AY 2024/2025 S1

EE6427 Video Signal Processing

Part 5 Video Analysis and Understanding

Dr Yap Kim Hui

Room: S2-B2b-53
Tel: 6790 4339
Email: ekhyap@ntu.edu.sg

References

- Xiong et. al., “Recent Advances in Deep Learning for Object Detection,” Neurocomputing 2020.
- Zaidi et. al., “Survey of Modern Deep Learning based Object Detection Models,” CoRR 2021. https://arxiv.org/abs/2104.11892
- Gioele Ciaparrone, “Deep Learning in Video Multi-Object Tracking: A Survey,” Neurocomputing, 2020.
- Ce Zheng et. al., “Deep Learning-Based Human Pose Estimation: A Survey,” CoRR 2020. https://doi.org/10.48550/arXiv.2012.13392
- Yi Zhu et. al., “A Comprehensive Study of Deep Video Action Recognition,” CoRR 2020. https://doi.org/10.48550/arXiv.2012.06567

Part 5 Outline

- Video Analysis & Understanding
  - Object Detection & Tracking
  - Pose Estimation
  - Video / Human Action Recognition

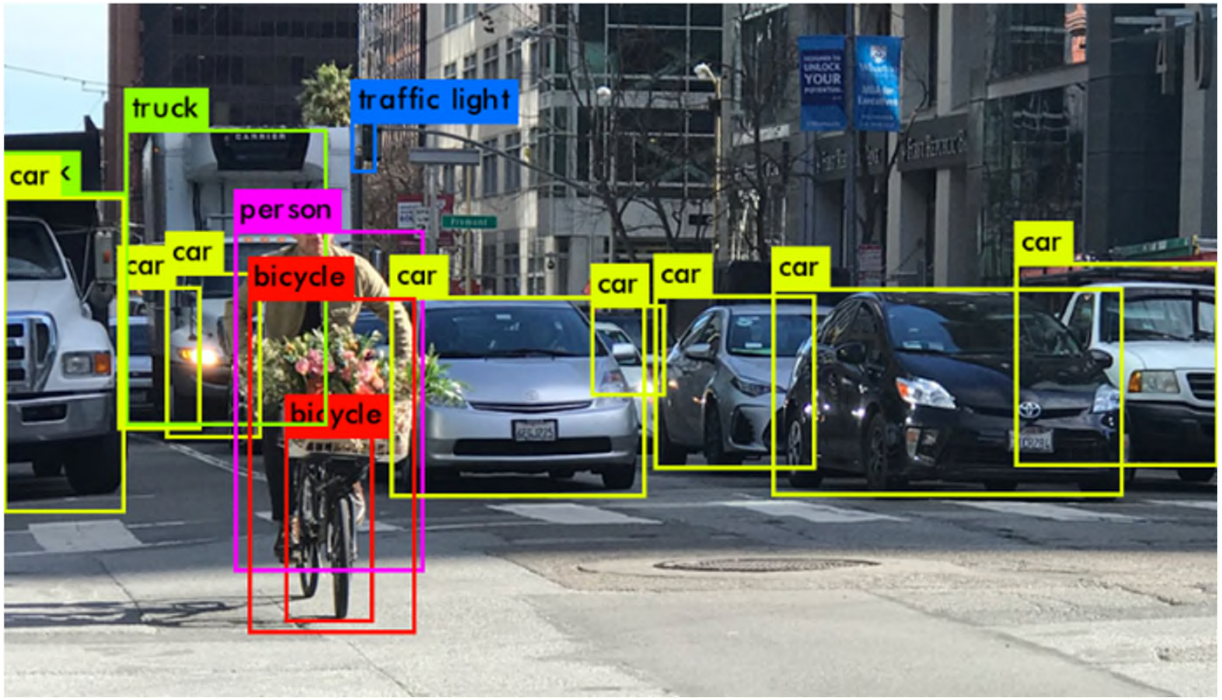
Object Detection

Object Detection Overview

- Introduction
- Object Detection Methods
  - Two Stage Detectors
  - One Stage Detectors
  - Lightweight Detectors
- New / Emerging Methods

Objective

- To predict object classes in an image / video and localize them using bounding boxes.
- A regression + classification task.



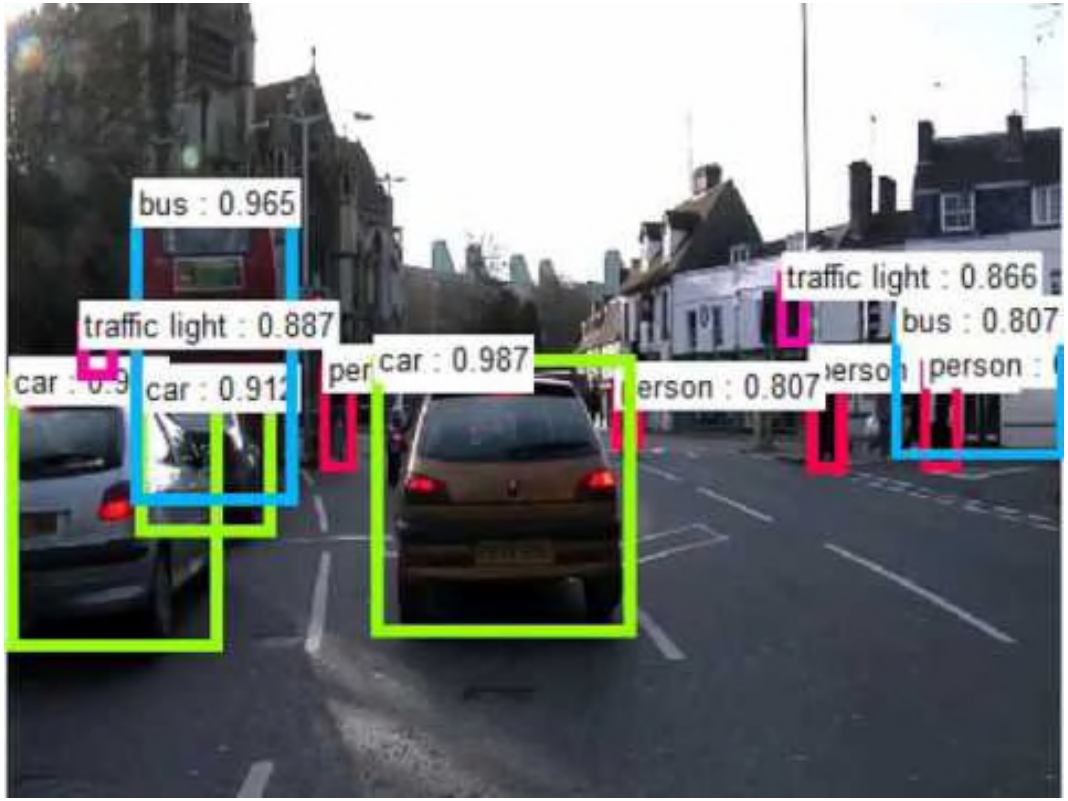
Source: https://17bce011.medium.com/complete-guide-to-object-detection-using-deep-learning-23ffc99ab072

Applications

- A key step for many downstream CV tasks:
  - Object tracking
  - Pose estimation
  - Action recognition
  - etc.

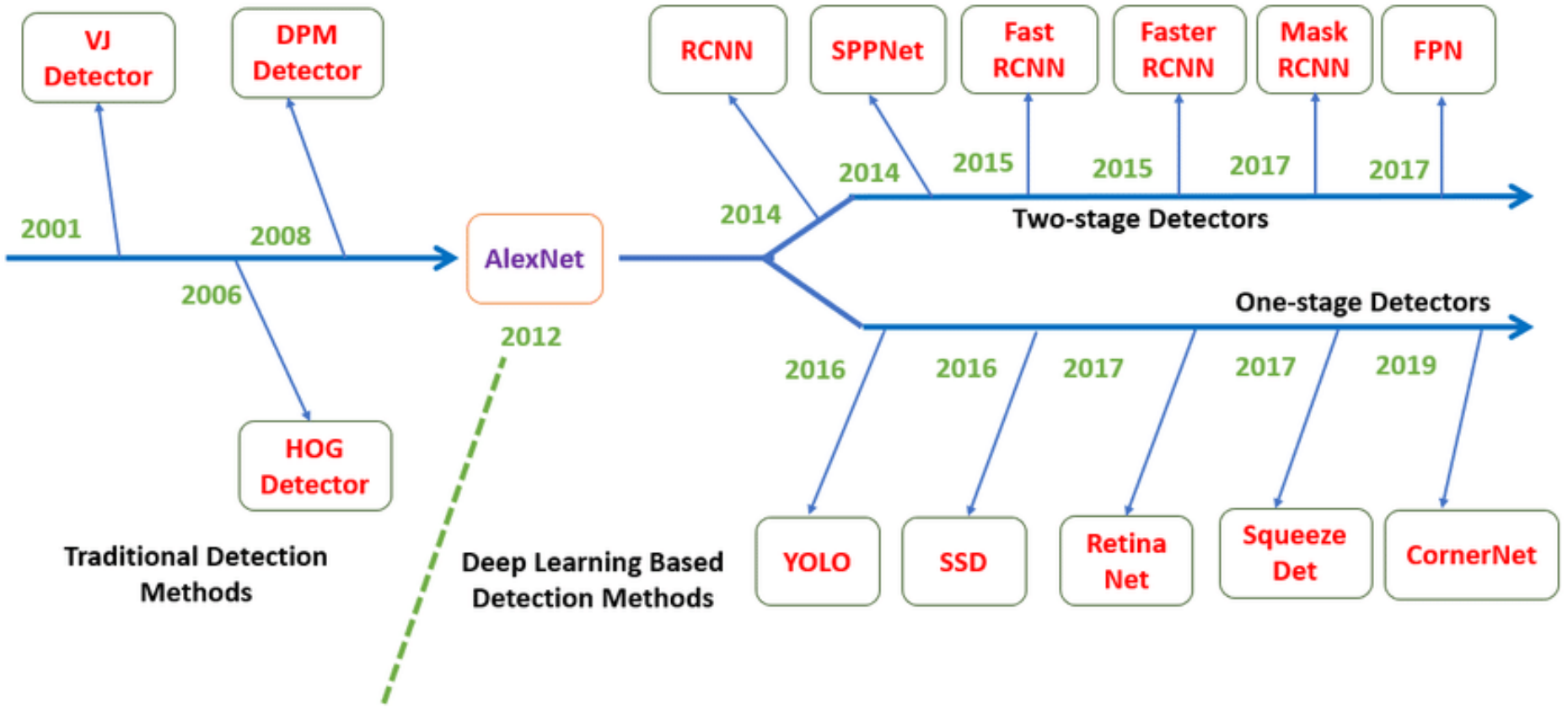
- Object detection has many real-life applications:
  - Autonomous driving
  - Video surveillance / monitoring
  - etc.

Application: Autonomous Driving



Source: https://blog.roboauto.cz/how-it-works-object-detection-2ae4448efa3f

Brief History & Milestones



Source: Investigations of Object Detection in Images/Videos Using Various Deep Learning Techniques and Embedded Platforms—A Comprehensive Review

Challenges

- Different imaging conditions including variations in viewpoints, scales, lighting, background clutter, occlusion, etc.
- Computational efficiency.
- Close-world vs open-world problem.
- Etc.

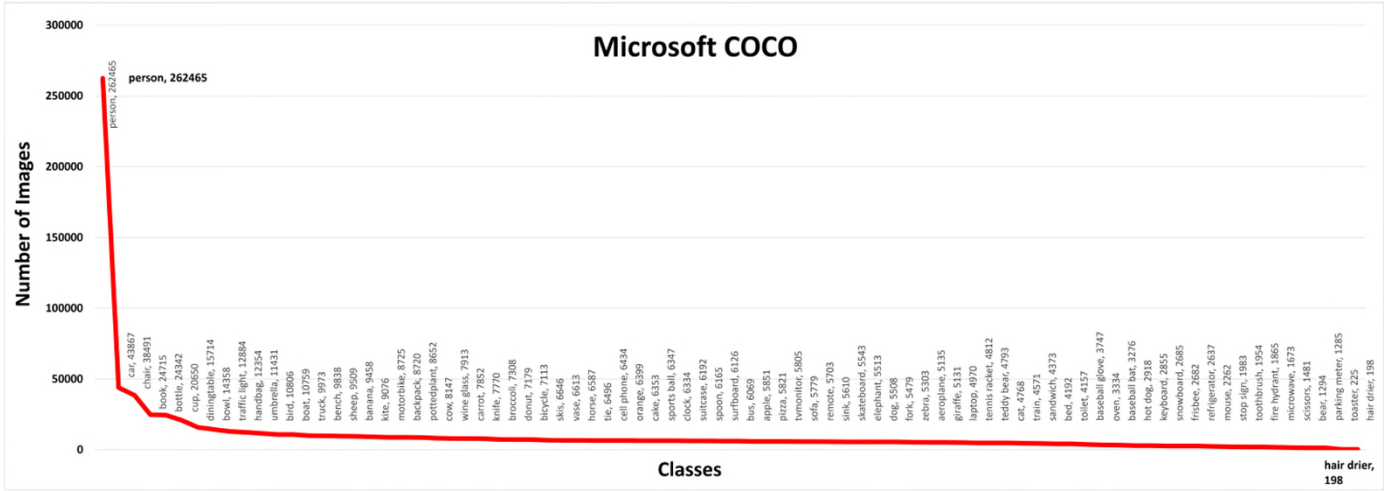
Common Datasets

- Microsoft Common Objects in Context (MS-COCO)
- ImageNet
- Google Open Images

Dataset	Classes	Train			Validation			Test
		Images	Objects	Objects/Image	Images	Objects	Objects/Image	
MS-COCO	80	118,287	860,001	7.27	5,000	36,781	7.35	40,670
ImageNet	200	456,567	478,807	1.05	20,121	55,501	2.76	40,152
Open Images	600	1,743,042	14,610,229	8.38	41,620	204,621	4.92	125,436

MS COCO

- Key benchmark dataset for training and evaluation
- Train, validation & test sets contain more than 200,000 images and 80 object categories.



ImageNet

- Consist of 1000 image classification classes.
- 200 classes hand-picked for object detection.
- More than 500k images for object detection.

Source: ImageNet Large Scale Visual Recognition Challenge 2017 (ILSVRC2017)

Google Open Images

- Contain 1.9 million images for object detection.
  - 16 million bounding boxes.
  - 600 categories.

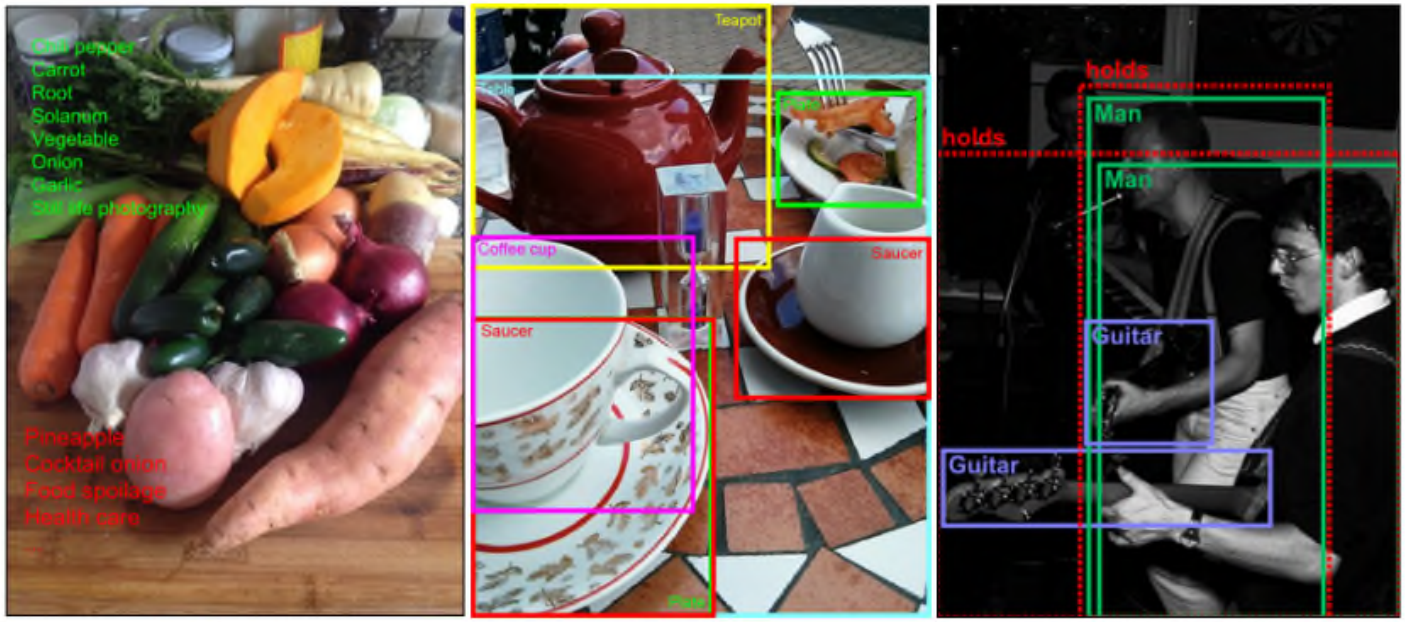


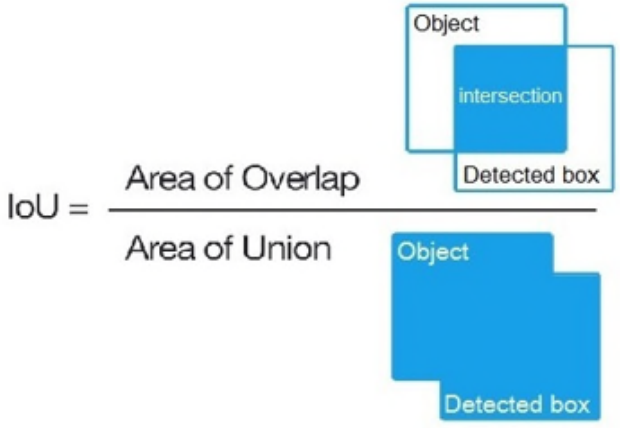
Image classification | Object detection | Visual relationship detection

Fig. 1 Example annotations in Open Images for image classification, object detection, and visual relationship detection. For image classification, positive labels (present in the image) are in green while negative labels (not present in the image) are in red. For visual relationship detection, the box with a dashed contour groups the two objects that hold a certain visual relationship.

Performance Metrics

- mean Average Precision (mAP), also loosely known as AP.
  - A metric used to evaluate the accuracy of object detection models.
  - Dependent on chosen value of Intersection over Union (IoU).
  - A prediction is considered as True Positive (TP) if IoU > threshold, and False Positive (FP) if IoU < threshold.
  - Common AP: AP50 or $AP_{0.5}$, $AP_{0.50:\ 0.05:\ 0.95}$

$$mAP_{\text{COCO}} = \frac{mAP_{0.50} + mAP_{0.55} + \ldots + mAP_{0.95}}{10}$$



mean Average Precision (mAP)

- mAP is calculated by finding AP for each class and then average over all classes.
- Note: some papers call mAP loosely as AP and assume the difference is clear from context.
- Steps to calculate mAP:
  - Generate the predicted object bounding boxes and labels using model
  - Calculate the confusion matrix: True Positives (TP) , False Positives (FP), False Negatives (FN)
  - Calculate the Precision and Recall metrics
  - Calculate the AP for current class: the area under the precision-recall curve
  - Calculate mAP as the Average Precision (AP) over all classes.

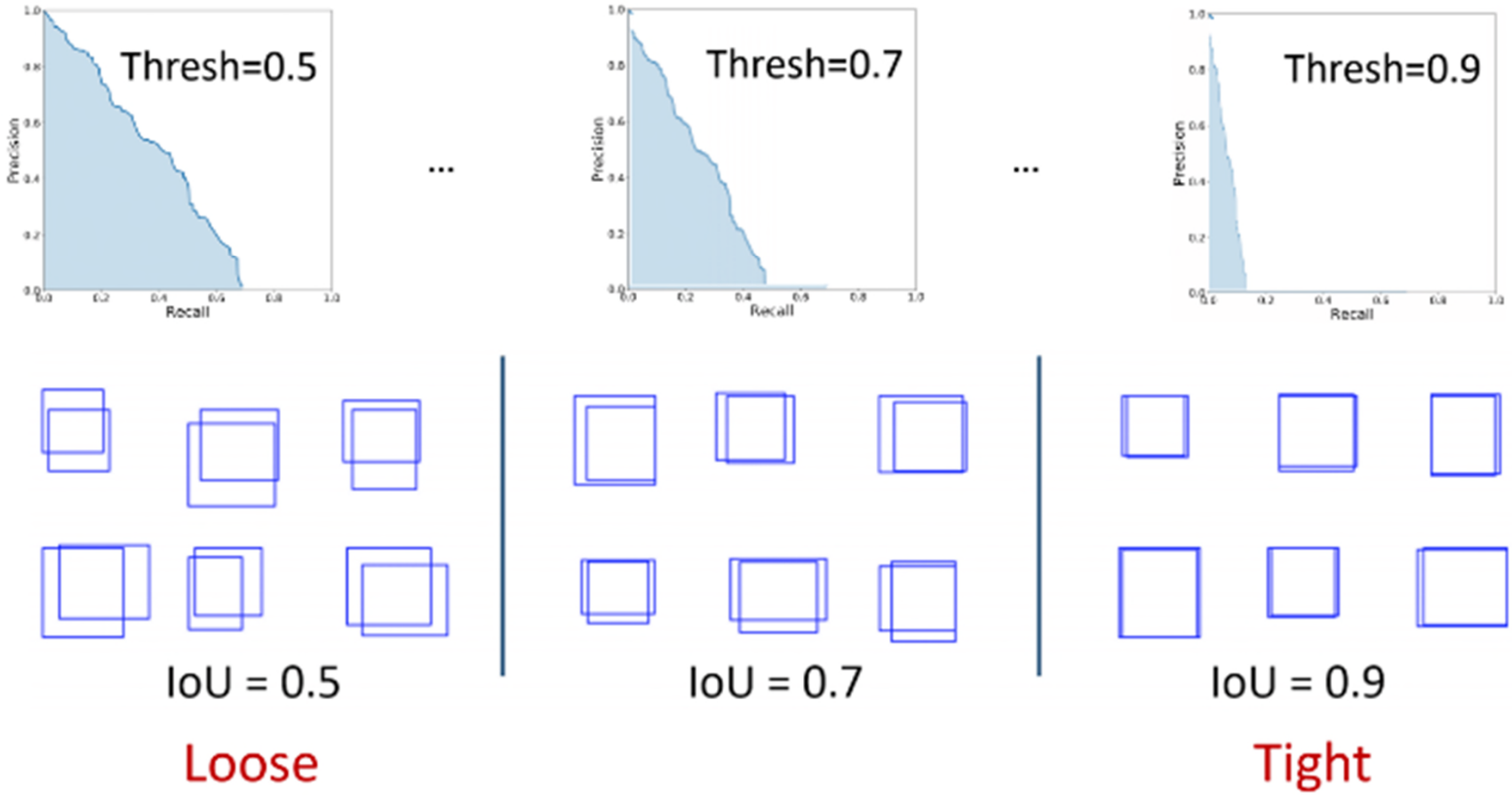
$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i$$

Mean Average Precision Formula

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\#\text{ ground truths}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\#\text{ predictions}}$$

mean Average Precision (mAP)



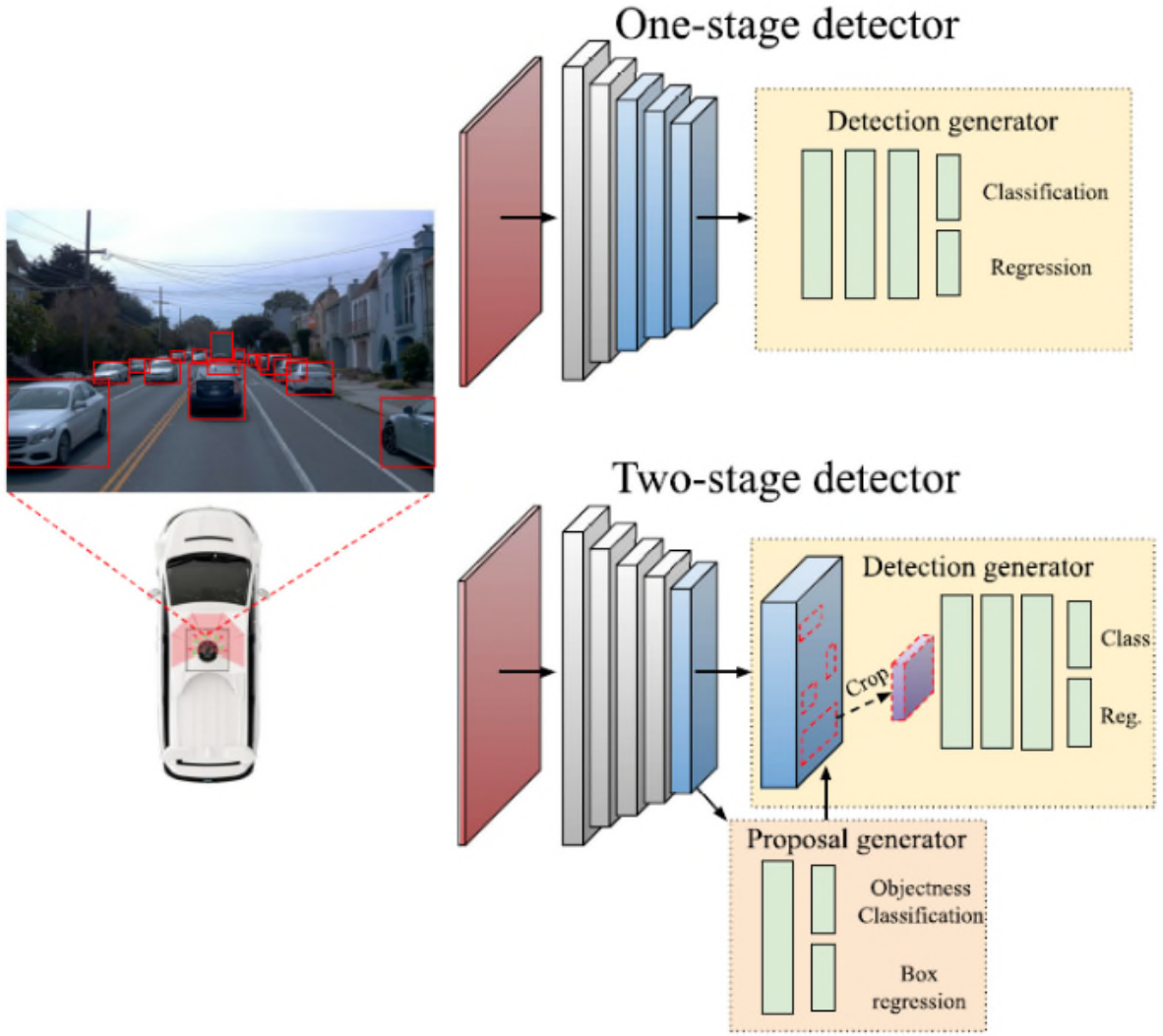
Source: https://kharshit.github.io/blog/2019/09/20/evaluation-metrics-for-object-detection-and-segmentation

Floating Point Operations (FLOPs)

- A metric used to evaluate computational complexity of any AI models.
- Often used to describe how many operations are required to run a single instance for a given model.
- Measure the total number of floating-point operations required for a single forward pass.
- Reflect the detection time of different models.

Object Detector Categories

- Main detector categories:
  - One-stage detectors
  - Two-stage detectors
- Lightweight detectors



Source: Manuel Carranza-García et. al, On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data

Two-Stage Detectors

Two-Stage Detectors

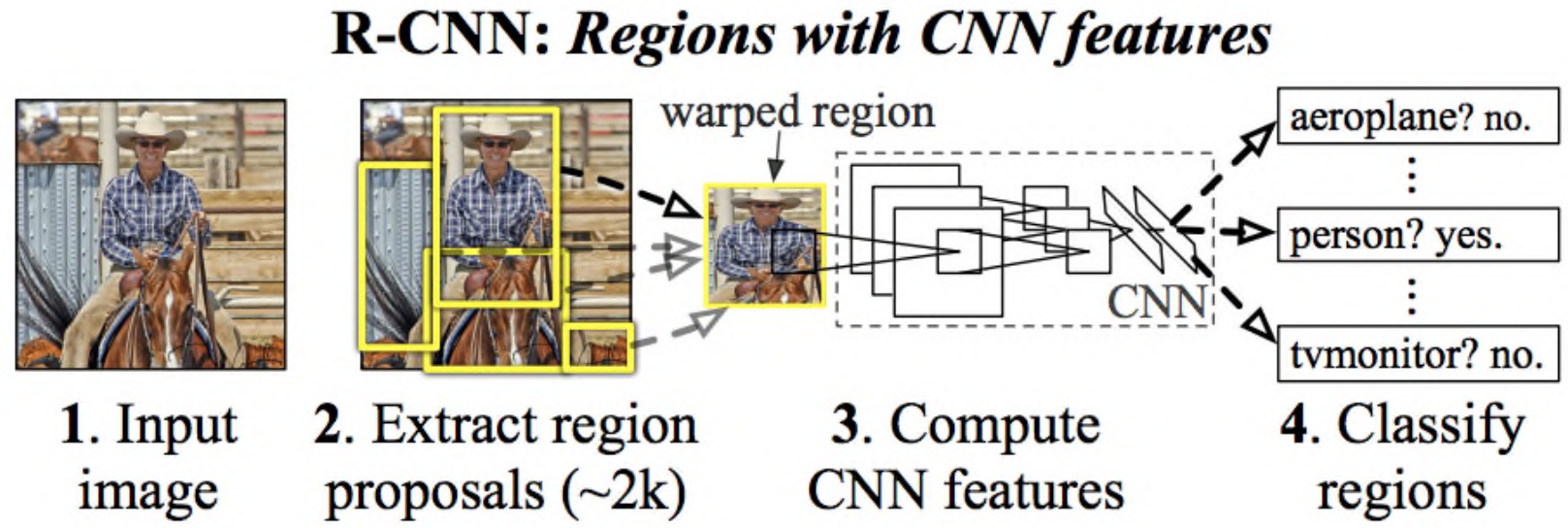
- Use techniques such as Region Proposal Network (RPN) to propose possible object regions.
- Representative methods: Faster-RCNN & Mask-RCNN.
- Other methods: SPP-Net, FPN, etc.
- Compared to one-stage detectors, generally:
  - Pros: higher accuracy (in the past).
  - Cons: more complex and slower.

Region-based Convolutional Neural Network (RCNN) Series Detectors

- R-CNN
- Fast R-CNN
- Faster R-CNN
- Mask R-CNN

Region-based Convolutional Neural Network (R-CNN)

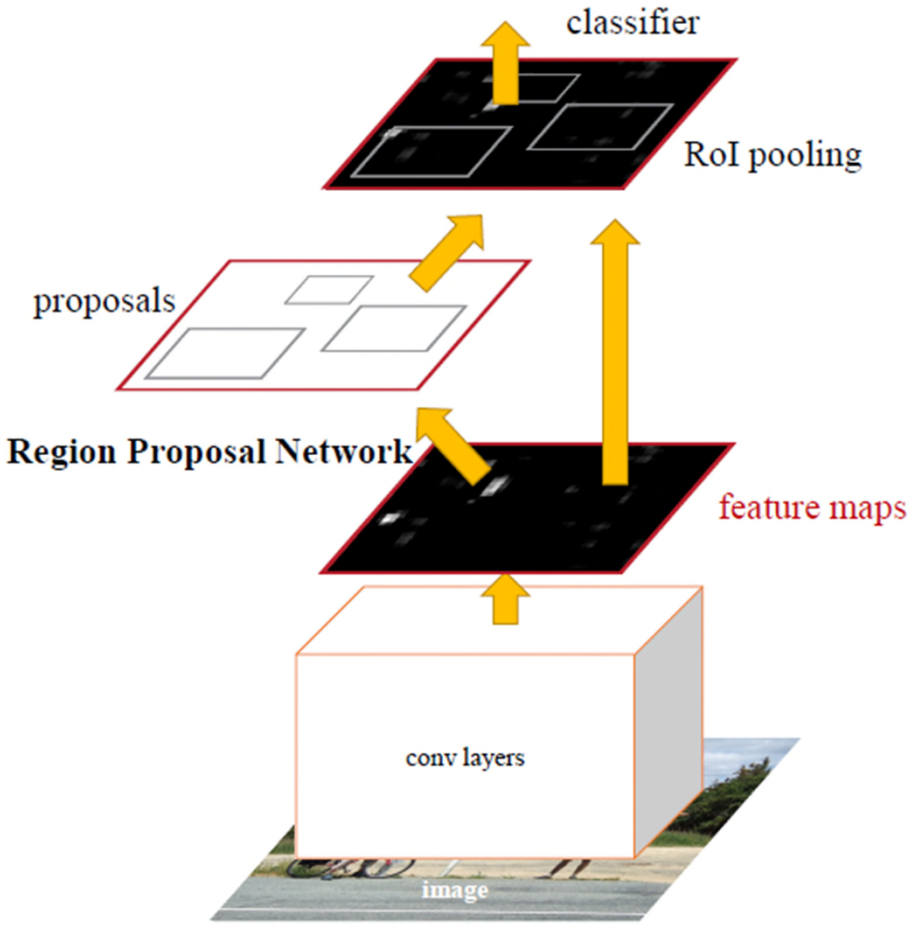
- Generate region proposals (~2000 regions).
- Extract feature vector from each region using CNN.
- Classify each region using class-specific linear Support Vector Machine (SVM) and perform bounding-box regressions.



Source: Rich feature hierarchies for accurate object detection and semantic segmentation (CVPR 2014)

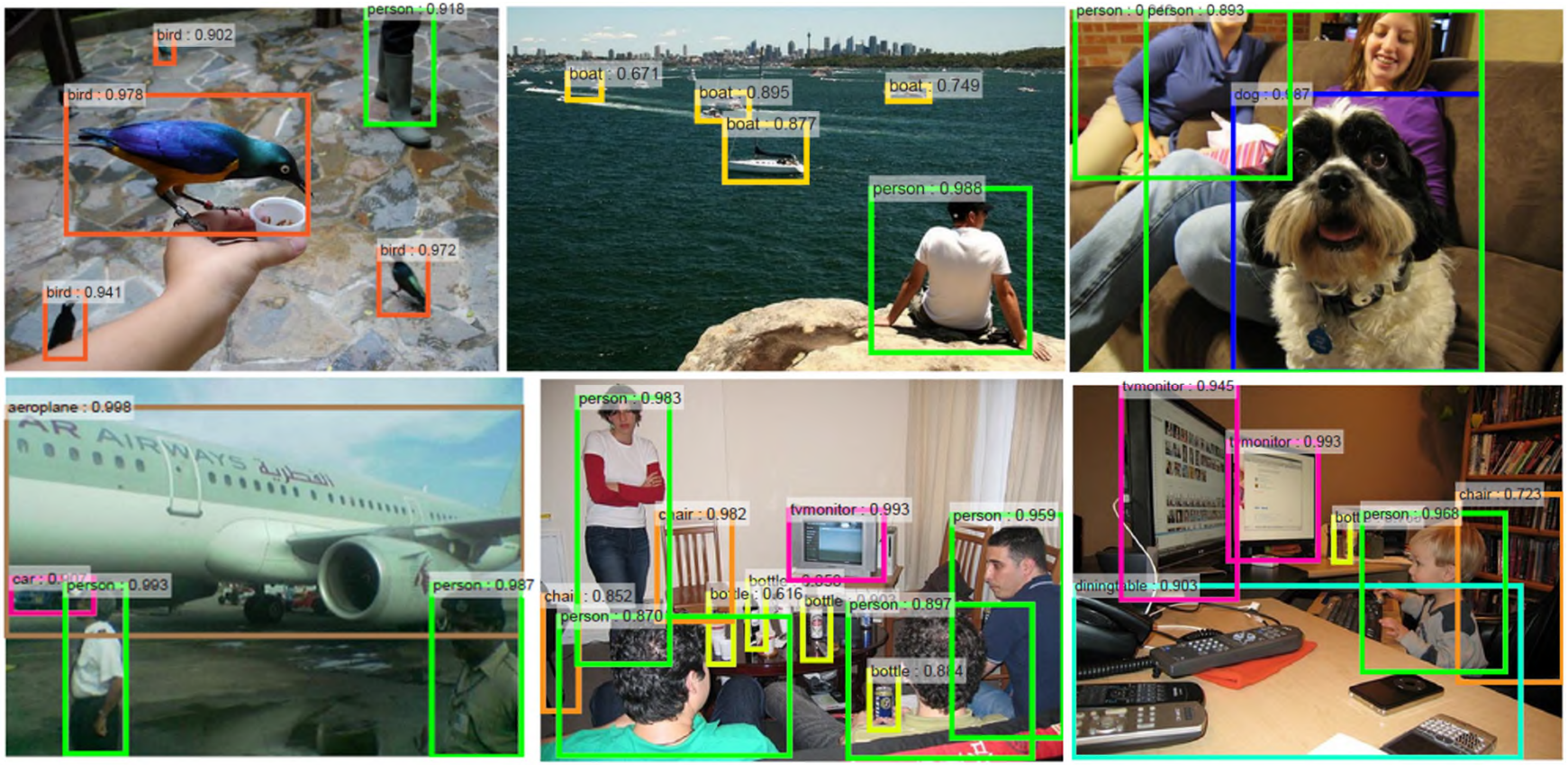
Faster R-CNN

- Introduce a Region Proposal Network (RPN).
- RPN shares full-image convolutional features with detection branch, hence enabling nearly cost-free region proposals.
- RPN simultaneously predicts object boundaries and objectness scores.

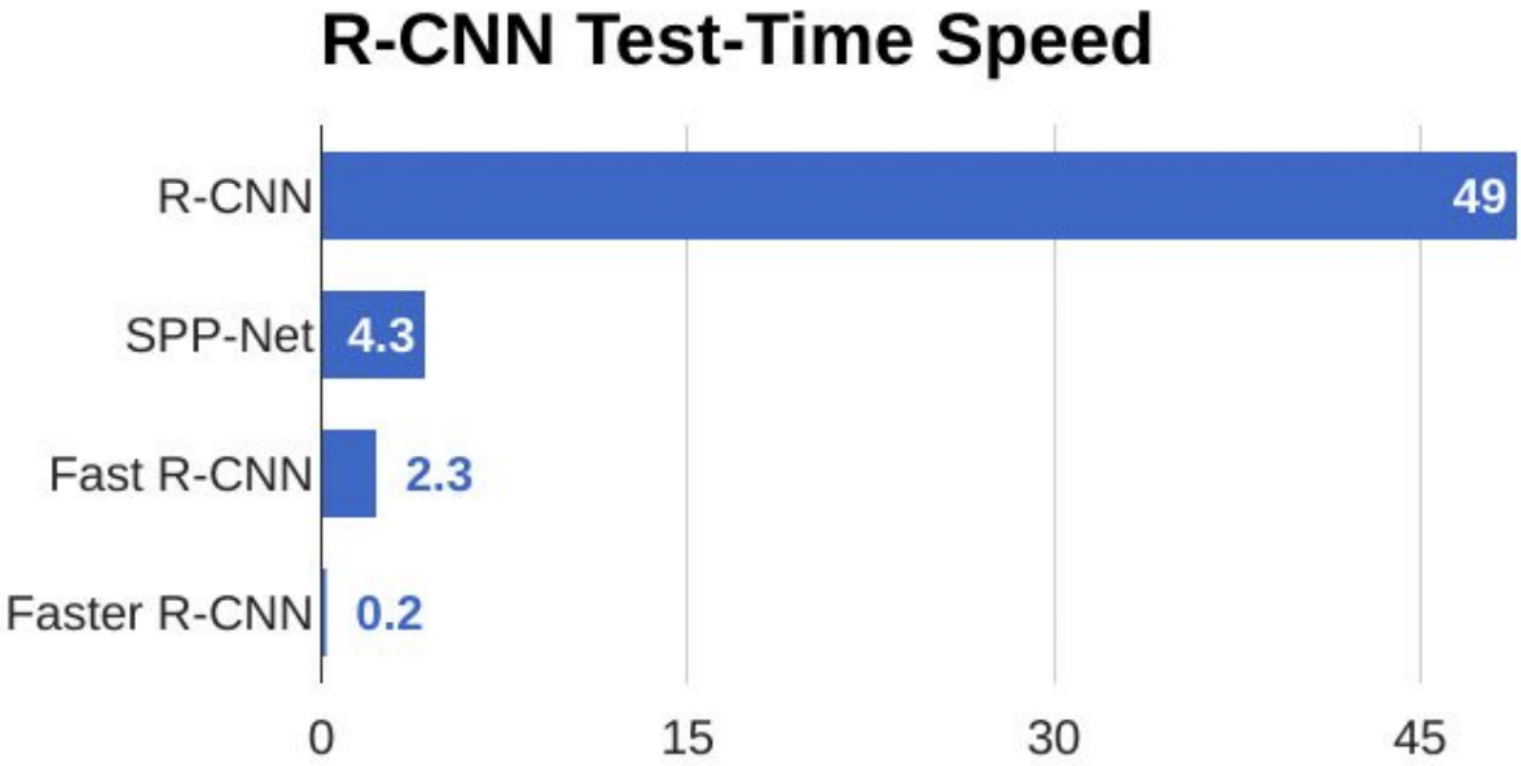


Source: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks (NIPS 2015)

Detection Results of Faster R-CNN



Speed Comparison

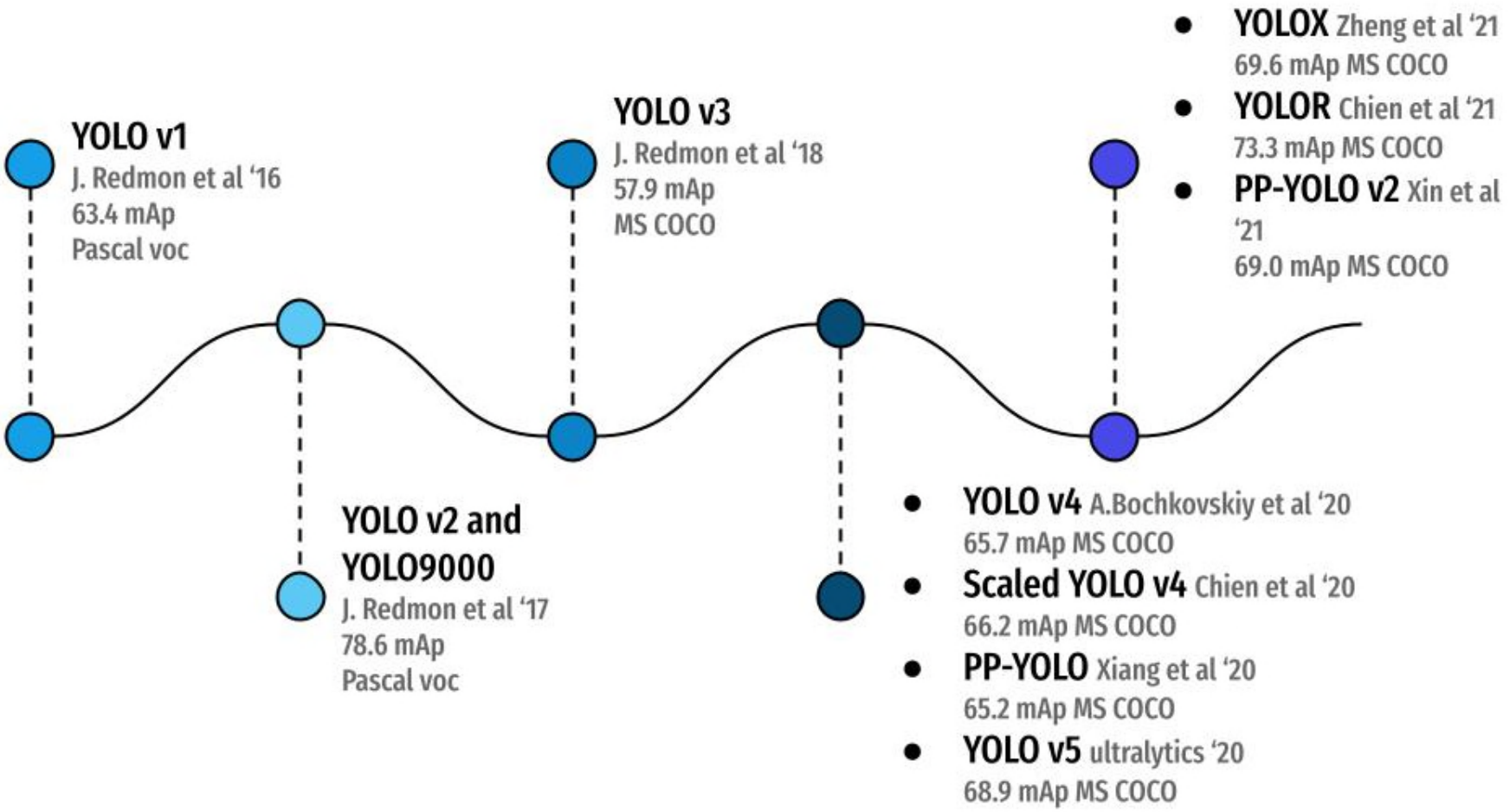


One-Stage Detectors

One-Stage Detectors

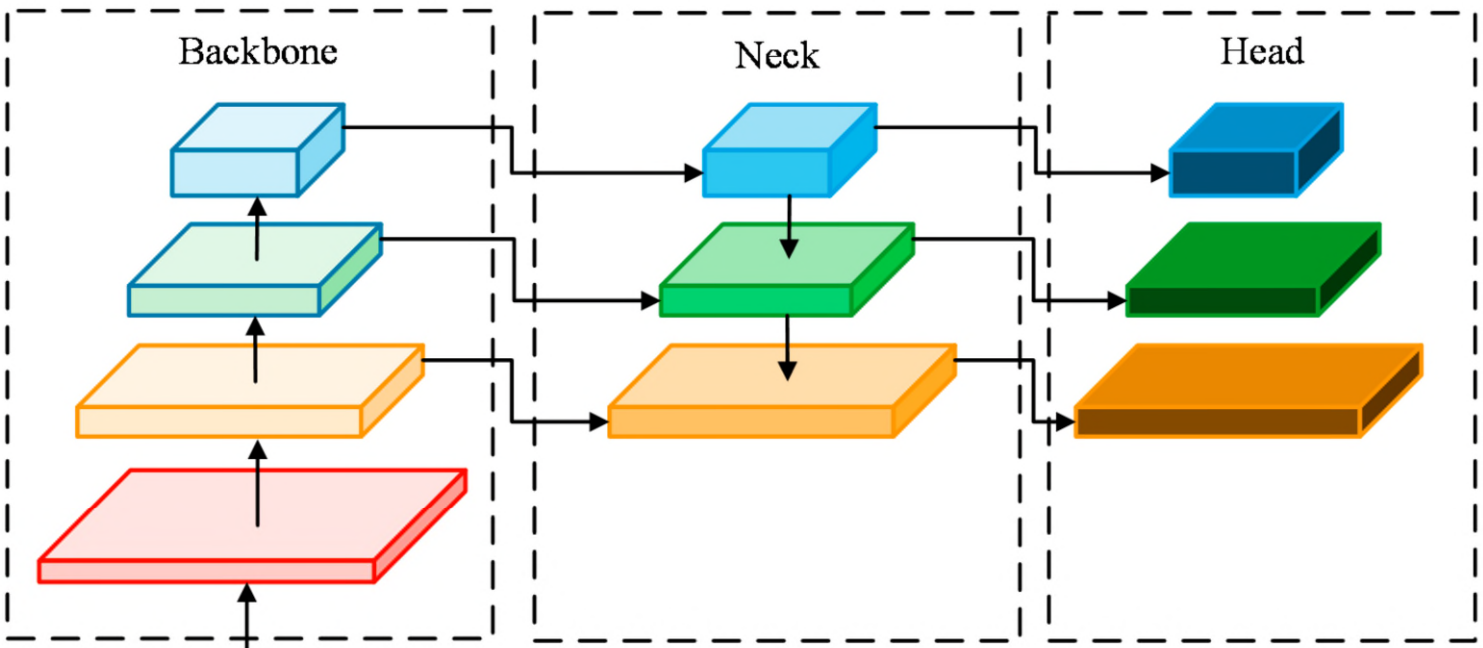
- One stage detectors regress and classify from the feature map directly.
- Representative methods: Yolo, SSD.
- Other methods: CenterNet, EfficientDet, etc.
- Compared to two-stage detectors, generally:
  - Pros: lower complexity and faster speed
  - Cons: lower accuracy (for older methods)

You Only Look Once (YOLO) Series Detectors



Source: https://viso.ai/deep-learning/yolov7-guide/

YOLO General Architecture



Input: images, patches by data augmentation;

Backbone: pre-trained classification models for feature extraction, various levels for different scales of objects;

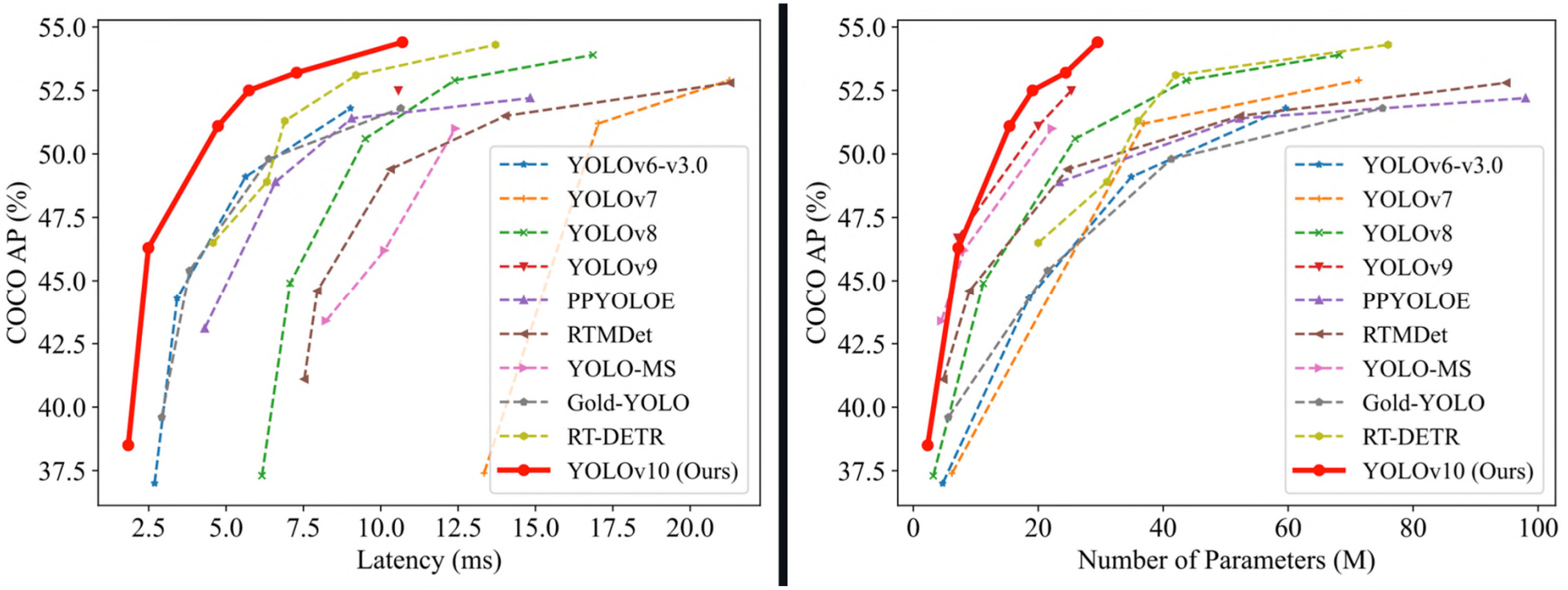
Neck: up-sampling and concatenation mechanisms to fusion different stage feature maps;

Head: predictions to classes and bounding boxes of detected objects.

Source: Automatic Ceiling Damage Detection in Large-Span Structures Based on Computer Vision and Deep Learning

Recent YOLO Detectors

- Current SOTA object detection models.
- YOLOv7: 2022; YOLOv8: 2023; YOLOv9, YOLOv10: 2024



Source: https://blog.roboflow.com/yolov10-how-to-train/

YOLO Comparison

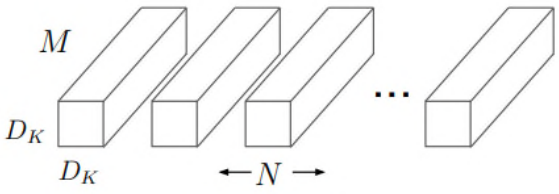
Lightweight Detectors

Lightweight Detectors

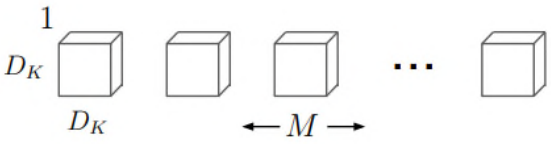
- Designed for resource limited devices.
- Small, fast and efficient networks.
- Representative method: MobileNetV2-based SSD (MediaPipe).
- Other methods: SqueezeNet, ShuffleNet, etc.
- Pros: ultra fast, efficient.
- Cons: less accurate.

MobileNet

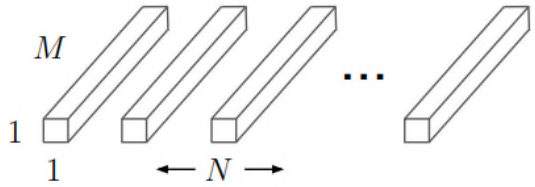
- Use depth-wise separable convolutions to reduce the computation.
  - Depth-wise convolution applies a single filter to each input channel.
  - Pointwise convolution is then applied using 1x1 convolution.



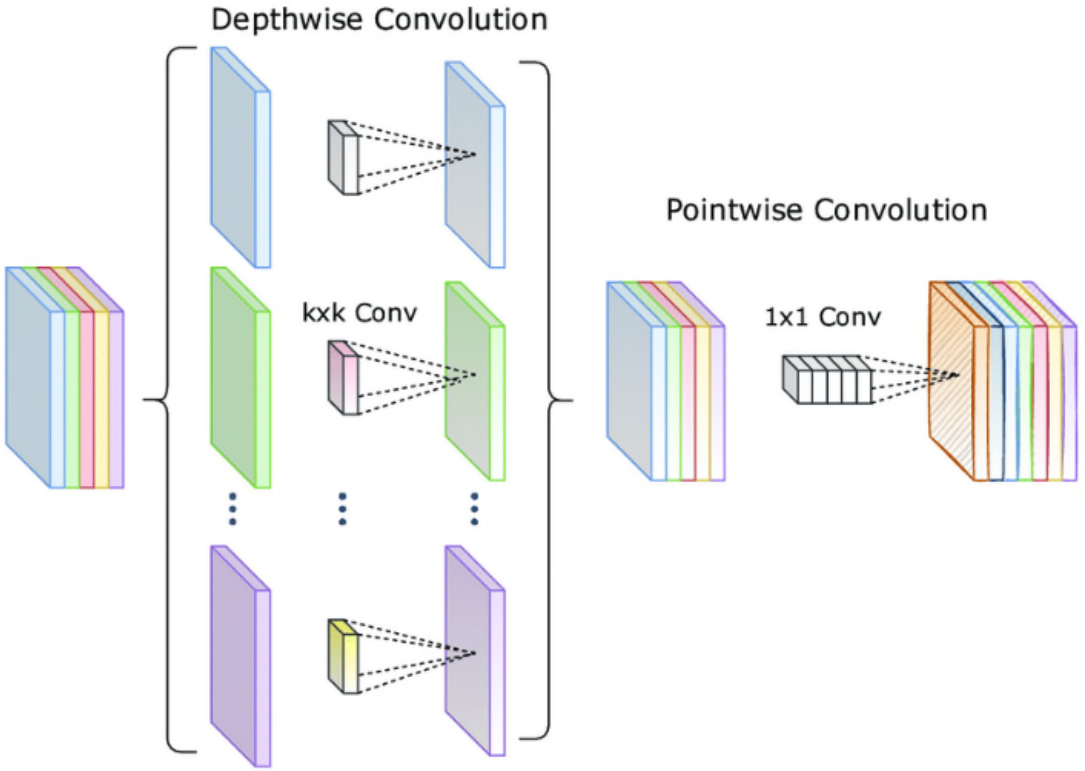
(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution



Depthwise separable convolutions.

Source: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications (arvix 2016)

Emerging / New Methods

Swin Transformer

- A popular recent image classification / object detection method.
- Serve as general-purpose backbone for computer vision.
- Hierarchical feature maps.

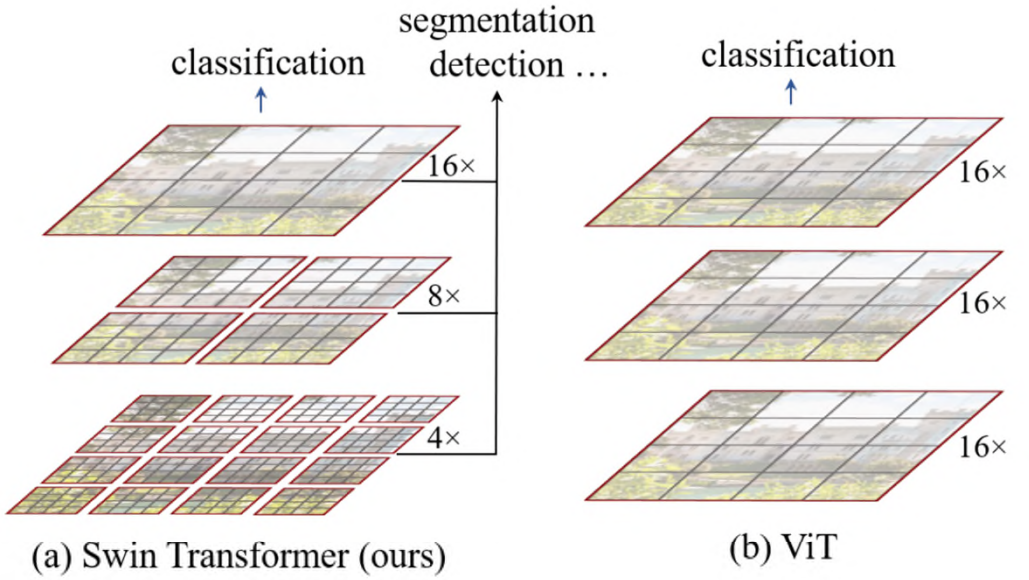
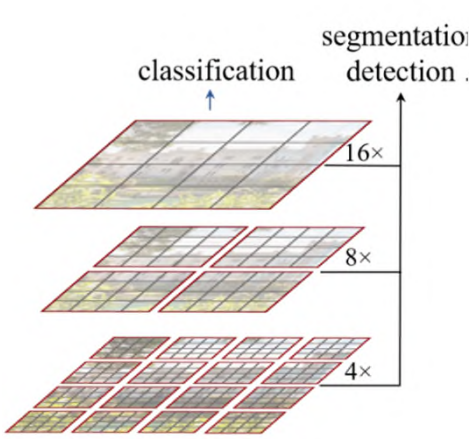


Figure 1. (a) The proposed Swin Transformer builds hierarchical feature maps by merging image patches (shown in gray) in deeper layers and has linear computation complexity to input image size due to computation of self-attention only within each local window (shown in red). It can thus serve as a general-purpose backbone for both image classification and dense recognition tasks. (b) In contrast, previous vision Transformers [19] produce feature maps of a single low resolution and have quadratic computation complexity to input image size due to computation of self-attention globally.

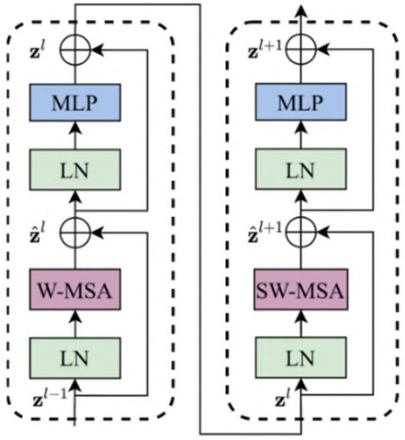
Source: Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows (ICCV 2021)

Swin Transformer Architecture

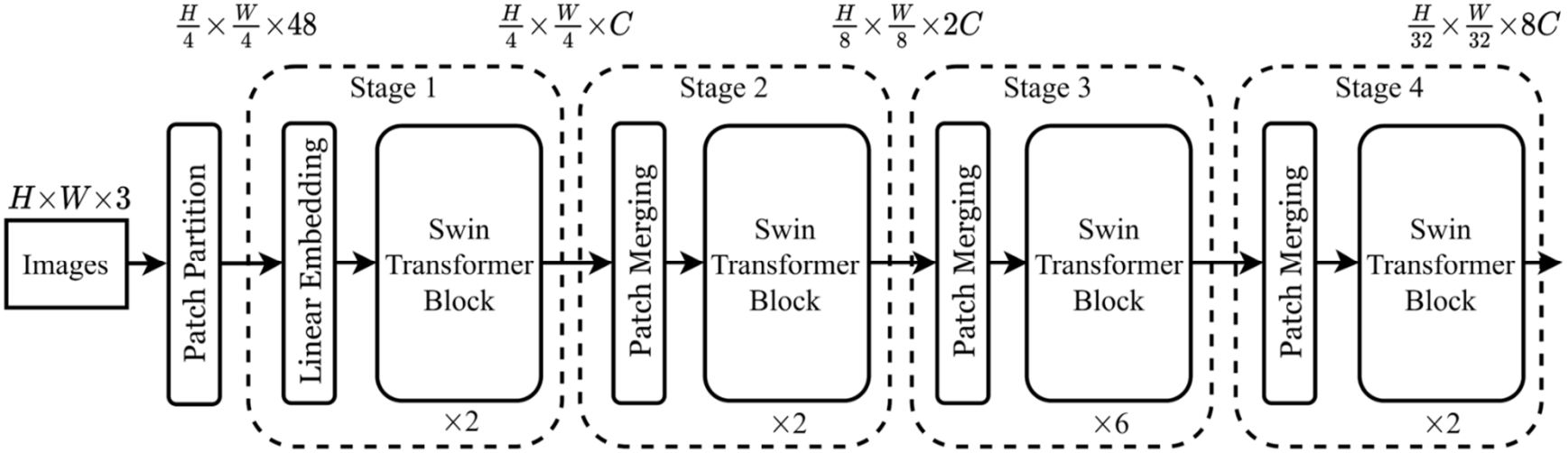
- Split an input RGB image into non-overlapping 4 x 4 patches.
- Propose Swin Transformer Block for learning.



(a) Swin Transformer



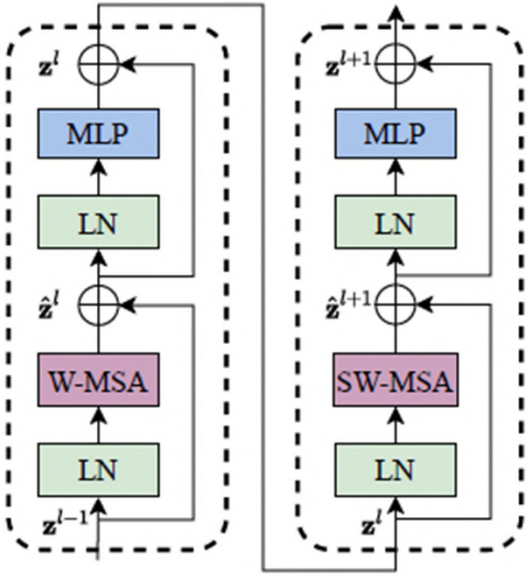
Two Successive Swin Transformer Blocks

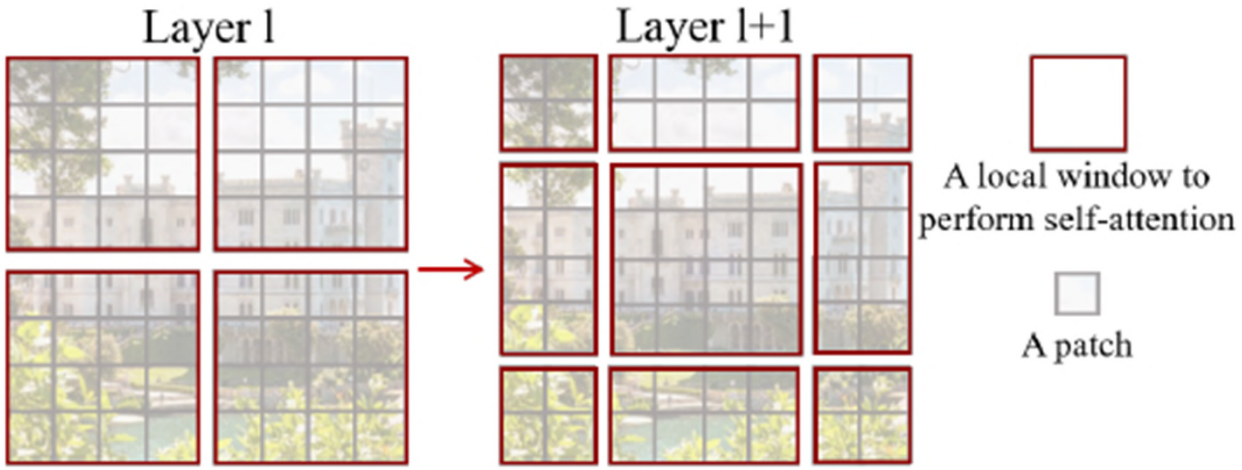


(d) Architecture

Swin Transformer Block

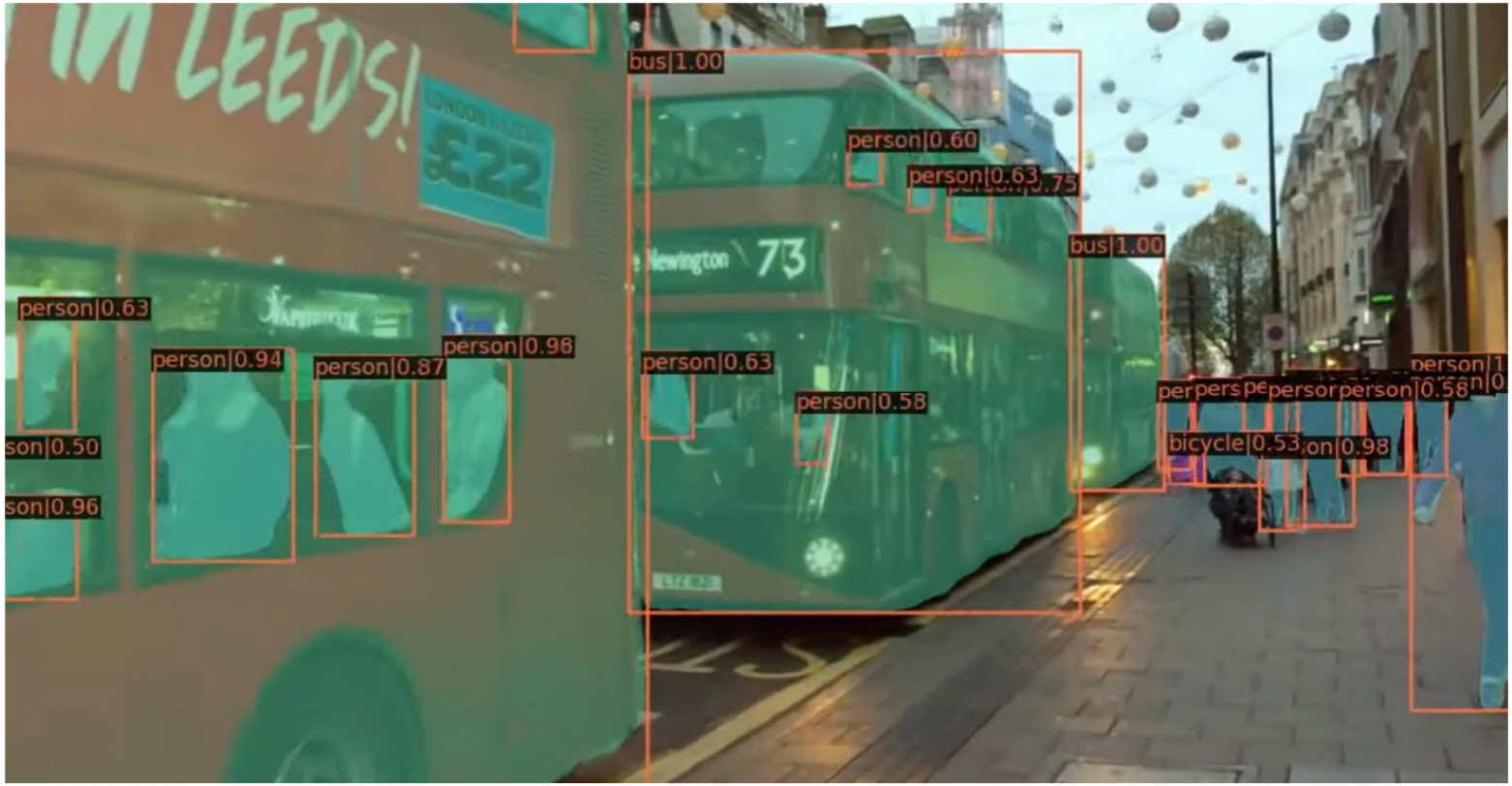
- W-MSA: Window-based Multi-head Self-Attention
  - In layer l, a regular window partitioning scheme is adopted, and self-attention is computed within each window.
- SW-MSA: Shifted Window-based Multi-head Self-Attention
  - In the next layer l + 1, the window partitioning is shifted, resulting in new shifted windows.





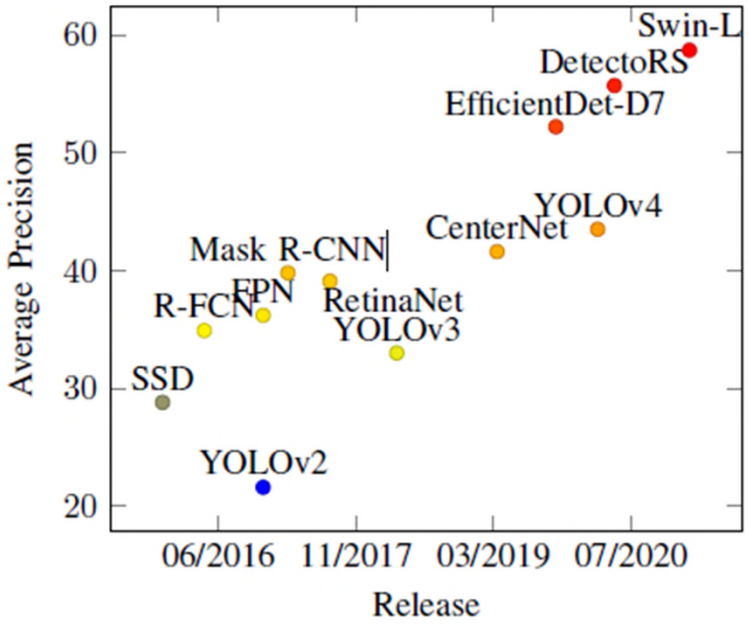
Source: Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows (ICCV 2021)

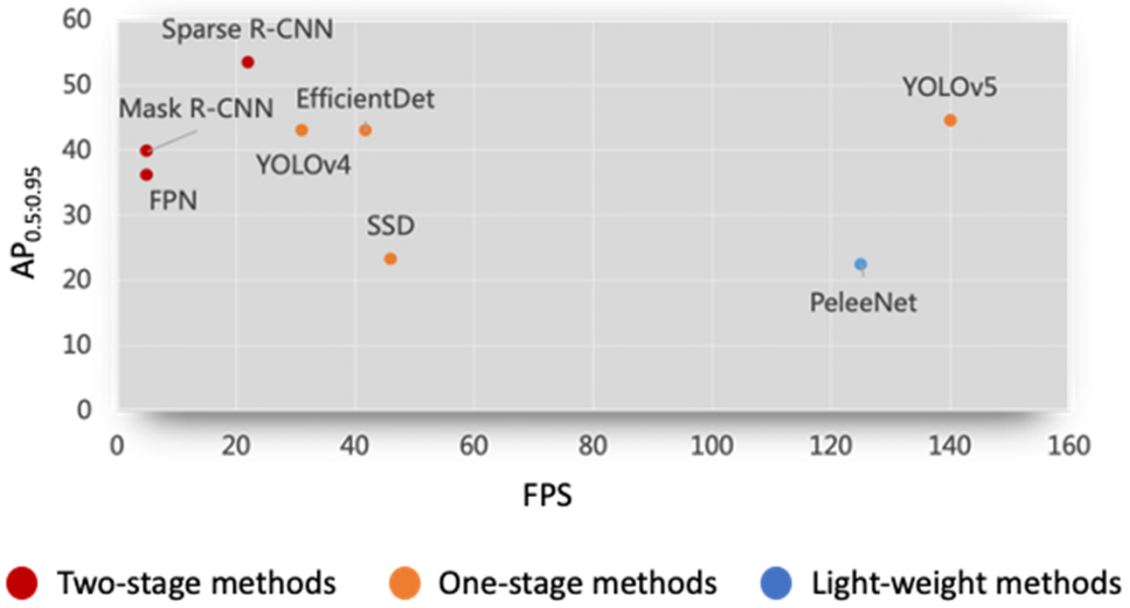
Demo: Swin Transformer



Source: https://www.youtube.com/watch?v=FQVS_0Bja6o&t=91s

Performance Comparison on COCO Dataset





Detection Transformer (DETR): End-to-End Object Detection With Transformers

- A popular recent object detection method.
- Predict in parallel the final set of detections by combining a common CNN with a transformer architecture.
- Decoder is different from original Attention is all you need design.
- Input to decoder is a given sequence of positional encoding (object queries).

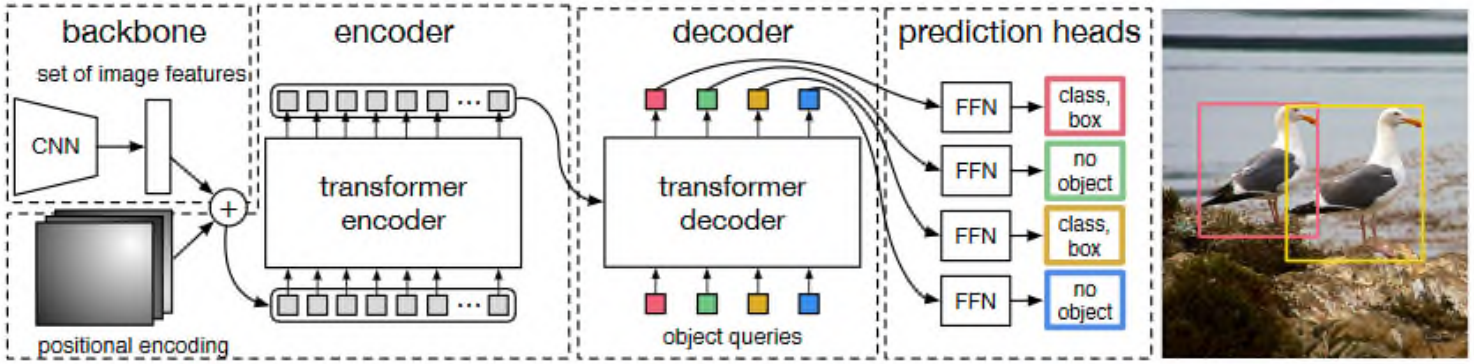


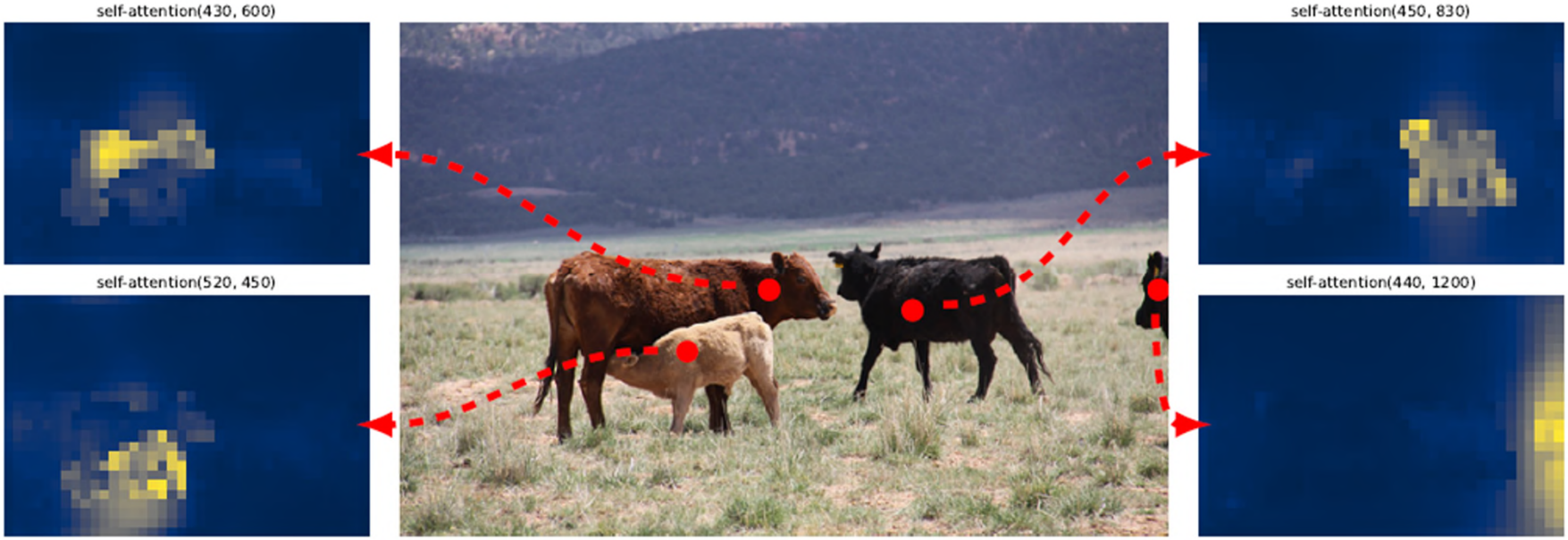
Fig. 2: DETR uses a conventional CNN backbone to learn a 2D representation of an input image. The model flattens it and supplements it with a positional encoding before passing it into a transformer encoder. A transformer decoder then takes as input a small fixed number of learned positional embeddings, which we call *object queries*, and additionally attends to the encoder output. We pass each output embedding of the decoder to a shared feed forward network (FFN) that predicts either a detection (class and bounding box) or a "`no object`" class.

DETR: Experimental Results

Model	GFLOPS/FPS	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster RCNN-DC5	320/16	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster RCNN-FPN	180/26	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster RCNN-R101-FPN	246/20	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster RCNN-DC5+	320/16	166M	41.1	61.4	44.3	22.9	45.9	55.0
Faster RCNN-FPN+	180/26	42M	42.0	62.1	45.5	26.6	45.4	53.4
Faster RCNN-R101-FPN+	246/20	60M	44.0	63.9	**47.8**	**27.2**	48.1	56.0
DETR	86/28	41M	42.0	62.4	44.2	20.5	45.8	61.1
DETR-DC5	187/12	41M	43.3	63.1	45.9	22.5	47.3	61.1
DETR-R101	152/20	60M	43.5	63.8	46.4	21.9	48.0	61.8
DETR-DC5-R101	253/10	60M	**44.9**	**64.7**	47.7	23.7	**49.5**	**62.3**

Source: DETR: End-to-End Object Detection With Transformers (ECCV 2020)

DETR: Visualization



Source: DETR: End-to-End Object Detection With Transformers (ECCV 2020)

Exercise: Object Detection

(a) State clearly whether the following object detectors are one-stage detector or two-stage detector: (i) R-CNN and (ii) YOLOv7. Which of the two object detectors above is a more suitable choice if speed is the key consideration in an object detection application? Briefly justify your answer.

(7 Marks)

Solution

Object Detection Summary

- This section covers the following:
  - Introduction
  - Object Detection Methods
    - Two Stage Detectors
    - One Stage Detectors
    - Lightweight Detectors
  - New / Emerging Methods

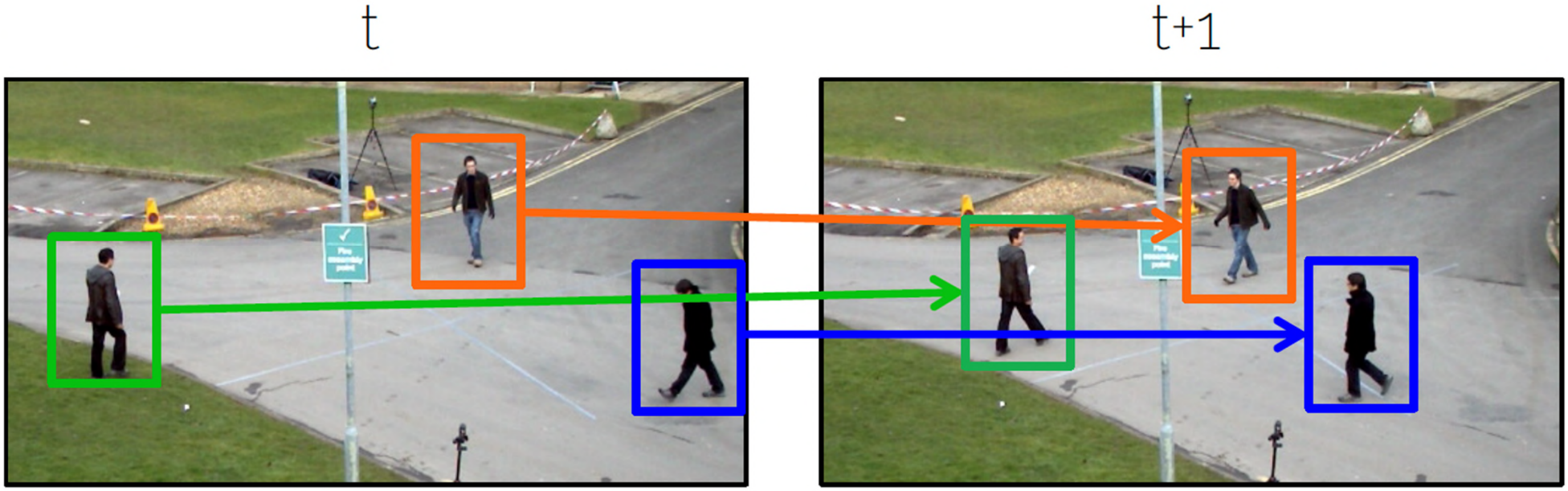
Object Tracking

Object Tracking Overview

- This section covers the following:
  - Introduction
  - Multiple Object Tracking (MOT)
  - Emerging / New Direction

Objective

- Track the same persons/objects in continuous frames of a video.
- Objects are represented by the bounding boxes with IDs. Typically, the detection results are the input to the tracking algorithm.



Source: https://dvl.in.tum.de/teaching/cv3dst-ss20

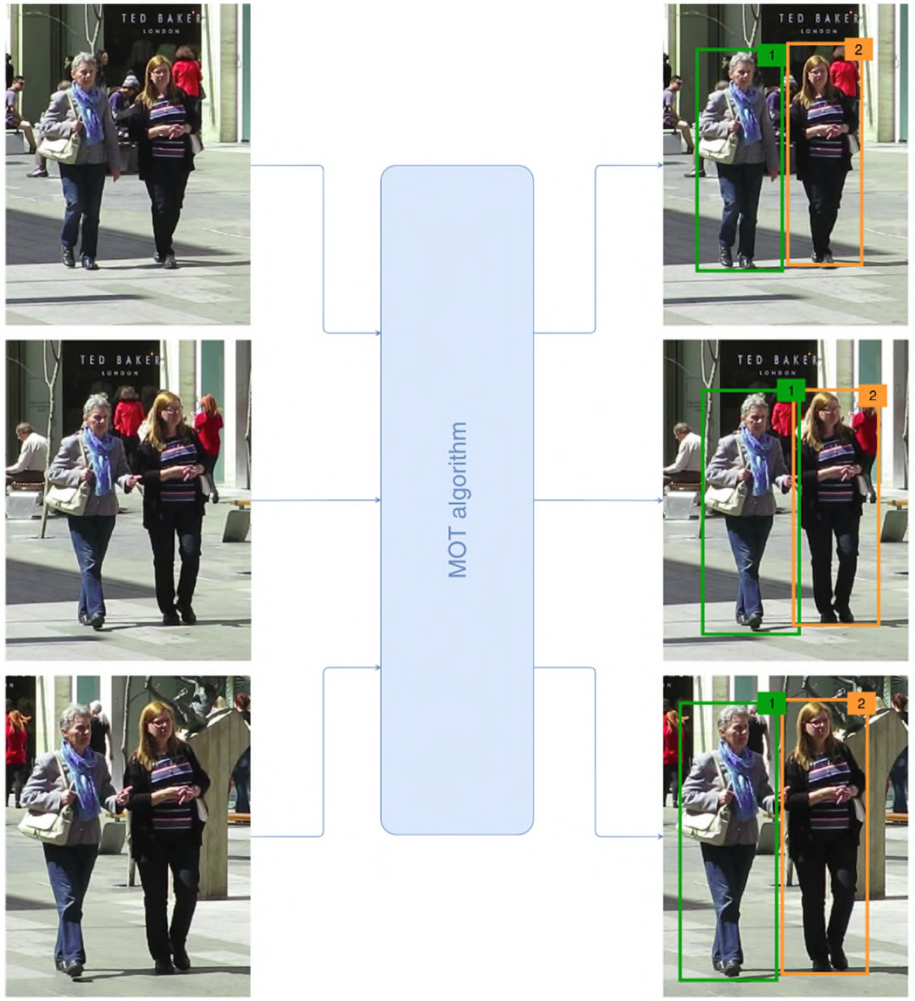
Tracking Issues

- Two important factors:
  - Appearance features
    - How does the object look like?
  - Motion features
    - What is the movement of objects?
    - Where will they be located in the next frame?

Multiple Object Tracking (MOT)

Multiple Object Tracking (MOT)

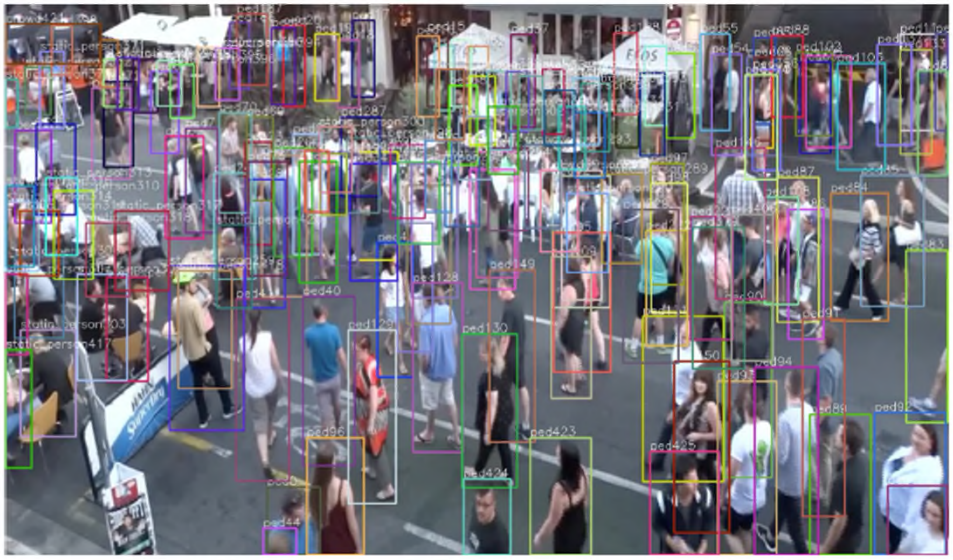
- Multiple object tracking (MOT) tracks all objects (e.g., persons) simultaneouly.



Source: Gioele Ciaparrone, Deep learning in video multi-object tracking: A survey, Neurocomputing, 2020

Challenges of MOT

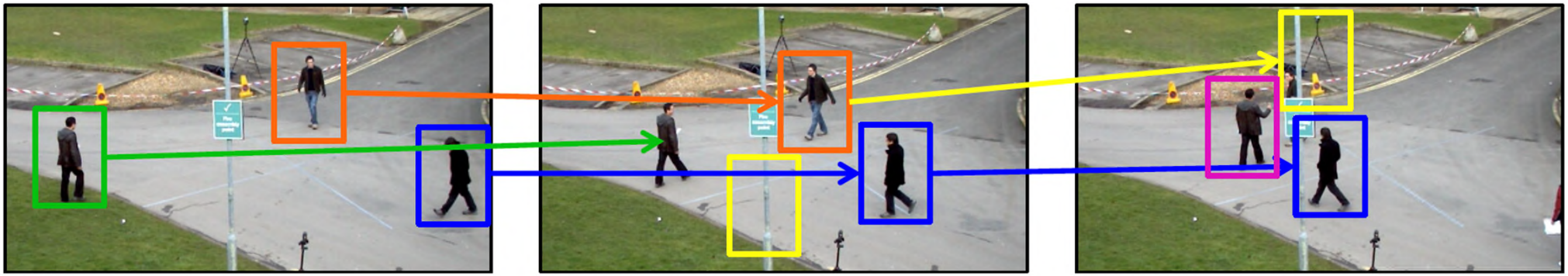
- Multiple objects with similar appearance, like pedestrians.
- Other uninterested objects (e.g., background, fire hydrants).
- Heavy occlusion.



Source: https://dvl.in.tum.de/teaching/cv3dst-ss20

Tracking-by-Detection

- Goal:
  - Detect objects in individual video frames.
  - Associate sets of detections between frames, thereby creating individual object tracks over time.
- Problem:
  - Detections are not always accurate, like false positive, missing detections.



Tracking-by-Detection (1)

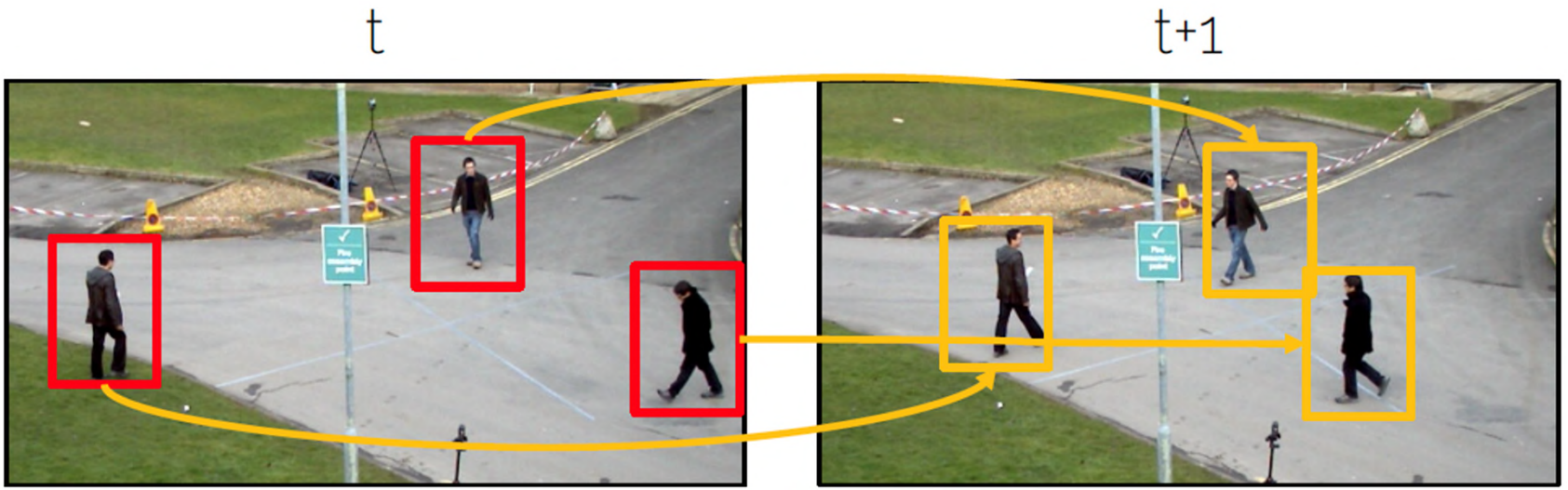
1. Detection: Use a detector to initialize tracks at frame t.

t

t+1

Tracking-by-Detection (2)

1. Detection: Use a detector to initialize tracks at frame t.
2. Motion prediction: predict the next positions of objects using motion model.



Tracking-by-Detection (3)

1. Detection: Use a detector to initialize tracks at frame t.
2. Motion prediction: predict the next positions of objects using motion model.
3. Data association: Match predicted positions with detections at frame t+1.

t

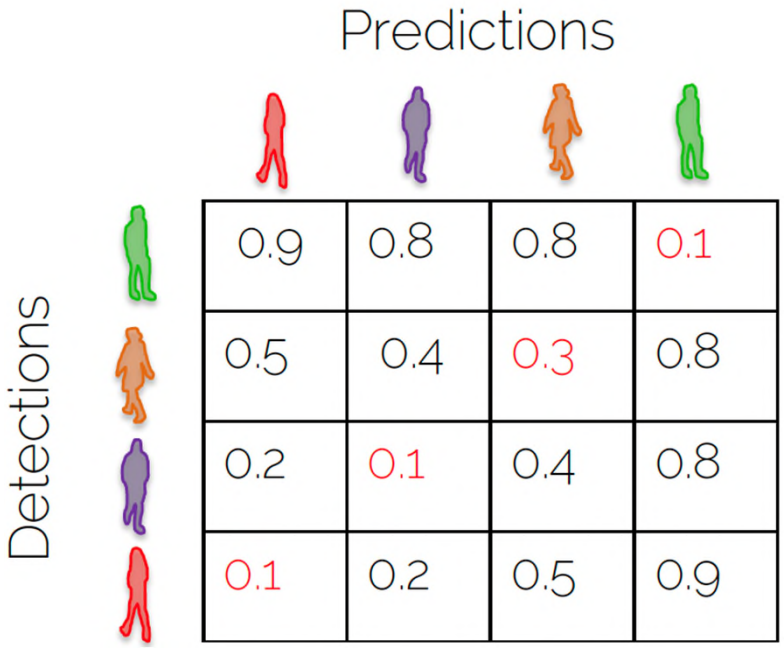
t+1

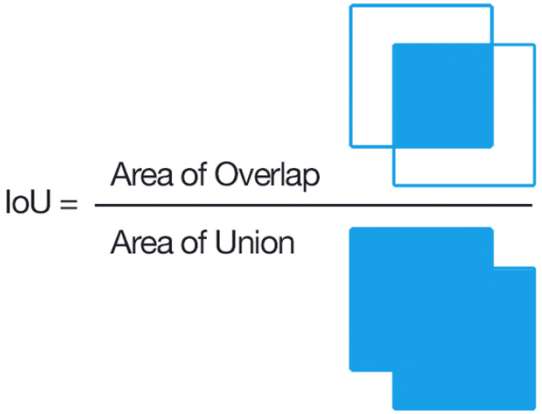
Motion Model

- Predictions are based on the movement of objects.
- Assumption: e.g., objects moves at a constant speed.
- Motion modelling
  - Kalman filtering
  - Particle filtering
  - Etc.

Data Association

- Simple matching of detections with the predicted boxes by Kalman Filtering.
  - Calculate the distances between boxes (e.g., based on IOU, pixel-wise)
  - Hungarian algorithm solves the one-to-one linear assignment problem.
- Distance based on Intersection over Union (IoU)
  - Distance between each detection with all predicted bounding boxes using Kalman filtering.

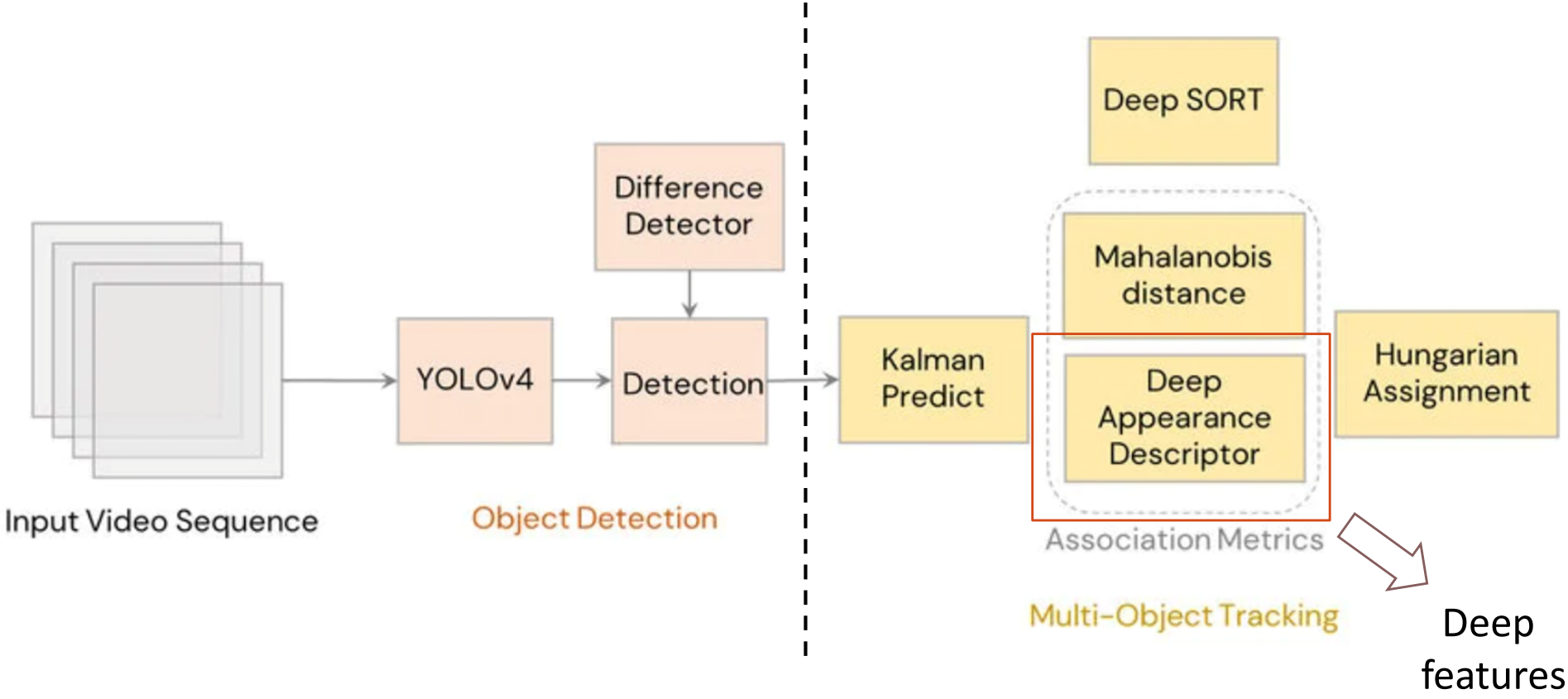




Source: https://dvl.in.tum.de/teaching/cv3dst-ss20
*Refer to http://www.hungarianalgorithm.com/solve.php to use Hungarian algorithm

Data Association

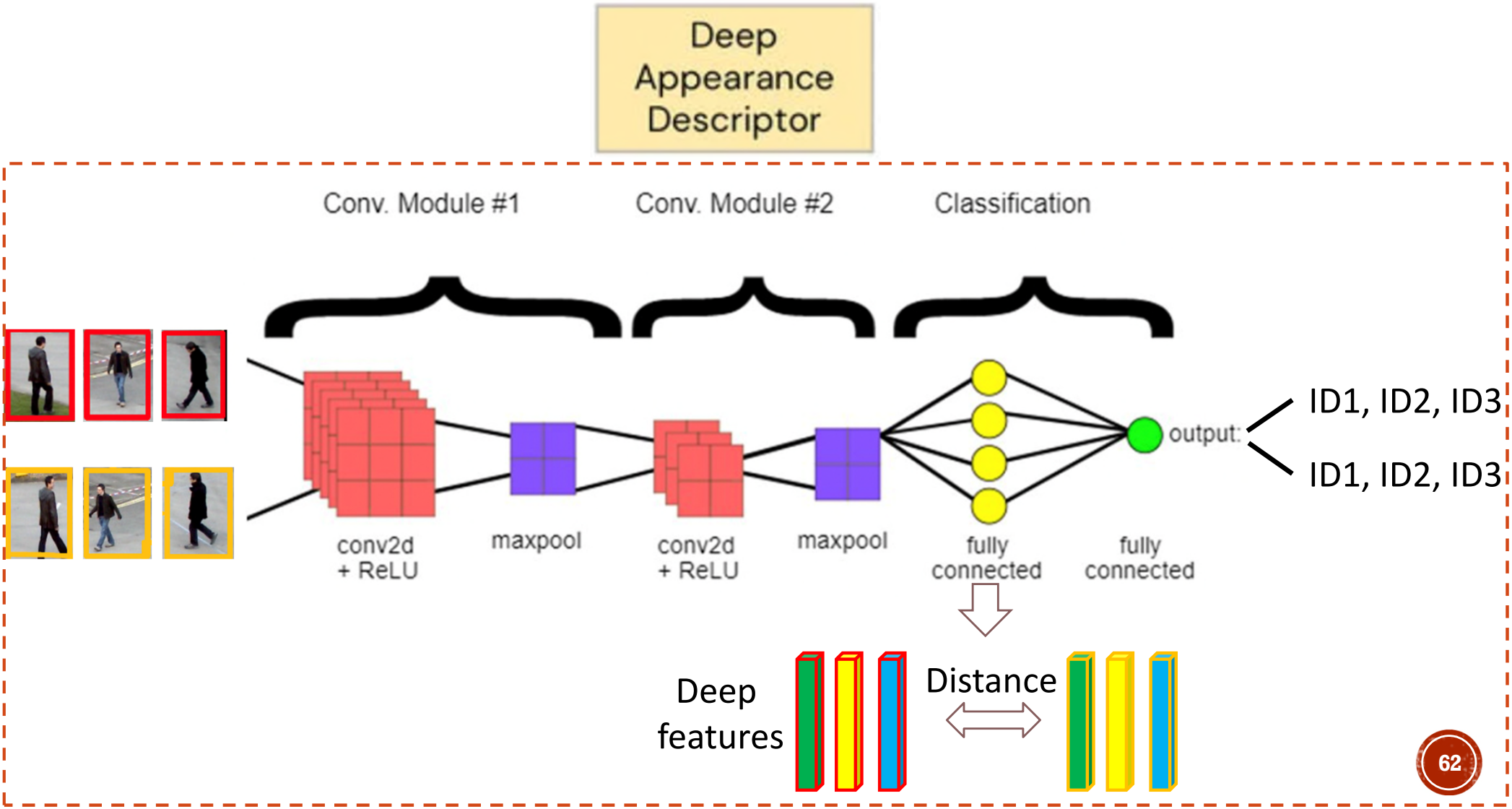
- Appearance model: Deep SORT (Simple Online and Real-time Tracking).
- Add deep appearance features to help data association between detections by object detector (e.g., YOLOv4) and predictions by Kalman filter.



Source: https://learnopencv.com/understanding-multiple-object-tracking-using-deepsort

Data Association

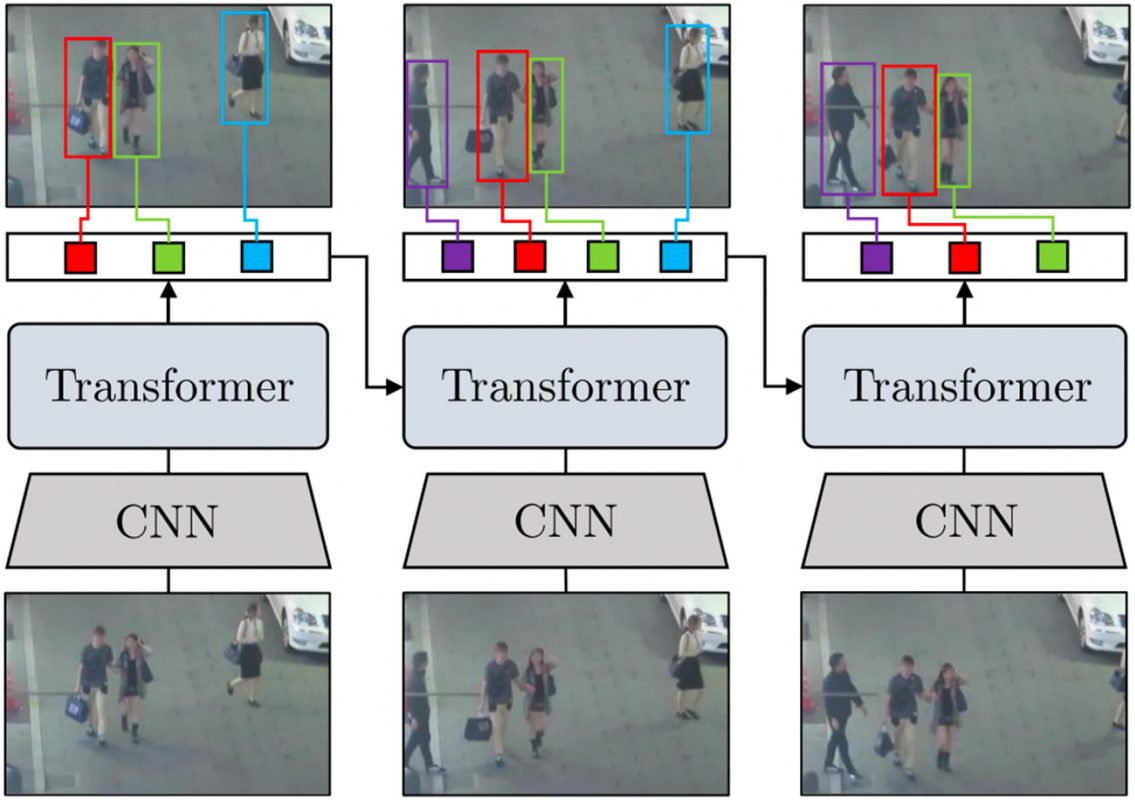
- Deep SORT: Deep appearance descriptor / feature / embedding (i.e., using CNN)



Emerging / New Methods in MOT

Trackformer

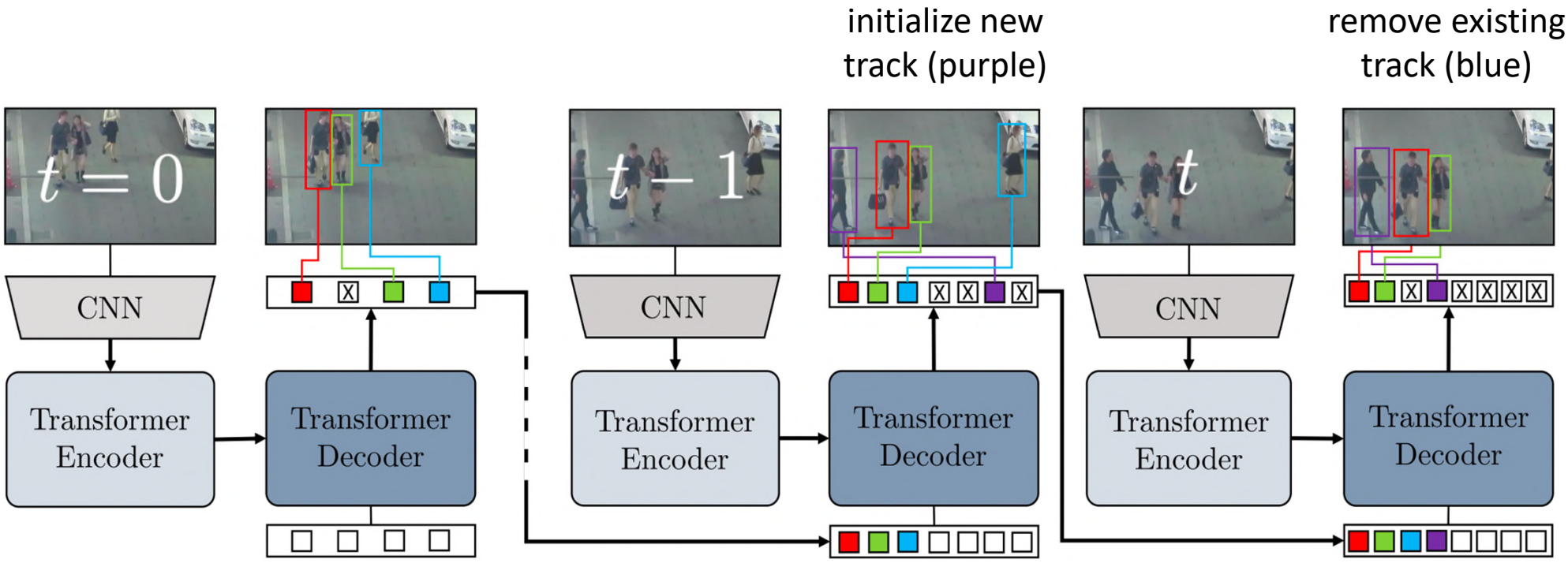
- Perform joint object detection and tracking-by-attention with Transformer.
- Object and autoregressive track queries reason about track initialization, identity, and spatiotemporal trajectories.



Source: Tim Meinhardt et al., TrackFormer: Multi-Object Tracking with Transformers. CVPR, 2022.

Trackformer Architecture

- Encoder: CNN + Transformer encoder.
- Decoder: Transformer decoder + object / track queries.
- Bounding box and class prediction: multilayer perceptron (MLP).



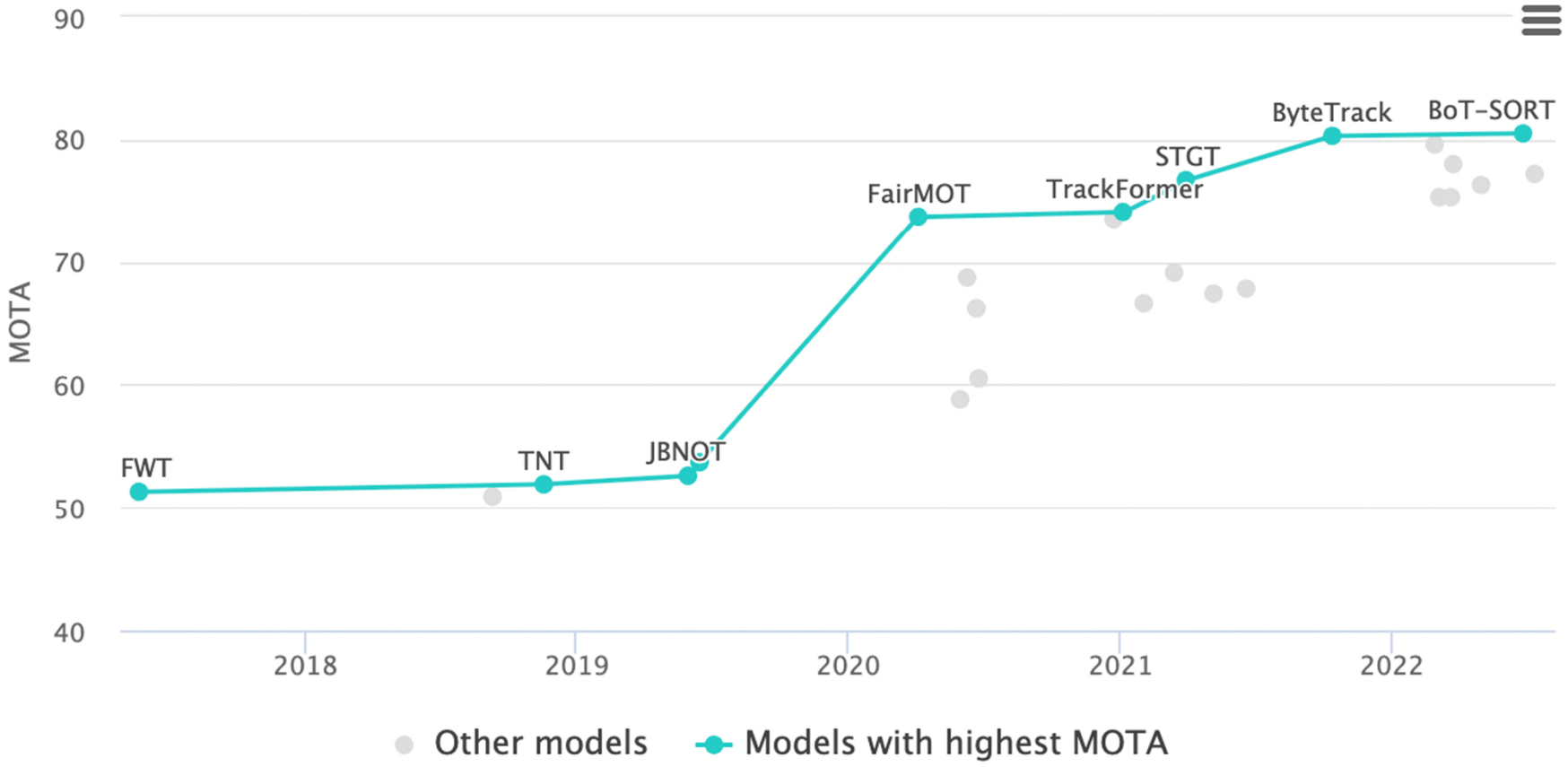
White boxes: object queries, Colored boxes: track queries, Crossed boxes: background classes

Trackformer Queries

- Object queries:
  - Detect new objects in current frame and then form tracks for the next frame.
- Track queries:
  - Follow objects through a video by carrying their identity information while adapting to their changing position.
  - Associate tracked objects with the bounding box of the current frame.

Demo: Trackformer

Performance Comparison on MOT16



- Comparison of state-of-the-art methods by MOTA on the MOT16 dataset. (Date: Aug 2022)

MOTA (Multiple Object Tracking Accuracy)

$$MOTA = 1 - \frac{(FN + FP + IDSW)}{GT} \quad \in (-\infty, 1]$$

GT is the number of ground truth boxes

Source: https://paperswithcode.com/sota/multi-object-tracking-on-mot17?metric=MOTA

Object Tracking Summary

- This section covers the following:
  - Introduction
  - Multiple Object Tracking (MOT)
  - Emerging / New Direction

Exercise: Multiple Object Tracking

(c) List the key steps in the tracking-by-detection multiple-object tracking in video.

(6 Marks)

Solution

Pose Estimation

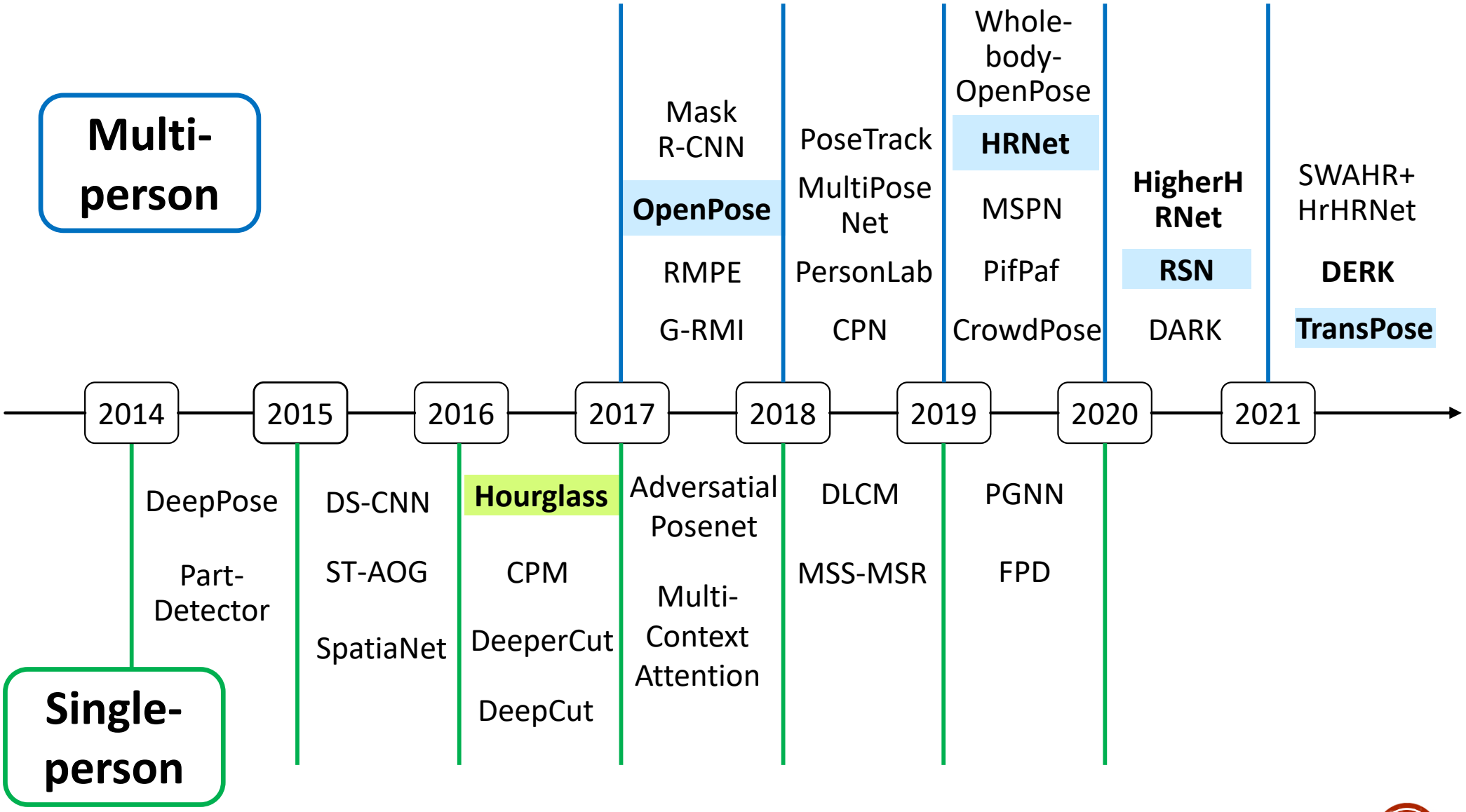
Pose Estimation Overview

- Introduction
- Single-Person Pose Estimation
- Multi-Person Pose Estimation
- Emerging / New Methods

Objective

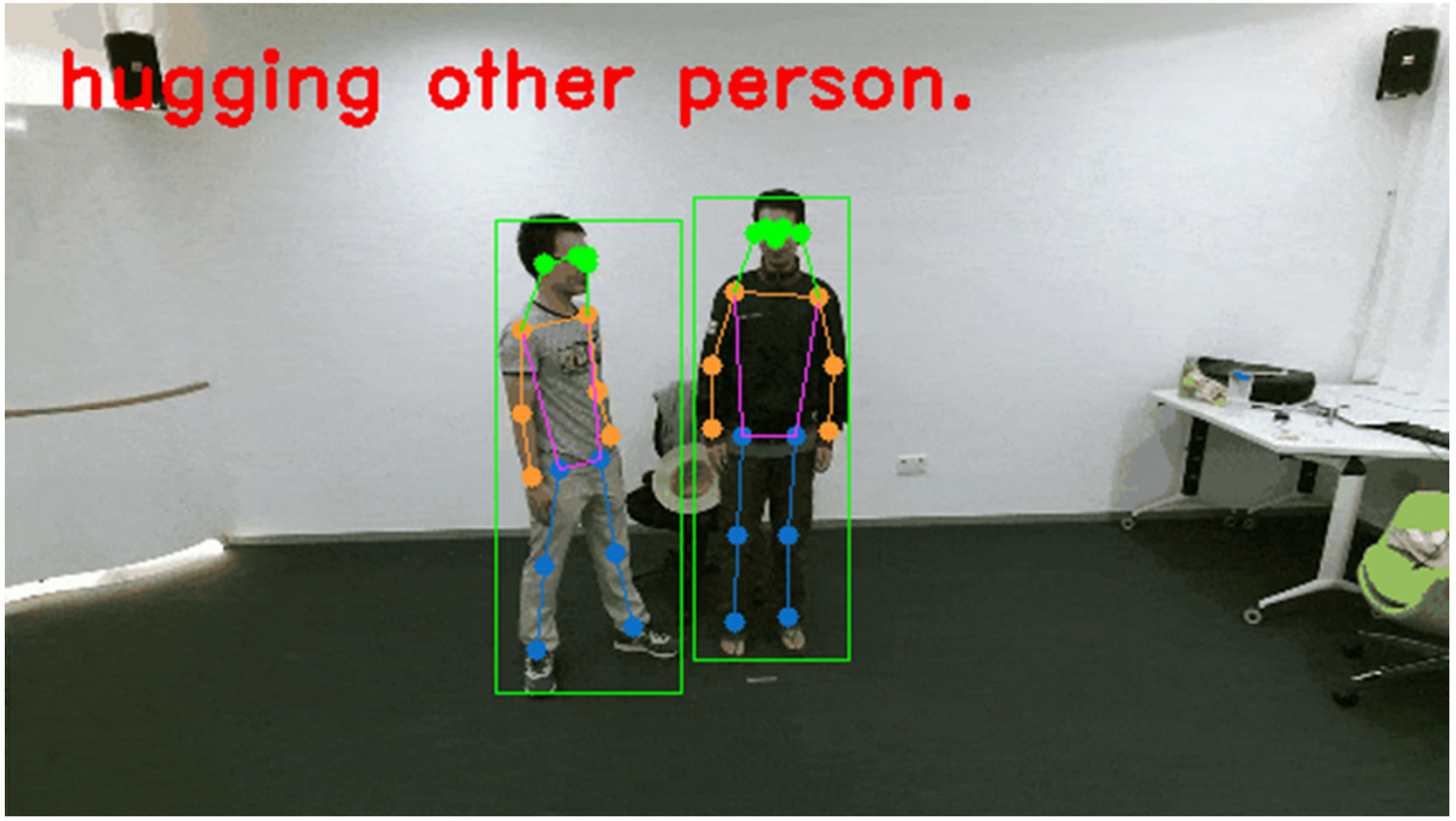
- To locate human body / facial keypoints from images or videos

Brief History & Milestones



Applications

- Action Recognition: use body keypoints to classify human actions



source: https://github.com/open-mmlab/mmaction2

Applications

- Animation and Motion Capture: build virtual character based on human pose.

source: http://www.norawillett.com/pose2pose/index.html

Dataset: Single-Person Pose Estimation

Dataset Name	Year	Single-Person	Multi-Person	Upper Poses	Full Poses	Various Poses	Number of Joints	Evaluation Metric	Number of Images / Videos		
									Train	Val	Test
Image-Based Datasets for Human Pose Estimation.											
LSP [71]	2010	✓			✓		14	PCP	1,000	-	1,000
LSP-Extended [72]	2011	✓			✓		14	PCP	10,000	-	-
Flic [142]	2013	✓		✓			10	PCP	5,000	-	1,016
Flic-Full [142]	2013	✓		✓			10	PCP	20,928	-	-
Flic-Plus [160]	2013	✓		✓			10	PCP	17,380	-	-
MPII [1]	2014	✓				✓	16	PCPm/PCKh	28,821	-	11,701
Video-Based Datasets for Human Pose Estimation.											
Penn Action [195]	2013	✓			✓		13	mAP	1,000	-	1,000
JHMDB [65]	2013	✓			✓		15	mAP	600	-	300

Source: Qi Dang, Jianqin Yin, Bin Wang, Wenqing Zheng, “Deep learning based 2D human pose estimation: A survey”

Dataset: Multi-Person Pose Estimation

Dataset Name	Year	Single-Person	Multi-Person	Upper Poses	Full Poses	Various Poses	Number of Joints	Evaluation Metric	Number of Images / Videos		
									Train	Val	Test
Image-Based Datasets for Human Pose Estimation.											
MPII [1]	2014		✓			✓	16	PCKh	3,800	-	1,700
COCO [93]	2017		✓			✓	17	AP	57,000	5,000	20,000
AIC-HKD [179]	2017		✓			✓	14	mAP	210,000	30,000	60,000
CrowdedPose [84]	2019		✓			✓	14	mAP	10,000	2,000	8,000
Video-Based Datasets for Human Pose Estimation.											
PoseTrack2017 [63]	2017		✓			✓	15	mAP	250	50	214
PoseTrack2018 [2]	2018		✓			✓	15	mAP	593	170	375
HiEve [94]	2020		✓			✓	14	mAP	19	-	13

Source: Qi Dang, Jianqin Yin, Bin Wang, Wenqing Zheng, “Deep learning based 2D human pose estimation: A survey”

Sample Images in MS-COCO Dataset

Source: https://cocodataset.org/#keypoints-2020

Performance Metrics

- PCK (Percentage of Correct Keypoint)
  - Measure the percentage of predicted keypoint that falls within a normalized distance of true joint, e.g.,
  - In Flic dataset, PCK@0.2: Distance between predicted and true joint < 0.2 x torso diameter.
  - In MPII dataset, PCKh@0.5 refers to the threshold = 50% of the head segment line.
- PCP (Percentage of Correct Parts)
  - Measure the localization accuracy of limbs.
  - If the distance between the two predicted joints and the true limb is less than a fraction of the limb length (e.g., 0.1-0.5), then the limb is considered detected correctly.

Performance Metrics

- AP and AR based on OKS (Object Keypoint Similarity)
  - Average Precision (AP), Average Recall (AR) and their variants are used in COCO, MPII and PoseTrack.
  - Object Keypoint Similarity (OKS) plays the same role as the IoU in detection.
  - OKS definition: the distance between predicted points and ground truth joints normalized by the scale of the person.

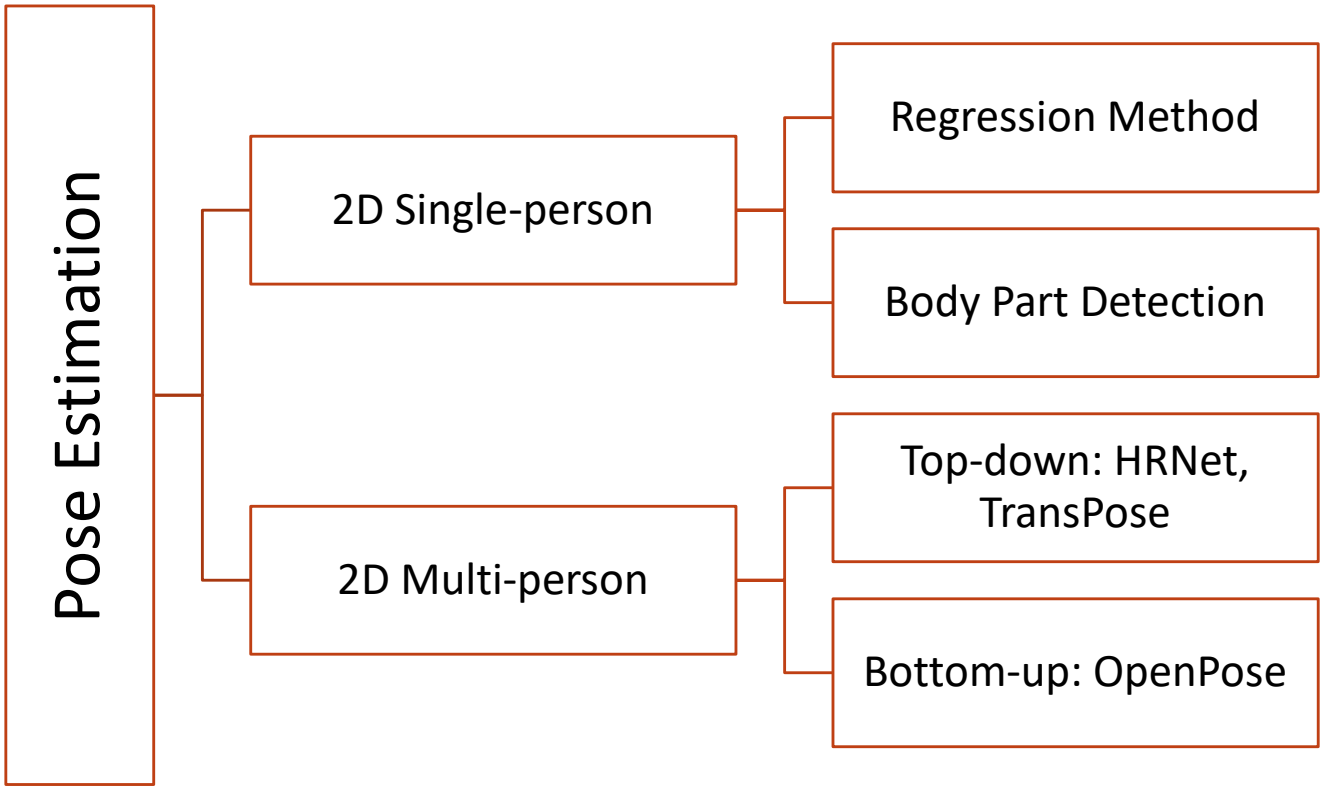
Metric of COCO dataset with OKS:

```
Average Precision (AP):
   AP                  % AP at OKS=.50:.05:.95 (primary challenge metric)
   AP^OKS=.50          % AP at OKS=.50 (loose metric)
   AP^OKS=.75          % AP at OKS=.75 (strict metric)
Average Recall (AR):
   AR                  % AR at OKS=.50:.05:.95
   AR^OKS=.50          % AR at OKS=.50
   AR^OKS=.75          % AR at OKS=.75
```

Challenges

- Computational efficiency
- Limited data for rare poses
- Occlusion

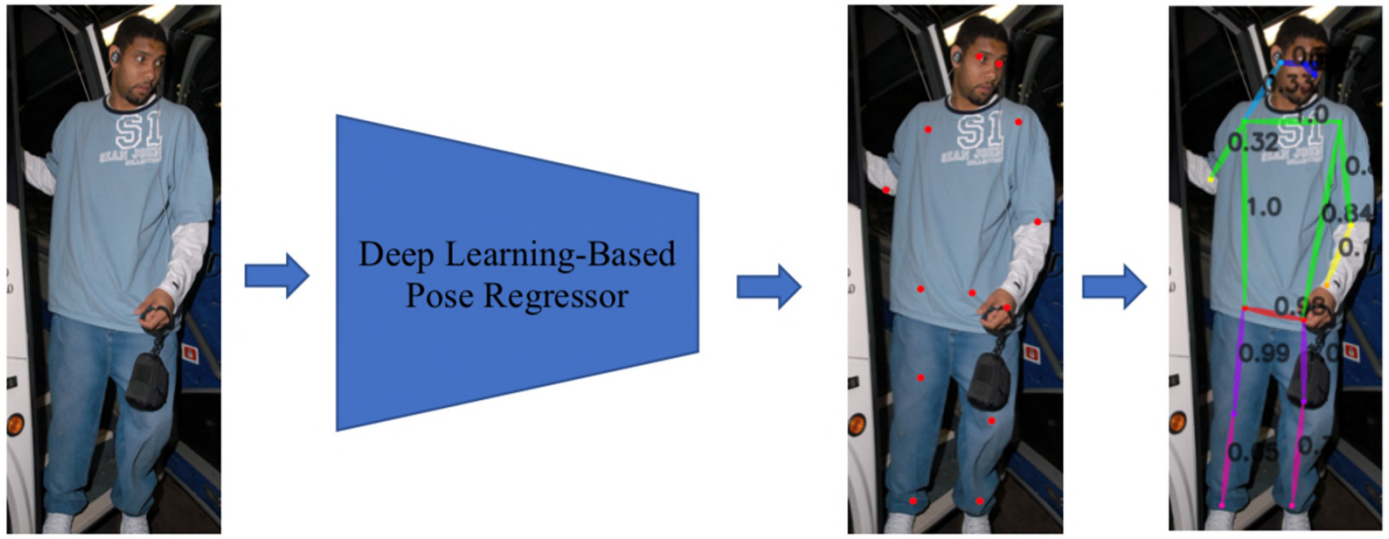
Pose Estimation Methods



Single-Person Pose Estimation

Regression Method

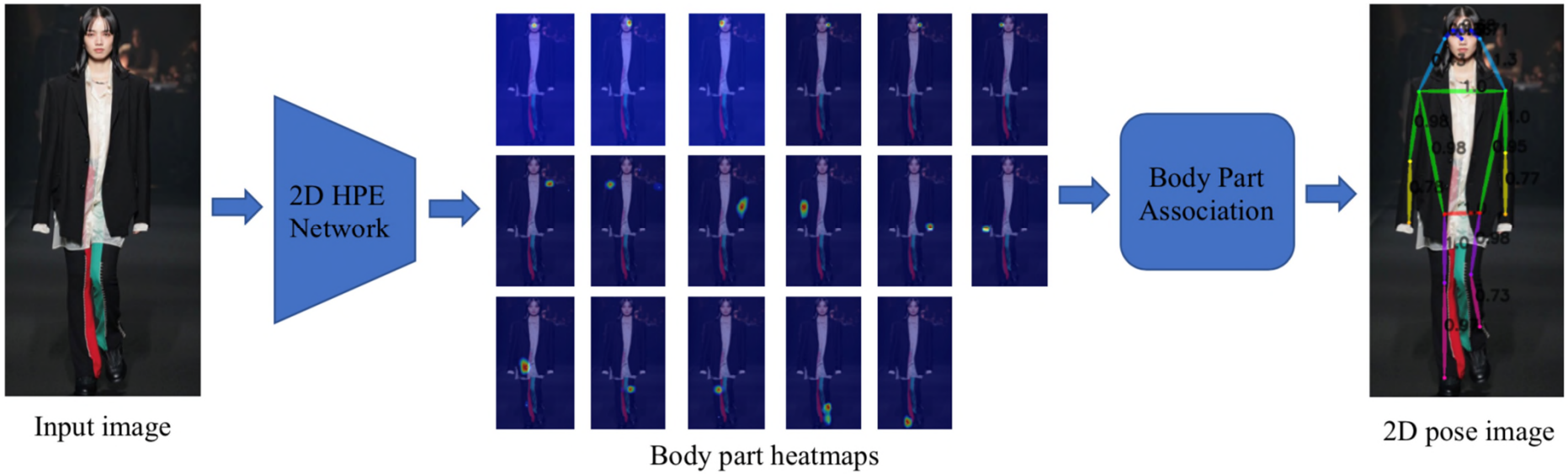
- Approach:
  - Obtain feature map from CNN
  - Calculate the regressor from the feature map
- Pros: easy and fast
- Cons: less accurate



Input image Keypoints (coordinates) 2D pose image

Body Part Detection

- Approach:
  - Estimate K heatmaps for a total of K keypoints
  - Assemble these detected keypoints into whole body pose or skeleton.
- Pros: accuracy
- Cons: slower and complex
- Representative work: Stacked Hourglass Networks for Human Pose Estimation (2016 ECCV)



Result Comparison on MPII dataset

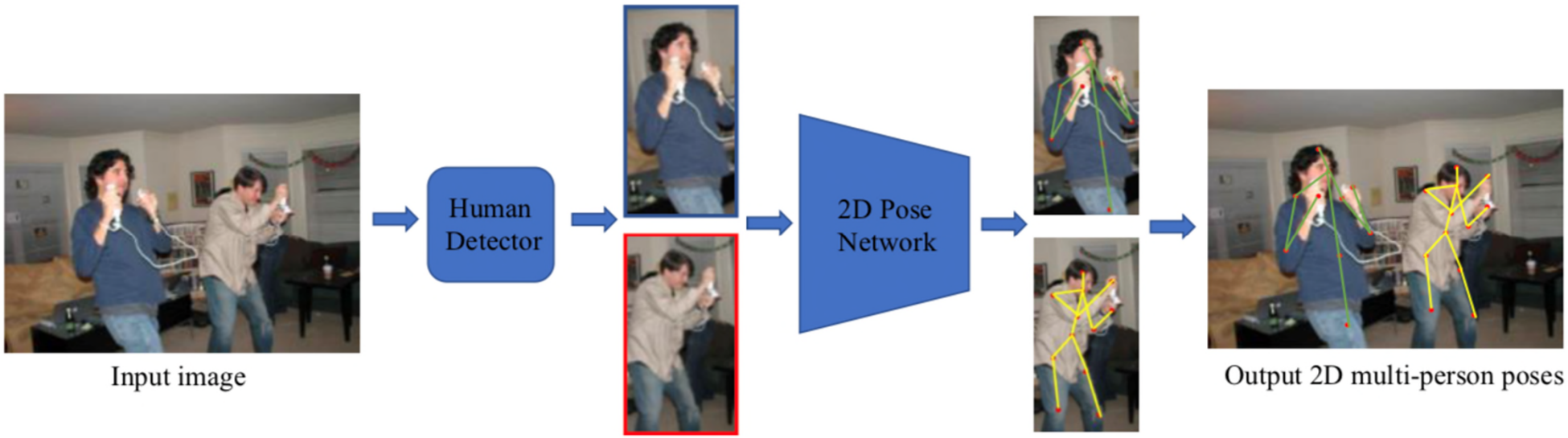
Results based on AP Using OKS

	Year	Model	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total
Body Part Detection	2016	Deepercut	96.8	95.2	89.3	84.4	88.4	83.4	78.0	88.5
	2016	CPM	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
	2016	Stacked hourglass	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
	2017	Multi-Context Attention	98.5	96.3	91.9	88.1	90.6	88.0	85.0	91.5
	2017	Adversatial Posenet	98.1	96.5	92.5	88.5	90.2	89.6	86.0	91.9
	2017	PRMs	98.5	96.7	92.5	88.7	91.1	88.6	86.0	92.0
	2018	MSS-MSR	98.5	96.8	92.7	88.4	90.6	89.3	86.3	92.1
	2018	DLCM	98.4	96.9	92.6	88.7	91.8	89.4	86.2	91.5
	2019	PGNN	98.6	97.0	92.8	88.8	91.7	89.9	86.6	92.5
	2019	MSPN	98.4	97.1	93.2	89.2	92.0	90.1	85.5	92.6
	2020	Unipose	-	-	-	-	-	-	-	92.7
Regression	2016	IEF	95.7	91.7	81.7	72.4	82.8	73.2	66.4	81.3
	2017	Compositional pose regression	97.5	94.3	87.0	81.2	86.5	78.5	75.4	86.4
	2019	FPD	98.3	96.4	91.5	87.4	90.0	87.1	83.7	91.1

Multi-Person Pose Estimation

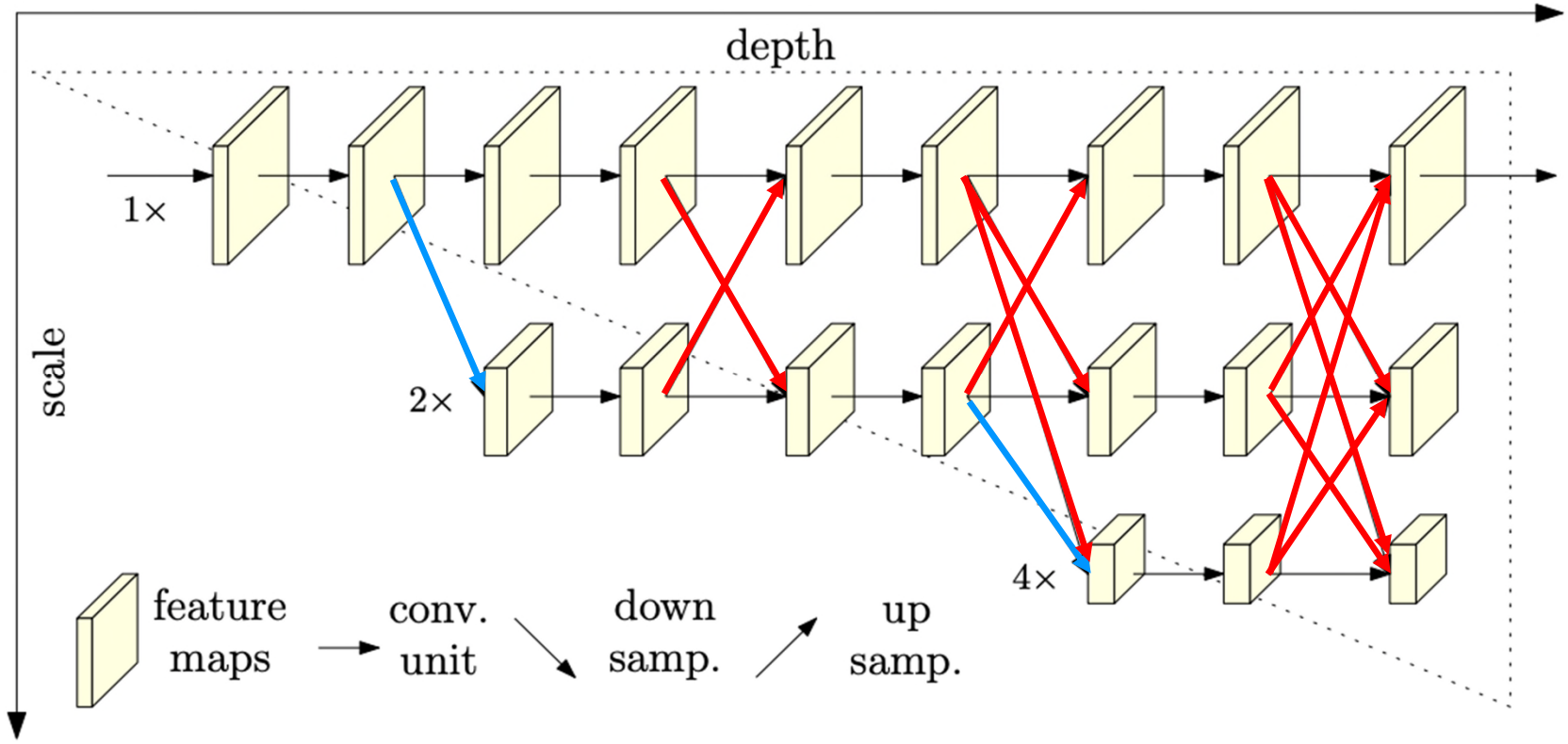
Top-Down Methods

- Approach:
  - Detect persons in the image with off-the-shelf detector
  - Estimate the pose for every single person
- Representative work: HRNet (High-Resolution Net)
- Pros: less network complexity, accurate
- Cons: runtime increases with the increase of the number of detected persons.
- Suitable for the scenes with few people.



HRNet (High-Resolution Net)

- Start from a high-resolution subnet, add high-to-low resolution subnet gradually.
- Connect multi-resolution subnets in parallel.
- Multi-scale fusions: exchanging information across parallel multi-resolution subnets.



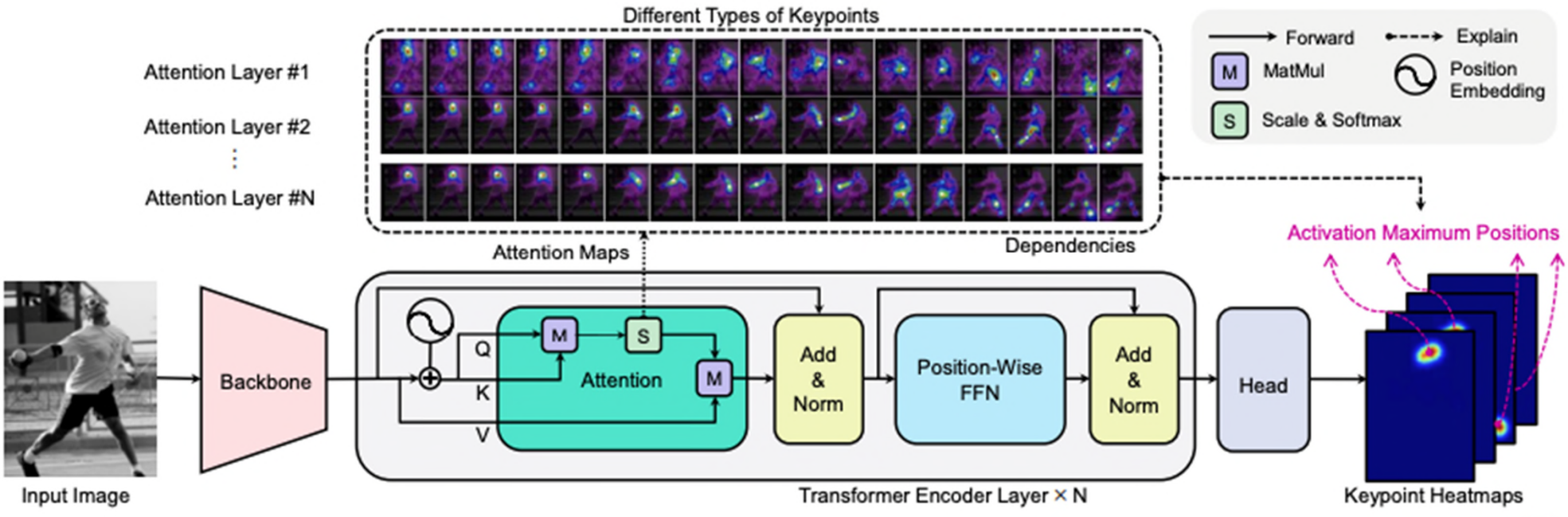
Bottom-up Methods

- Approach:
  - Detect all body parts in the image
  - Assemble the body parts into poses for different persons
- Representative work: OpenPose
- Pros: runtime is mostly constant with respect to different number of detected persons.
- Cons: complex
- Suitable for crowded scene

Emerging / New Methods

TransPose

- A transformer-based pose estimation method
- Backbone: common CNNs to extract features, e.g., ResNet and HRNet, but only keep initial several layers
- Transformer: only use encoder, with N attention layers and feed-forward networks (FFN)
- Head: use convolution layers to predict keypoint heatmaps



Comparison on MS-COCO for Multi-Person Pose Estimation

	Year	Model	Backbone	Input size	AP	AP.5	AP.75	AP(M)	AP(L)
Top-down	2017	Mask-RCNN	ResNet50	-	63.1	87.3	68.7	57.8	71.4
	2017	G-RMI	ResNet101	353x257	68.5	87.1	75.5	65.8	73.3
	2018	CPN*	ResNet-Inception	384x288	72.1	91.4	80.0	68.7	77.2
	2018	SimpleBaseline	ResNet152	384x288	73.7	91.9	81.1	70.3	80.0
	2019	HRNet-W32	HRNet-W32	384x288	74.9	92.5	82.8	71.3	80.9
	2019	HRNet-W48*	HRNet-W48	384x288	77.0	92.7	84.5	73.4	83.1
	2020	DARK*	HRNet-W48	384x288	77.4	92.6	84.6	73.6	83.7
	2020	RSN	4xRSN50	384x288	78.6	94.3	86.6	75.5	83.3
	2021	TransPose	HRNet-Small-W48	256x192	75.0	92.2	83.6	72.5	82.4
Bottom-up	2017	OpenPose	-	-	61.8	84.9	67.5	57.1	68.1
	2017	AE	Hourglass	512x512	65.5	86.8	72.3	60.6	72.6
	2018	PersonLab	ResNet152	1401x1401	68.7	89.0	75.4	64.1	75.5
	2020	HGG	Hourglass	512x512	67.6	85.1	73.7	62.7	74.6
	2020	HrHRNet-W48+AE	HRNet-W48	640x640	70.5	89.3	77.2	66.6	75.8
	2021	DERK-W48	HRNet-W48	640x640	71.0	89.2	78.0	67.1	76.9
	2021	SWAHR+HrHRNet	HRNet-W48	-	72.0	90.7	78.8	67.8	77.7

* using extra data

Pose Estimation Summary

- This section covers the following:
  - Introduction
  - Single-Person Pose Estimation
  - Multi-Person Pose Estimation
  - Emerging / New Methods

Video / Human Action Recognition (HAR)

Human Action Recognition Overview

- Introduction
- Action Recognition Methods
  - Two-Stream Networks
  - 3D CNNs
  - Efficient Video Modelling
- New / Emerging Methods

Objective

- To recognize human actions in a video.

Shaking hands

Cricket bowling

Skate boarding

Cutting in kitchen

Stretching leg

Riding a bike

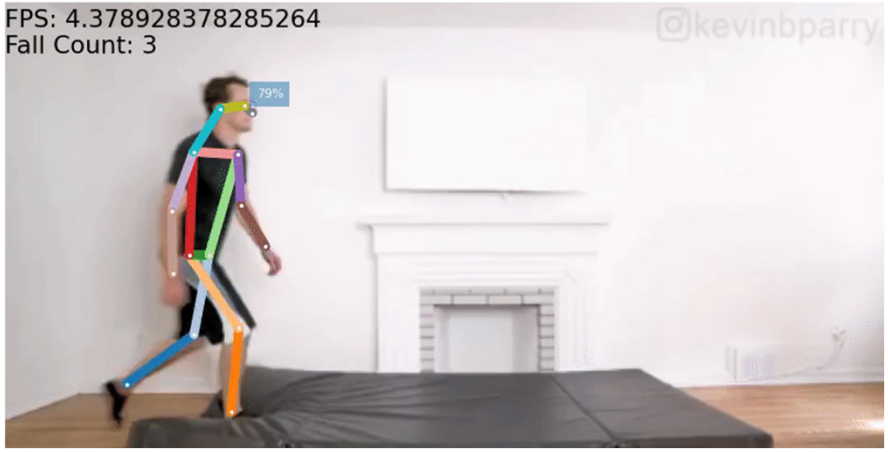
Playing violin

Dribbling basketball

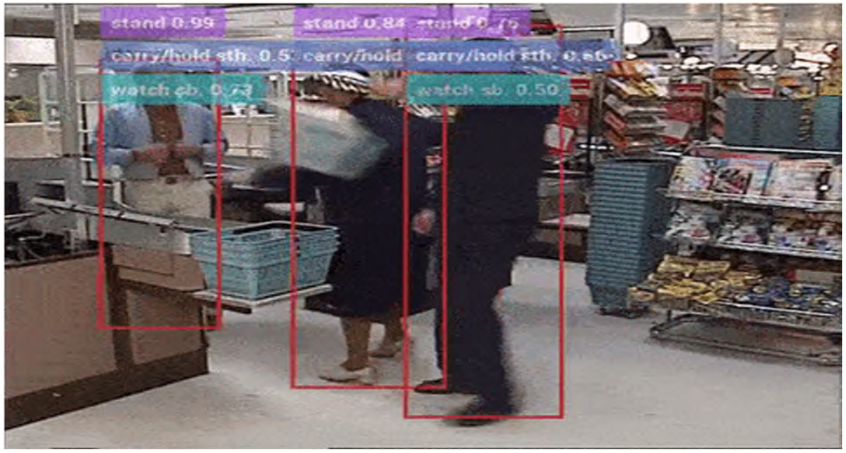
Applications

Monitoring

Control

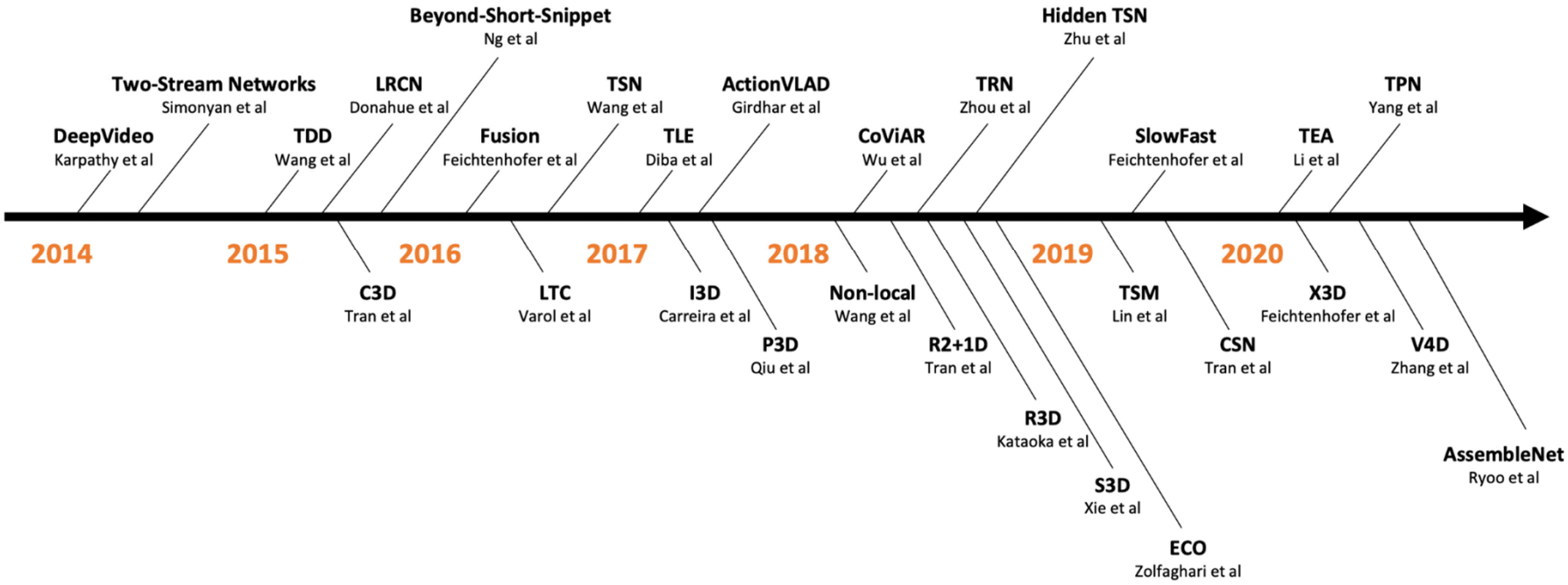


Danger detection



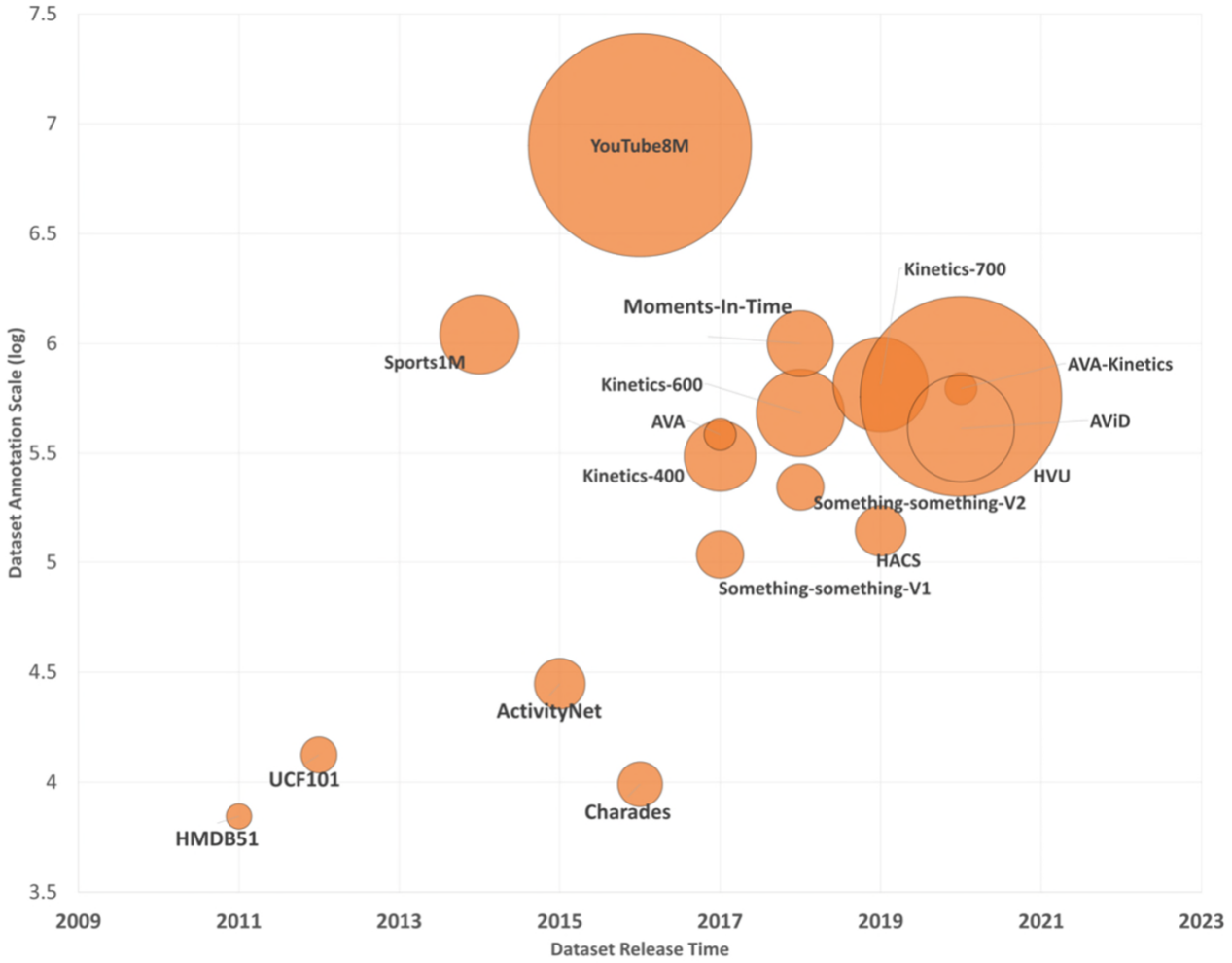
Event analysis

Brief History & Milestones



HAR Datasets

- High-quality large-scale action recognition datasets have emerged over the last decade.



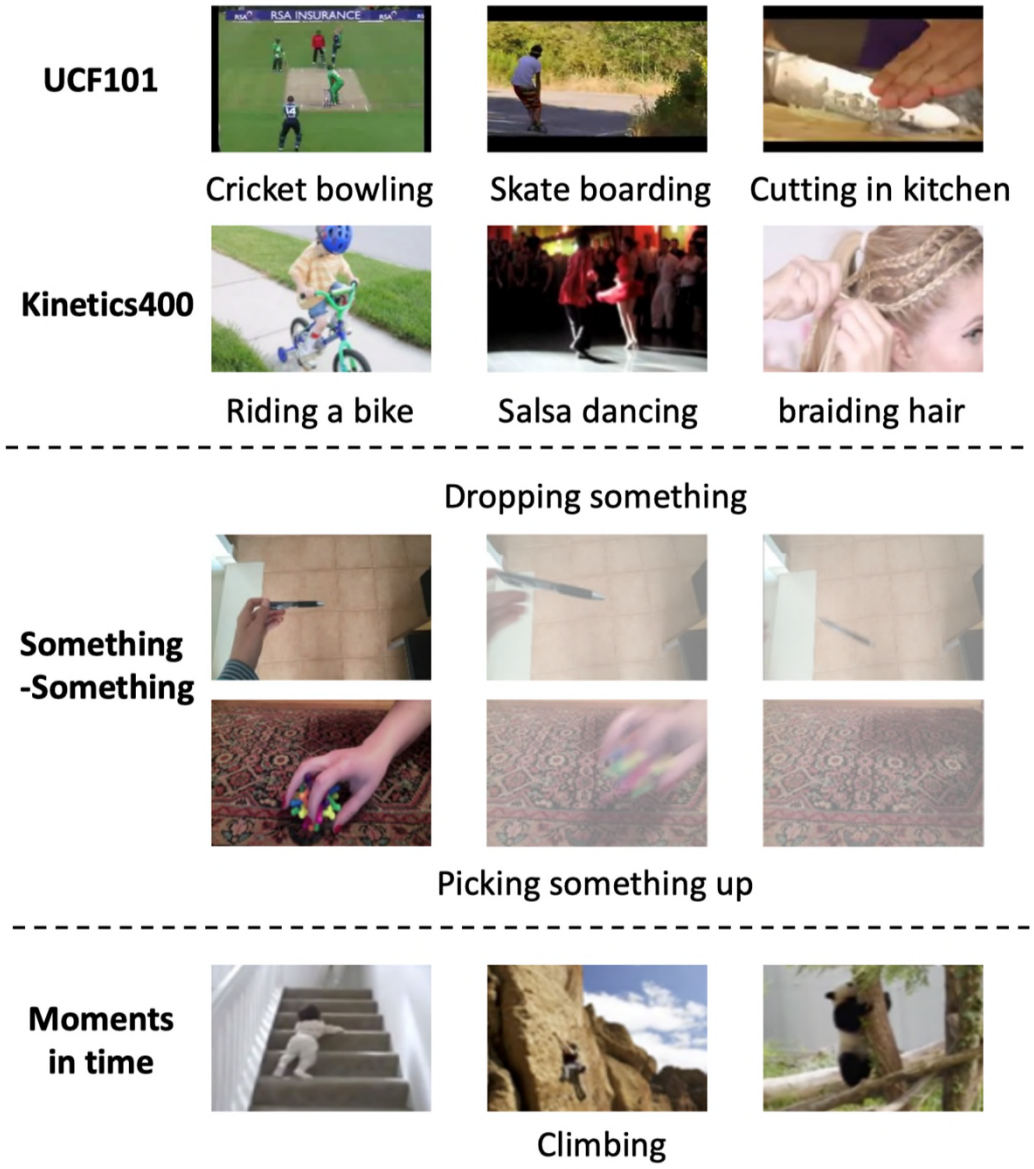
HAR Datasets

Dataset	Year	# Samples	Ave. Len	# Actions	Comments
HMDB51	2011	7K	~5s	51	-
UCF101	2012	13.3K	~6s	101	-
Sports1M	2014	1.1M	~5.5m	487	First large-scale video action dataset
ActivityNet	2015	28K	[5, 10]m	200	Human daily living actions
YouTube8M	2016	8M	229.6s	3862	Largest-scale video dataset
Charades	2016	9.8K	30.1s	157	Multi-label daily indoor activities
Kinetics400	2017	306K	10s	400	Most widely adopted benchmark
Kinetics600	2018	482K	10s	600	
Kinetics700	2019	650K	10s	700	

HAR Datasets

Dataset	Year	# Samples	Ave. Len	# Actions	Comments
Sth-Sth V1	2017	108.5K	[2, 6]s	174	Requires strong temporal modeling
Sth-Sth V2	2017	220.8K	[2, 6]s	174	
AVA	2017	385K	15m	80	First large-scale spatio-temporal action detection dataset
AVA-kinetics	2020	624K	15m	80	
Moment in Time	2018	1M	3s	339	Designed for event understanding
HACS Clips	2019	1.55M	2s	200	Consist of segments
HVU	2020	572K	10s	739	This dataset has six task categories: scene, object, action, event, attribute, and concept
AViD	2020	450K	[3, 15]s	887	Remove face identities to protect privacy of video makers
NTU RGB+D (S)	2016	-	-	120	Introduce depth and skeleton information

Samples in HAR Datasets



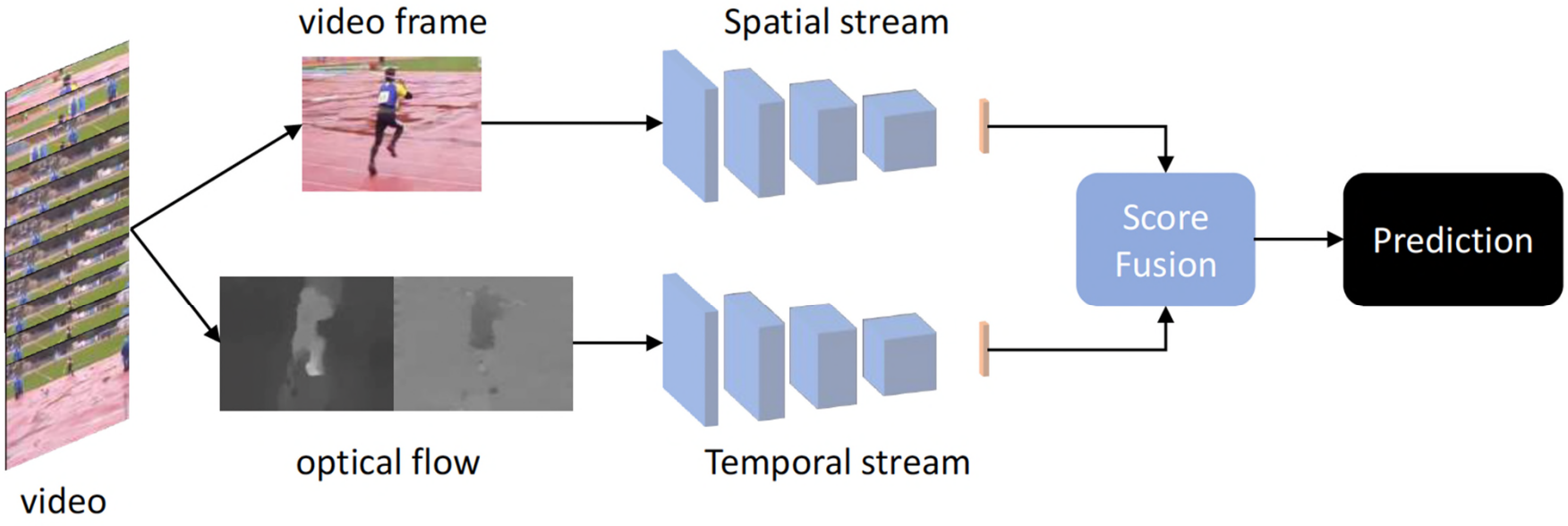
HAR Methods

- Two-Stream Networks
- 3D CNNs
- Efficient Video Modelling
- Etc.

Two-Stream Networks

Two-Stream Networks

- Consist of a spatial stream, a temporal stream and a score fusion module:
  - Spatial stream captures appearance information from sampled RGB frames.
  - Temporal stream recognizes motion from optical flow.
  - Score fusion module combines scores from the spatial and temporal streams.

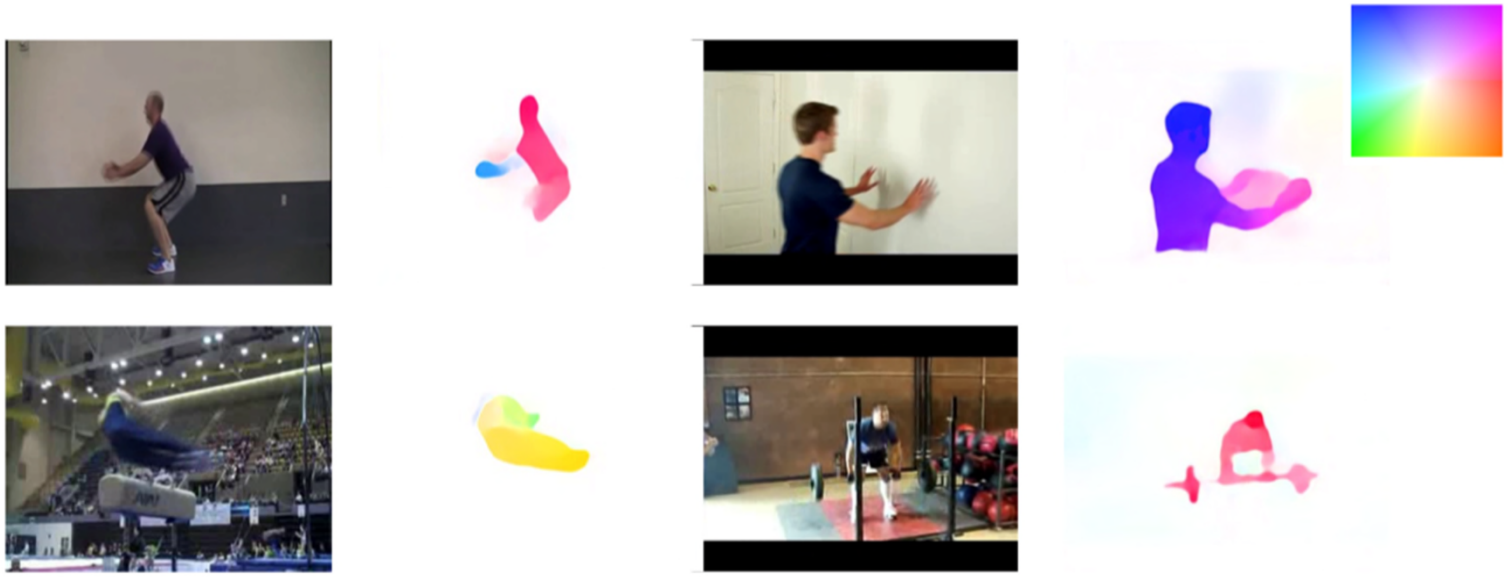


Two-stream Networks

Source: A Comprehensive Study of Deep Video Action Recognition

Optical Flow

- An effective motion representation to describe object/scene movement.
- Optical flow can describe the motion pattern of each action.
- Pros: provide orthogonal information to RGB image.
- Cons: Optical flow is computationally intensive & has large storage requirement.



Two-Stream Networks

- Spatial stream
  - Sample video frame from video clips.
  - Input the sampled frame to a CNN network.
  - Output the spatial prediction score from a fully-connected layer.
- Temporal stream
  - Estimate the horizontal and vertical optical flows from stacked frames.
  - Concatenate the two optical flow images and input to a CNN.
  - Output the temporal prediction score from a fully-connected layer.
- Fusion
  - Use different fusion techniques (e.g., average, weighted average, etc.) to obtain final prediction score from both spatial and temporal scores.

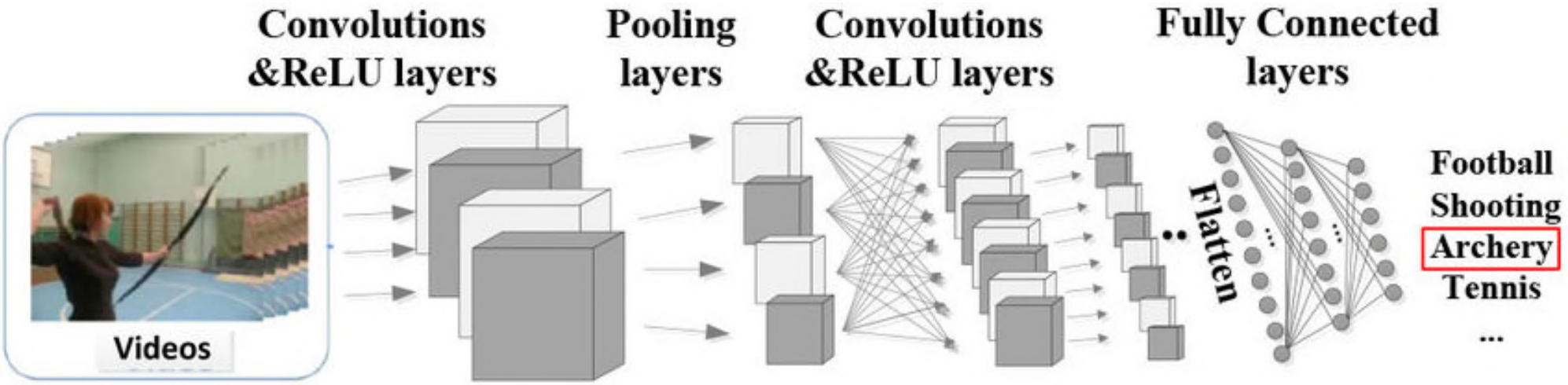
Pros and Cons in Two-Stream Networks

- Pros
  - Simple network structure.
  - Optical flow is an accurate way to represent motion information.
- Cons
  - Optical flow is expensive to extract and store.

3D CNNs

3D CNNs

- 3D CNN treats a video as a 3D tensor with two spatial and one time dimension.

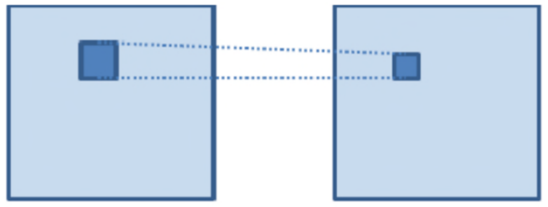


Source: https://www.researchgate.net/figure/Real-life-3-D-CNN-model-for-video-classification_fig1_332517153

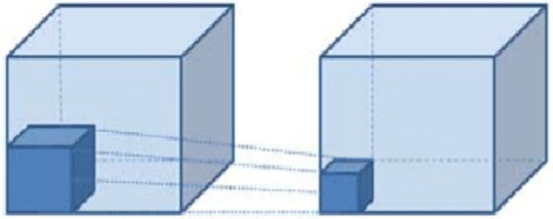
3D CNNs

- Designed to handle 3D tensors.
- Traditional 2D Convolution: dot product between input volume with different filters, where each filter and input are 2D matrices (ignore the depth dimension).
- 3D Convolution: dot product on 3D tensors.
- Representative work: SlowFast Network

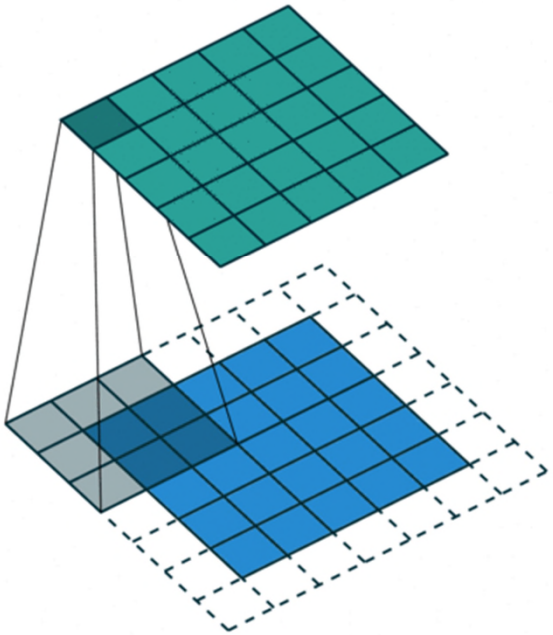
2D CNNs vs. 3D CNNs

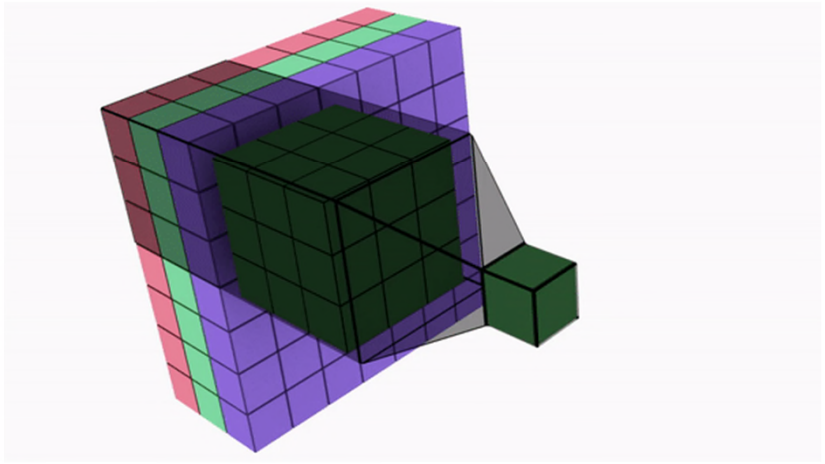


(*a*) 2D convolution



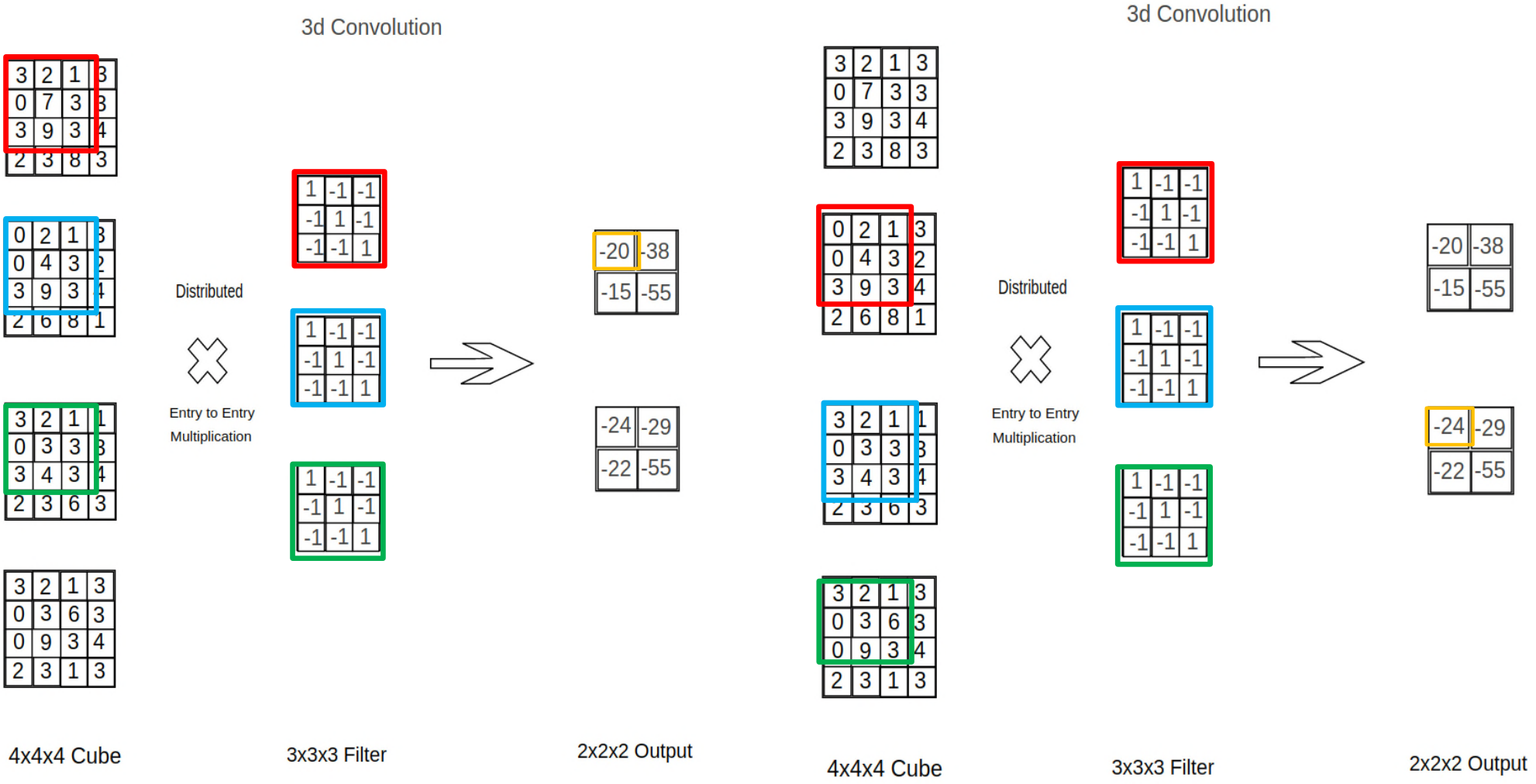
(*b*) 3D convolution





3D Convolution

- Assume depth channel = 1



Pros and Cons

- Pros:
  - Feature extractor can obtain temporal and spatial information simultaneously.
- Cons:
  - Hard to train and deploy due to large model and low inference efficiency, e.g. a standard SlowFast network trained on Kinetics400 dataset using a high-end 8-GPU machine takes 10 days to complete.

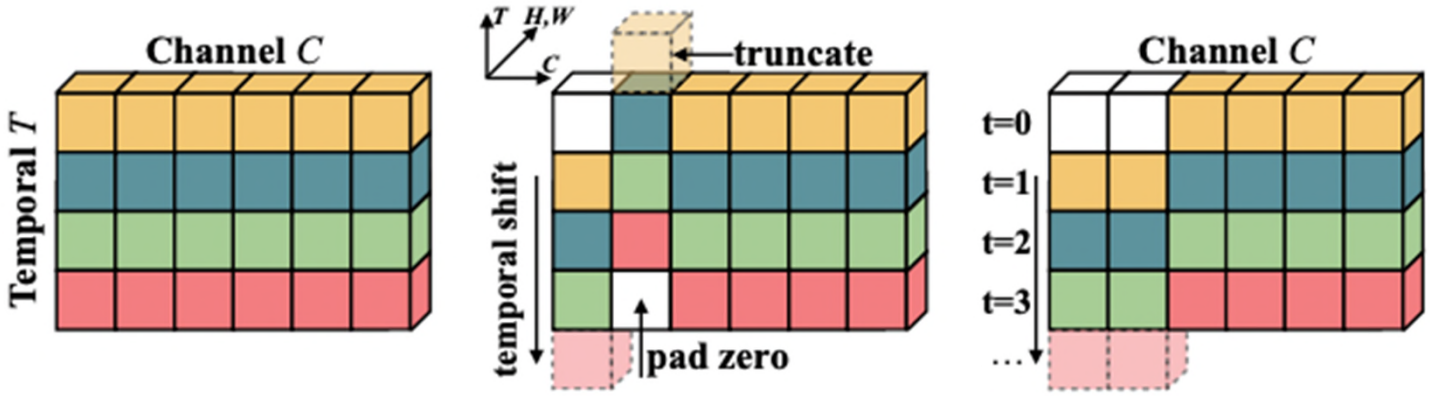
Efficient Video Modelling

Efficient Video Modelling

- Improve accuracy and efficiency at the same time for video action recognition.
- Features:
  - Encode motion information without using optical flow.
  - Temporal modeling without 3D convolution.
- Representative work: Temporal Shift Module (TSM)
- Pros
  - Fast and efficient.
- Cons
  - Not as accurate as 3D CNNs.

Temporal Shift Module (TSM)

- Shifting Operation
  - It shifts part of the channels along the temporal dimension, thus facilitating information exchange among neighboring frames.
  - Two shift options: (1) bi-direction for offline settings and (2) uni-direction for online settings.



(a) The original tensor without shift. **(b)** Offline temporal shift (bi-direction). **(c)** Online temporal shift (uni-direction).

TSM: Temporal Shift Module for Efficient Video Understanding. ICCV, 2019.

Emerging / New Methods

Transformer-based Methods

- Existing approaches encode only short-range dependencies.
- Long-range dependency modelling is desirable in many applications such as action recognition.
- Representative work: Video Swin transformer
- Pros
  - Capture long-term dependencies.
- Cons
  - Computationally intensive and data hungry.

Video Swin Transformer

- ViT-based methods require large computational resource as self-attention module consider all tokens.
- Video Swin Transformer aims to reduce model complexity while keeping the ability to capture long-term dependencies.

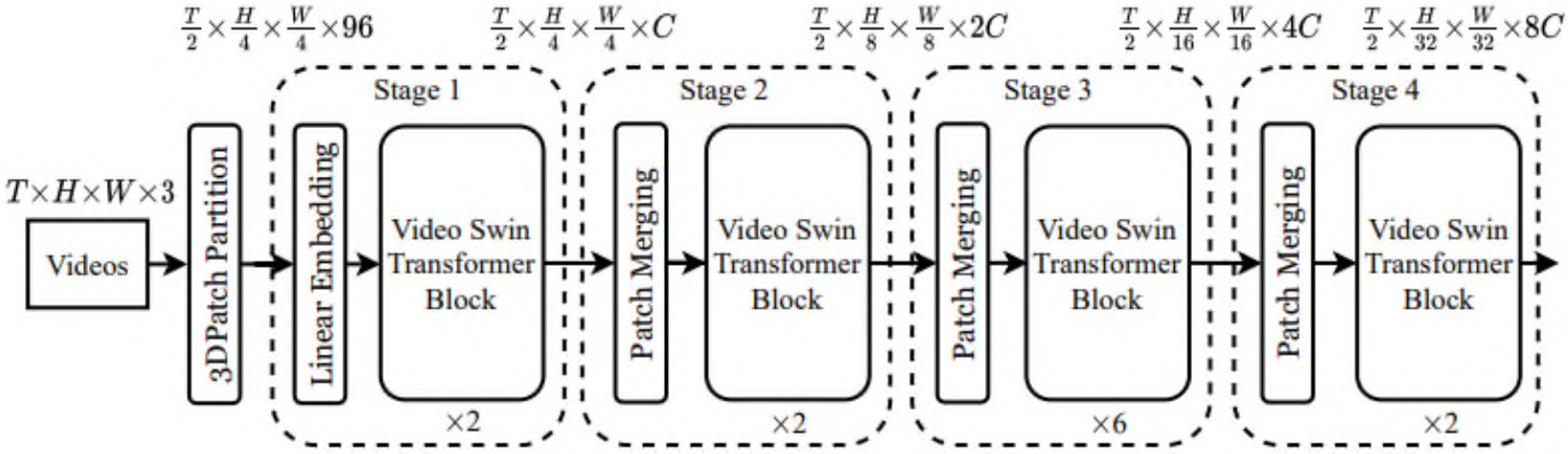
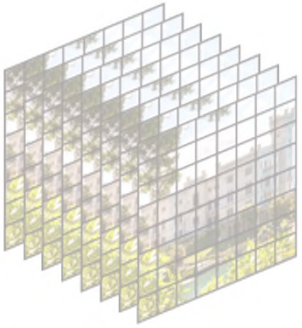


Figure 1: Overall architecture of Video Swin Transformer (tiny version, referred to as Swin-T).

Source: Video Swin transformer. CVPR, 2022.

Video Swin Transformer

- 3D Patch Partition
  - Input video size: T×H×W×3.
  - 3D token / patch size: 2×4×4×3.



3D tokens: T'×H'×W' = 8×8×8

- Linear Projection / Embedding
  - Use a linear layer to project the 3D token from number T'×H'×W'×96 to T'×H'×W'×C in channel dimension.

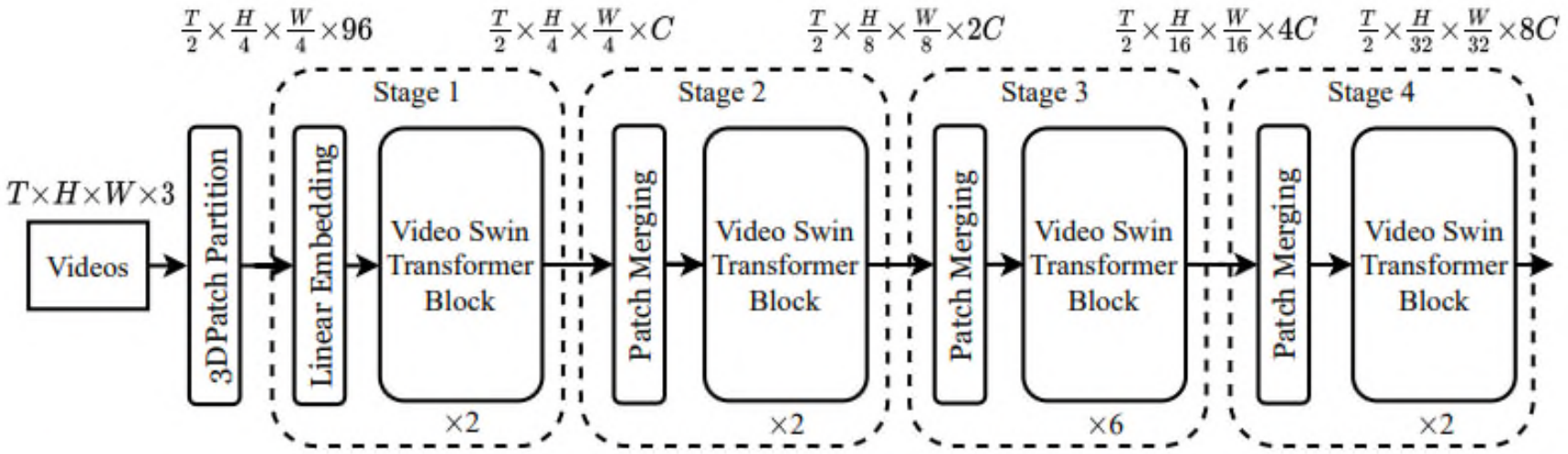


Figure 1: Overall architecture of Video Swin Transformer (tiny version, referred to as Swin-T).

Video Swin Transformer Block

- In layer l, partition the 3D tokens into windows of size P×M×M, and perform self-attention in every local window.
- In layer l+1, shift the window with a certain step (P/2 x M/2 x M/2), organize tokens into new windows, and calculate self-attention within each shift window.

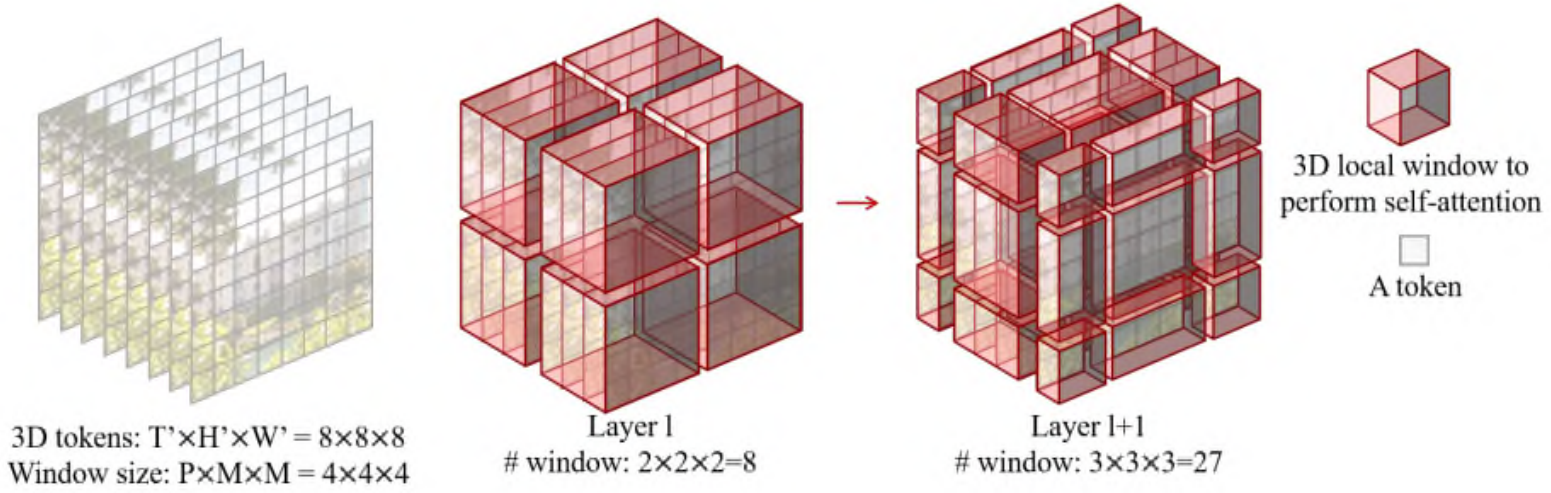


Figure 3: An illustrated example of 3D shifted windows. The input size $T' \times H' \times W'$ is $8 \times 8 \times 8$, and the 3D window size $P \times M \times M$ is $4 \times 4 \times 4$. As layer l adopts regular window partitioning, the number of windows in layer l is $2 \times 2 \times 2 = 8$. For layer $l+1$, as the windows are shifted by $(\frac{P}{2}, \frac{M}{2}, \frac{M}{2}) = (2, 2, 2)$ tokens, the number of windows becomes $3 \times 3 \times 3 = 27$. Though the number of windows is increased, the efficient batch computation in [28] for the shifted configuration can be followed, such that the final number of windows for computation is still 8.

Video Swin Transformer

- Patching merging
  - Perform 2× spatial downsampling and concatenate features of each 2×2 spatially neighboring patches.
  - Do not downsample along the temporal dimension.
  - Apply a linear layer to project the concatenated features.

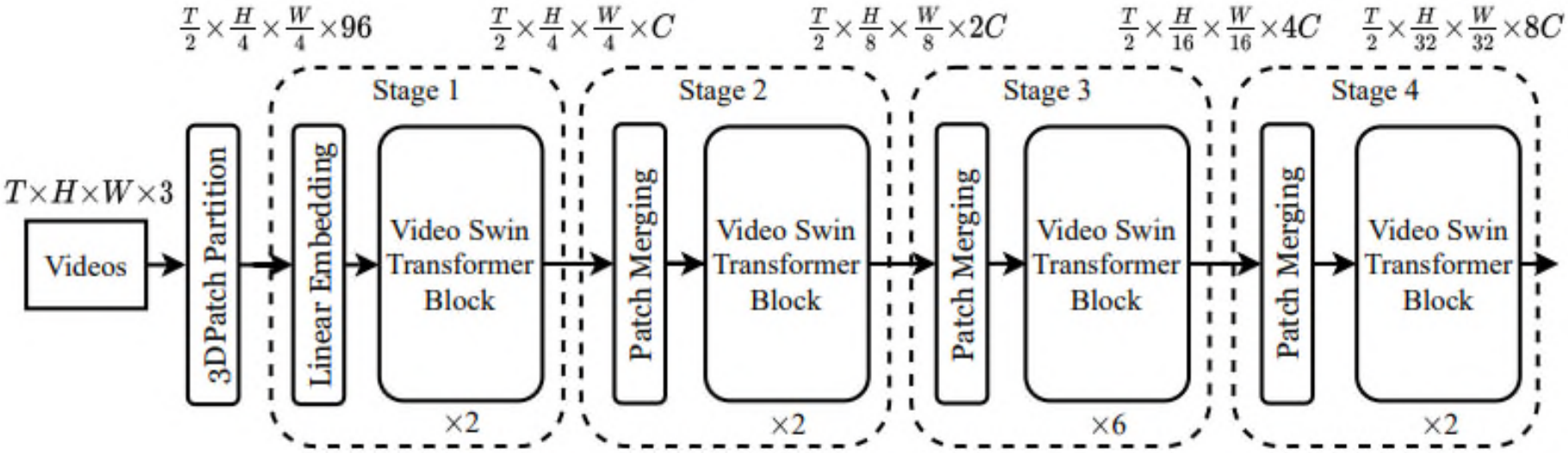


Figure 1: Overall architecture of Video Swin Transformer (tiny version, referred to as Swin-T).

Architecture Variants

- Swin-T: $C = 96$, layer numbers = $\{2, 2, 6, 2\}$
- Swin-S: $C = 96$, layer numbers =$\{2, 2, 18, 2\}$
- Swin-B: $C = 128$, layer numbers =$\{2, 2, 18, 2\}$
- Swin-L: $C = 192$, layer numbers =$\{2, 2, 18, 2\}$

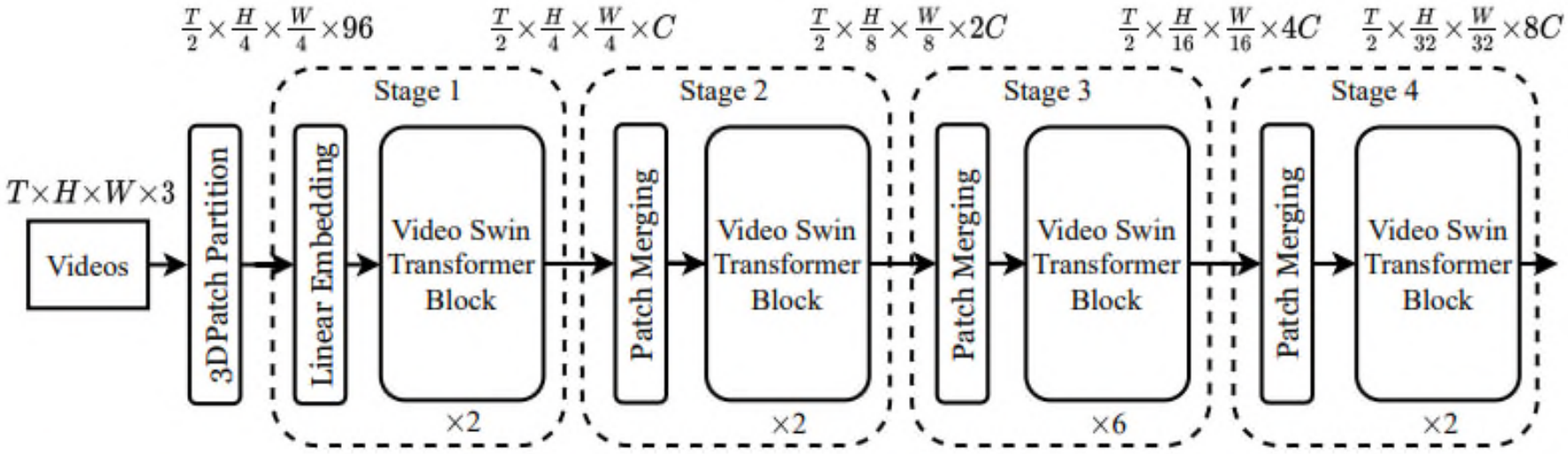


Figure 1: Overall architecture of Video Swin Transformer (tiny version, referred to as Swin-T).

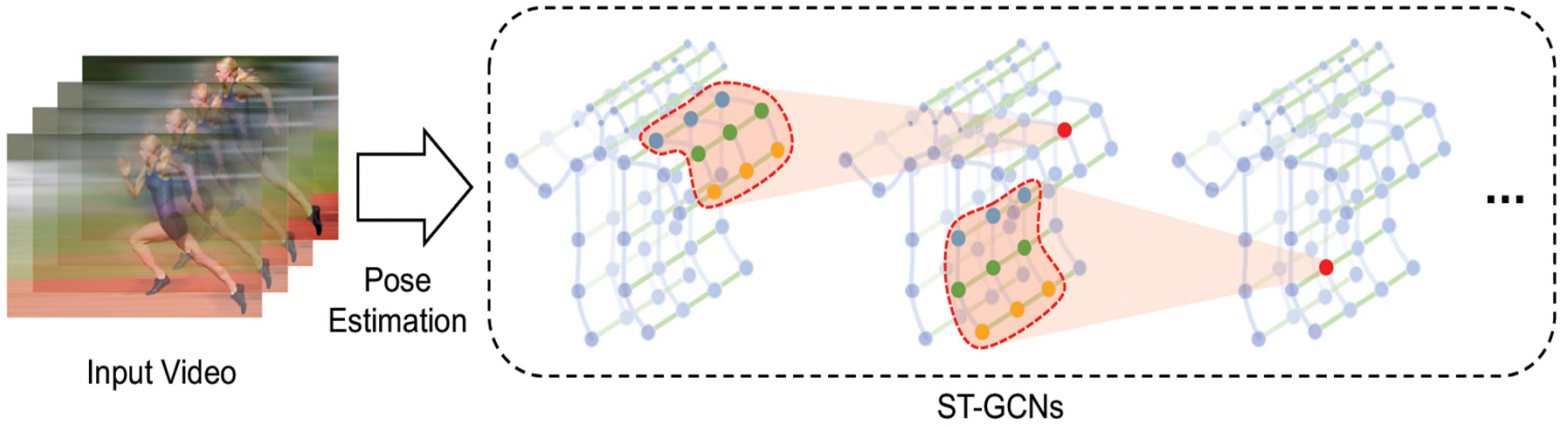
Performance Comparison

Table 2: Comparison to state-of-the-art on Kinetics-600.

Method	Pretrain	Top-1	Top-5	Views	FLOPs	Param
SlowFast R101+NL [13]	-	81.8	95.1	10 × 3	234	59.9
X3D-XL [12]	-	81.9	95.5	10 × 3	48	11.0
MViT-B-24, 32×3 [9]	-	83.8	96.3	5 × 1	236	52.9
TimeSformer-HR [3]	ImageNet-21K	82.4	96	1 × 3	1703	121.4
ViViT-L/16x2 320 [1]	ImageNet-21K	83.0	95.7	4 × 3	3992	310.8
ViViT-H/16x2 [9]	JFT-300M	85.8	96.5	4 × 3	8316	647.5
Swin-B	ImageNet-21K	84.0	96.5	4 × 3	282	88.1
Swin-L (384↑)	ImageNet-21K	85.9	97.1	4 × 3	2107	200.0
Swin-L (384↑)	ImageNet-21K	**86.1**	**97.3**	10 × 5	2107	200.0

Skeleton-based Networks

- Utilize temporal pose information to predict actions.
- Methods
  - Extract pose information from videos.
  - Use techniques such as LSTM or Graph Neural Network to classify actions.

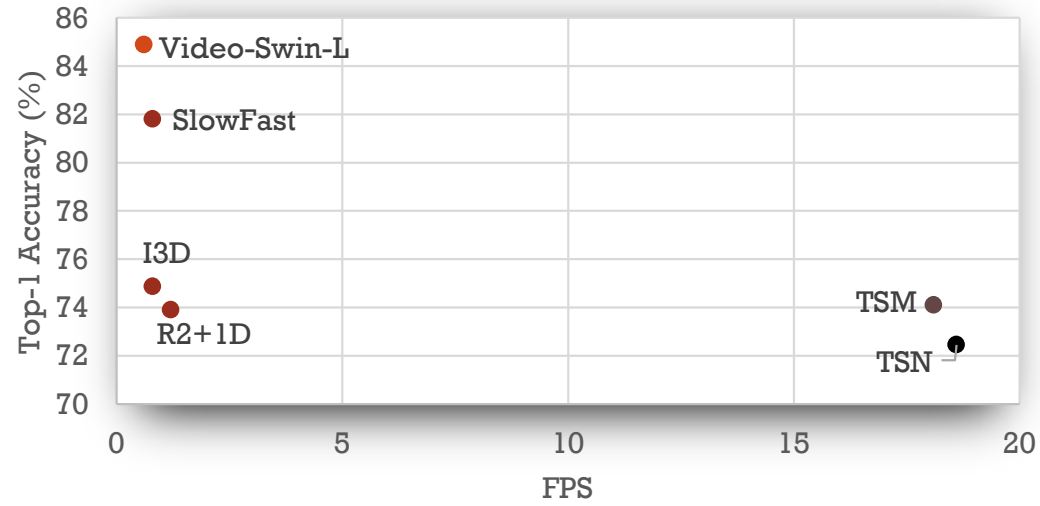


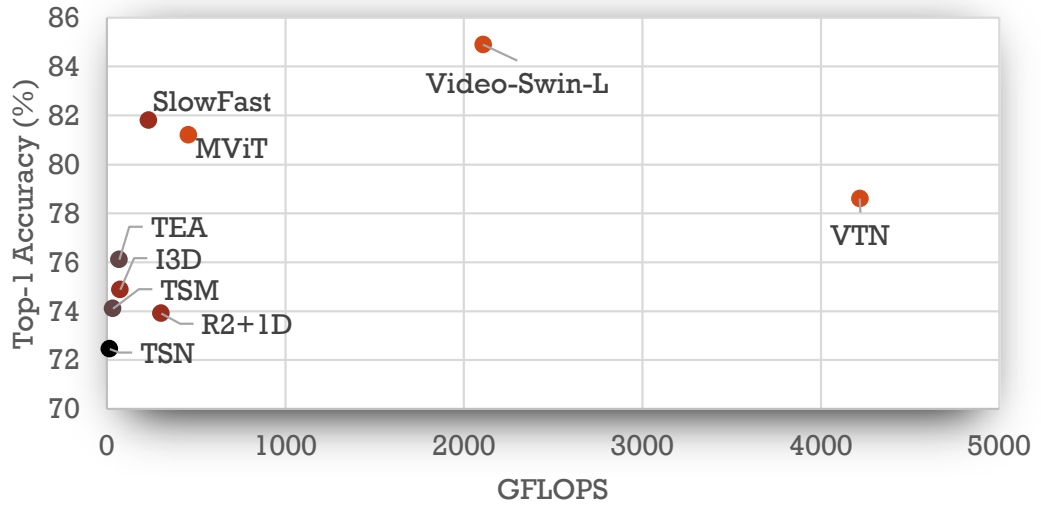
Source: Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. AAAI, 2018.

Performance Comparison

Methods	Representative Models			Input Size	GFLOPs × views	Accuracy (Top1 %) (Kinetics 400)	FPS	Remarks
	Model	Year	Venue					
Two-stream networks	TSN	2016	ECCV	8×3×224×224	16×250	72.45	18.6	Simple design, significant computation and storage requirement for optical flows
3D CNNs	I3D	2017	CVPR	32×3×224×224	108×N/A	74.87	0.8	Complex, very large computation consumption in training
	SlowFast	2019	CVPR	32×3×224×224	234×30	81.8	0.8	
Efficient video modeling	TSM	2019	ICCV	16×3×224×224	33×30	74.1	18.1	Simple, efficient training, fast runtime, relatively accurate
	TDN	2021	CVPR	24×3×224×224	198×30	79.4	-	
Transformers	VTN	2021	ICCV	250×3×224×224	4218 × 1	78.6	-	Accurate, large data requirement, high computational cost
	Video Swin-L	2022	CVPR	32×3×384×384	2107 × 50	84.9	0.6	
Skeleton based networks	ST-GCN	2018	AAAI	-	-	30.7 (Kinetics 400) 86.9 (NTU60_XSub)	-	Leverage on human pose, lower accuracy in broad domain applications.
	PoseC3D	2021	ArXiv	8×3×224×224	-	47.4 (Kinetics 400) 94.3 (NTU60_XSub)	-	

Performance Comparison





Two-stream Methods Efficient Video Modeling 3D CNNs Transformer-based Methods

Exercise: Human Action Recognition

(d) Briefly describe how Temporal Shift Module (TSM) can achieve good computational efficiency in video action recognition.

(6 Marks)

Solution

Human Action Recognition Summary

- The section cover the following:
  - Introduction
  - Action Recognition Methods
    - Two-Stream Networks
    - 3D CNNs
    - Efficient Video Modelling
  - New / Emerging Methods

Part 5 Summary

- The section covers the following topics:
  - Object Detection & Tracking
  - Pose Estimation
  - Video / Human Action Recognition