# Transformers In Computer Vision

## Transformers in Image Segmentation

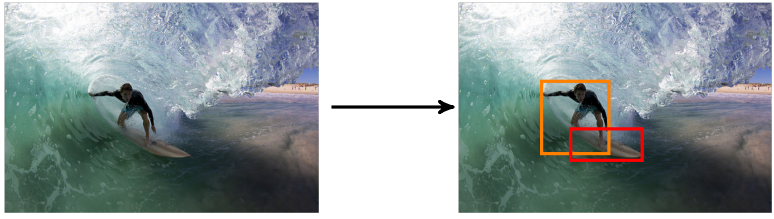
Sergey Zagoruyko

October 13, 2025

# Object Detection Task

- Given an input image, predict **a set** of bounding boxes with corresponding categories

set(("person", $\text{bbox}_1$), ("surfboard", $\text{bbox}_2$))



# CV problems: Segmentation

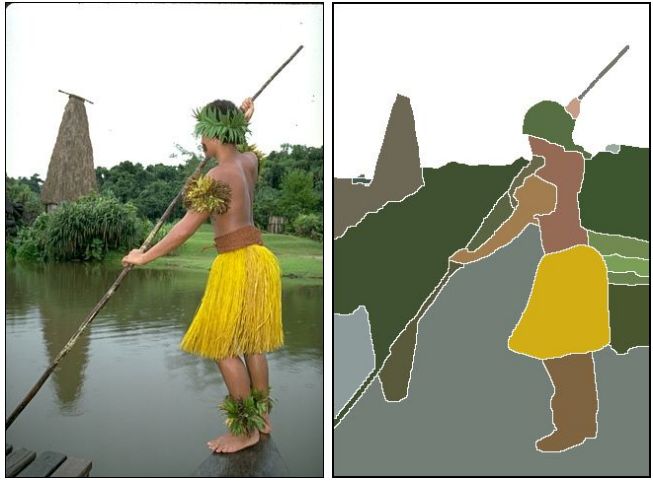


Image source: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html

# CV problems: Semantic Segmentation

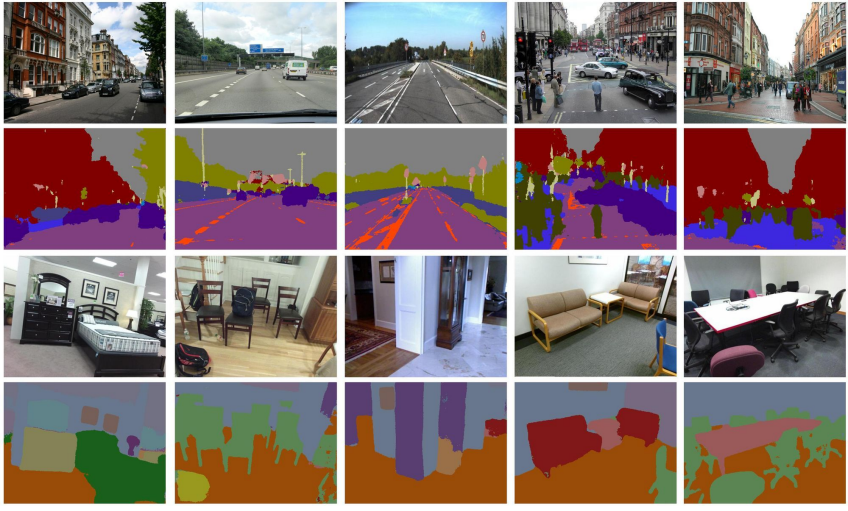
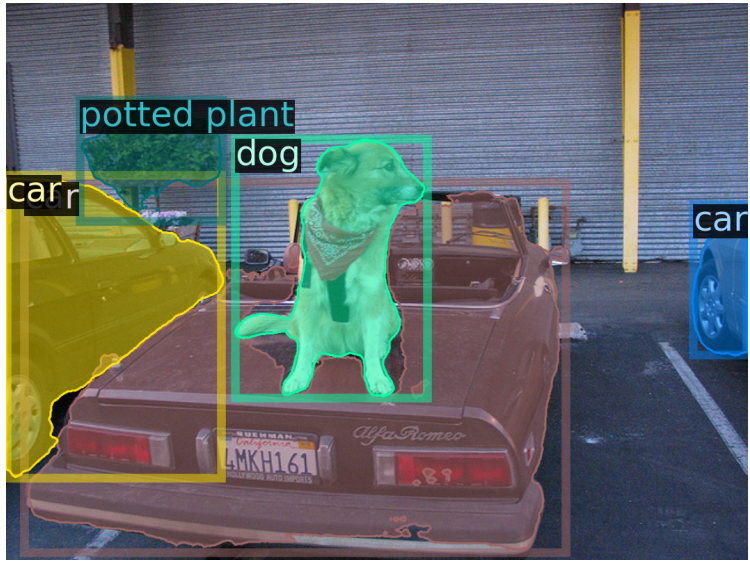
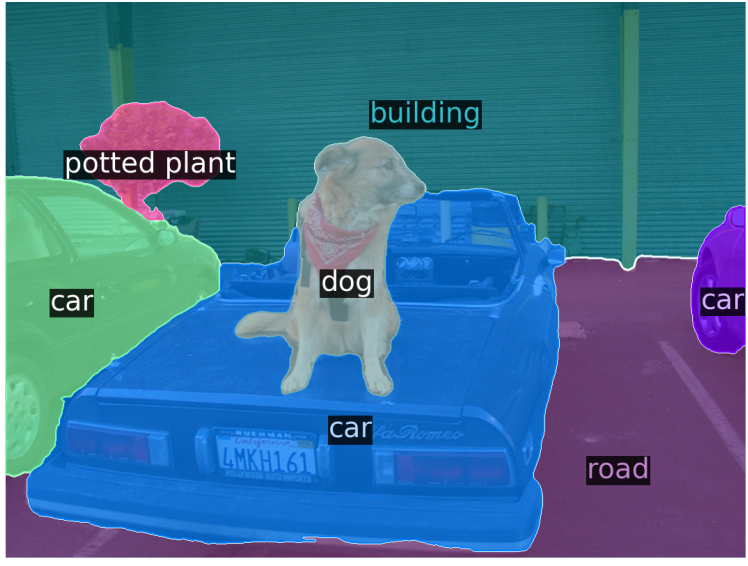


Image source: https://mi.eng.cam.ac.uk/projects/segnet/

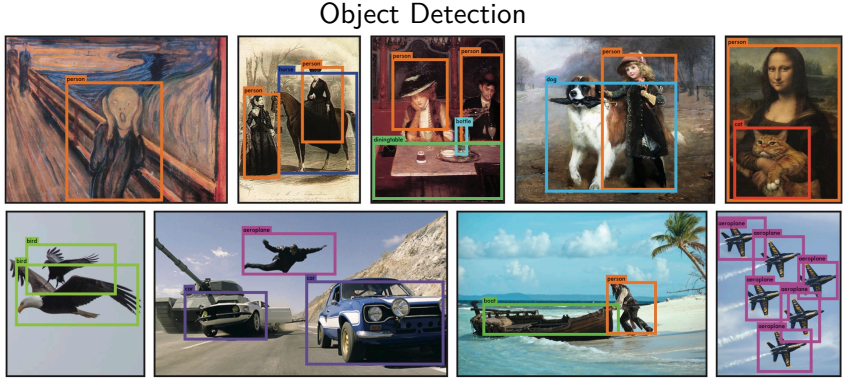
# CV problems: Instance Segmentation



# CV problems: Panoptic Segmentation



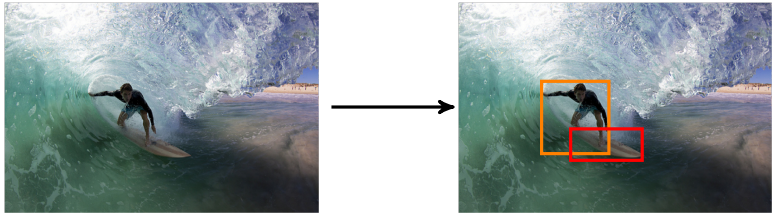
# Object Detection



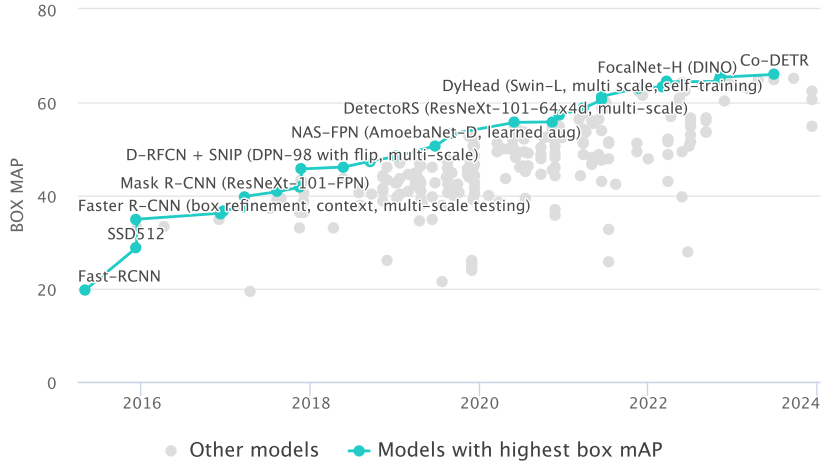
# Object Detection Task

- Given an input image, predict **a set** of bounding boxes with corresponding categories

set(("person", $\text{bbox}_1$), ("surfboard", $\text{bbox}_2$))

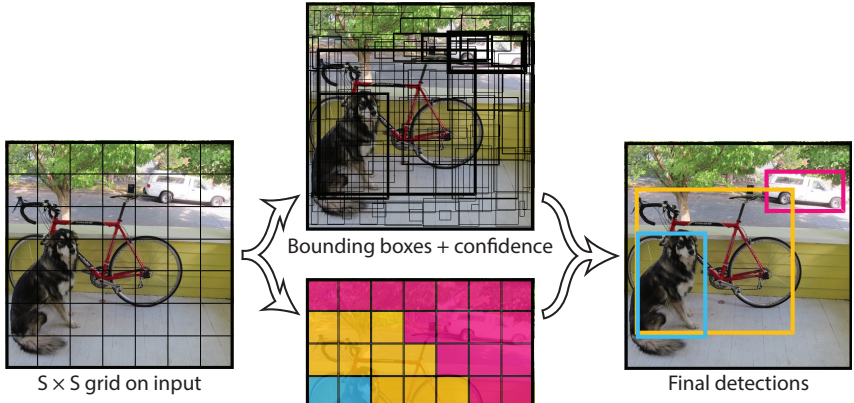


# Object Detection




https://paperswithcode.com/sota/object-detection-on-coco

# YOLO



# Introduction to Non-Maximum Suppression (NMS)

- NMS is a technique used in object detection to select one object out of many overlapping detections.
- It helps reduce redundancy by eliminating less likely bounding boxes.
- The process involves:

# Introduction to Non-Maximum Suppression (NMS)

- NMS is a technique used in object detection to select one object out of many overlapping detections.
- It helps reduce redundancy by eliminating less likely bounding boxes.
- The process involves:
  1. Sorting the detections by confidence scores.

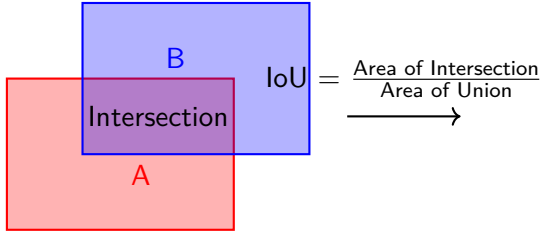
# Introduction to Non-Maximum Suppression (NMS)

- NMS is a technique used in object detection to select one object out of many overlapping detections.
- It helps reduce redundancy by eliminating less likely bounding boxes.
- The process involves:
  1. Sorting the detections by confidence scores.
  2. Selecting the highest scoring detection.

# Introduction to Non-Maximum Suppression (NMS)

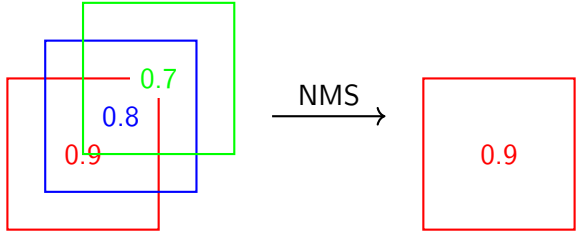
- NMS is a technique used in object detection to select one object out of many overlapping detections.
- It helps reduce redundancy by eliminating less likely bounding boxes.
- The process involves:
  1. Sorting the detections by confidence scores.
  2. Selecting the highest scoring detection.
  3. Suppressing all other detections that have a high overlap (IoU) with the selected one.

# Intersection over Union (IoU)




- IoU measures the overlap between two bounding boxes.
- It is used to determine how much two boxes overlap.

# Illustration of NMS




- The red box is selected as it has the highest score.
- The blue and green boxes are suppressed due to high overlap.

# Classical approach to detection

- Popular approach: detection := classification of boxes

# Classical approach to detection

- Popular approach: detection := classification of boxes
- Requires selecting a subset of candidate boxes

# Classical approach to detection

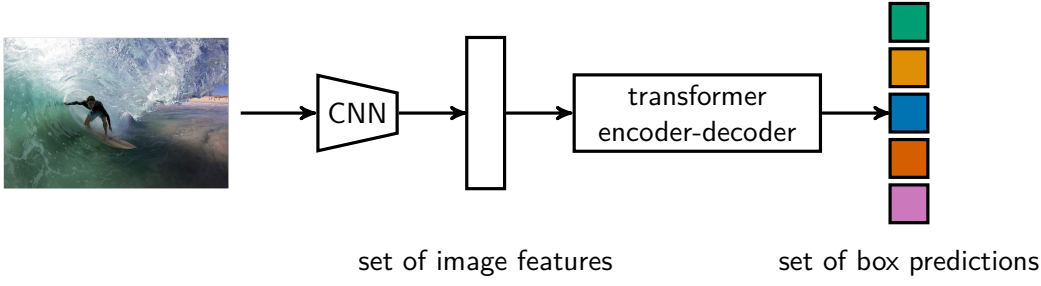
- Popular approach: detection := classification of boxes
- Requires selecting a subset of candidate boxes
- Regression step to refine the predictions

# Classical approach to detection

- Popular approach: detection := classification of boxes
- Requires selecting a subset of candidate boxes
- Regression step to refine the predictions
- Typically non-differentiable

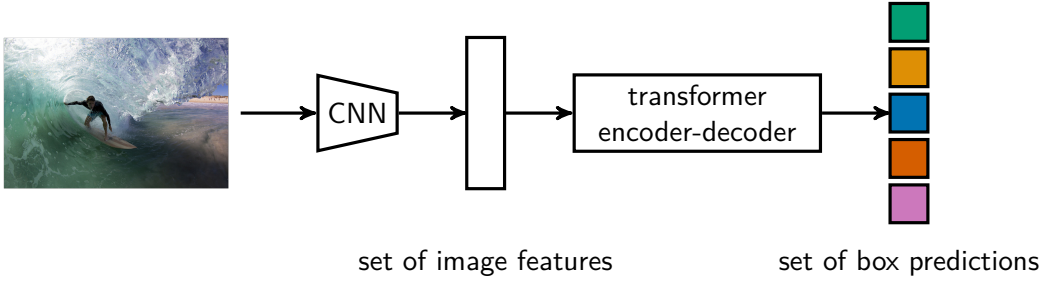
# DETR: Rethinking object detection

- *Neural machine translation* from features to boxes



# DETR: Rethinking object detection

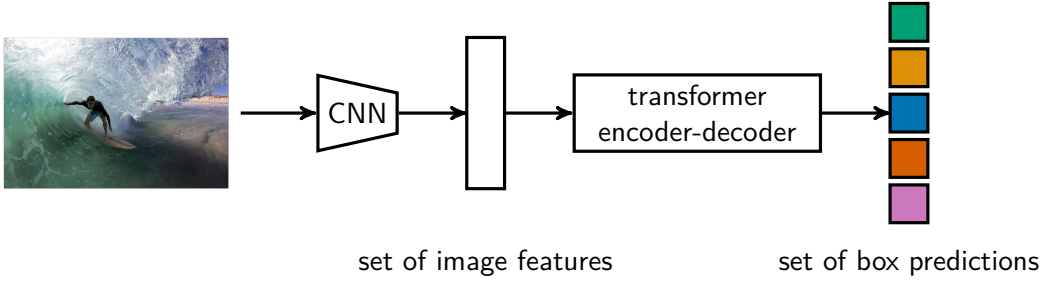
- *Neural machine translation* from features to boxes




- End-to-end parallel set prediction

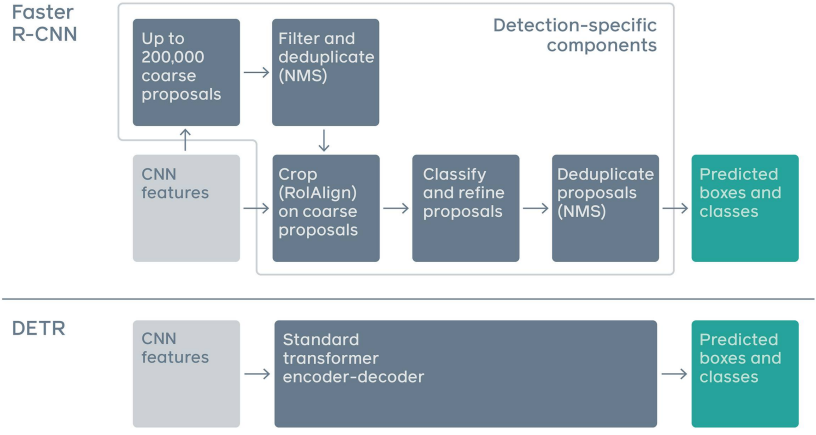
# DETR: Rethinking object detection

- *Neural machine translation* from features to boxes

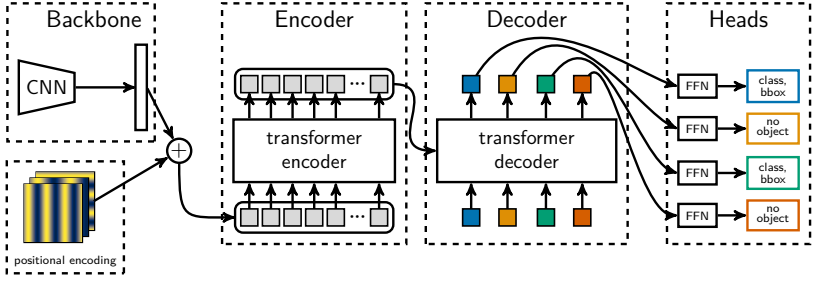



- End-to-end parallel set prediction
- Global scene reasoning

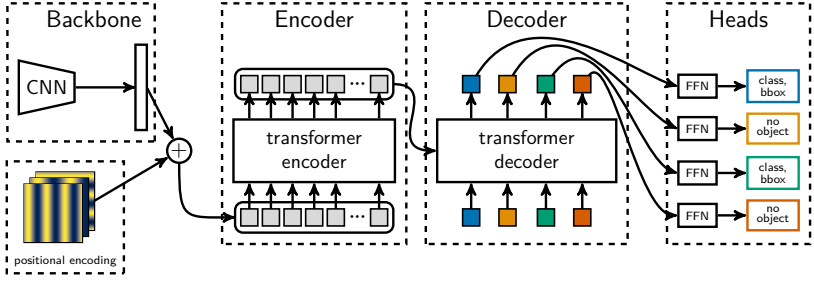
# Streamlined detection pipeline



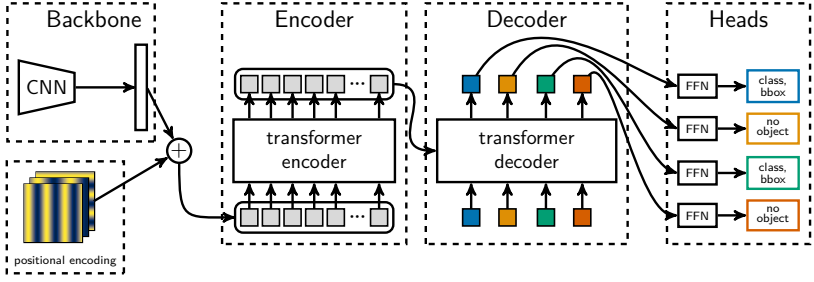
- We use standard ResNet from torchvision



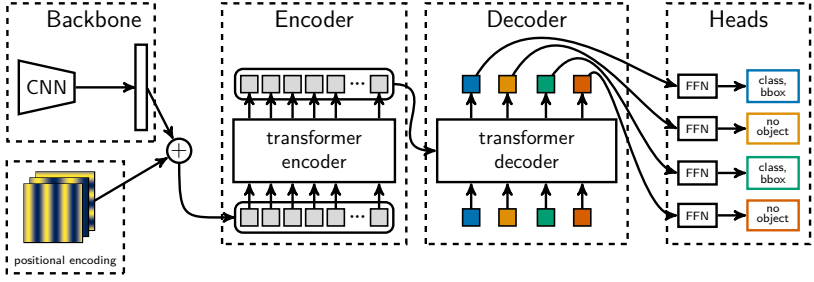
- We use standard ResNet from torchvision
- Pretraining on Imagenet is key (labels or SSL)



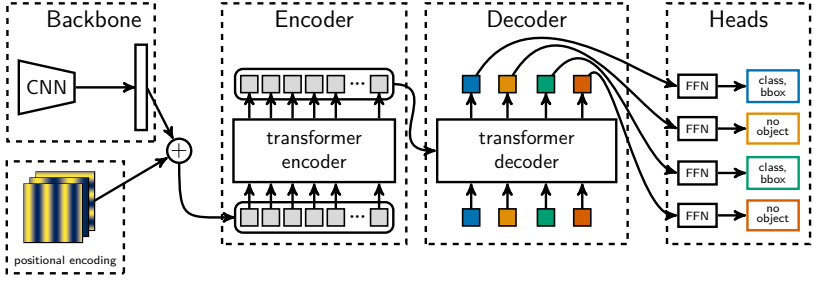
- We use 2D sine/cosine embeddings



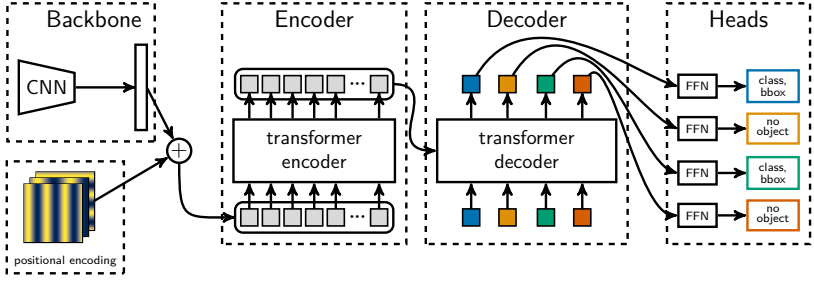
- We use 2D sine/cosine embeddings
- Embeddings are added to features



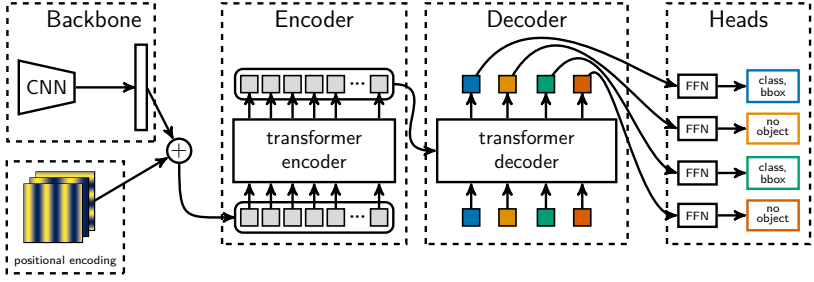
- We use 6 layers of transformer encoder



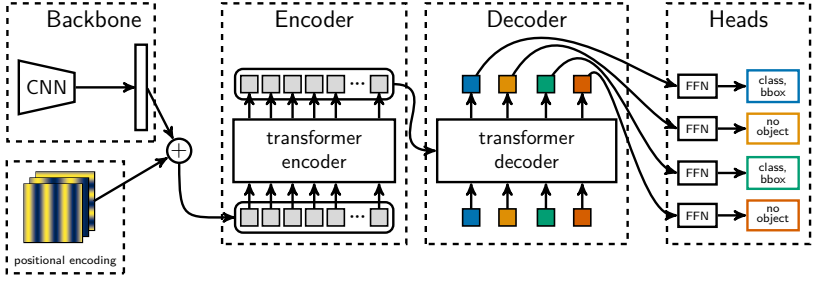
- We use 6 layers of transformer encoder
- Global reasoning through attention



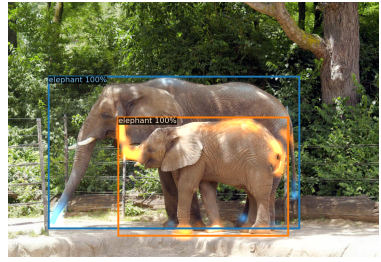
- We use 6 layers of transformer encoder
- Global reasoning through attention
- Starts separating instances

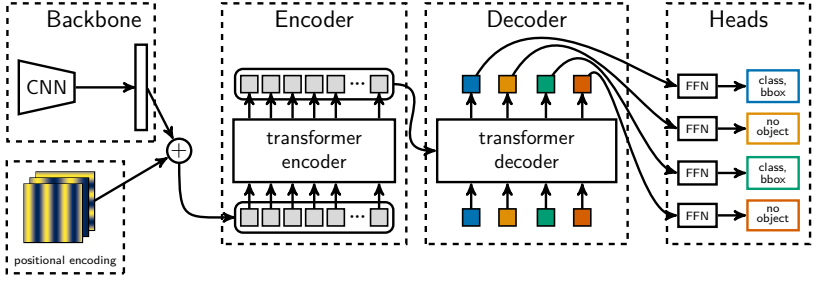



- We use 6 layers of transformer decoder;

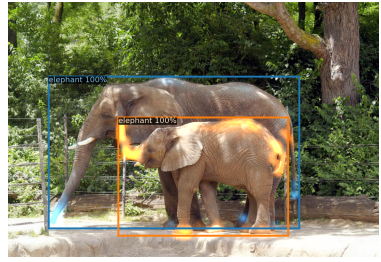


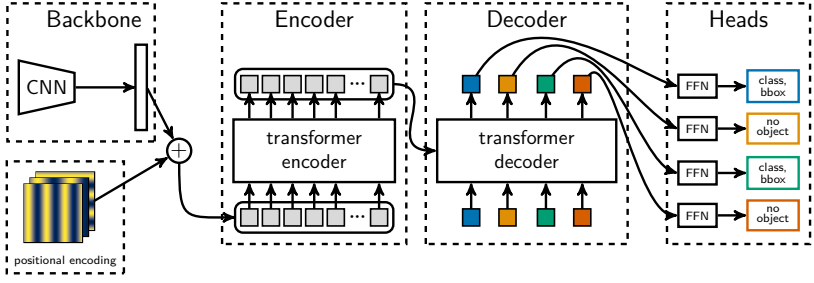
- We use 6 layers of transformer decoder;
- Attention focuses on extremities



- We use 6 layers of transformer decoder;
- Attention focuses on extremities
- Predictions are refined at each layer **in parallel**



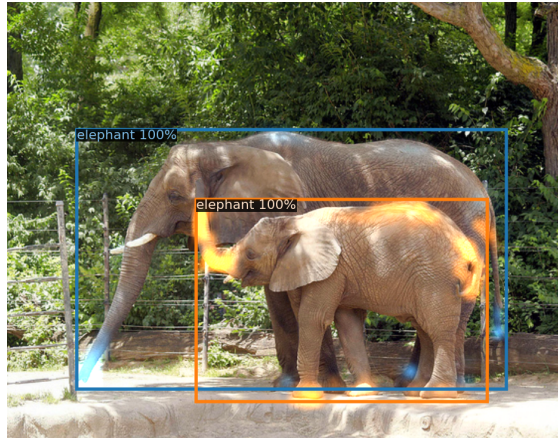
# Decoder attention weights

6 decoder layers of:

- self-attention
- enc-dec attention
- FFN
- LayerNorm

All outputs are decoded *in parallel*

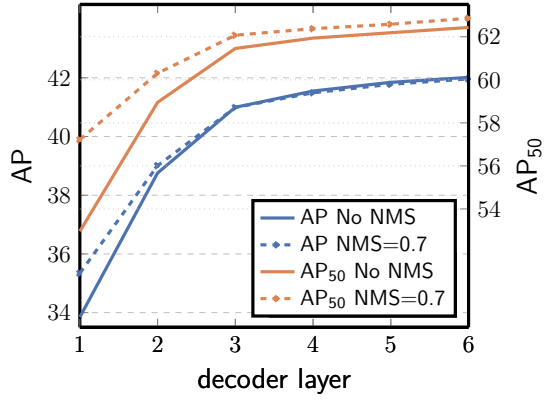
Attention focuses on extremities



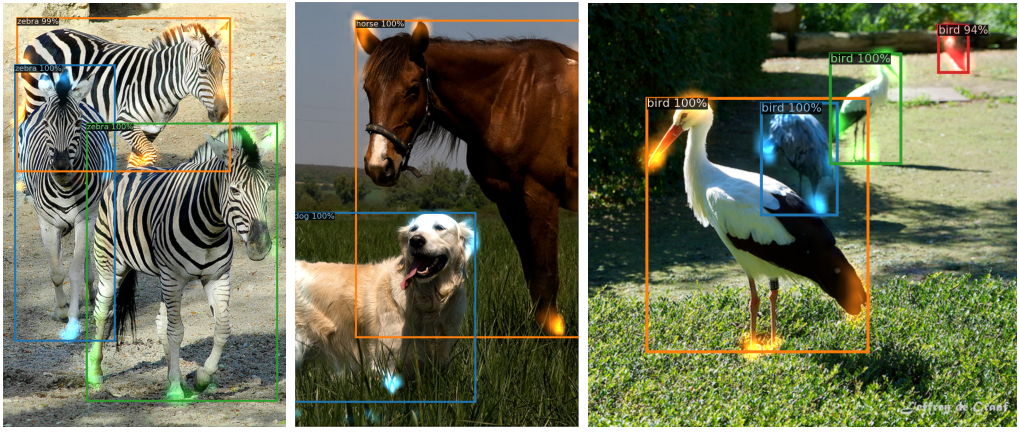
# Decoder: no NMS needed

- AP/$AP_{50}$ goes up in lower layers (no communication)
- AP goes down in the last layers
- $AP_{50}$ goes up slightly

**There is no need for NMS in DETR**

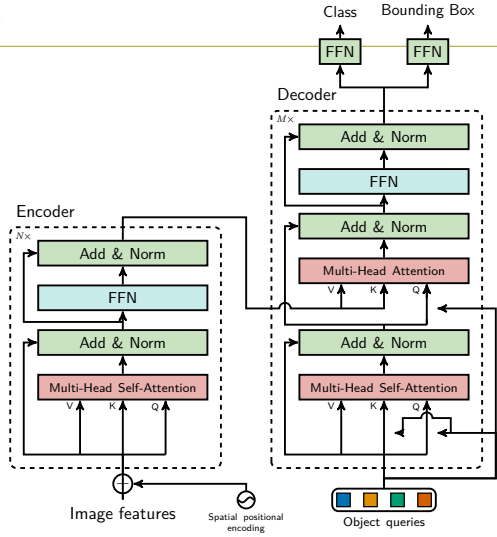


# Decoder attention weights



# Transformer architecture variations

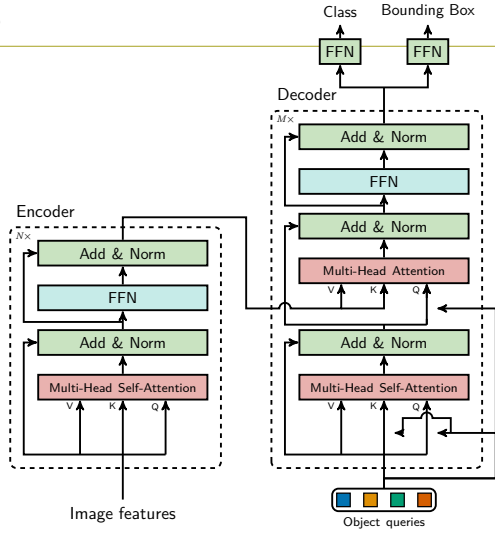
- Base transformer
  39.2 AP[a]




[a] Ashish Vaswani et al. (2017). "Attention is All you Need". In: *NeurIPS*.

# Transformer architecture variations

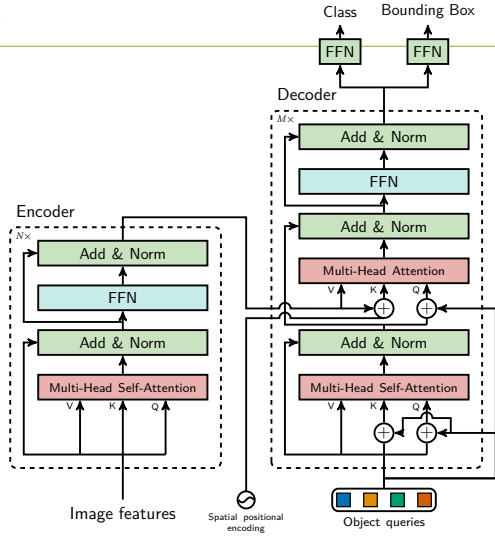
- Base transformer
  39.2 AP[a]
- No input positional encoding
  32.8 AP




[a] Ashish Vaswani et al. (2017). "Attention is All you Need". In: *NeurIPS*.

# Transformer architecture variations

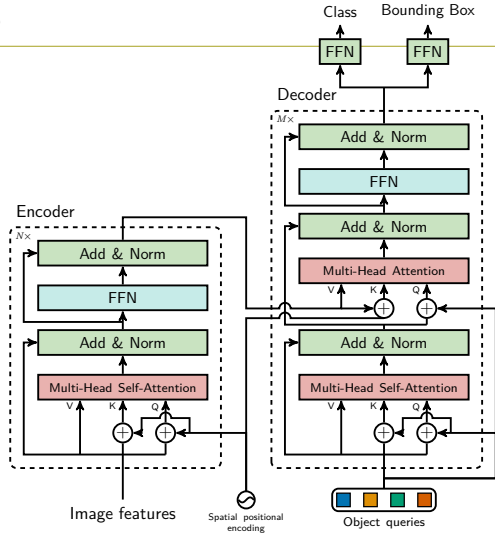
- Base transformer
  39.2 AP[a]
- No input positional encoding
  32.8 AP
- Encodings in decoder attentions only
  39.3 AP




[a] Ashish Vaswani et al. (2017). "Attention is All you Need". In: *NeurIPS*.

# Transformer architecture variations

- Base transformer
  39.2 AP[a]
- No input positional encoding
  32.8 AP
- Encodings in decoder attentions only
  39.3 AP
- Encodings in all attentions
  **40.6 AP**

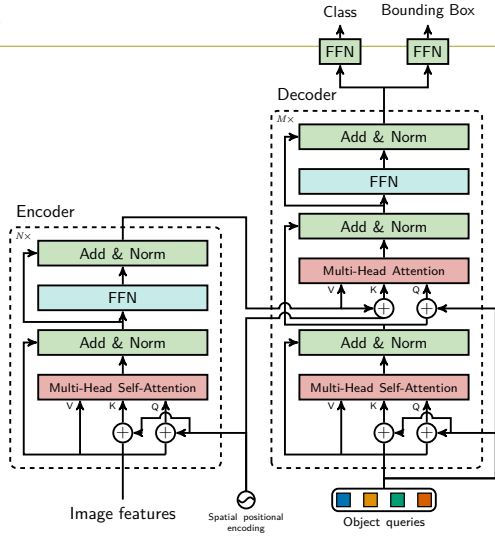



[a] Ashish Vaswani et al. (2017). "Attention is All you Need". In: *NeurIPS*.

# Transformer architecture variations

- Base transformer
  39.2 AP[a]
- No input positional encoding
  32.8 AP
- Encodings in decoder attentions only
  39.3 AP
- Encodings in all attentions
  **40.6 AP**

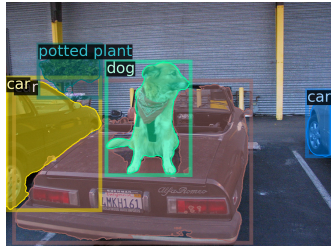
**All transformer parts are contributing!**




[a] Ashish Vaswani et al. (2017). "Attention is All you Need". In: *NeurIPS*.

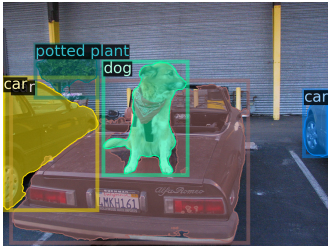
# Panoptic segmentation

Instance segmentation



# Panoptic segmentation

Instance segmentation

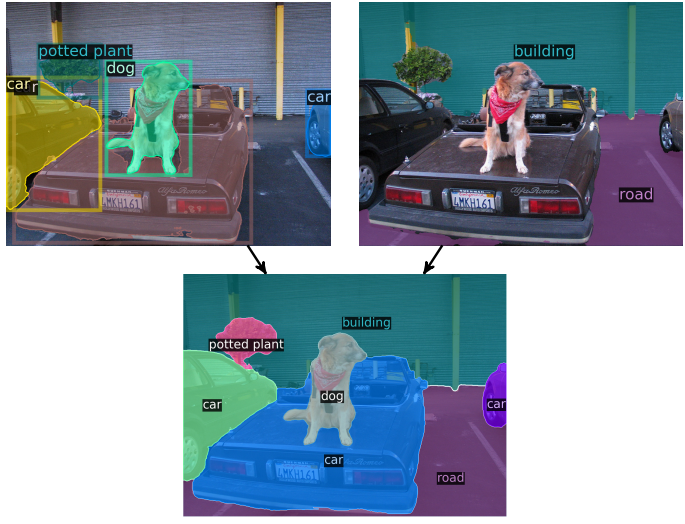



Semantic segmentation

# Panoptic segmentation

Instance segmentation

Semantic segmentation

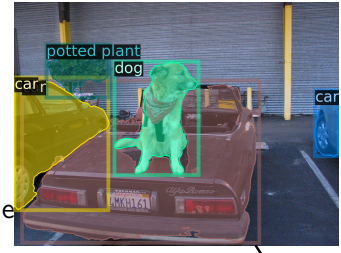



Panoptic segmentation

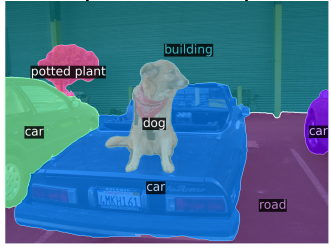
# Panoptic segmentation

- Each pixel belongs to exactly one segment

Instance segmentation




Semantic segmentation

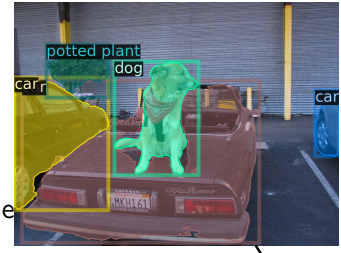




Panoptic segmentation

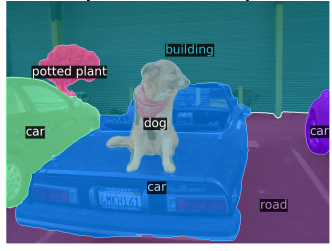
# Panoptic segmentation

- Each pixel belongs to exactly one segment
- Useful for self-driving

Instance segmentation




Semantic segmentation

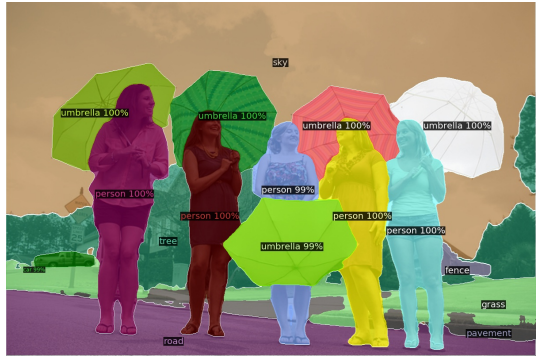




Panoptic segmentation

# From attention to mask

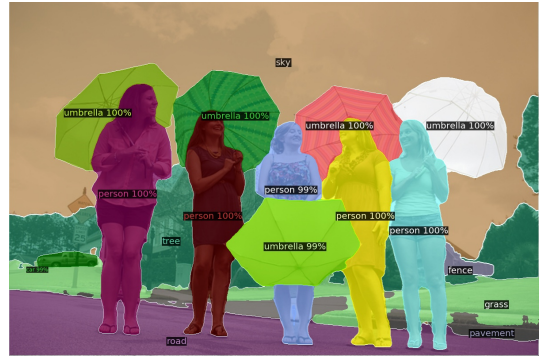
# Object-centric detection paradigm

- Object instances are first class citizens in DETR



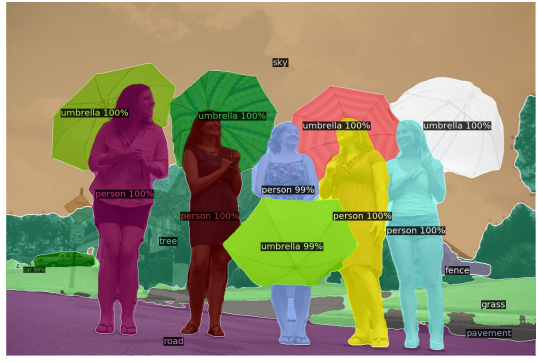
# Object-centric detection paradigm

- Object instances are first class citizens in DETR
- Application: panoptic segmentation

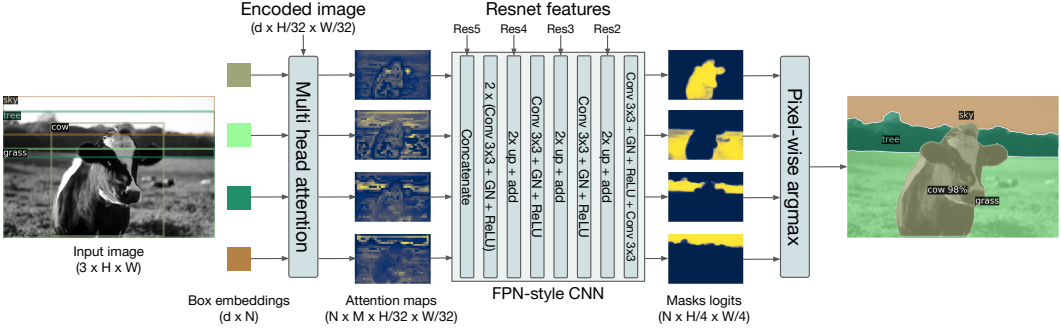


# Object-centric detection paradigm

- Object instances are first class citizens in DETR
- Application: panoptic segmentation
- DETR improves over state-of-the-art results



# Attention-based mask head



# SOLQ

- How to predict masks more efficiently?

SOLQ (Dong et al. 2021)

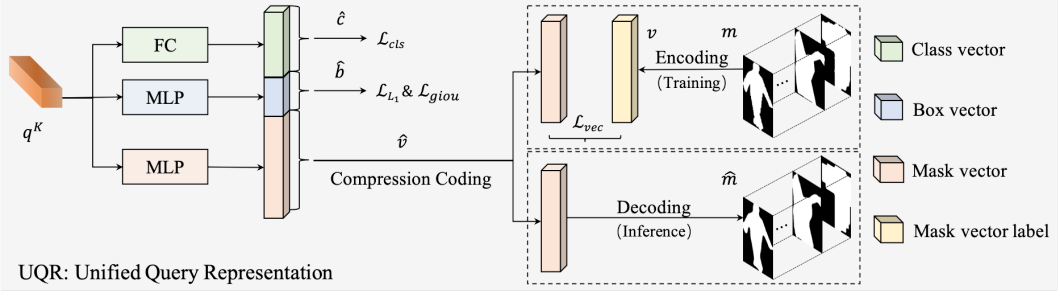
# SOLQ

- How to predict masks more efficiently?
- Encode masks into lower dimension and predict similar to bounding boxes (Dong et al. 2021)

SOLQ (Dong et al. 2021)

# SOLQ

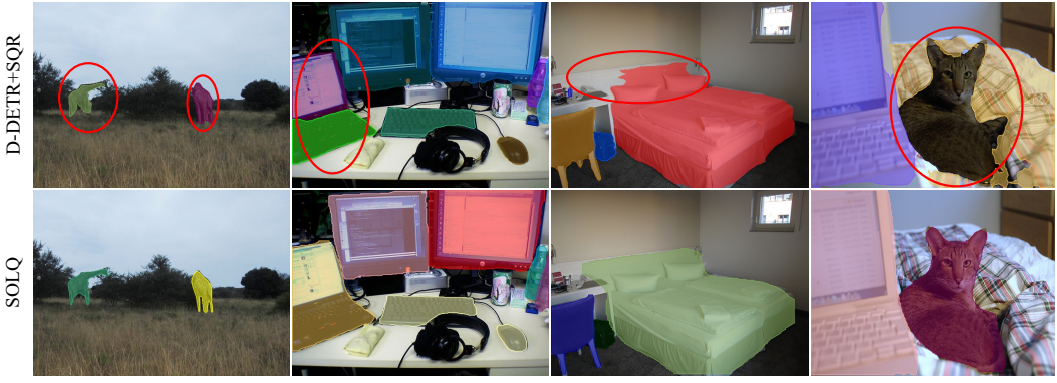
- How to predict masks more efficiently?
- Encode masks into lower dimension and predict similar to bounding boxes (Dong et al. 2021)




**SOLQ**

SOLQ (Dong et al. 2021)

# SOLQ

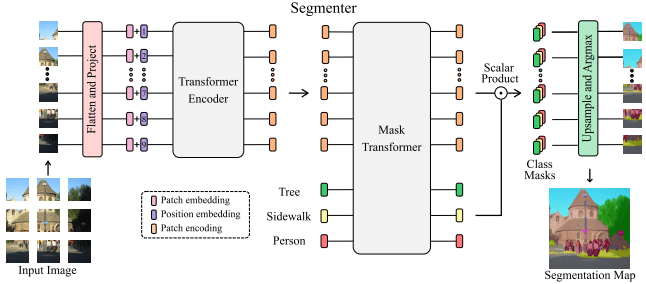



SOLQ (Dong et al. 2021)

# Table of Contents

# Segmenter Overview

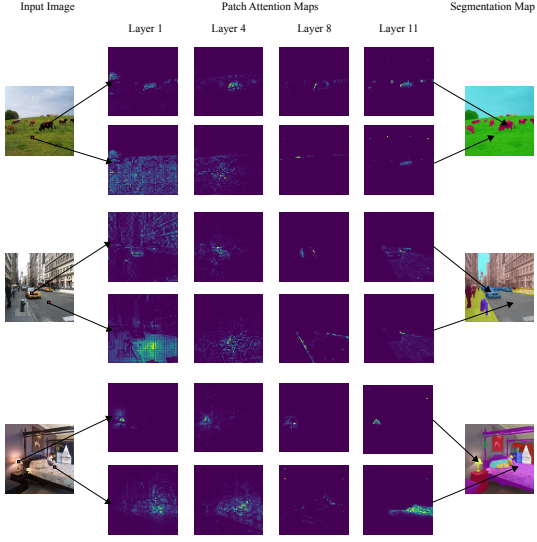
**Segmenter: Transformer-based Semantic Segmentation** (Strudel et al. 2021)




- **Patch-based encoding**: Divides image into patches and processes them with Vision Transformer (ViT)
- **Transformer decoder**: Uses learnable class embeddings to generate segmentation masks
- **End-to-end training**: Trained directly for semantic segmentation without CNN backbone

# Segmenter Attention Maps

## Visualizing Attention in Segmenter



# Table of Contents

# Segment Anything

Segment Anything (Kirillov et al. 2023)

**Huggingface demo**

## Segmentation Problem in Computer Vision

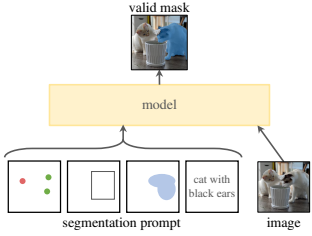
**What is Image Segmentation?**

- Image segmentation is the process of partitioning an image into multiple segments or regions.
- The goal is to simplify or change the representation of an image into something more meaningful and easier to analyze.

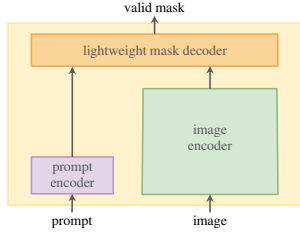
**Purpose of Segment Anything:**

- To create a model capable of *segmenting any object in an image without needing task-specific training*.
- It aims to generalize across diverse object types and scenes.

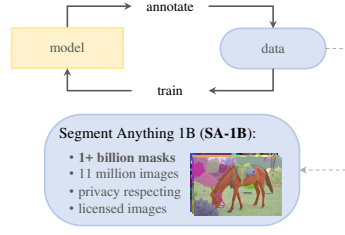
# SAM components




(a) **Task**: promptable segmentation




(b) **Model**: Segment Anything Model (**SAM**)




(c) **Data**: data engine (top) & dataset (bottom)

# SAM components