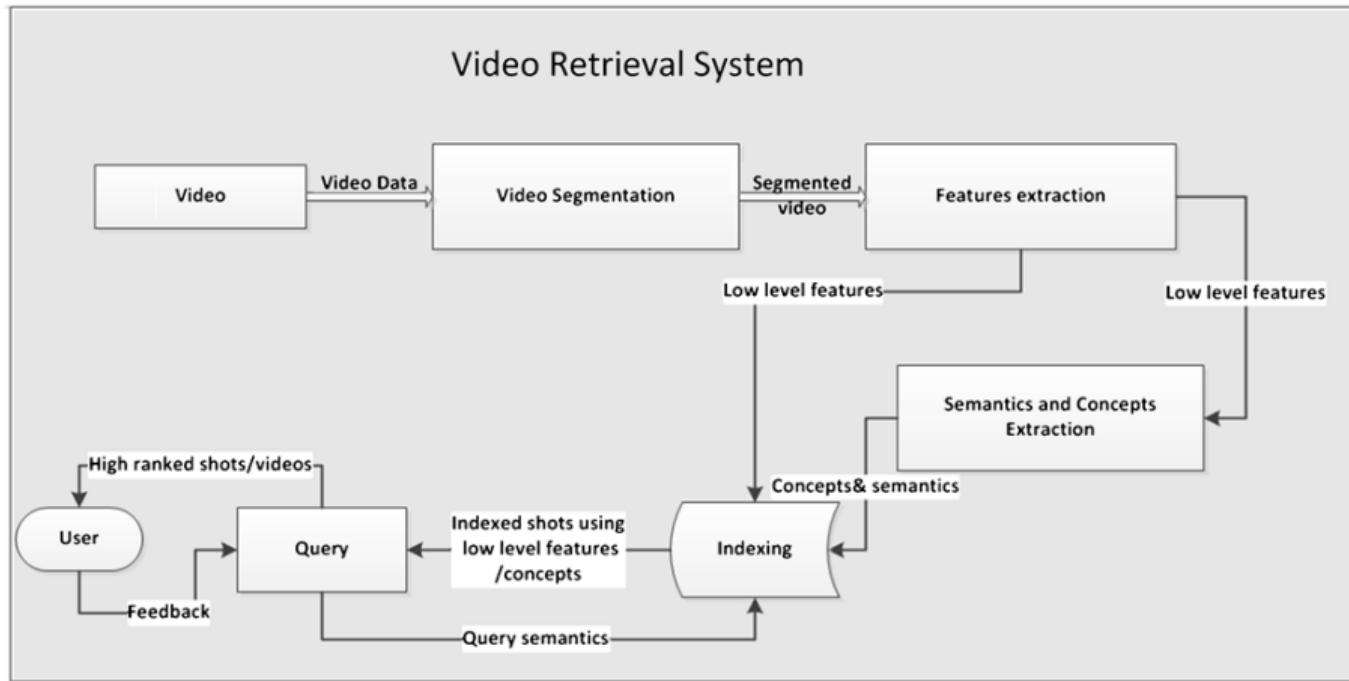
# VIDEO RETRIEVAL TECHNIQUES

---

METADATA-BASED RETRIEVAL leverages metadata associated with videos, such as titles, descriptions, tags, categories, or timestamps, to search for and filter videos. Users can specify search criteria or filter options based on metadata attributes to retrieve videos matching specific criteria.

SEMANTIC RETRIEVAL uses semantic analysis to understand the meaning and context of video content. Natural language processing (NLP) techniques, semantic indexing, or ontology-based approaches are applied to analyze video content and retrieve videos based on their semantic relevance to user queries.

# SEMANTIC-BASED VIDEO RETRIEVAL



# VIDEO RETRIEVAL TECHNIQUES

---

**RELEVANCE FEEDBACK** involve iterative refinement of search results based on user feedback. Users initially submit a query, and the system retrieves an initial set of results. Users then provide feedback on the relevance of the retrieved videos, which is used to refine the search and retrieve more relevant videos in subsequent iterations.

**TEMPORAL-BASED RETRIEVAL** consider the temporal structure and dynamics of video content when retrieving videos. Features such as shot boundaries, scene changes, or temporal patterns are analyzed to retrieve videos based on their temporal characteristics.

# TABLE OF CONTENTS

---

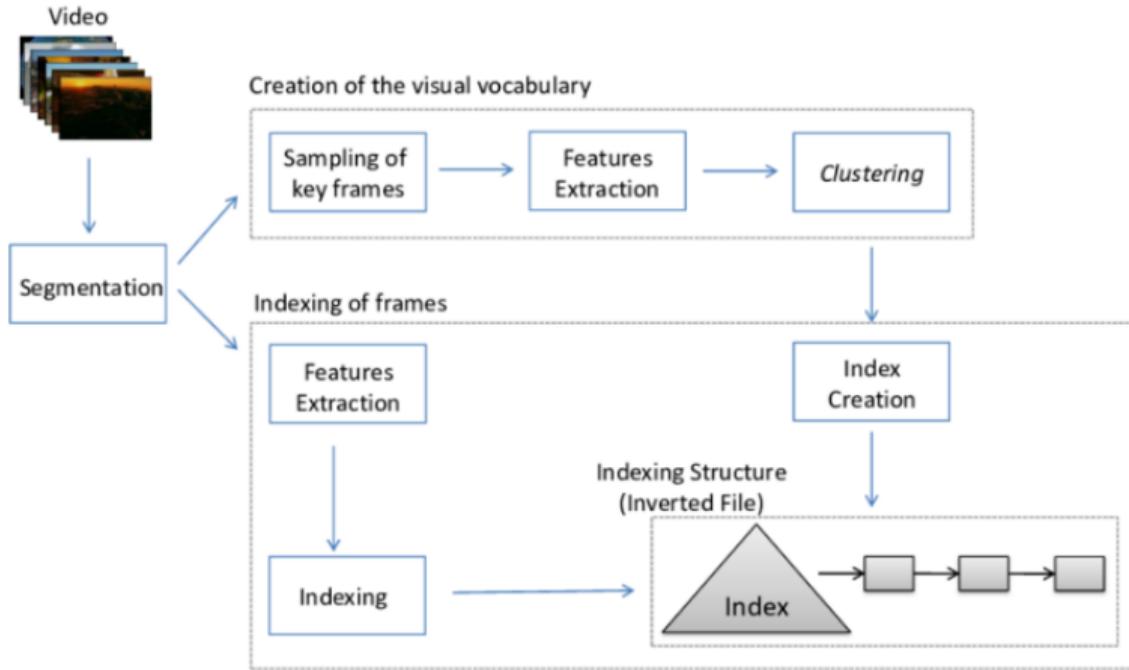
- 1 MODULE 6 TOPICS
- 2 MULTIMODAL ANALYSIS
- 3 IMAGE AND VIDEO FEATURES
- 4 VIDEO ANALYSIS
- 5 VIDEO REPRESENTATION
- 6 VIDEO BROWSING
- 7 VIDEO RETRIEVAL
- 8 VIDEO INDEXING
- 9 VIDEO SUMMARIZATION
- 10 VIDEO SKIMMING
- 11 VIDEO SYNOPSIS

# VIDEO INDEXING

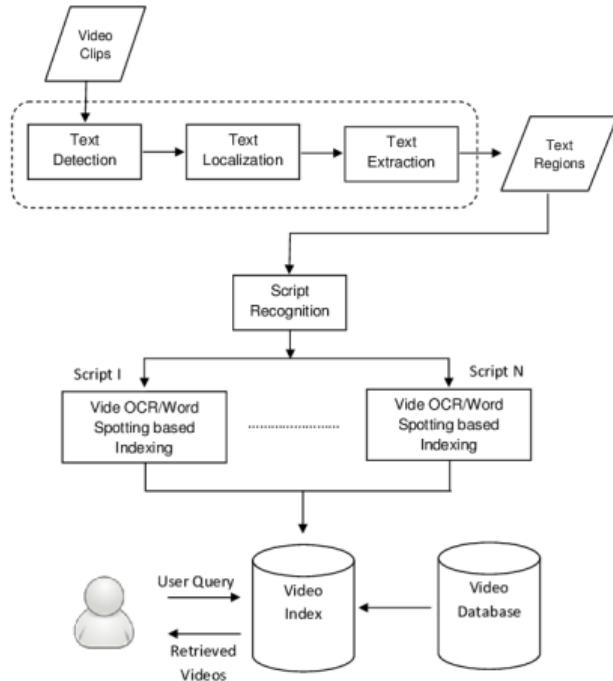
---

- Video indexing is the process of creating an organized and searchable representation of the content within a video or a collection of videos.
- It involves analyzing the visual, auditory, and temporal aspects of videos to extract relevant information and metadata.
- Video indexing facilitates efficient retrieval, browsing, and analysis of video content.

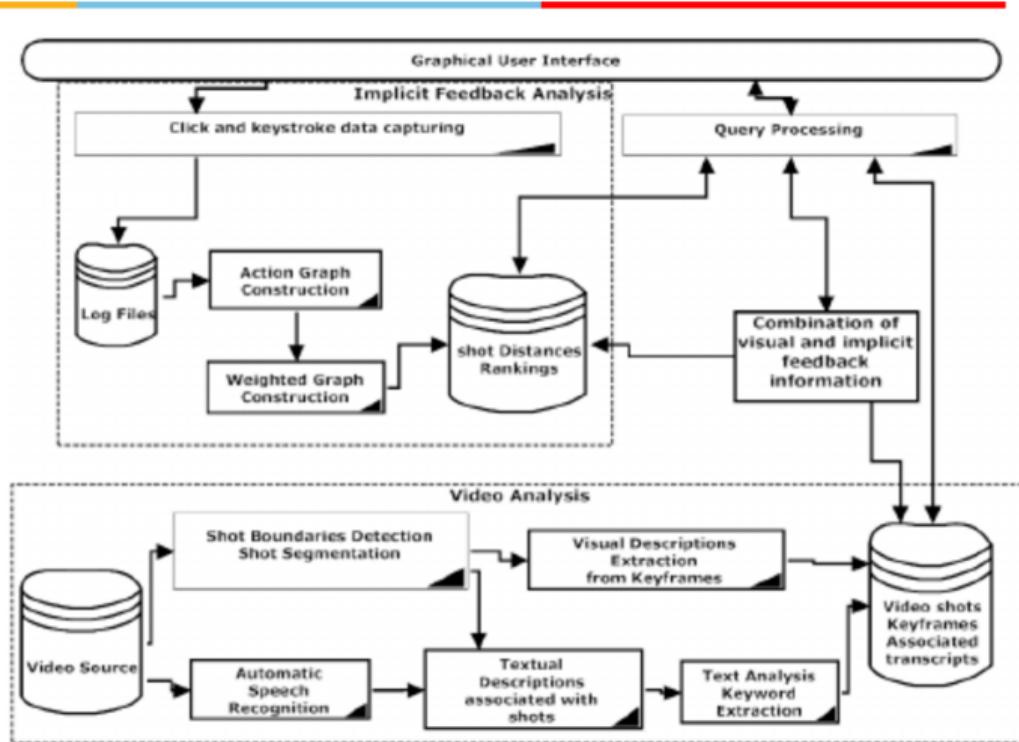
# VIDEO INDEXING FRAMEWORK



# VIDEO INDEXING FRAMEWORK



# VIDEO INDEXING FRAMEWORK



# VIDEO INDEXING TECHNIQUES

---

**SHOT BOUNDARY DETECTION** helps segment the video into coherent units for indexing and retrieval.

**KEYFRAME EXTRACTION** Keyframes serve as visual anchors and provide a compact representation of the video content for indexing and browsing.

**FEATURE EXTRACTION** Extracting low-level visual, auditory, or temporal features from video data. Visual features may include color histograms, texture descriptors, edge maps, or deep learning embeddings. Auditory features may include audio spectrograms, speech transcripts, or sound event classifications. Temporal features may capture motion dynamics, scene changes, or activity patterns.

# VIDEO INDEXING TECHNIQUES

---

**OBJECT DETECTION AND RECOGNITION** Object detection algorithms identify and localize objects of interest, while recognition algorithms classify the detected objects into predefined categories.

**SCENE CLASSIFICATION** Classifying video frames or segments into semantic categories, such as indoor/outdoor scenes, action types, or event categories.

**SPEECH RECOGNITION** enables indexing and searching for videos based on spoken content, facilitating content discovery and retrieval.

**SEMANTIC ANNOTATION** provides additional context and metadata to describe the video content, enhancing search and retrieval capabilities.

# VIDEO INDEXING TECHNIQUES

---

**TEMPORAL SEGMENTATION** helps organize the video into coherent segments for indexing and retrieval.

**METADATA EXTRACTION** Extracting metadata from video files or external sources, such as titles, descriptions, tags, categories, timestamps, or geolocation information. Metadata provides context and additional information about the video content for indexing and retrieval.

# TABLE OF CONTENTS

---

- 1 MODULE 6 TOPICS
- 2 MULTIMODAL ANALYSIS
- 3 IMAGE AND VIDEO FEATURES
- 4 VIDEO ANALYSIS
- 5 VIDEO REPRESENTATION
- 6 VIDEO BROWSING
- 7 VIDEO RETRIEVAL
- 8 VIDEO INDEXING
- 9 VIDEO SUMMARIZATION
- 10 VIDEO SKIMMING
- 11 VIDEO SYNOPSIS

# VIDEO SUMMARIZATION

---

- Video summarization is the process of creating a concise and representative summary of a longer video, capturing its essential content, key events, and highlights.
- Goal: provide a condensed version of the original video that conveys its main points and allows viewers to quickly grasp its content without watching the entire video.
- Video summarization techniques aim to automatically select and extract salient segments, keyframes, or shots from the video, considering factors such as content importance, relevance, diversity, and coherence.
- The resulting summary may consist of a sequence of selected shots, keyframes, or a compact summary video that represents the most informative and significant parts of the original video.

# VIDEO SUMMARIZATION - 2 APPROACHES

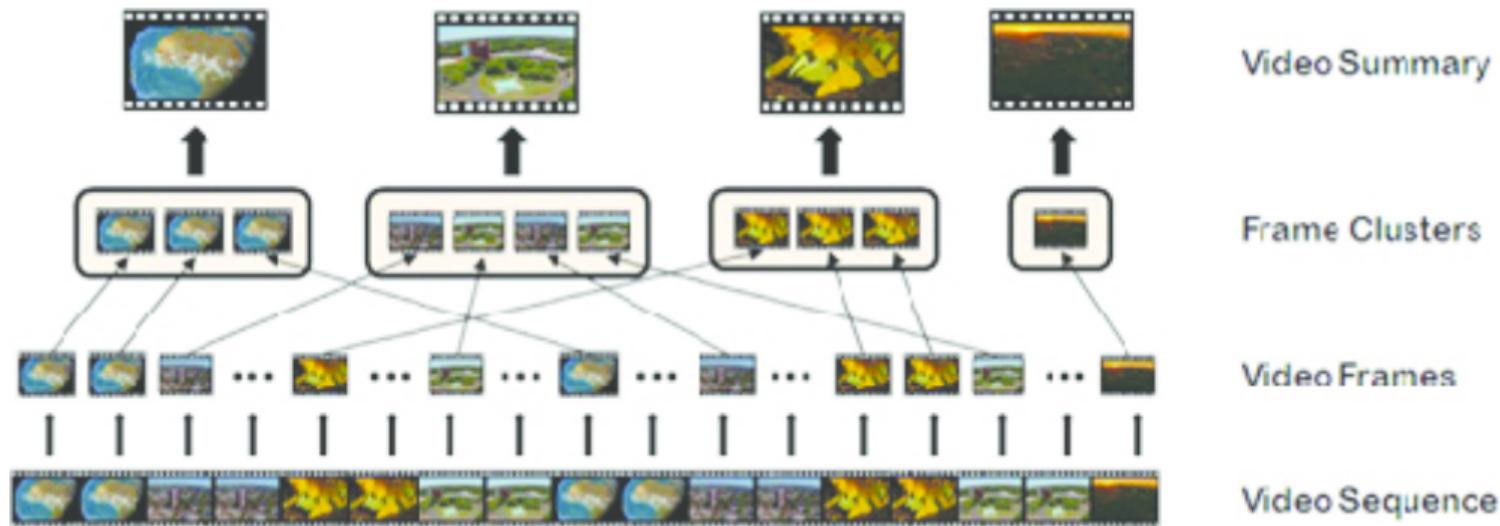
---

**EXTRACTIVE SUMMARIZATION** select and extract existing segments, shots, or keyframes from the original video to compose the summary.

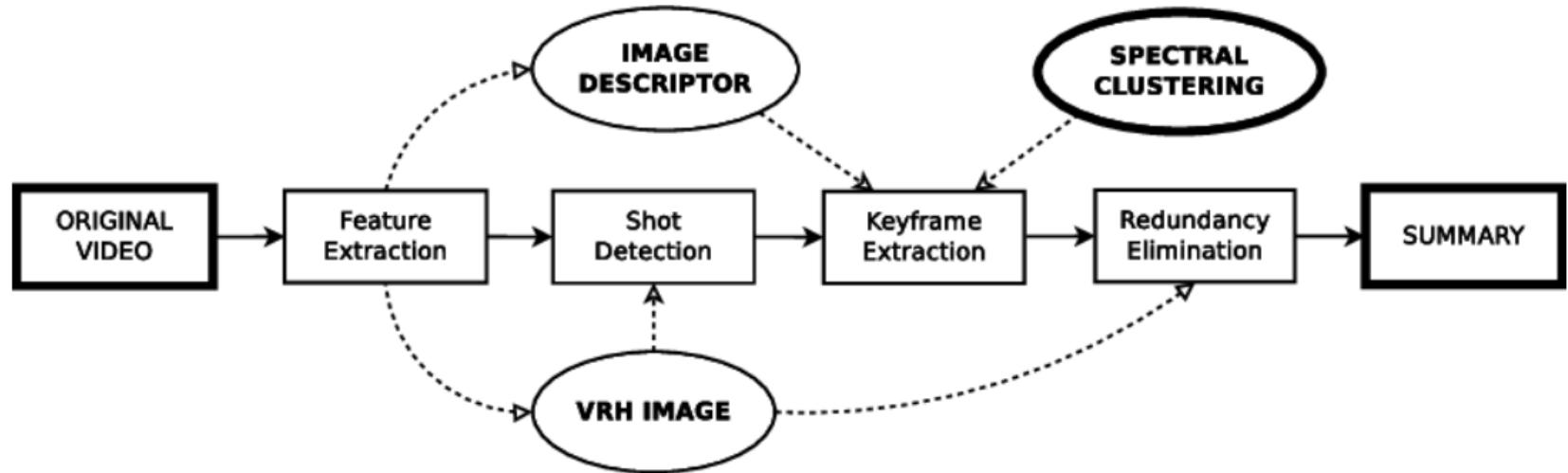
- Evaluate the relevance of each segment based on features such as visual content, audio characteristics, or temporal dynamics.
- Include keyframe selection, shot selection, and sequence reordering to create a coherent summary while preserving the original content.

**ABSTRACTIVE SUMMARIZATION** generate a summary by synthesizing new content or descriptions that capture the essence of the original video. Abstractive summarization aims to provide a more concise and informative summary by capturing the underlying meaning and context of the video content.

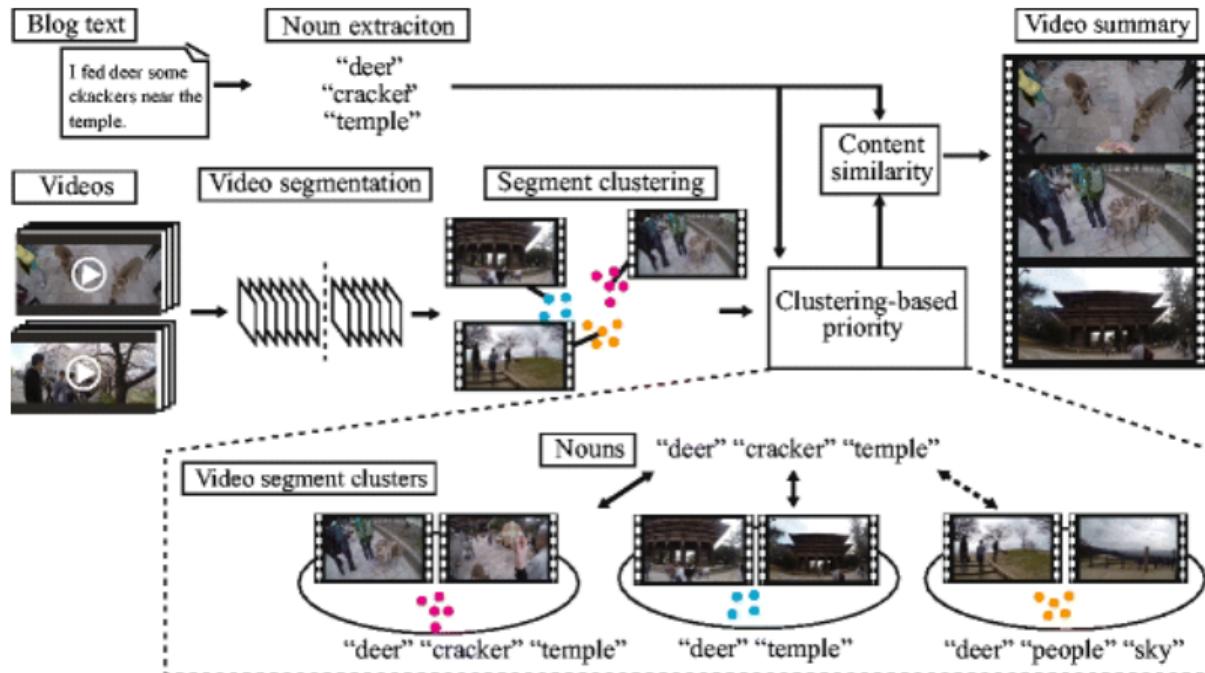
# VIDEO SUMMARIZATION FRAMEWORK



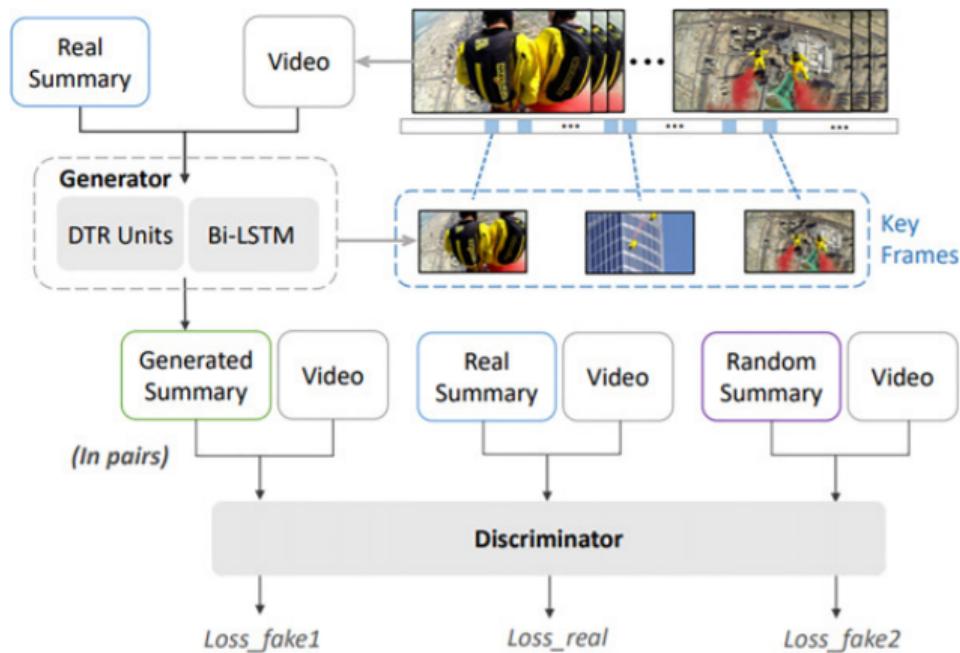
# VIDEO SUMMARIZATION FRAMEWORK



# TEXT BASED VIDEO SUMMARIZATION



# VIDEO SUMMARIZATION BASED ON GAN



# TABLE OF CONTENTS

---

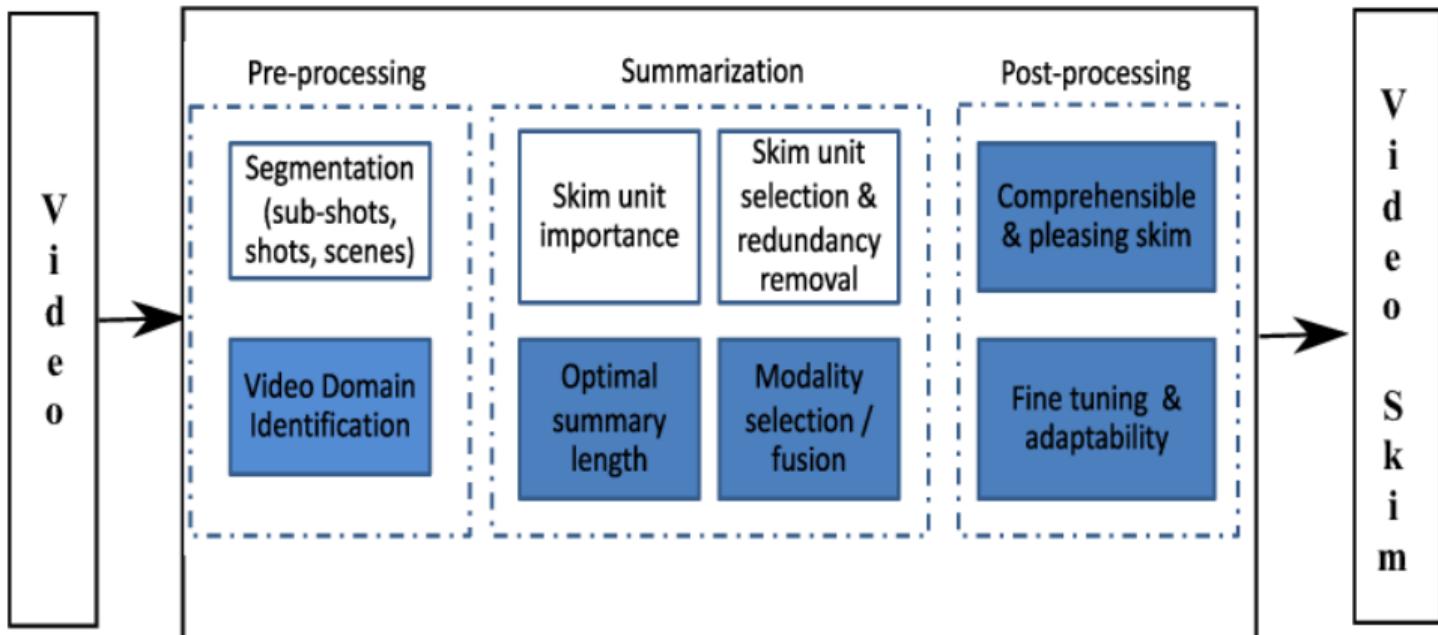
- 1 MODULE 6 TOPICS
- 2 MULTIMODAL ANALYSIS
- 3 IMAGE AND VIDEO FEATURES
- 4 VIDEO ANALYSIS
- 5 VIDEO REPRESENTATION
- 6 VIDEO BROWSING
- 7 VIDEO RETRIEVAL
- 8 VIDEO INDEXING
- 9 VIDEO SUMMARIZATION
- 10 VIDEO SKIMMING
- 11 VIDEO SYNOPSIS

# VIDEO SKIMMING

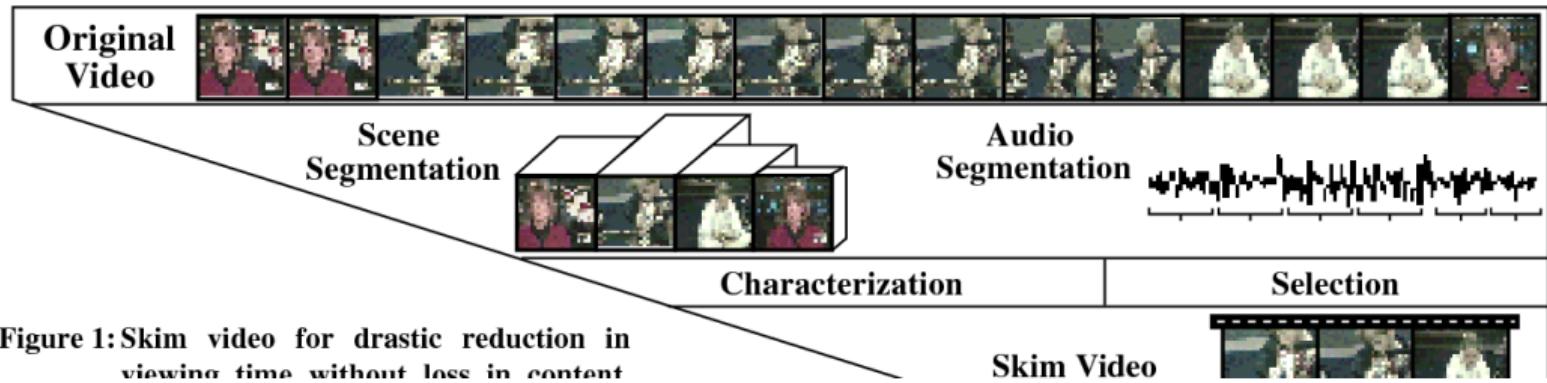
---

- Video skimming is a process of quickly browsing through a video to get an overview of its content without watching the entire video in real-time.
- Condensed video content
- Video skimming enables users to efficiently navigate and explore video content, helping them identify relevant segments or areas of interest within the video.
- Applications include video browsing, content summarization, video surveillance, and multimedia retrieval, where users need to quickly review and understand the content of large video collections.

# VIDEO SKIMMING

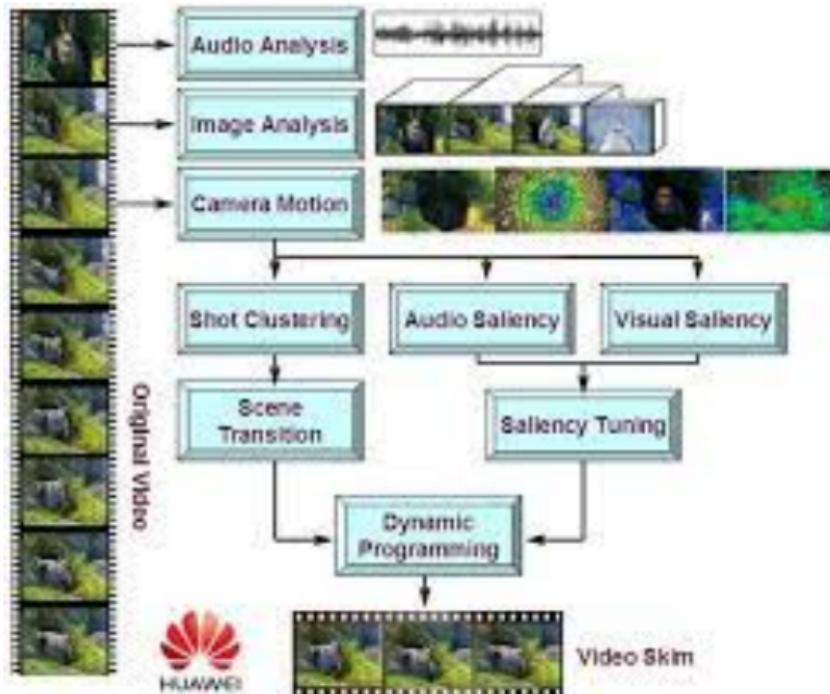


# VIDEO SKIMMING



**Figure 1:** Skim video for drastic reduction in viewing time without loss in content

# VIDEO SKIMMING



# VIDEO SKIMMING



# TABLE OF CONTENTS

---

- 1 MODULE 6 TOPICS
- 2 MULTIMODAL ANALYSIS
- 3 IMAGE AND VIDEO FEATURES
- 4 VIDEO ANALYSIS
- 5 VIDEO REPRESENTATION
- 6 VIDEO BROWSING
- 7 VIDEO RETRIEVAL
- 8 VIDEO INDEXING
- 9 VIDEO SUMMARIZATION
- 10 VIDEO SKIMMING
- 11 VIDEO SYNOPSIS

# VIDEO SYNOPSIS

---

- Video synopsis is a condensed and compact representation of a longer video sequence, created by summarizing and compressing its content while preserving the essential information and key events.
- Generate a concise summary that captures the main storyline, significant events, and transitions within the video.
- Video synopsis techniques involve selecting and summarizing salient segments, keyframes, or events from the original video, while also maintaining temporal coherence and context.
- Synopsis Generation combine the selected keyframes, events, or segments into a condensed summary that represents the main storyline and significant events of the video.

# VIDEO SYNOPSIS

## CAMERA IN A BILLIARD CLUB (SEE 9 HOURS IN 20 SECONDS)

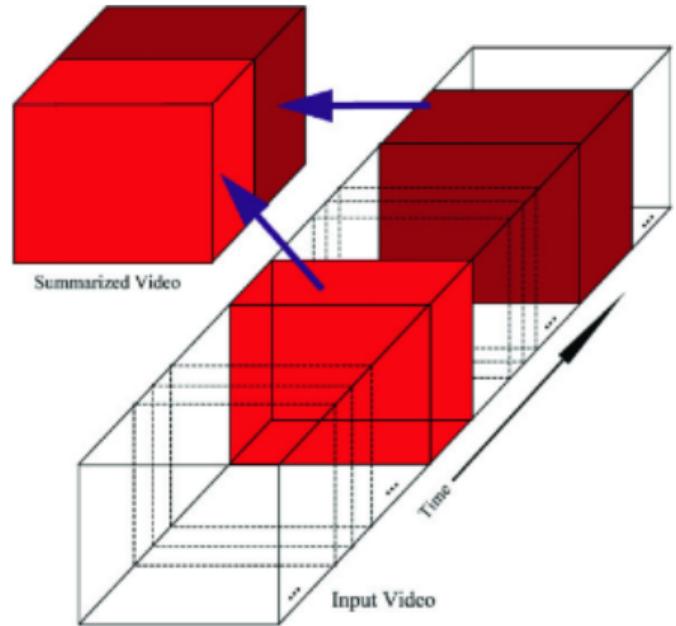
Typical Video Stream (9 Hours)



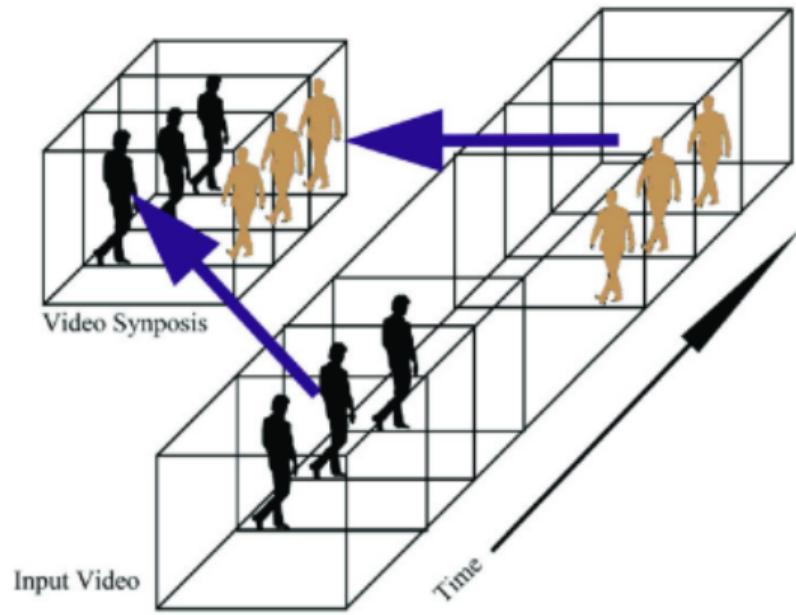
Video Synopsis (20 Seconds)



# VIDEO SYNOPSIS

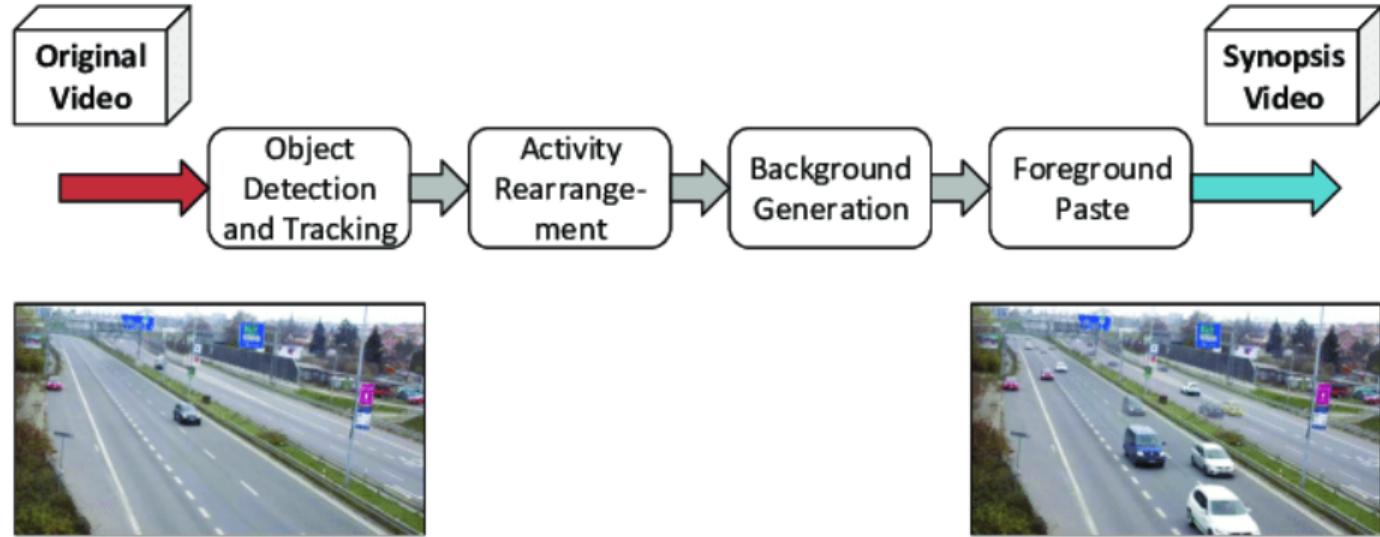


(a)

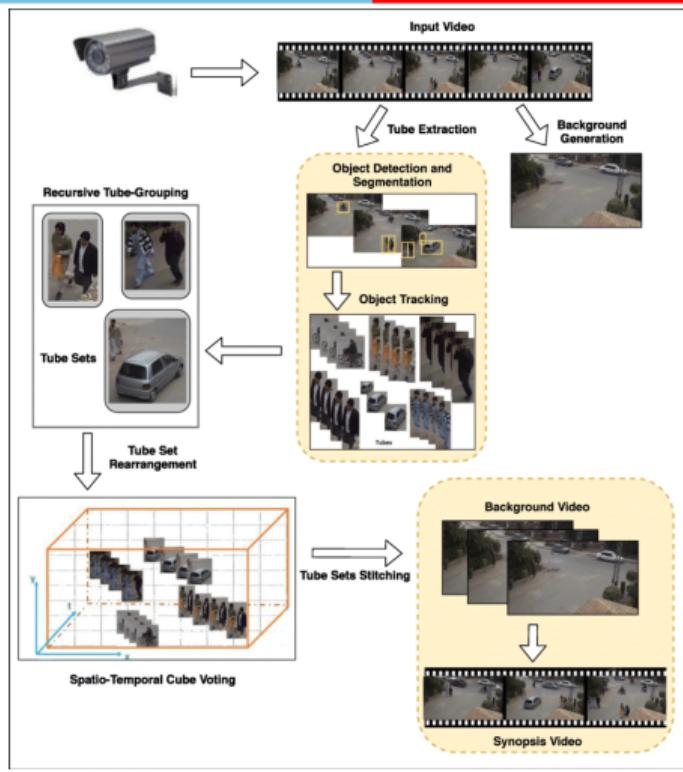


(b)

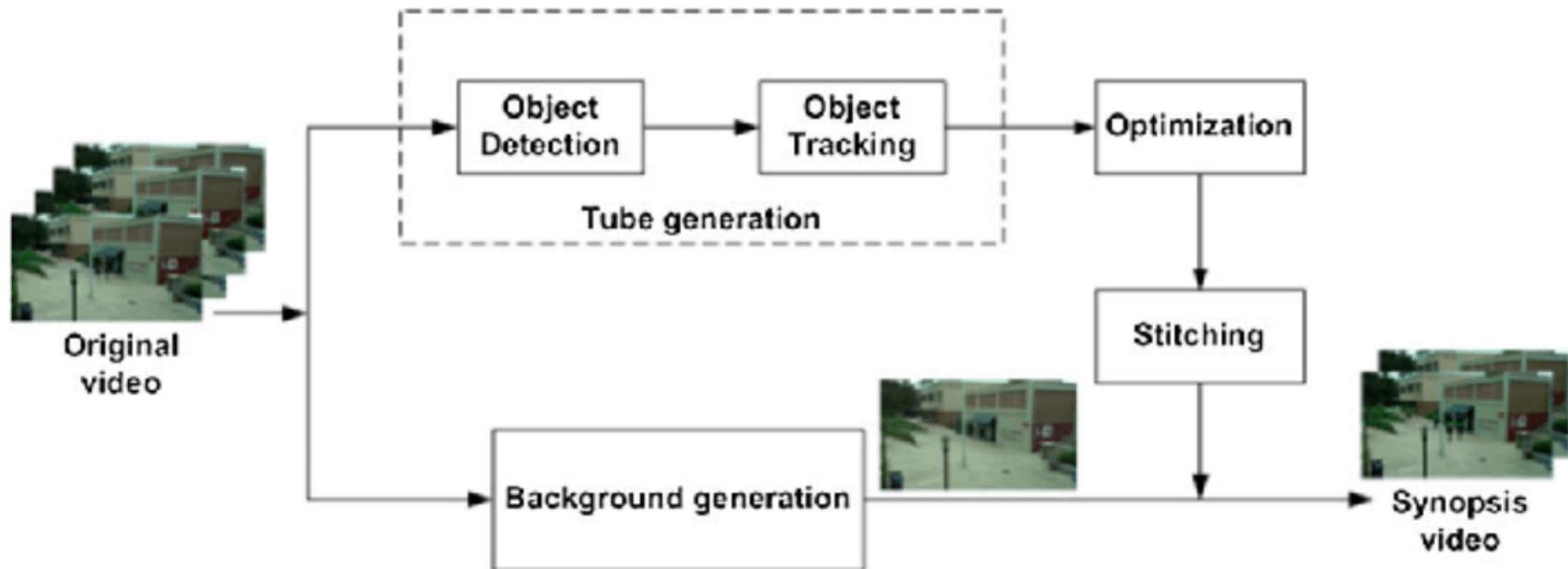
# VIDEO SYNOPSIS



# VIDEO SYNOPSIS FRAMEWORK



# VIDEO SYNOPSIS USING OBJECT TRACKING



# REFERENCES

- ① Bovik, Alan C. The essential guide to video processing. (T1)



**Thank You!**



# VIDEO ANALYTICS

## MODULE # 7 DEEP LEARNING FOR VIDEO ANALYTICS



**BITS Pilani**  
Pilani | Dubai | Goa | Hyderabad

Seetha Parameswaran  
BITS Pilani

---

The instructor is gratefully acknowledging  
the authors who made their course  
materials freely available online.

This deck is prepared by Seetha Parameswaran.

# TABLE OF CONTENTS

---

- 1 MODULE 7 TOPICS
- 2 VISION TRANSFORMER
- 3 3D CONVOLUTIONAL NEURAL NETWORK (3D CNN)
- 4 TWO-STREAM CONVOLUTIONAL NETWORKS
- 5 SIAMESE NETWORK

# MODULE TOPICS....

---

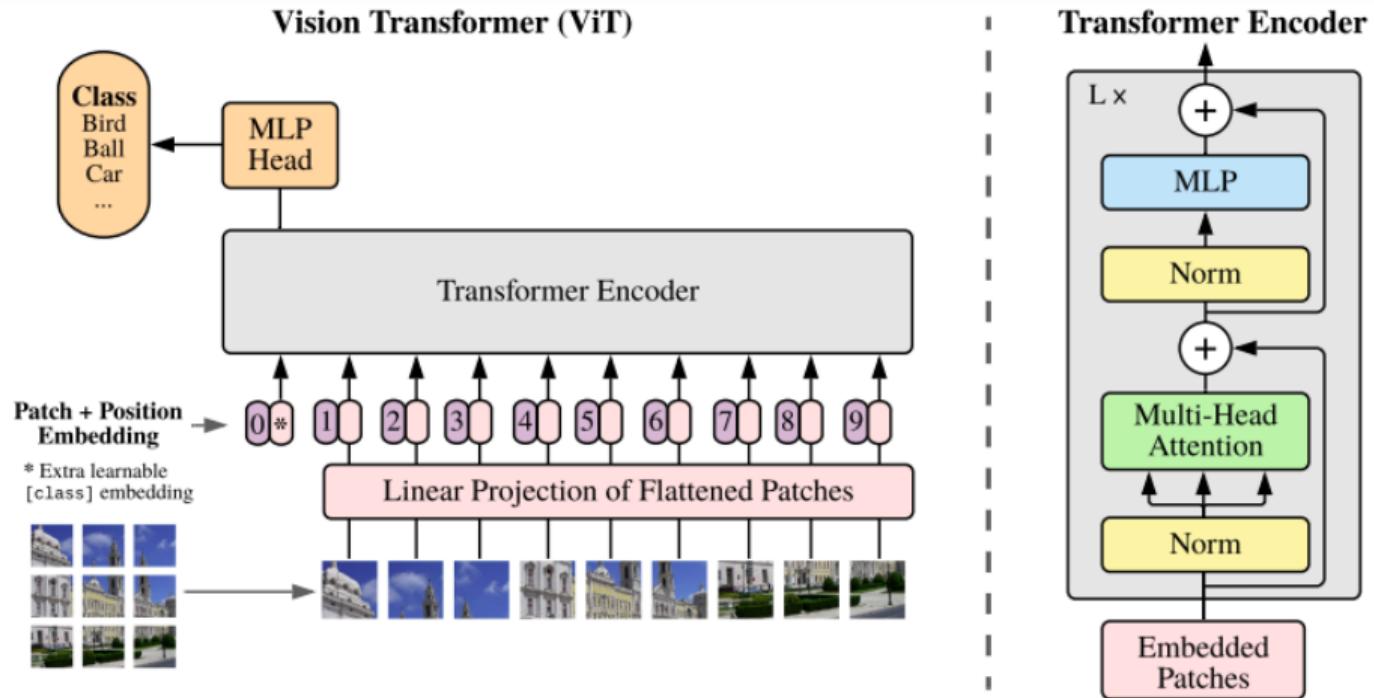
- Vision Transformer
- 3D CNN
- Two Stream CNN
- Siamese Network

# TABLE OF CONTENTS

---

- 1 MODULE 7 TOPICS
- 2 VISION TRANSFORMER
- 3 3D CONVOLUTIONAL NEURAL NETWORK (3D CNN)
- 4 TWO-STREAM CONVOLUTIONAL NETWORKS
- 5 SIAMESE NETWORK

# VISION TRANSFORMER



# TRANSFORMER: A SELF-ATTENTION NETWORK

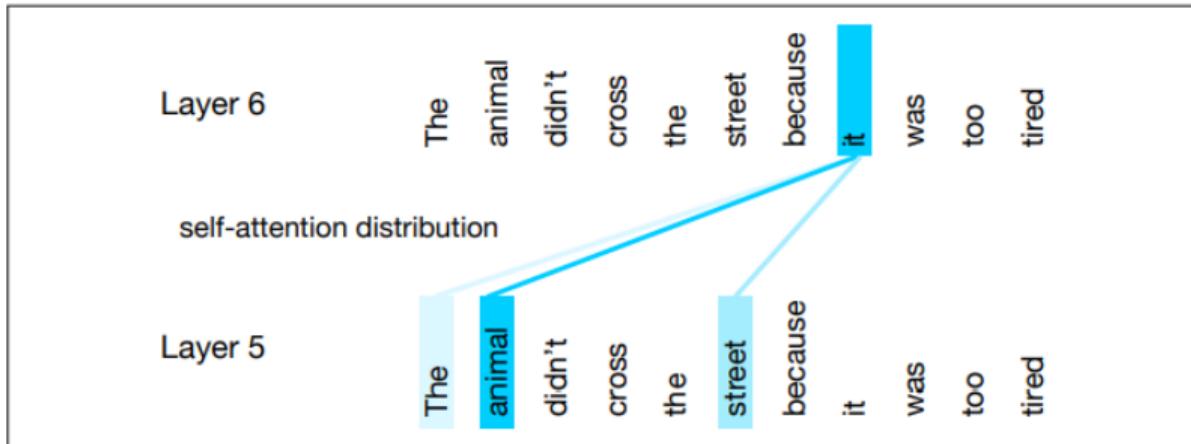
- Causal language modeling
  - ▶ input to a transformer is a sequence of words / tokens ( $x_1, \dots, x_n$ )
  - ▶ output
    - ★ prediction for what word comes next ( $z_1, \dots, z_n$ )
    - ★ sequence of contextual embedding that represents the contextual meaning of each of the input words.
- Transformers are made up of stacks of **transformer blocks**.
- Each transformer block is a multilayer network that maps sequences of input vectors to sequences of output vectors of the same length.
- Transformer blocks are made by combining simple linear layers, feedforward networks, and **self-attention layers**.

# SELF-ATTENTION

---

- Self-attention allows a network to directly extract and use information from arbitrarily large contexts.
- Attention helps the large language model to understand the whole contents of the text as opposed to few words at a time.
- Words have rich linguistic relationships with words that can be many sentences away. Even within the sentence, words have important linguistic relationships with contextual words.
  - ▶ The **keys** to the cabinet **are** on the table.
  - ▶ The **chicken** crossed the road because **it** wanted to get to the other side.
  - ▶ I walked along the **pond**, and noticed that one of the trees along the **bank** had fallen into the **water** after the storm.

# SELF-ATTENTION



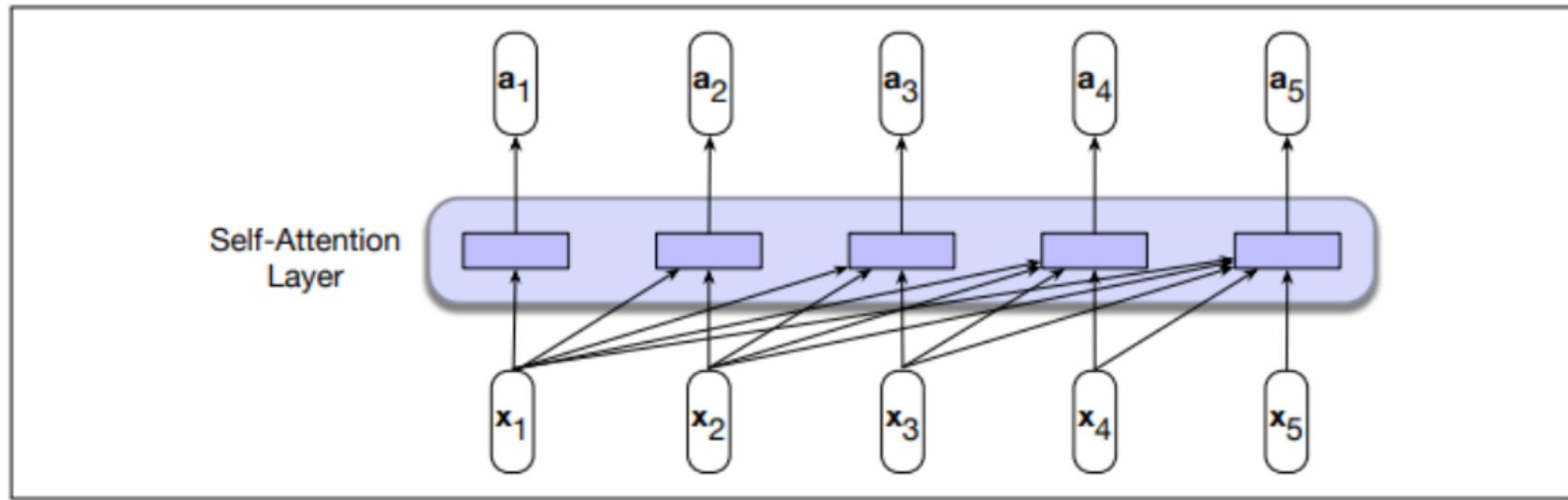
**Figure 10.1** The self-attention weight distribution  $\alpha$  that is part of the computation of the representation for the word *it* at layer 6. In computing the representation for *it*, we attend differently to the various words at layer 5, with darker shades indicating higher self-attention values. Note that the transformer is attending highly to *animal*, a sensible result, since in this example *it* corefers with the animal, and so we'd like the representation for *it* to draw on the representation for *animal*. Figure simplified from ([Uszkoreit, 2017](#)).

# CAUSAL, BACKWARD LOOKING SELF-ATTENTION

---

- A self-attention layer maps input sequences  $(x_1, \dots, x_n)$  to output sequences of the same length  $(a_1, \dots, a_n)$ .
- Create language models and use them for autoregressive generation.
  - ▶ When processing each item in the input, the model has access to all of the inputs up to and including the one under consideration, but no access to information about inputs beyond the current one.
- Parallelize both forward inference and training of such models
  - ▶ The computation performed for each item is independent of all the other computations.

# SELF-ATTENTION



**Figure 10.2** Information flow in a causal (or masked) self-attention model. In processing

# SELF-ATTENTION

- Compare a token to a collection of other tokens in a way that reveals their relevance in the current context.
- In self-attention for language, the set of comparisons are to other words (or tokens) within a given sequence. The result of these comparisons is then used to compute an output sequence for the current input sequence.

$$\text{score}(x_i, x_j) = x_i \cdot x_j$$

# SELF-ATTENTION

---

- Normalize the scores with a softmax to create a vector of weights,  $\alpha_{ij}$ , that indicates the proportional relevance of each input to the input element  $i$  that is the current focus of attention.

$$\begin{aligned}\alpha_{ij} &= \text{softmax(score}(x_i, x_j)) \quad \forall j \leq i \\ &= \frac{\exp(\text{score}(x_i, x_j))}{\sum_{k=1}^i \exp(\text{score}(x_i, x_k))} \quad \forall j \leq i\end{aligned}$$

- Generate output value  $a_i$  by summing the inputs seen so far, each weighted by its attention value.

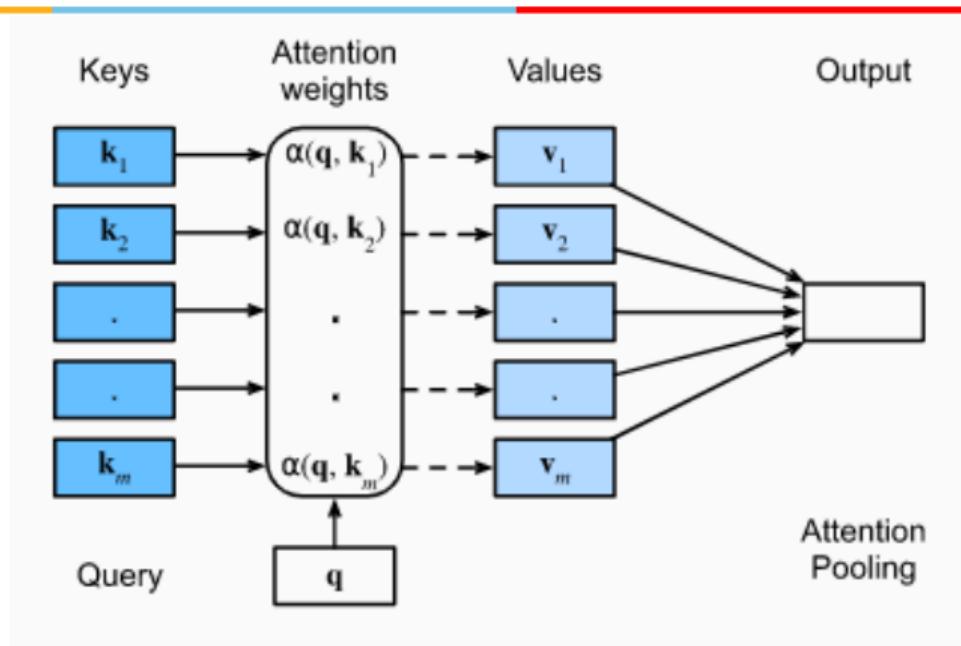
$$a_i = \sum_{j \leq i} \alpha_{ij} x_j$$

# SELF-ATTENTION

---

- The inputs  $x$  and outputs  $y$  of transformers, and attention vector  $a$ , all have the same dimensionality  $1 \times d$ .
- Let  $\mathcal{D} = \{(k_1, v_1), \dots, (k_m, v_m)\}$  represent  $m$  tuples of keys  $k$  and values  $v$ .
- Minibatches for computing attention for  $n$  queries and  $m$  key-value pairs.
- Queries and keys are of length  $d$  and values are of length  $v$ .
- Let  $q$  be a query.

# SELF-ATTENTION



**FIGURE:** The attention mechanism computes a linear combination over values  $v_i$  via attention pooling, where weights are derived according to the compatibility between a query  $q$  and keys  $k_i$ .

# SELF-ATTENTION

---

- Define the attention

$$\text{Attention}(q, \mathcal{D}) = a_i = \sum_{i=1}^m \alpha(q, k_i) v_i$$

- $\alpha(q, k_i)$  are scalar attention weights.
- The operation is referred to as **attention pooling**.
- The name **attention** derives from the fact that the operation pays particular attention to the terms for which the weight  $\alpha$  is significant (i.e., large).

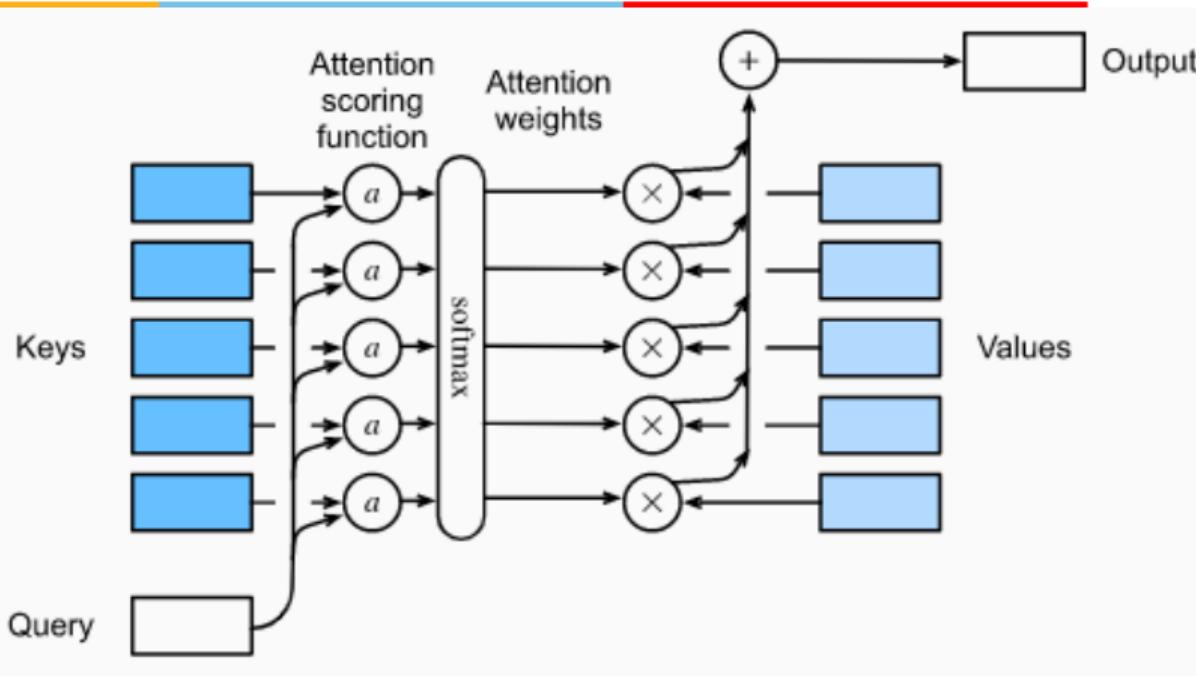
# SELF-ATTENTION

---

- All weights are equal,  $\alpha(q, k_i) = \frac{1}{m} \quad \forall i$ . This is average pooling in deep learning.
- The weights  $\alpha(q, k_i)$  are nonnegative. The output of the attention mechanism is contained in the convex cone spanned by the values  $v_i$ .
- The weights  $\alpha(q, k_i)$  form a convex combination, i.e.,  $\sum_i \alpha(q, k_i) = 1$  and  $\alpha(q, k_i) > 0 \quad \forall i$ . This is the most common setting in deep learning.
- Apply the softmax operation to ensure that weights are non-negative. Ensure that the weights sum up to 1 by normalizing.

$$\alpha(q, k_i) = \frac{\exp(\alpha(q, k_i))}{\sum_j \exp(\alpha(q, k_j))}$$

# SELF-ATTENTION



**FIGURE:** Computing the output of attention pooling as a weighted average of values, where weights are computed with the attention scoring function  $\alpha$  and the softmax operation.

# SELF-ATTENTION

---

- Query = current focus of attention  $q_i = x_i W^Q$
- Key = preceding input  $k_i = x_i W^K$
- Value  $v_i = x_i W^V$
- Score

$$score(x_i, x_j) = \frac{q_i \cdot k_j}{\sqrt{d}}$$

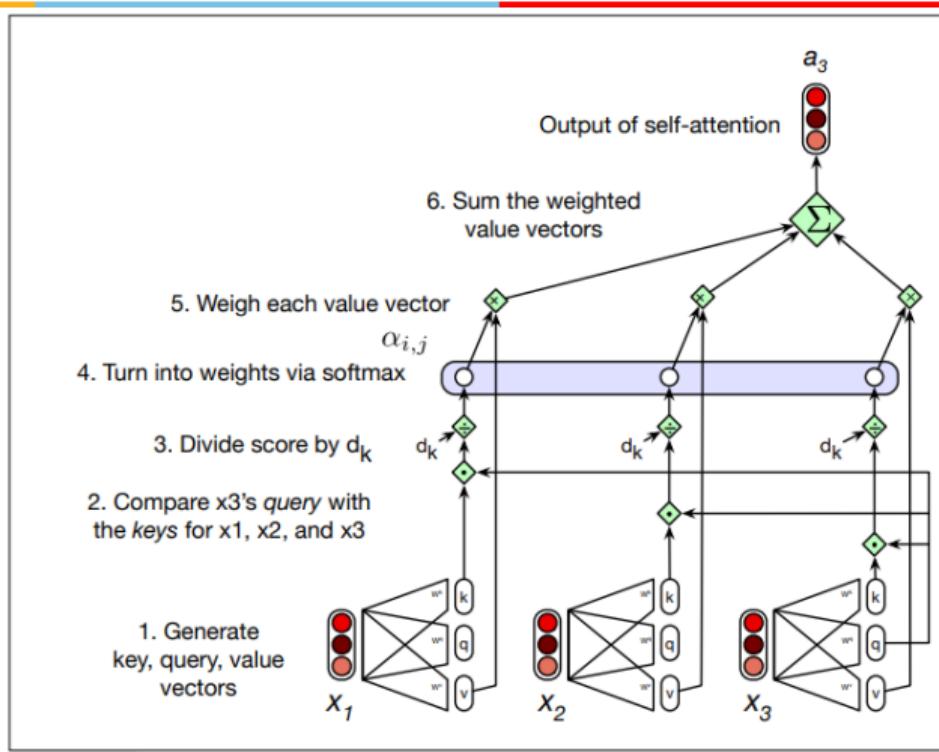
- Attention weight  $\alpha \in \mathbb{R}^{n \times v}$

$$\alpha_{ij} = softmax(score(x_i, x_j))$$

- Attention output

$$a_i = \sum_{j \leq i} \alpha_{ij} v_j$$

# SELF-ATTENTION



**Figure 10.3** Calculating the value of  $a_3$ , the third element of a sequence using causal (left-to-right) self-attention.

# PARALLELIZING SELF-ATTENTION

- Input  $X \in \mathbb{R}^{n \times d}$
- Query  $\mathbf{Q} \in \mathbb{R}^{n \times d}$
- Key  $\mathbf{K} \in \mathbb{R}^{n \times d}$
- Value  $\mathbf{V} \in \mathbb{R}^{n \times v}$

$$\mathbf{Q} = XW^Q$$

$$\mathbf{K} = XW^K$$

$$\mathbf{V} = XW^V$$

$$\mathbf{A} = \text{SelfAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}$$

# MASKED SOFTMAX OPERATION

- To deal with sequences of different lengths, padding with dummy tokens for shorter sequences is necessary
  - ▶ Dive into Deep Learning
  - ▶ Learn to code  $< \text{blank} >$
  - ▶ Hello world  $< \text{blank} >< \text{blank} >$
- Limit

$$\sum_{i=1}^n \alpha(q, k_i) v_i \quad \text{to} \quad \sum_{i=1}^l \alpha(q, k_i) v_i \quad \text{where } l \leq n$$

This is masked softmax operation.

- Implementation: set the values of  $v_i$  for  $i > l$  to zero.

# MASK THE FUTURE TOKENS

- The calculation in  $\mathbf{Q}\mathbf{K}^T$  results in a score for each query value to every key value, including those that follow the query. This is undesirable.
- To fix this, the elements in the upper-triangular portion of the matrix are zeroed out, thus eliminating any knowledge of words that follow in the sequence.
  - ▶ Dive into Deep Learning
  - ▶ Learn to code  $< \text{blank} >$
  - ▶ Hello world  $< \text{blank} >< \text{blank} >$
- Limit

$$\sum_{i=1}^n \alpha(q, k_i) v_i \quad \text{to} \quad \sum_{i=1}^l \alpha(q, k_i) v_i \quad \text{where } l \leq n$$

This is masked softmax operation.

- Implementation: set the values of  $v_i$  for  $i > l$  to zero.

# MASK THE FUTURE TOKENS

	q1·k1	$-\infty$	$-\infty$	$-\infty$	$-\infty$
	q2·k1	q2·k2	$-\infty$	$-\infty$	$-\infty$
N	q3·k1	q3·k2	q3·k3	$-\infty$	$-\infty$
	q4·k1	q4·k2	q4·k3	q4·k4	$-\infty$
	q5·k1	q5·k2	q5·k3	q5·k4	q5·k5

**Figure 10.4** The  $N \times N$   $\mathbf{QK}^T$  matrix showing the  $q_i \cdot k_j$  values, with the upper-triangle portion of the comparisons matrix zeroed out (set to  $-\infty$ , which the softmax will turn to zero).

# MULTIHEAD ATTENTION

---

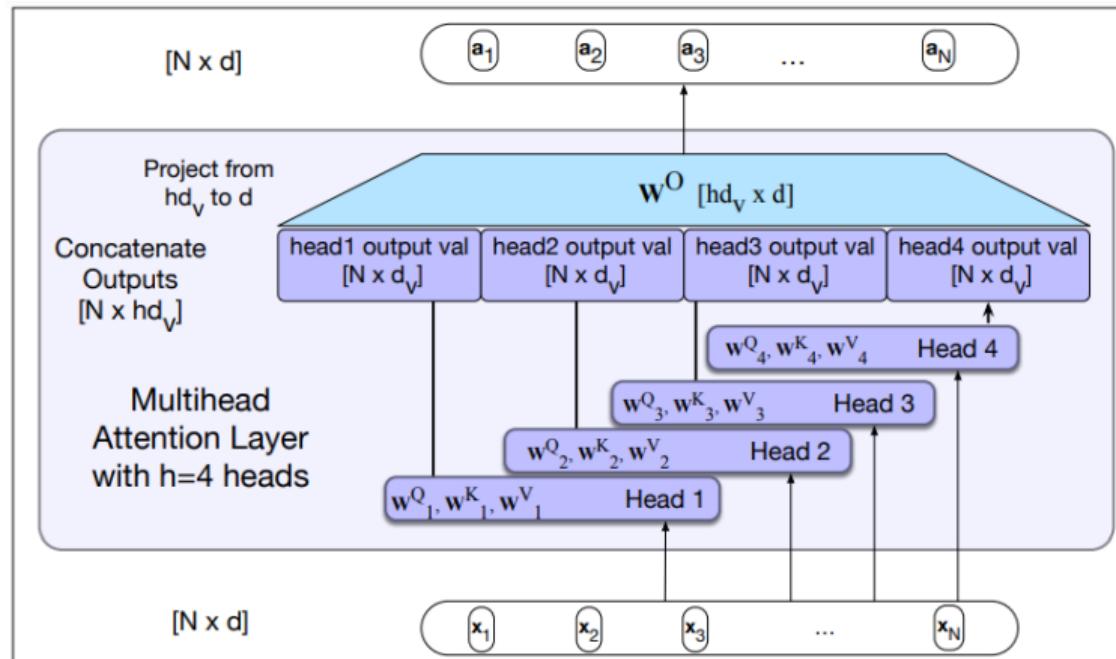
- Different words in a sentence can relate to each other in many different ways simultaneously.
- To capture all of the different kinds of parallel relations among its inputs, use multihead self-attention layers.
- **Multihead self-attention layers** are sets of self-attention layers, called **heads**, that reside in parallel layers at the same depth in a model, each with its own set of parameters. Each head can learn different aspects of the relationships among inputs at the same level of abstraction, using these parameters.

# MULTIHEAD ATTENTION

---

- Each head  $i$  in a self-attention layer is provided with its own set of key, query and value matrices:  $W_i^K$ ,  $W_i^Q$  and  $W_i^V$ . These are used to project the inputs into separate key, value, and query embeddings separately for each head.
- Input and output dimension -  $d$
- Key and query embeddings have dimensionality  $d_k$
- Value embeddings are of dimensionality  $d_v$
- Number of heads -  $h$
- Original paper -  $d_k = d_v = 64$ ,  $h = 8$ ,  $d = 512$

# MULTIHEAD ATTENTION



**Figure 10.5** Multihead self-attention: Each of the multihead self-attention layers is provided with its own

# MULTIHEAD ATTENTION

- Input  $X$  has a dimension of  $N \times d$

$$W_i^Q \in \mathbb{R}^{d \times d_k}$$

$$W_i^K \in \mathbb{R}^{d \times d_k}$$

$$W_i^V \in \mathbb{R}^{d \times d_v}$$

$$Q \in \mathbb{R}^{N \times d_k}$$

$$K \in \mathbb{R}^{N \times d_k}$$

$$V \in \mathbb{R}^{N \times d_v}$$

- Output of each of the  $h$  heads is of shape  $N \times d_v$ .
- Output of the multi-head layer with  $h$  heads consists of  $h$  matrices.

# MULTIHEAD ATTENTION

---

- $h$  matrices are concatenated to produce a single output with dimensionality  $N \times hd_v$ .
- Another linear projection  $W^O \in \mathbb{R}^{hd_v \times d}$  that reshape it to the original output dimension for each token.
- Multiplying the concatenated  $N \times hd_v$  matrix output by  $W^O \in \mathbb{R}^{hd_v \times d}$  yields the self-attention output  $A$  of shape  $N \times d$ , suitable to be passed through residual connections and layer norm.

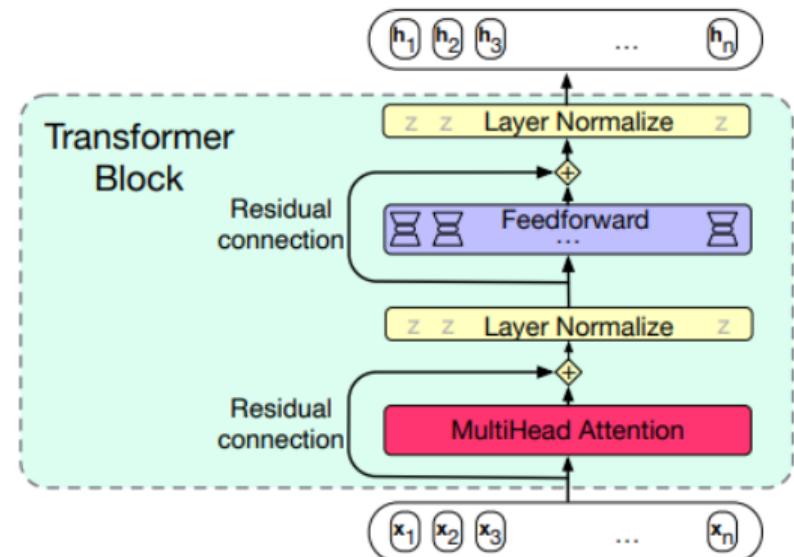
$$Q = XW_i^Q \quad K = XW_i^K \quad V = XW_i^V$$

$$\text{head}_i = \text{SelfAttention}(Q, K, V)$$

$$A = \text{Multihead}(X) = (\text{head}_1 \oplus \dots \oplus \text{head}_h)W^O$$

# TRANSFORMER BLOCK

- Self-attention layer
- Feedforward layer
- Residual connections
- Normalizing layers (layer norm)



# FEEDFORWARD LAYER

---

- The feedforward layer contains  $N$  position-wise networks, one at each position.
- Each is a fully-connected 2-layer network, i.e., one hidden layer, two weight matrices.
- The weights are the same for each position, but the parameters are different from layer to layer.
- Feedforward networks are independent for each position and so can be computed in parallel.
- Dimensionality  $d_{ff}$  of the hidden layer of the feedforward network be larger than the model dimensionality  $d$ . (original transformer model,  $d = 512$  and  $d_{ff} = 2048$ )

# RESIDUAL CONNECTIONS

---

- Residual connections are connections that pass information from a lower layer to a higher layer without going through the intermediate layer.
- Improves learning and gives higher level layers direct access to information from lower layers.
- Residual connections in transformers are implemented simply by adding a layer's input vector to its output vector before passing it forward.
- Residual connections are used with both the attention and feedforward sublayers.

# LAYER NORM

---

- Summed vectors are then normalized using layer normalization.
- Improve training performance in deep neural networks by keeping the values of a hidden layer in a range that facilitates gradient-based training.
- Layer norm is a variation of z-score, applied to a single vector in a hidden layer.
- The input to layer norm is a single vector of dimensionality  $d$ , for a particular token position  $i$ , and the output is that vector normalized. Output is a single vector of dimensionality  $d$ .
- Calculate the mean  $\mu$  and standard deviation  $\sigma$ , over the elements of the vector to be normalized. Assume a  $d_h$ .

# LAYER NORM

$d_h$  = dimensionality of hidden layer

$$\mu = \frac{1}{d_h} \sum_{i=1}^{d_h} x_i$$

$$\sigma = \sqrt{\frac{1}{d_h} \sum_{i=1}^{d_h} (x_i - \mu)^2}$$

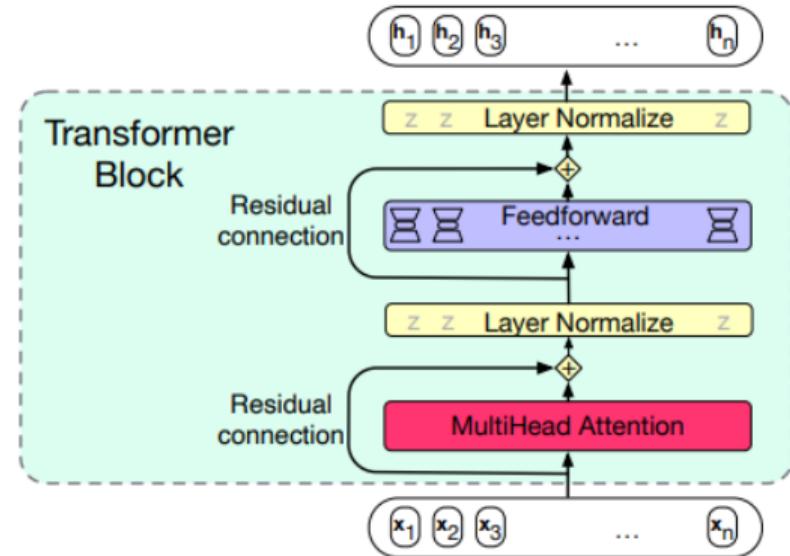
$$\hat{x} = \frac{x - \mu}{\sigma}$$

$$LayerNorm = \gamma \hat{x} + \beta$$

In standard implementation of layer normalization, two parameters,  $\gamma$  and  $\beta$  representing gain and offset values are learned.

# TRANSFORMER BLOCK

$$Z = \text{LayerNorm}(X + \text{SelfAttention}(X))$$
$$H = \text{LayerNorm}(Z + \text{FFN}(Z))$$

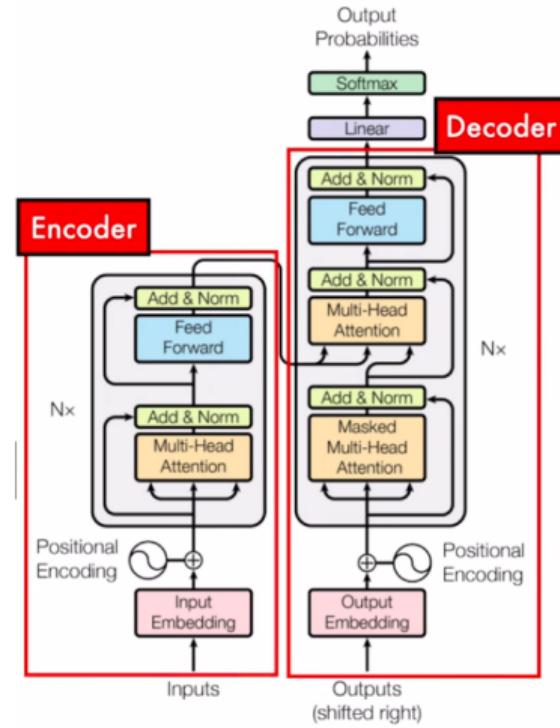


Transformers for large language models stack many of these blocks, from 12 layers (used for the T5 or GPT-3-small language models) to 96 layers (used for GPT-3 large).

# TRANSFORMER

ENCODER learns useful representation of input.

DECODER decodes encoded representation and combines with other input to predict output.

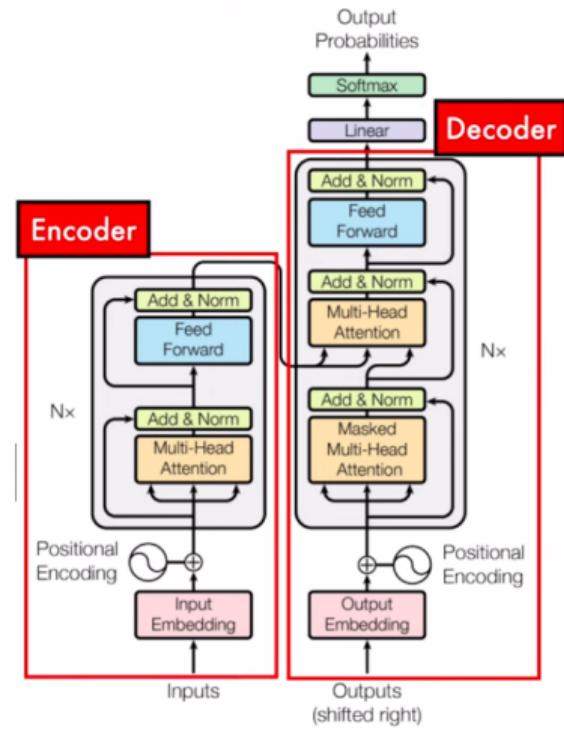


# TRANSFORMER

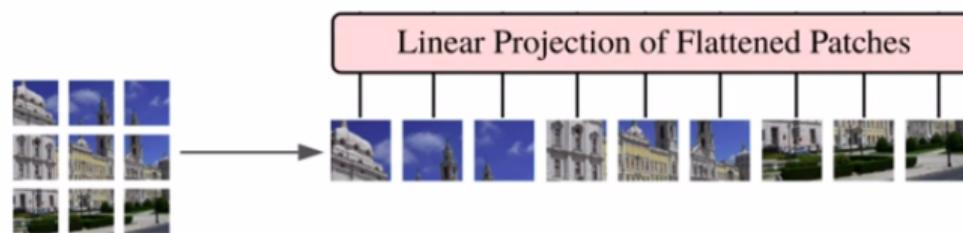
ENCODER ONLY learning representations. eg:  
BERT

DECODER ONLY generate tokens eg: GPT-3

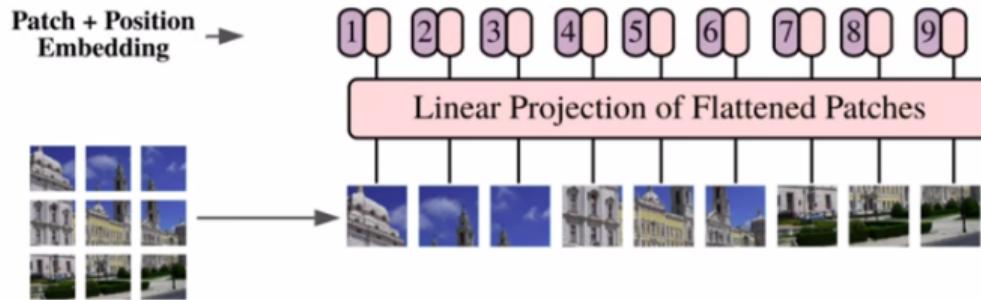
ENCODER-DECODER sequence-to-sequence



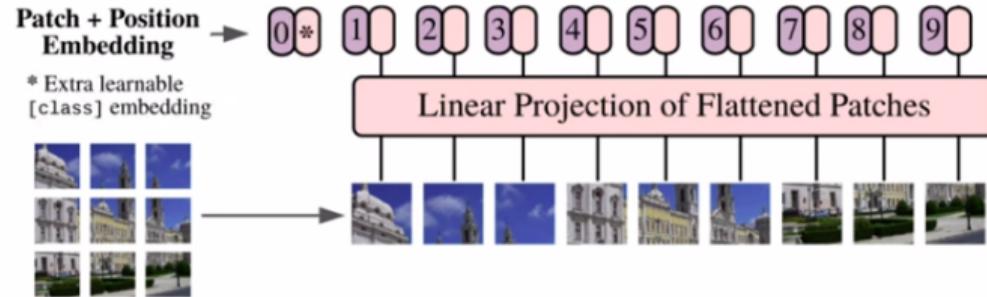
# VIT - ENCODER



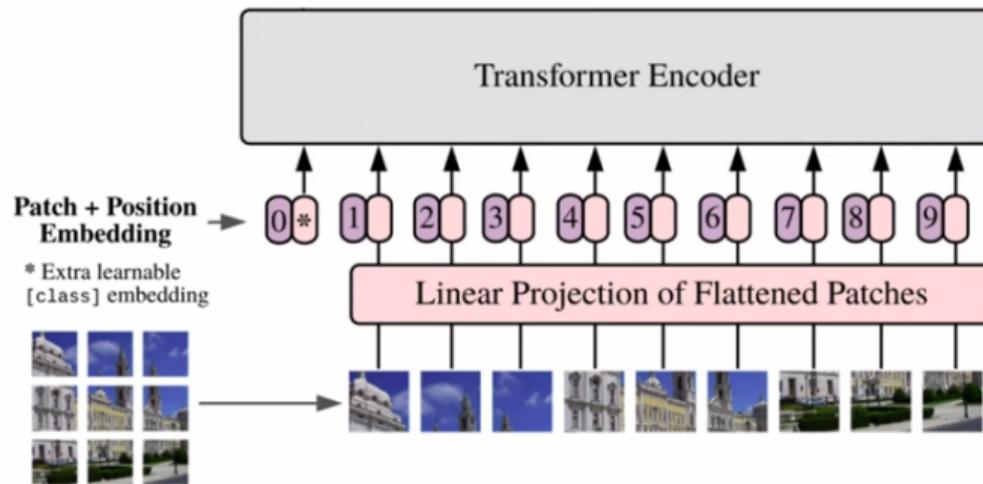
# ViT - ENCODER



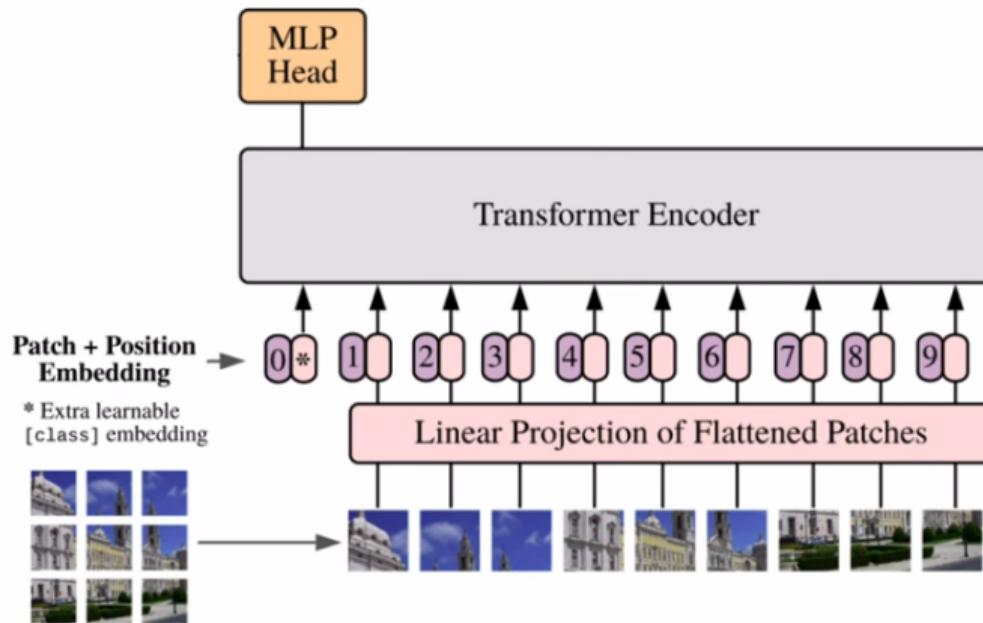
# VIT - ENCODER



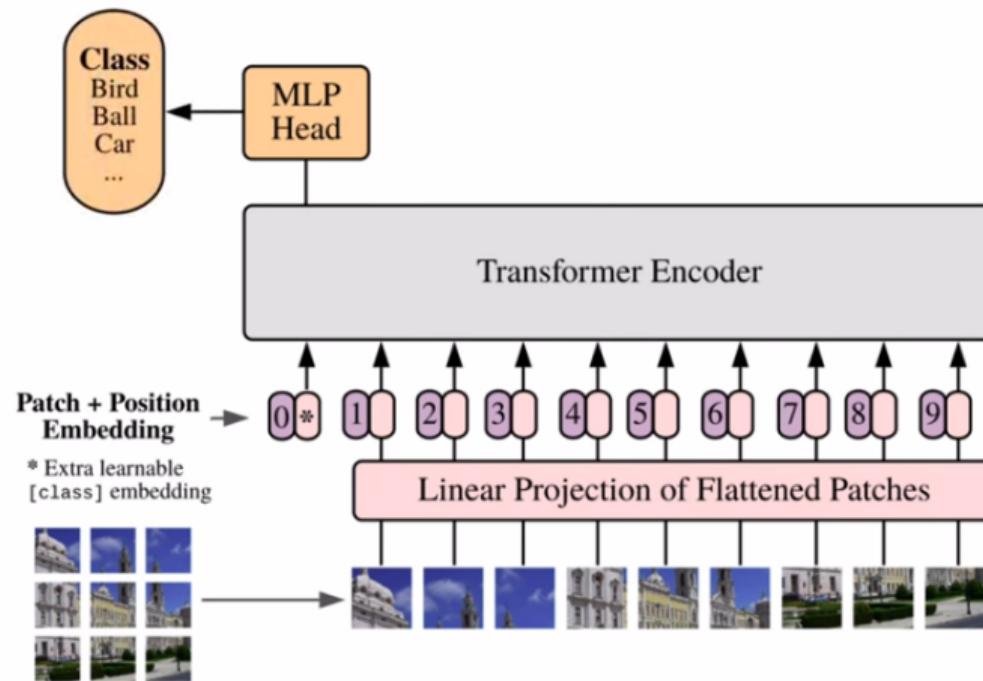
# VIT - ENCODER



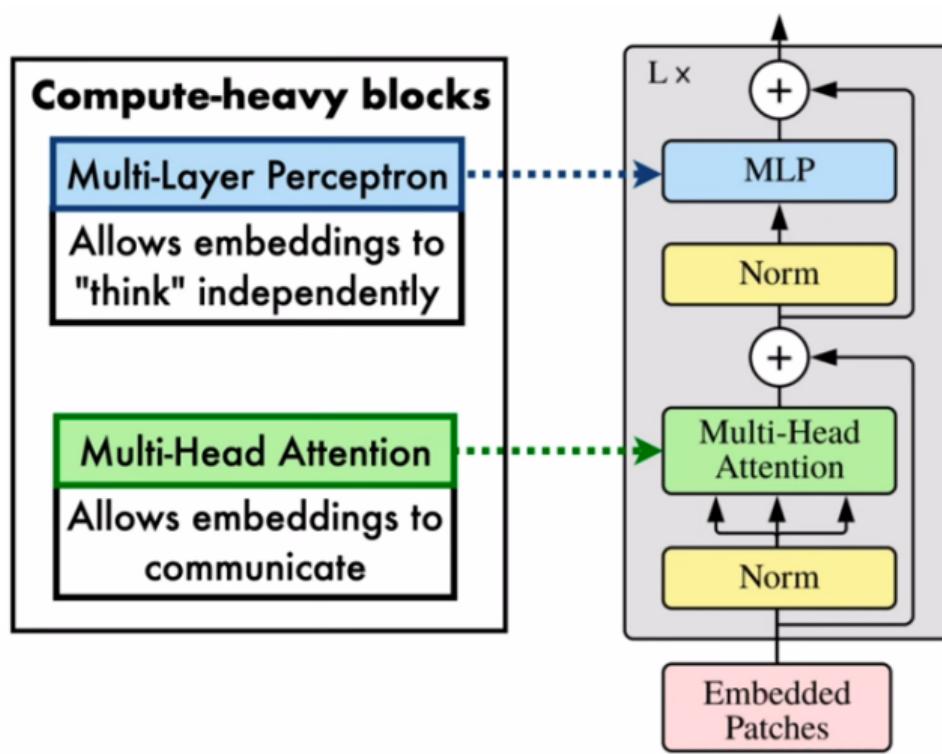
# VIT - ENCODER



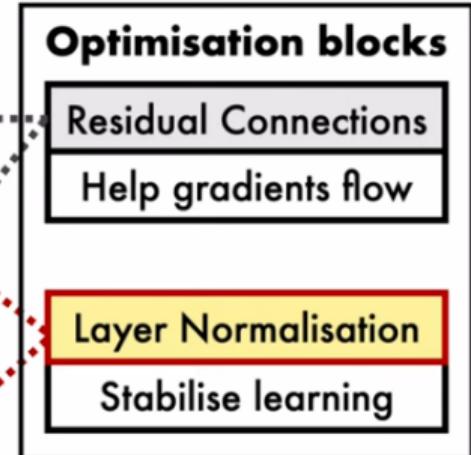
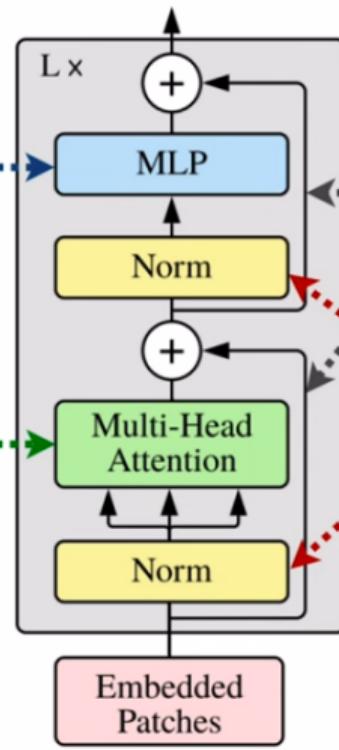
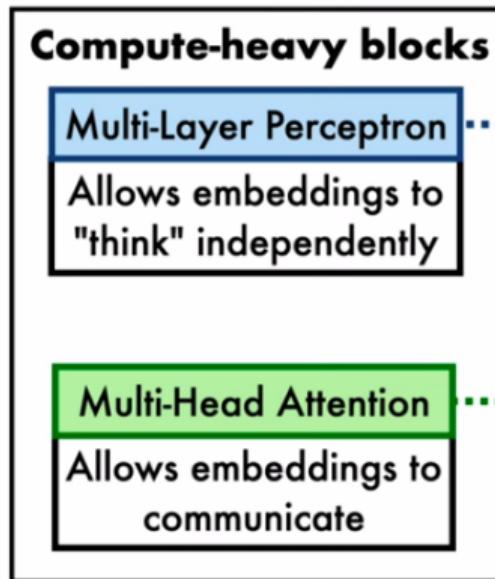
# VIT - ENCODER



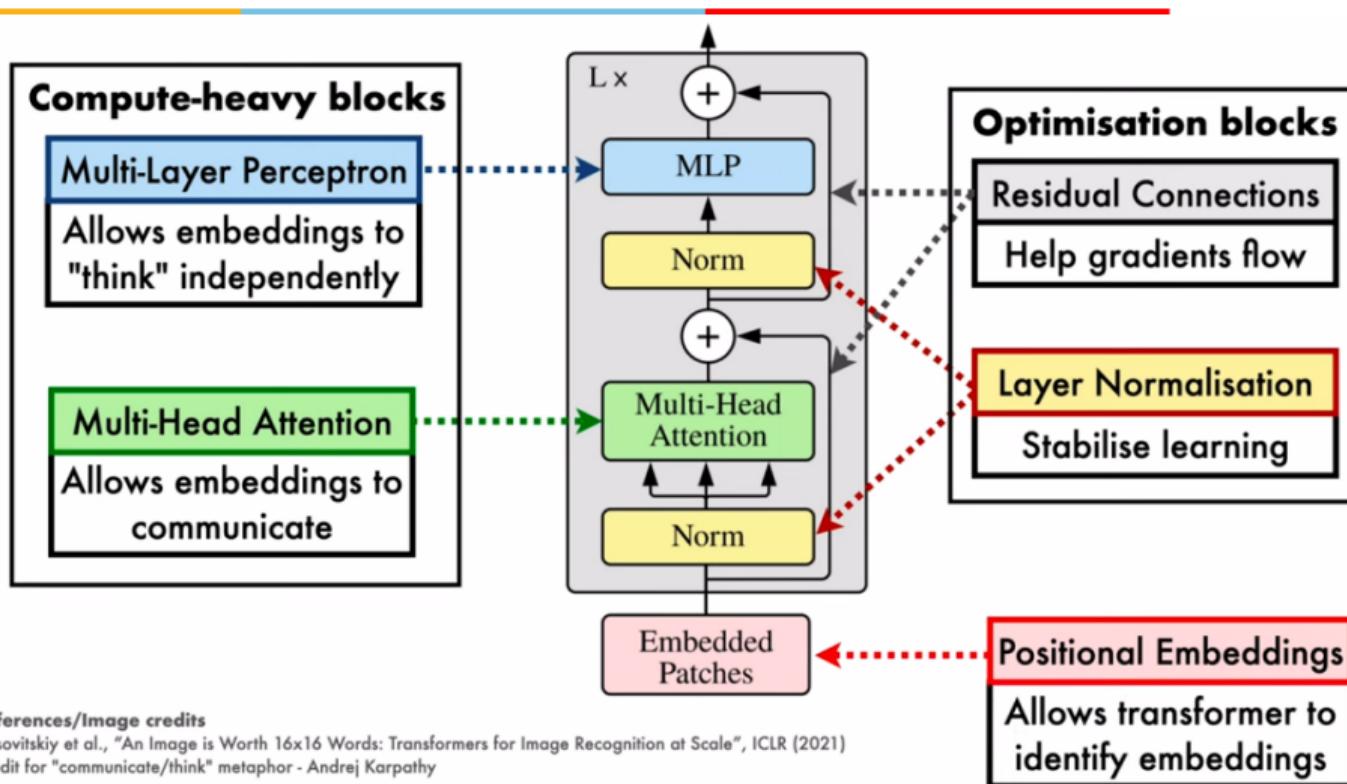
# TRANSFORMER ENCODER



# TRANSFORMER ENCODER



# TRANSFORMER ENCODER



ferences/Image credits

sovit斯基 et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR (2021)  
edit for "communicate/think" metaphor - Andrej Karpathy

# TABLE OF CONTENTS

---

- 1 MODULE 7 TOPICS
- 2 VISION TRANSFORMER
- 3 3D CONVOLUTIONAL NEURAL NETWORK (3D CNN)
- 4 TWO-STREAM CONVOLUTIONAL NETWORKS
- 5 SIAMESE NETWORK

# 3D CNN

---

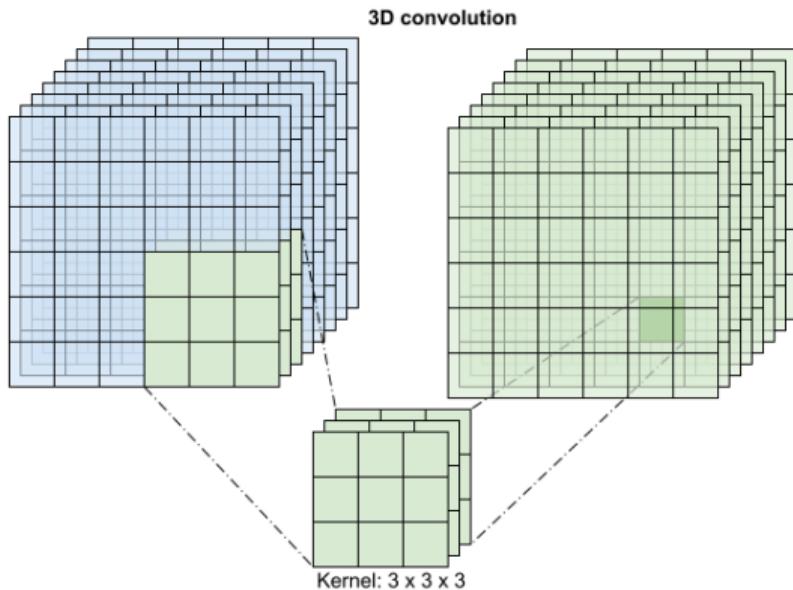
- 3D CNN is used in three-dimensional data, such as medical volumetric images (e.g., CT scans, MRI scans) or video sequences.
- 2D CNNs operate on two-dimensional data (e.g., images), 3D CNNs process volumetric data and are designed to capture spatial and temporal dependencies in 3D images.

# 3D CNN

---

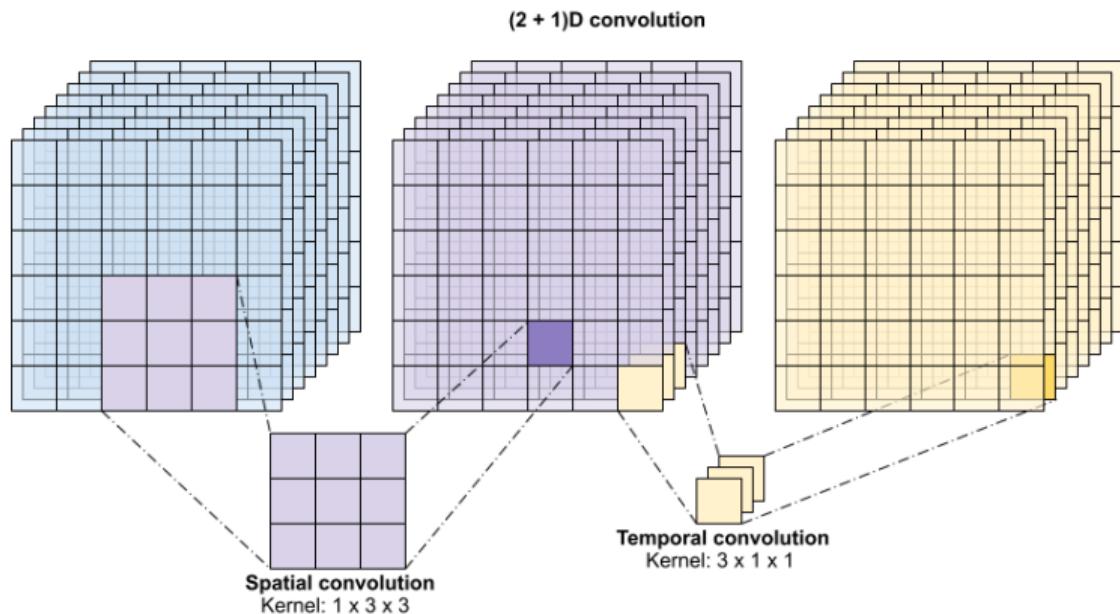
- A 3D CNN uses a three-dimensional filter to perform convolutions.
- The kernel is able to slide in three directions, whereas in a 2D CNN it can slide in two dimensions.
- Instead of operating on a single image with dimensions (height, width), these operate on video volume (time, height, width).
- Replace each 2D convolution (layers.Conv2D) with a 3D convolution (layers.Conv3D).

# 3D CNN



3D convolution layer with a kernel size of  $(3 \times 3 \times 3)$  would need a weight-matrix with  $27 * \text{channels}^{**} 2$  entries.

# (2 + 1)D CONVOLUTION



In the (2 + 1)D convolution the spatial convolution takes in data of the shape (1, width, height), while the temporal convolution takes in data of the shape (time, 1, 1).

A (2 + 1)D convolution with kernel size (3 x 3 x 3) would need weight matrices of size  $(9 * \text{channels}^{**2}) + (3 * \text{channels}^{**2})$

# 3D CNN LAYERS

---

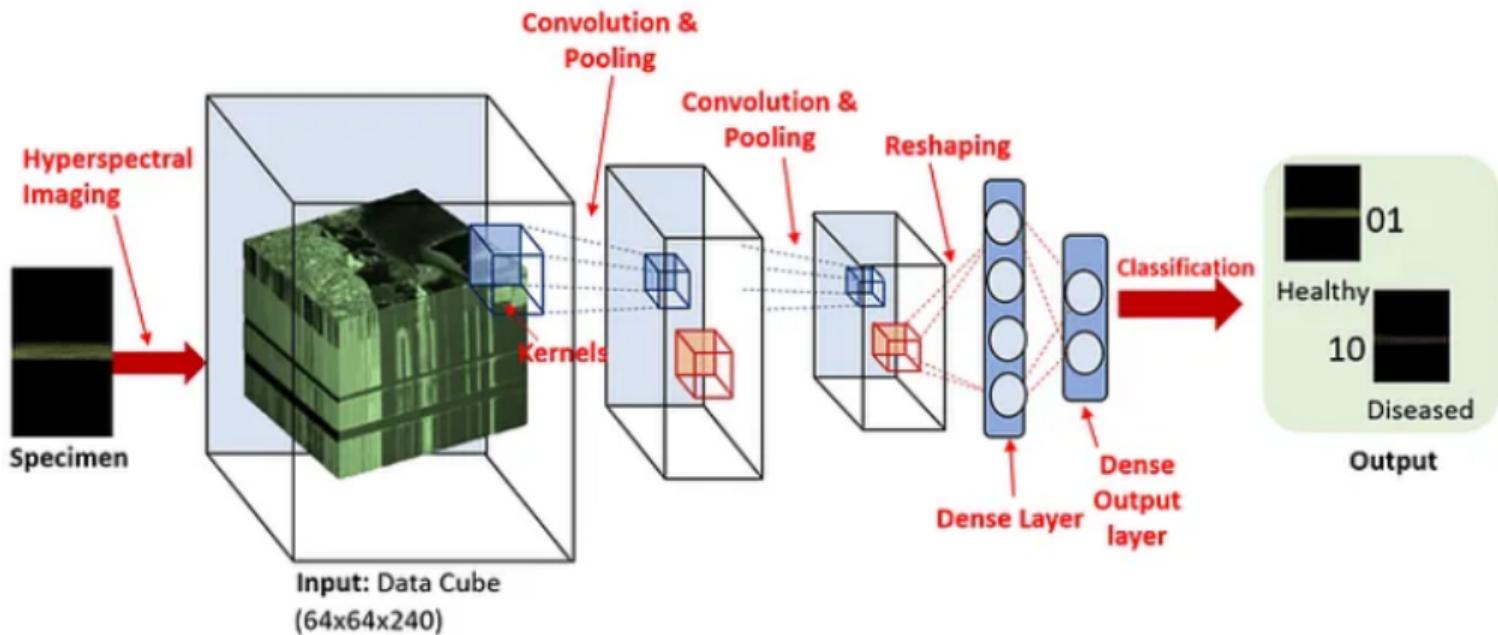
**3D CONVOLUTION** layers extract features from volumetric data. These layers slide a 3D kernel over the input volume to detect patterns in all three dimensions.

**POOLING AND STRIDING** use 3D max-pooling layers and strides to downsample the spatial dimensions of the data, reducing the computational load.

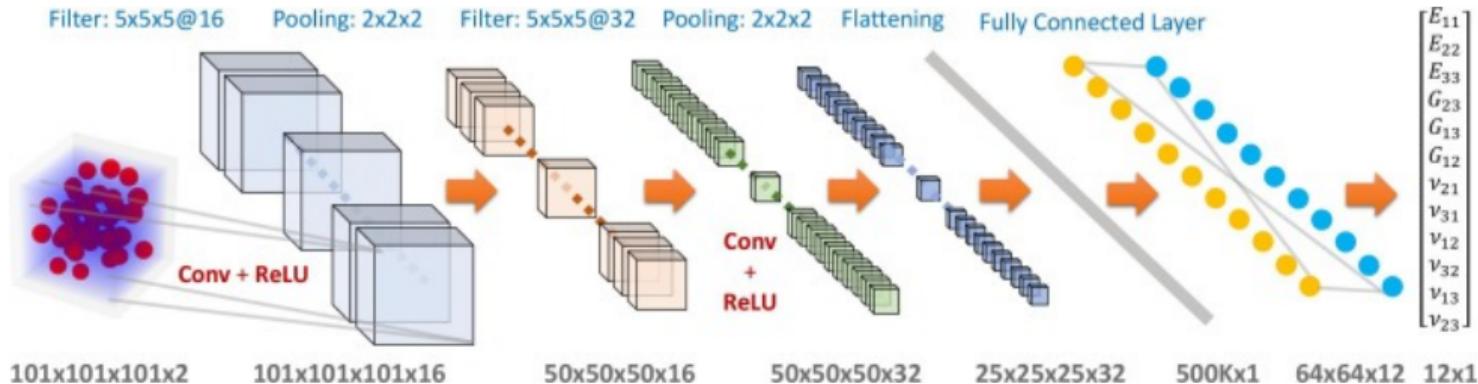
**SKIP CONNECTIONS** can be applied to 3D CNNs to improve segmentation accuracy.

**FULLY CONNECTED LAYERS** At the end of the network, fully connected layers are often used for classification or regression, depending on the segmentation task.

# 3D CNN



# 3D CNN



# 3D CNN ARCHITECTURE

---

**V-NET** for medical image segmentation. It includes skip connections and is known for its segmentation accuracy.

**3D U-NET** extends the popular 2D U-Net architecture to 3D data, making use of skip connections.

**3D RESNET** adaptations of the well-known ResNet architecture for 3D data, incorporating residual blocks to handle deep networks.

**3D DENSENET** connects each layer to every other layer, promoting feature reuse and gradient flow.

**3D INCEPTION** utilize multi-scale convolutional filters to capture features at various resolutions.

# TABLE OF CONTENTS

---

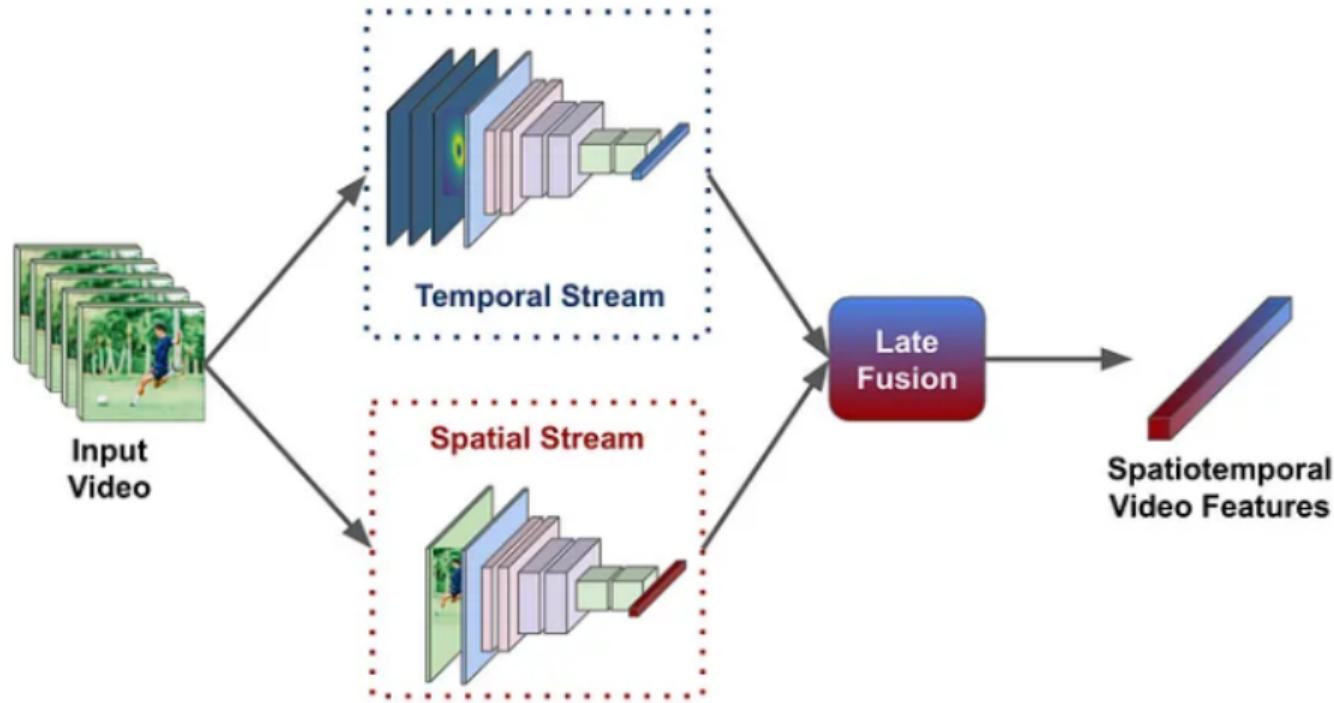
- 1 MODULE 7 TOPICS
- 2 VISION TRANSFORMER
- 3 3D CONVOLUTIONAL NEURAL NETWORK (3D CNN)
- 4 TWO-STREAM CONVOLUTIONAL NETWORKS
- 5 SIAMESE NETWORK

# TWO-STREAM CNN

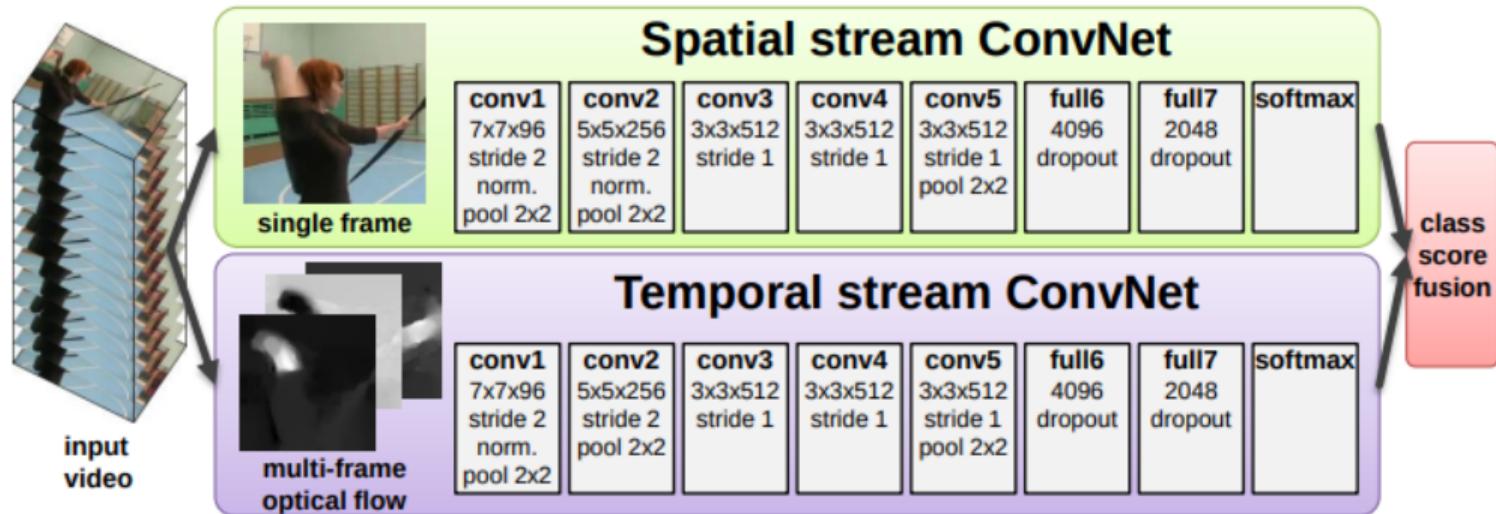
---

- Used for action recognition in videos.
- Video can naturally be decomposed into spatial and temporal components.
- The spatial part, individual frame appearance, carries information about scenes and objects depicted in the video.
- The temporal part, in the form of motion across the frames, conveys the movement of the observer (the camera) and the objects.

# TWO-STREAM CNN



# TWO-STREAM CNN



# TWO-STREAM CNN

---

- Spatial stream ConvNet operates on individual video frames, effectively performing action recognition from still images.
- Temporal stream ConvNet exploits motion and significantly improves accuracy. Motion representation inspired by the trajectory-based descriptors, replaces the optical flow, sampled at the same locations across several frames.
- SVM-based fusion of softmax scores outperforms fusion by averaging.

# MULTI-TASK LEARNING

---

- Learn a (video) representation, which is applicable not only to the task in question (HMDB-51 classification), but also to other tasks (UCF-101 classification).
- ConvNet architecture is modified so that it has two softmax classification layers on top of the last fullyconnected layer: one softmax layer computes HMDB-51 classification scores, and the UCF-101 scores.
- Each of the layers is equipped with its own loss function, which operates only on the videos, coming from the respective dataset.
- The overall training loss is computed as the sum of the individual tasks' losses, and the network weight derivatives can be found by back-propagation.

# TABLE OF CONTENTS

---

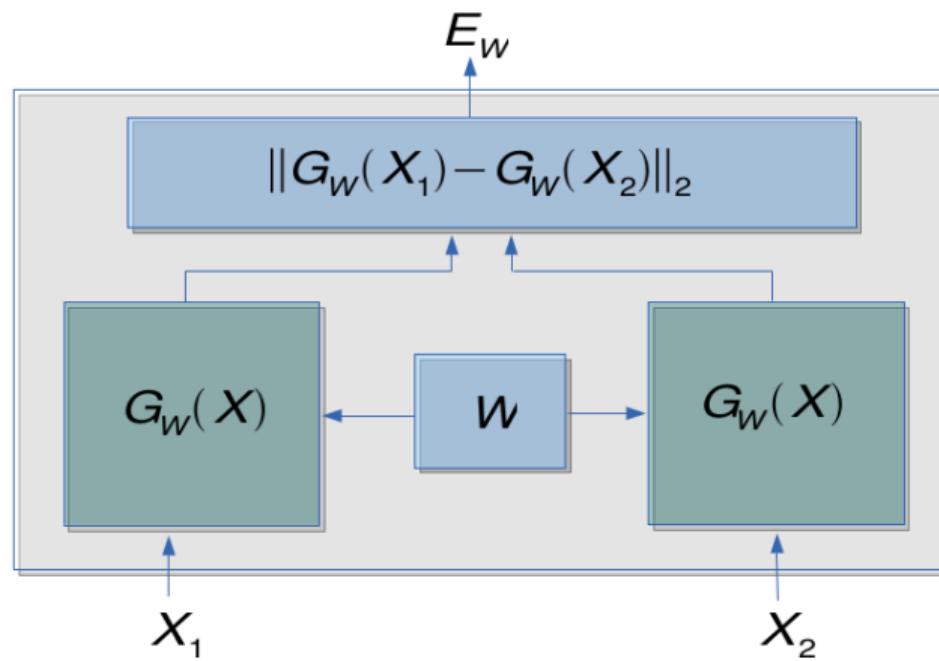
- 1 MODULE 7 TOPICS
- 2 VISION TRANSFORMER
- 3 3D CONVOLUTIONAL NEURAL NETWORK (3D CNN)
- 4 TWO-STREAM CONVOLUTIONAL NETWORKS
- 5 SIAMESE NETWORK

# SIAMESE NETWORK

---

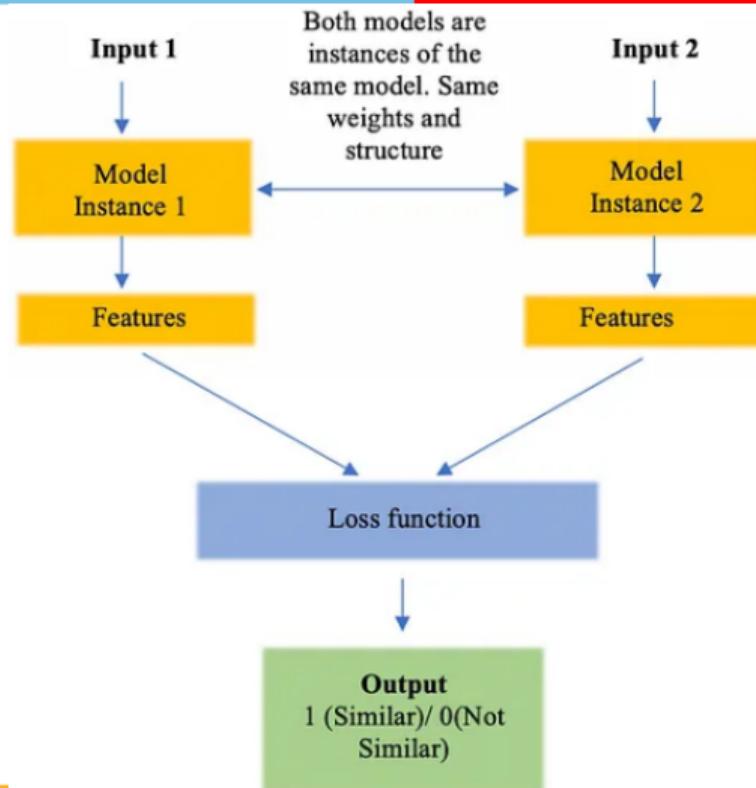
- Two identical networks.
- Feed a pair of inputs to these networks.
- Each network computes the features of one input.
- Similarity of features is computed using their difference or the dot product.
- For same class input pairs, target output is 1 and for different classes input pairs, the output is 0.
- The identical networks share the parameters.

# SIAMESE NETWORK



Siamese Architecture [Bromley, Sackinger, Shah, LeCun 1994]

# SIAMESE NETWORK



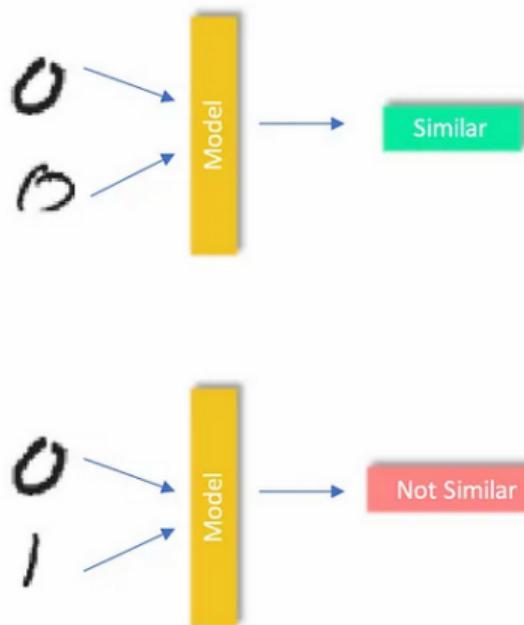
# SIAMESE NETWORK - CONTRASTIVE LOSS

---

- Image pairs are compared using Contrastive loss.
- Distance is less between same class pair and distance is more for different pairs, distance is more.

Ref: <https://towardsdatascience.com/siamese-networks-introduction-and-implementation-2140e3443dee>

# SIAMESE NETWORK - CONTRASTIVE LOSS

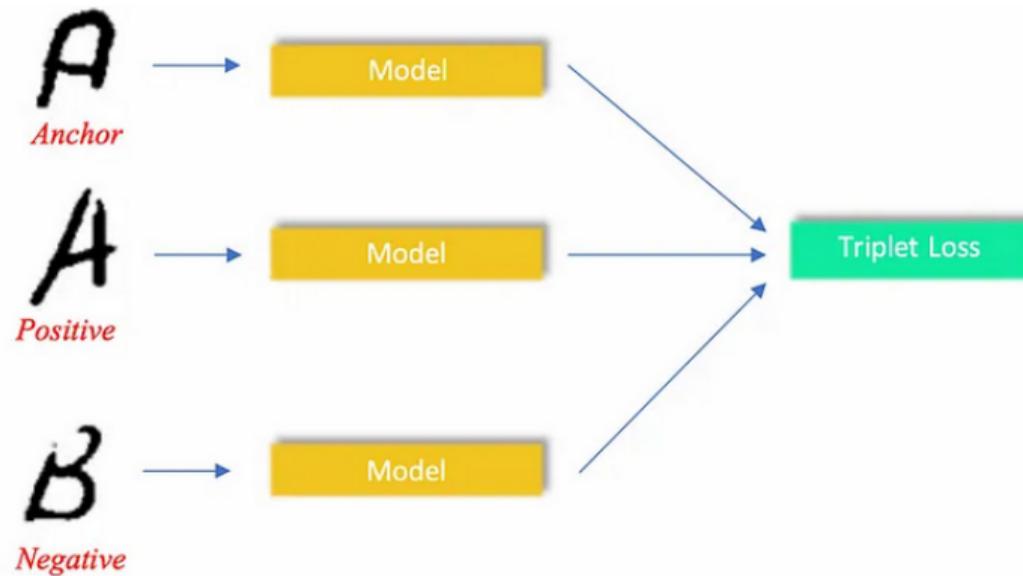


# SIAMESE NETWORK - TRIPLET LOSS

---

- Three buckets of data are required – anchor, positive, and negative are taken by the model.
- The anchor is a reference input.
- Positive input and anchor input belongs to same class.
- Random class is assigned with negative inputs.

# SIAMESE NETWORK - TRIPLET LOSS

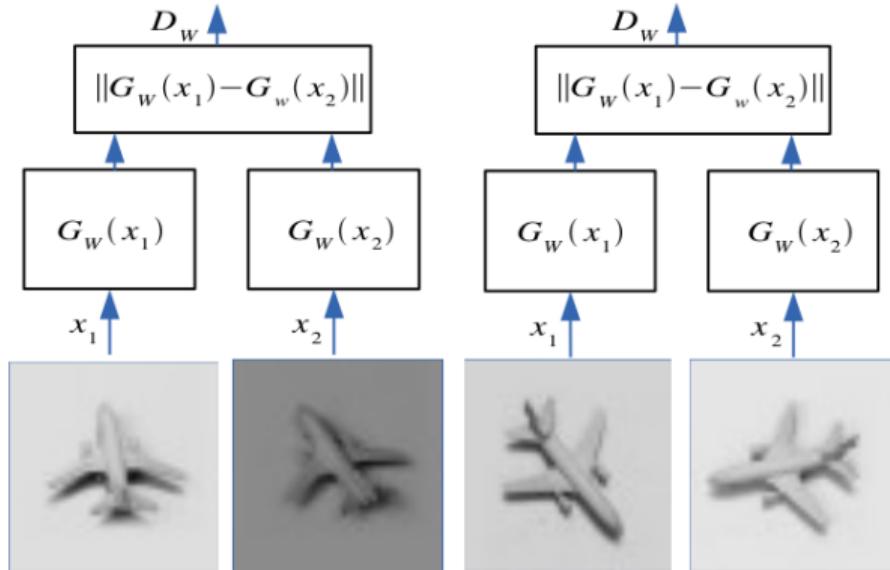


# SIAMESE NETWORK ON IMAGES

## Loss function:

- Outputs corresponding to input samples that are neighbors in the neighborhood graph should be nearby
- Outputs for input samples that are not neighbors should be far away from each other

Make this small



Make this large

## REFERENCES

- ① Dosovitskiy, A., et.al (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
- ② Vaswani, A., et.al (2017). Attention is all you need. Advances in neural information processing systems, 30.
- ③ Liu, Z., et.al (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 10012-10022).
- ④ A Closer Look at Spatiotemporal Convolutions for Action Recognition by D. Tran et al. (2017)
- ⑤ [https://proceedings.neurips.cc/paper\\_files/paper/1993/file/288cc0ff022877bd3df94bc9360b9c5d-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/1993/file/288cc0ff022877bd3df94bc9360b9c5d-Paper.pdf)
- ⑥ <https://proceedings.neurips.cc/paper/2014/file/00ec53c4682d36f5c4359f4ae7bd7ba1-Paper.pdf>

Thank You!