Dauphine | PSL
UNIVERSITÉ PARIS

Deep Learning for Image Analysis

Mehyar MLAWEH

🏫 Master BDIA - Dauphine Tunis
PhD student - Université PSL

Resources: https://mehyarmlaweh.github.io/mehyar-mlaweh/teaching

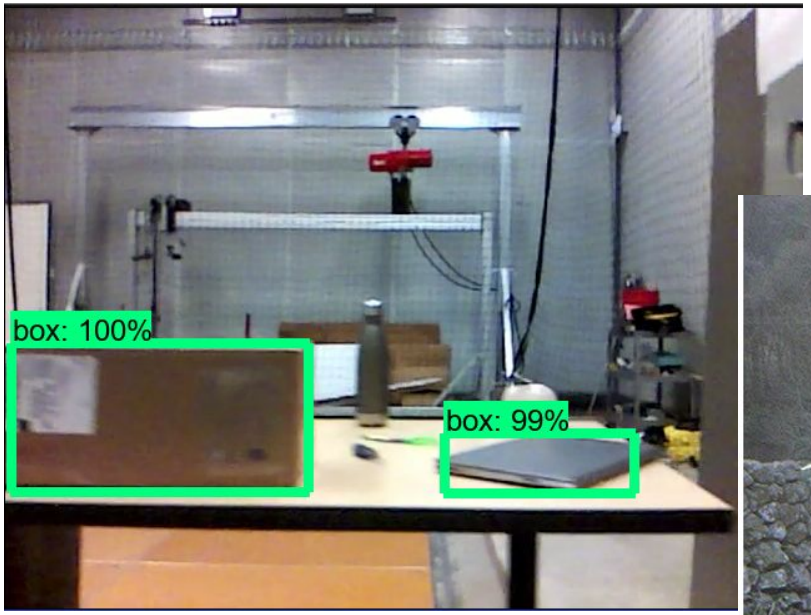
Module Overview

1. Computer Vision Foundations
2. CNNs
3. Transfer Learning
4. Vision Transformers
5. Object Detection
6. Segment Anything Model (SAM)
7. Final Project

Lecture 6 :

Segment Anything

Last Time



Original image

Introduction

Original image

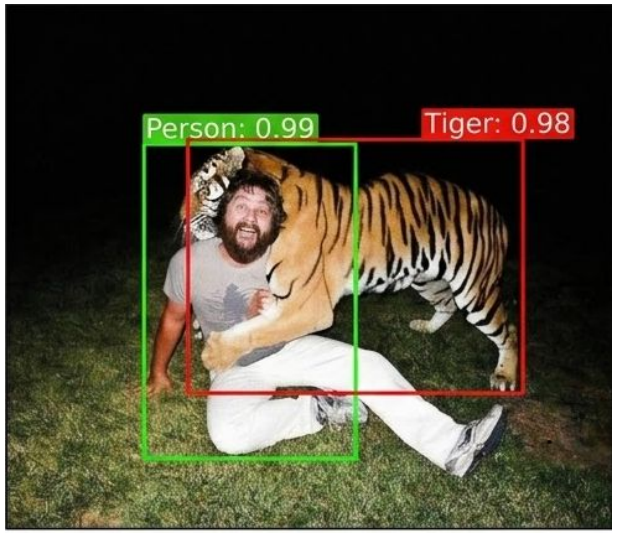
Classification

A person- 81%

No location. No boundary

Original image

Object Detection

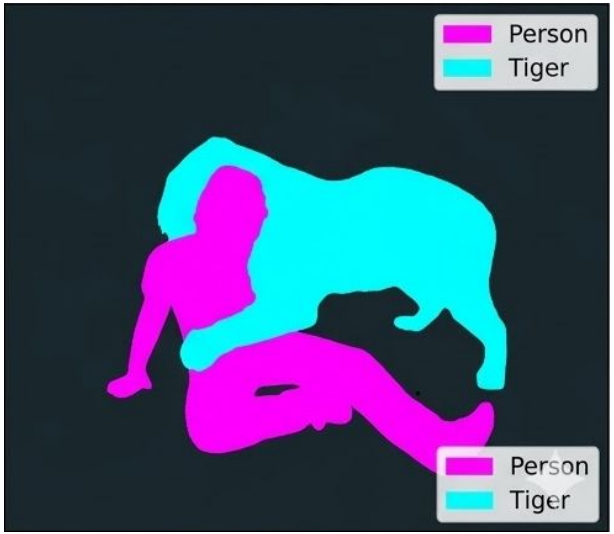


Bounding boxes around the tiger and then person
Approximate location

Introduction

Original image

Segmentation



Precise pixel masks

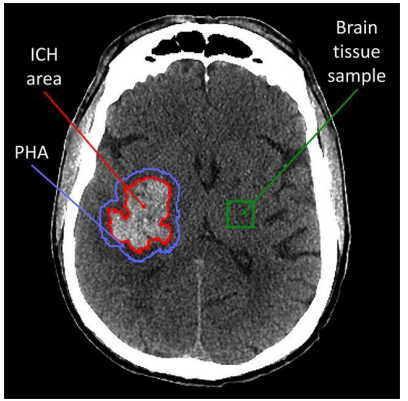
Segmentation ?

Assigning a label to every single pixel in the image

Segmentation ?

Assigning a label to every single pixel in the image

Not where roughly, where exactly



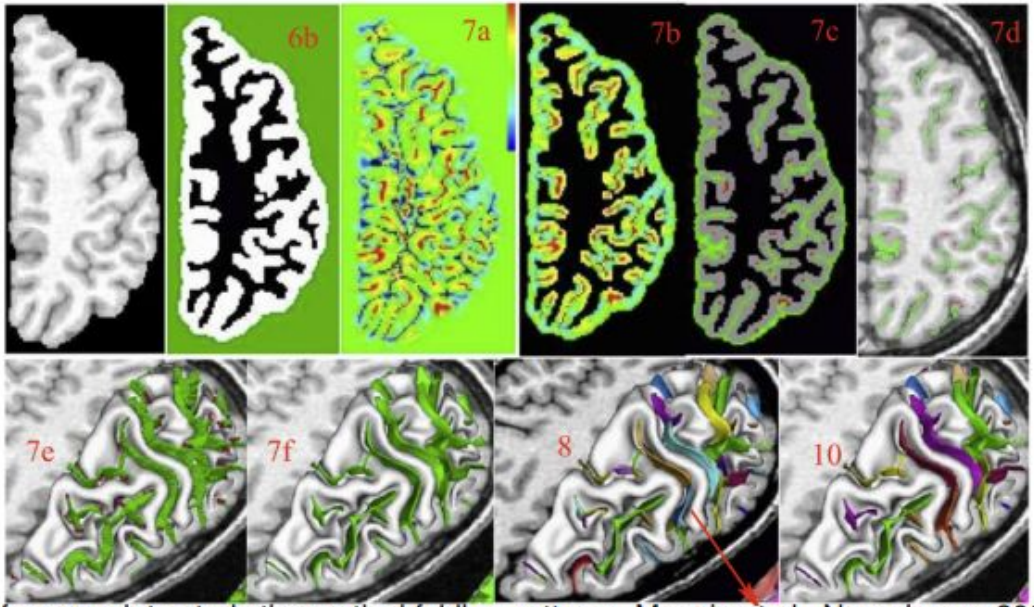
Segmentation ?

Some examples

Face Segmentation Using Skin-Color Map in Videophone Applications, Douglas Chai, and King N. Ngan, 1999

Segmentation ?

Some examples

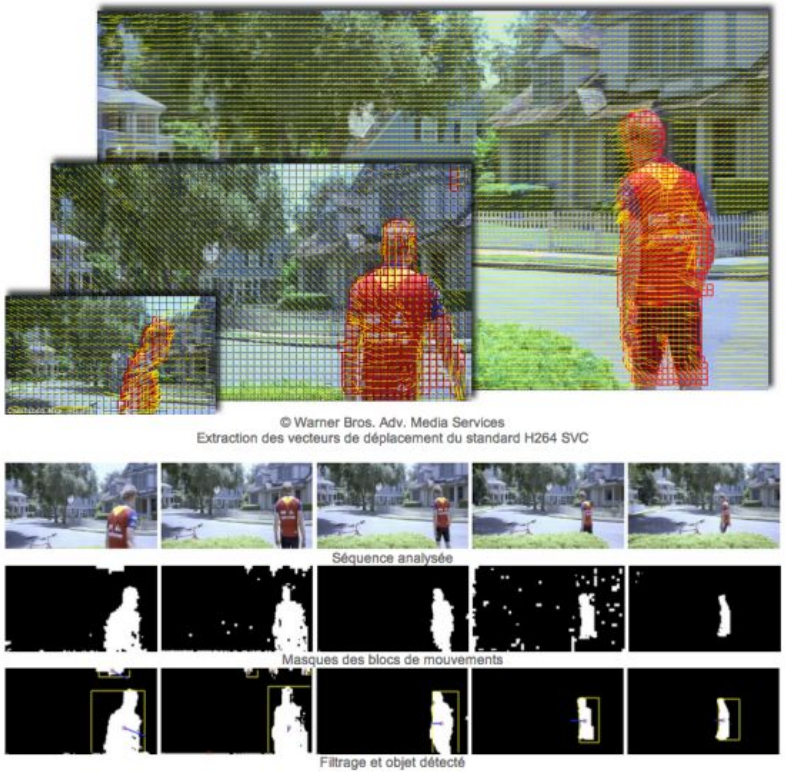


A framework to study the cortical folding patterns, Mangin et al., NeuroImage 2004

- Quantification of tissue and organ volumes
- Localization of a pathology
- Treatment planning
- ..

Segmentation ?

Some examples



- Sports analysis
- Autonomous driving
- ..

Three Flavors of Segmentation

Image

Semantic Segmentation

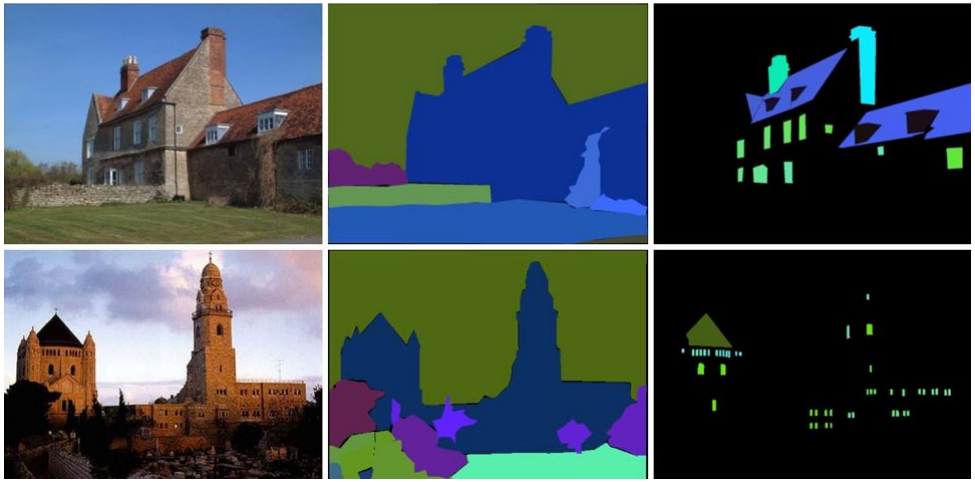
Instance Segmentation

Panoptic Segmentation

Semantic Segmentation dataset: ADE20k- 20K

- 20K scene-centric images, annotated with pixel-level objects and object parts labels.

- 150 semantic categories: sky, road, grass, and discrete objects like person, car, bed



[Source:https://www.kaggle.com/datasets/awsaf49/ade20k-dataset]

Instance & Panoptic Segmentation dataset: MS COCO

Dauphine | PSL UNIVERSITÉ PARIS

328K images

2.5M instances

80 object category

Microsoft COCO: Common Objects in Context

Tsung-Yi Lin Michael Maire Serge Belongie Lubomir Bourdev Ross Girshick
James Hays Pietro Perona Deva Ramanan C. Lawrence Zitnick Piotr Dollár

Abstract—We present a new dataset with the goal of advancing the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. This is achieved by gathering images of complex everyday scenes containing common objects in their natural context. Objects are labeled using per-instance segmentations to aid in precise object localization. Our dataset contains photos of 91 objects types that would be easily recognizable by a 4 year old. With a total of 2.5 million labeled instances in 328k images, the creation of our dataset drew upon extensive crowd worker involvement via novel user interfaces for category detection, instance spotting and instance segmentation. We present a detailed statistical analysis of the dataset in comparison to PASCAL, ImageNet, and SUN. Finally, we provide baseline performance analysis for bounding box and segmentation detection results using a Deformable Parts Model.

1 Introduction

One of the primary goals of computer vision is the understanding of visual scenes. Scene understanding involves numerous tasks including recognizing what objects are present, localizing the objects in 2D and 3D, determining the objects' and scene's attributes, characterizing relationships between objects and providing a semantic description of the scene. The current object classification and detection datasets [1], [2], [3], [4] help us explore the first challenges related to scene understanding. For instance the ImageNet dataset [1], which contains an unprecedented number of images, has recently enabled breakthroughs in both object classification and detection research [5], [6], [7]. The community has also created datasets containing object attributes [8], scene attributes [9], keypoints [10], and 3D scene information [11]. This leads us to the obvious question: what datasets will best continue our advance towards our ultimate goal of scene understanding?

We introduce a new large-scale dataset that addresses

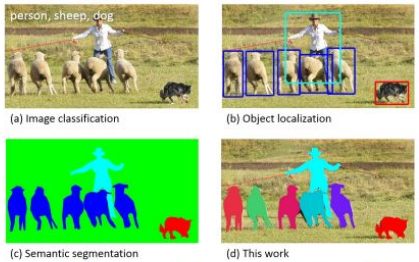
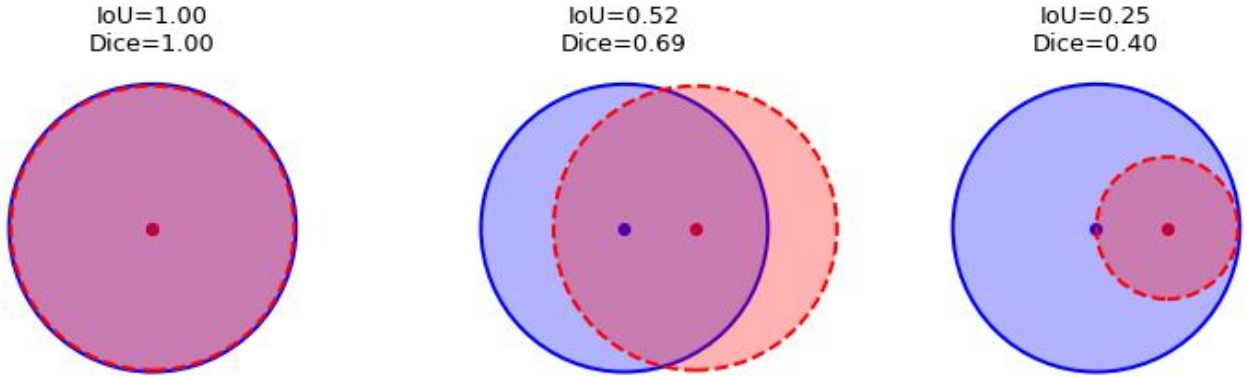


Fig. 1: While previous object recognition datasets have focused on (a) image classification, (b) object bounding box localization or (c) semantic pixel-level segmentation, we focus on (d) segmenting individual object instances. We introduce a large, richly-annotated dataset comprised of images depicting complex everyday scenes of common objects in their natural context.

[Source:https://arxiv.org/abs/1405.0312]

How Do We Measure Success

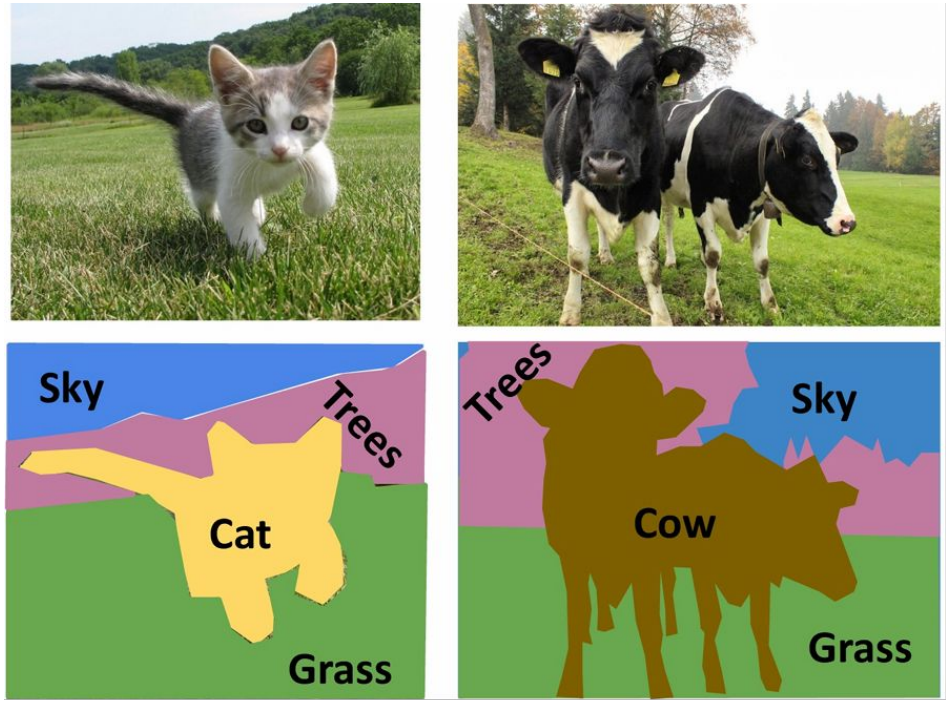


[Source:https://ilmonteux.github.io/2019/05/10/segmentation-metrics.html]

$$Dice(A,B) = \frac{2\|A \cap B\|}{\|A\| + \|B\|}, \qquad Jaccard(A,B) = \frac{\|A \cap B\|}{\|A \cup B\|}$$

Semantic Segmentation

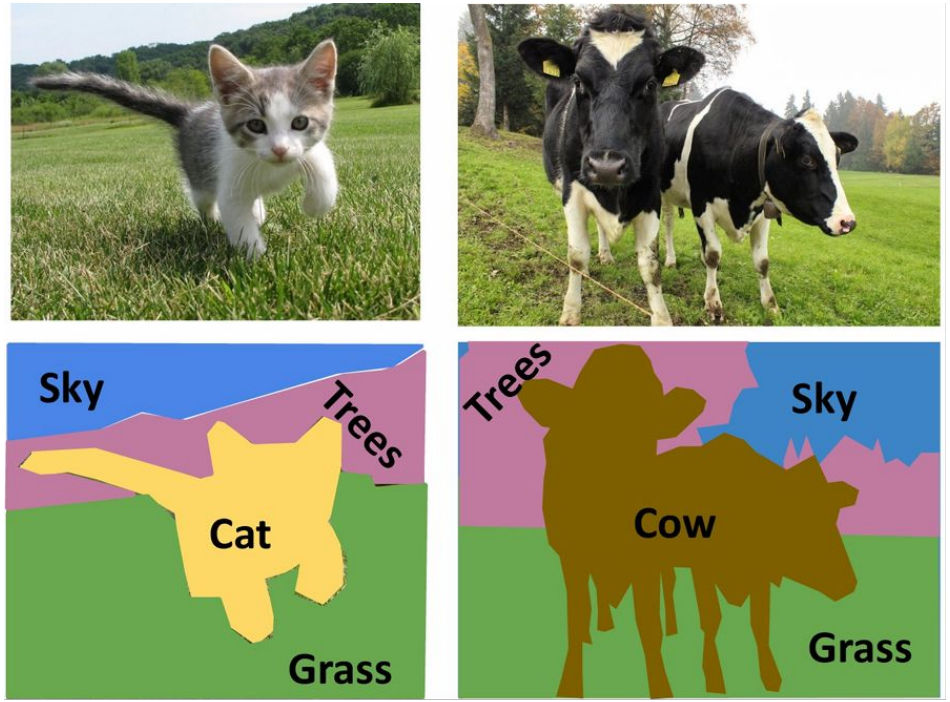
Label each pixel in the image with a category label



Semantic Segmentation

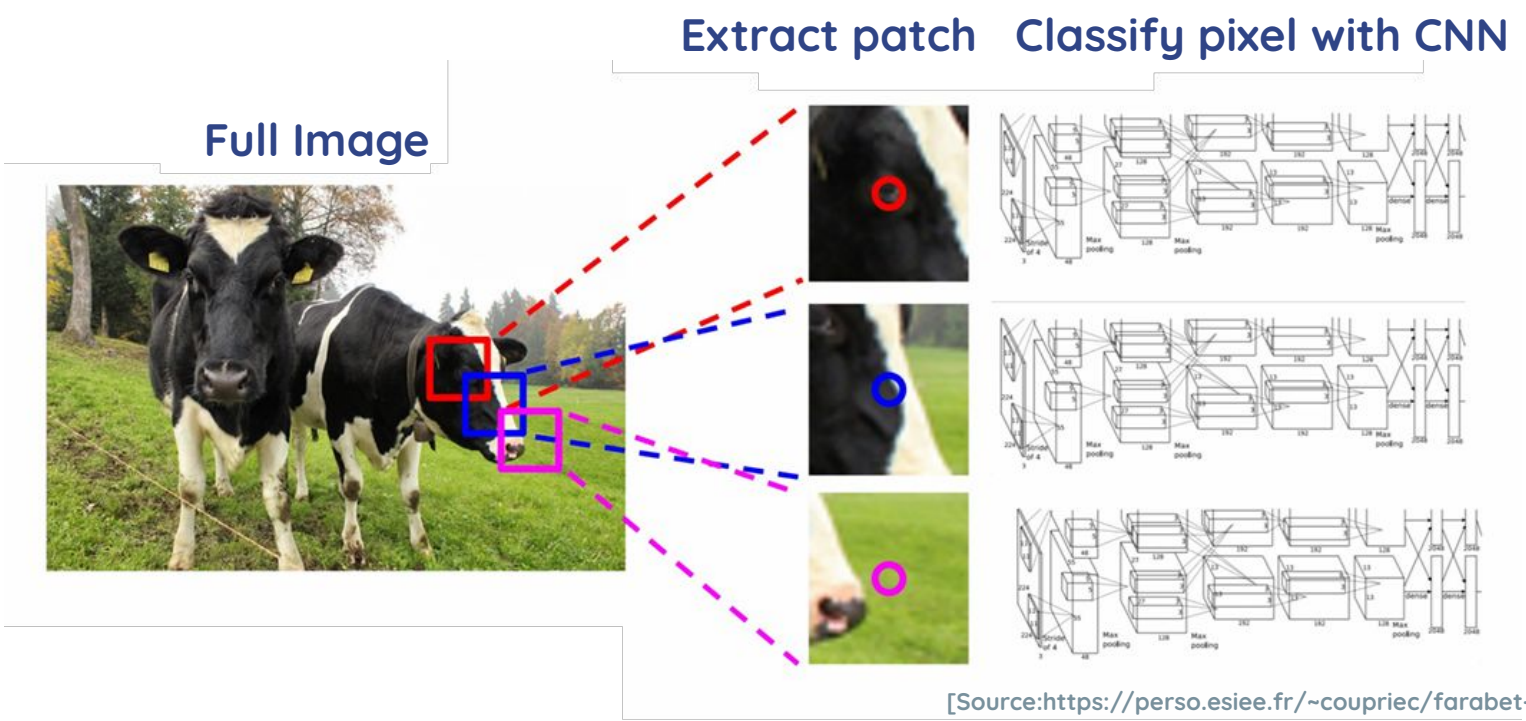
Label each pixel in the image with a category label

Don’t differentiate instances, only care about pixels



Semantic Segmentation w/ Sliding Window

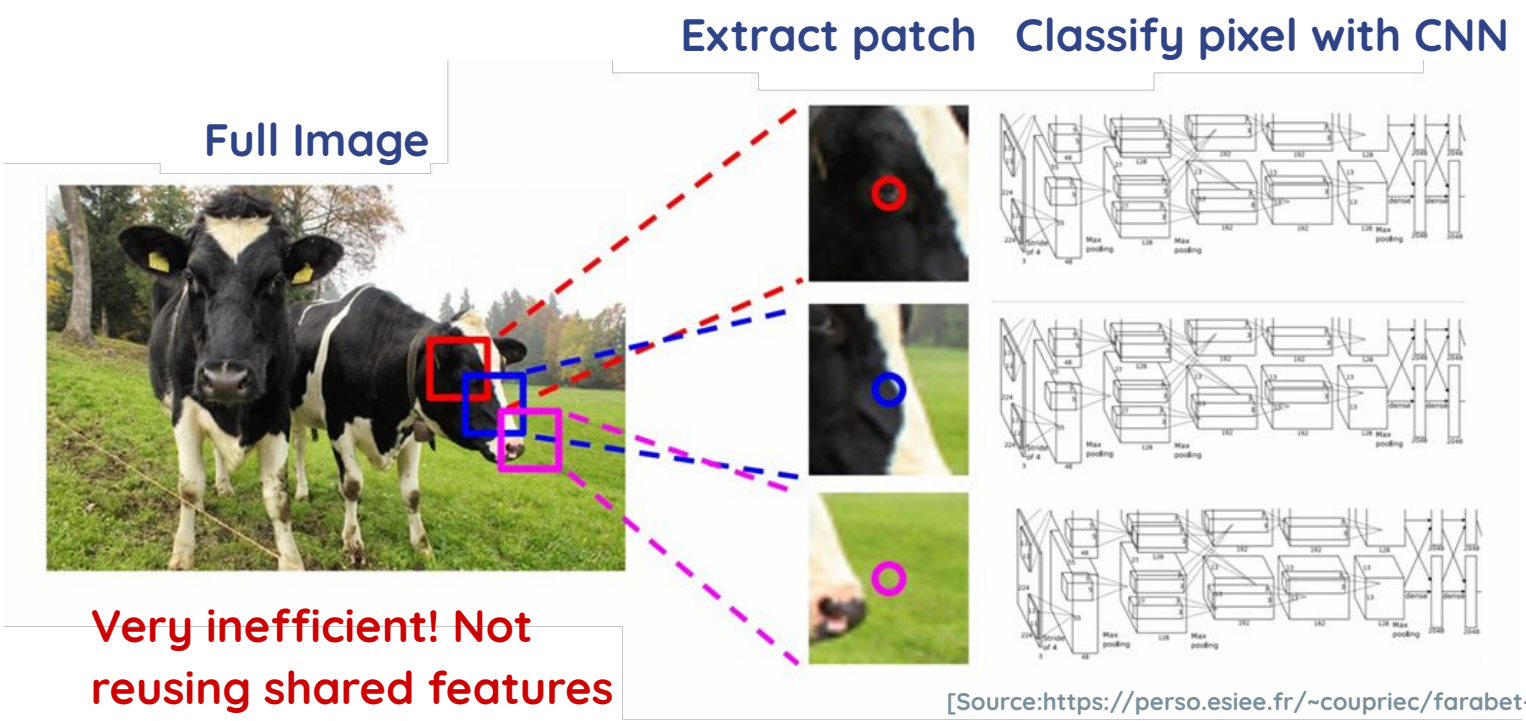
Dauphine | PSL UNIVERSITÉ PARIS



[Source:https://perso.esiee.fr/~coupriec/farabet-pami-13.pdf]

Semantic Segmentation w/ Sliding Window

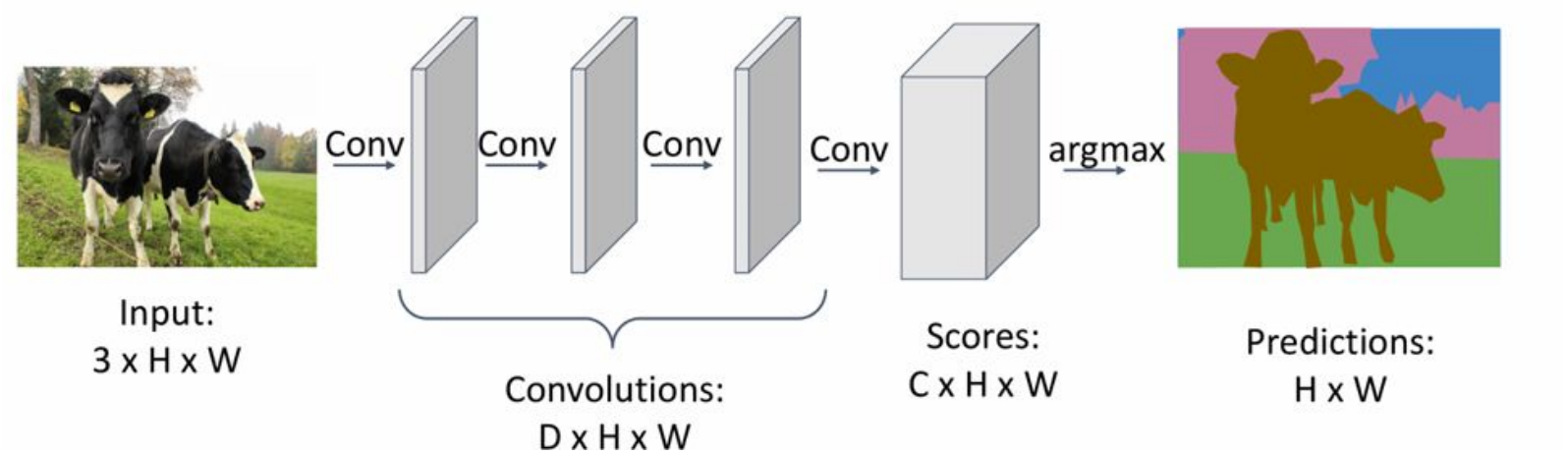
Dauphine | PSL UNIVERSITÉ PARIS



Very inefficient! Not reusing shared features between overlapping patches

[Source:https://perso.esiee.fr/~coupriec/farabet-pami-13.pdf]

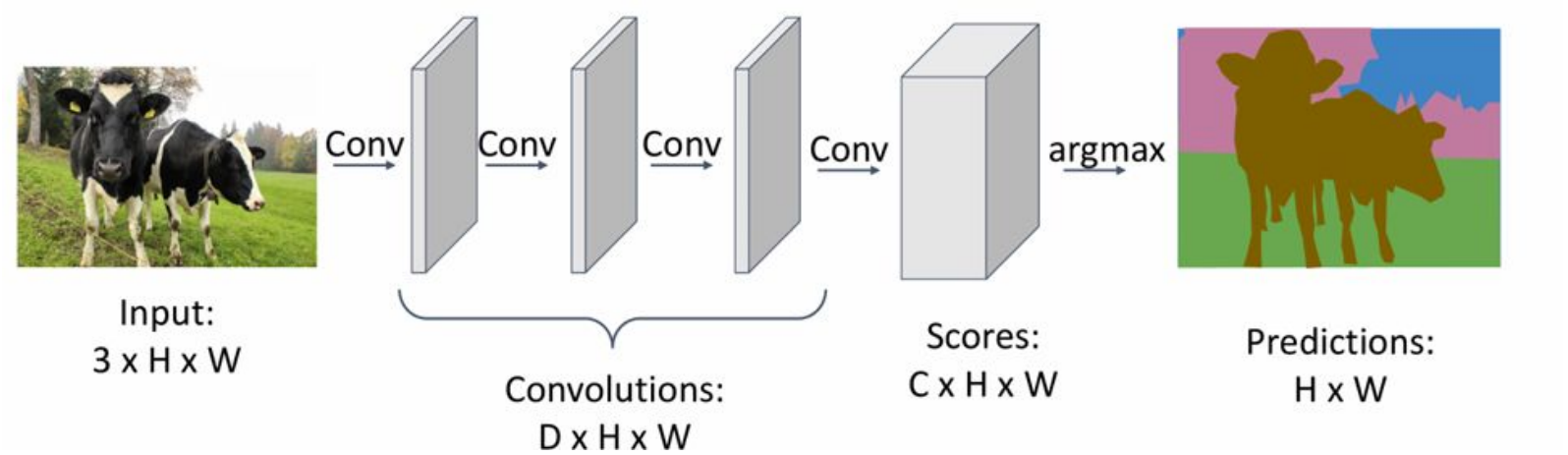
Design a network as a bunch of conv layers to make predictions for pixels at once



Loss Function : Per-Pixel cross-entropy

[Source: Long et al. 'Fully Convolutional Networks for Semantic Segmentation'
CVPR 2015]

Design a network as a bunch of conv layers to make predictions for pixels at once

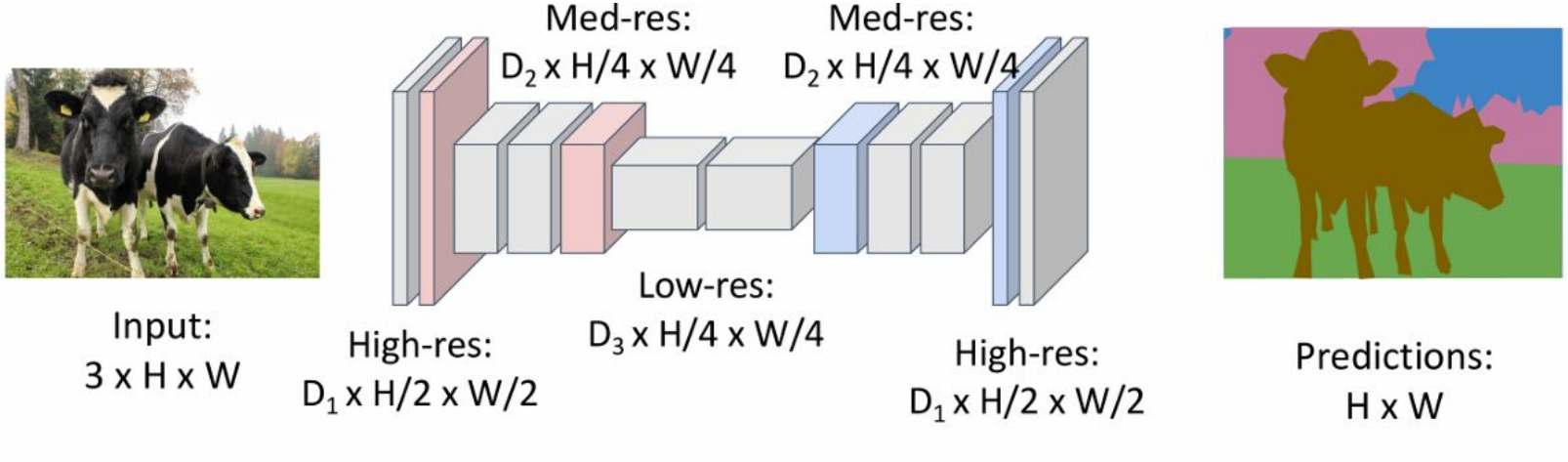


Problem: Convolution on high resolution images is expensive!

Loss Function : Per-Pixel cross-entropy

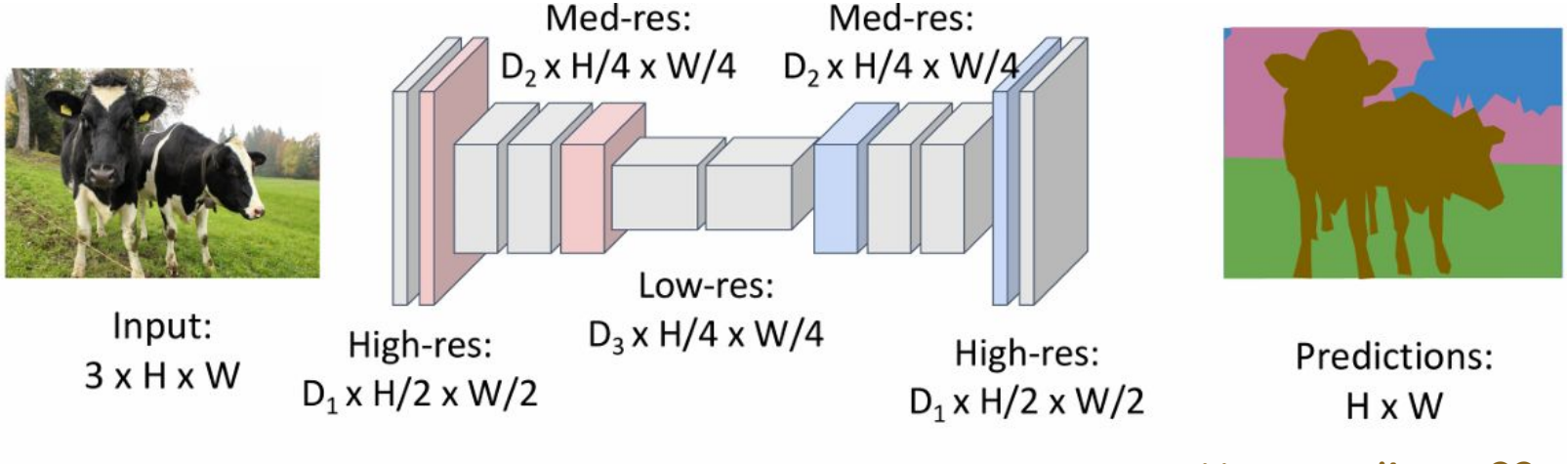
[Source: Long et al. 'Fully Convolutional Networks for Semantic Segmentation'
CVPR 2015]

Design a network as a bunch of conv layers, with downsampling and upsampling inside the network



[Sources:
- Long et al. 'Fully Convolutional Networks for Semantic Segmentation' CVPR 2015
- Noh et al. Learning Deconvolution Network for Semantic Segmentation, ICCV 2015]

Design a network as a bunch of conv layers, with downsampling and upsampling inside the network

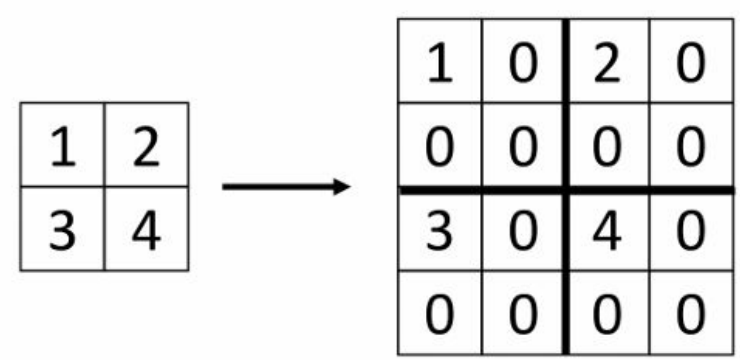


Upsampling: ??

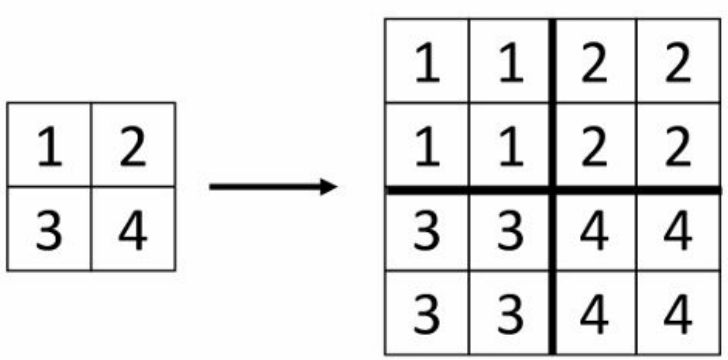
Downsampling: Pooling, strided convolution

[Sources:
- Long et al. 'Fully Convolutional Networks for Semantic Segmentation' CVPR 2015
- Noh et al. Learning Deconvolution Network for Semantic Segmentation, ICCV 2015]

Bed of Nails

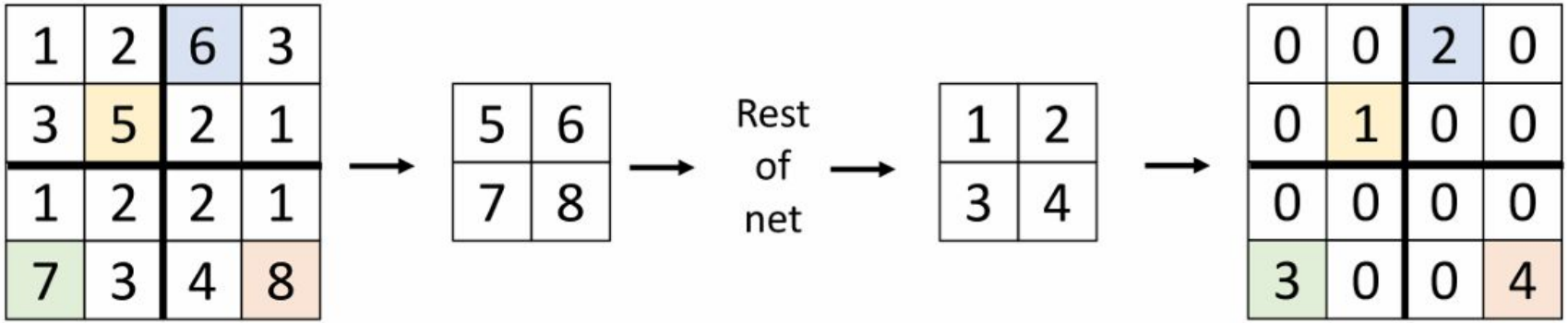


Nearest Neighbor



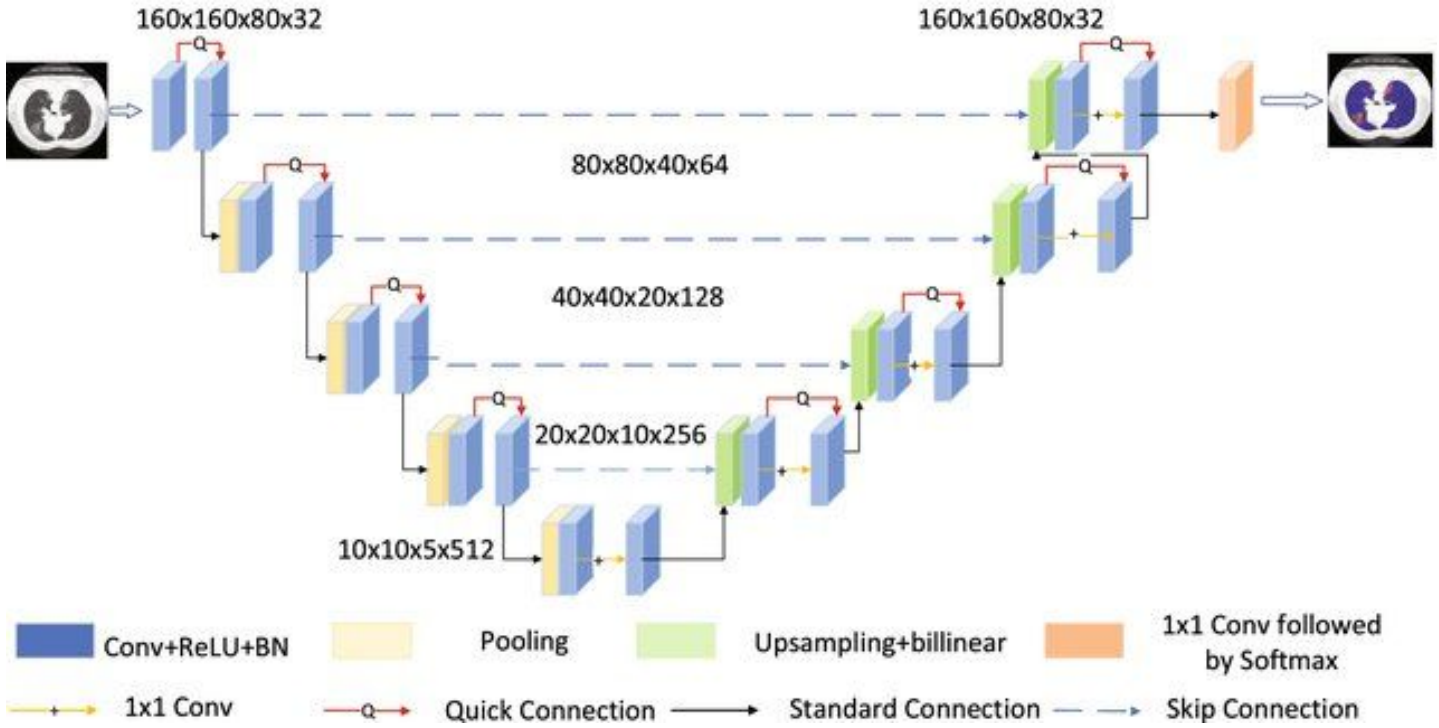
But in Network they used : **Max Unpooling**

During the max pooling of the downsampling, remember which position had the max



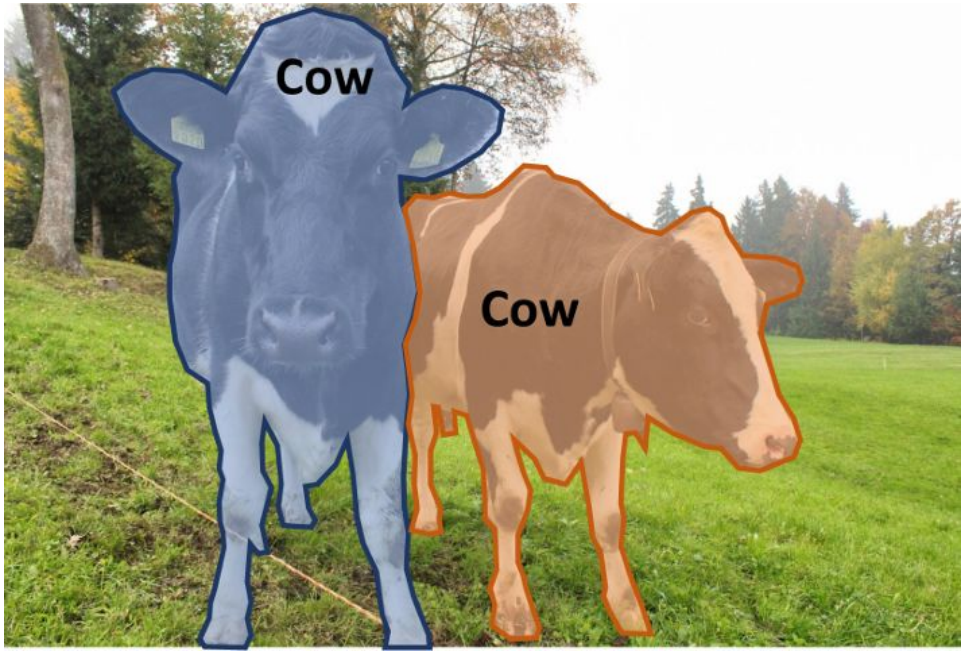
During the max unpooling of the upsampling, place into remembered positions

Semantic Segmentation w/ Fully Convolutional Network



Instance Segmentation

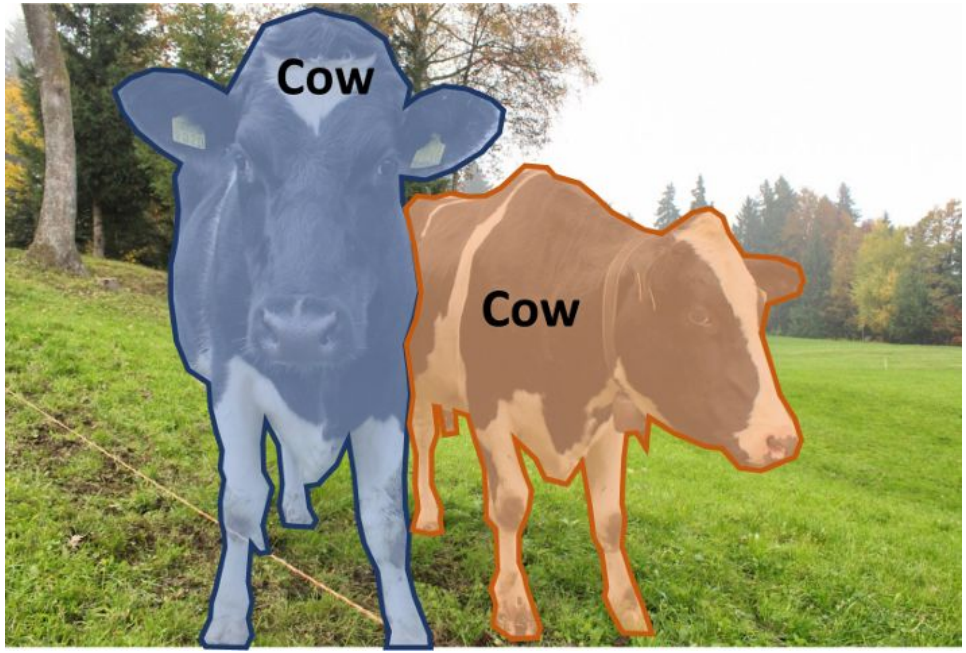
Detect all objects and identify the pixels that belong to each object



Instance Segmentation

Detect all objects and identify the pixels that belong to each object

Approach : Object detection then predict a segmentation mask for each object



Instance Segmentation w/ Mask R-CNN

Mask R-CNN

Kaiming He Georgia Gkioxari Piotr Dollár Ross Girshick

Facebook AI Research (FAIR)

Abstract

We present a conceptually simple, flexible, and general framework for object instance segmentation. Our approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel *with the existing branch for bounding box recognition. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover,*

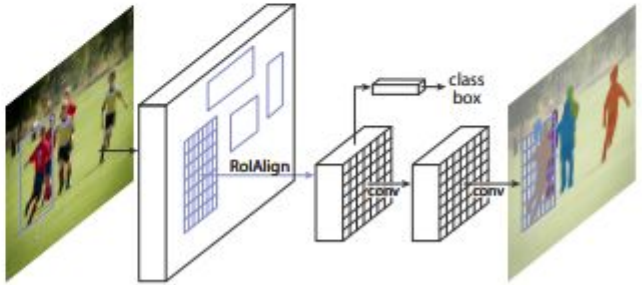
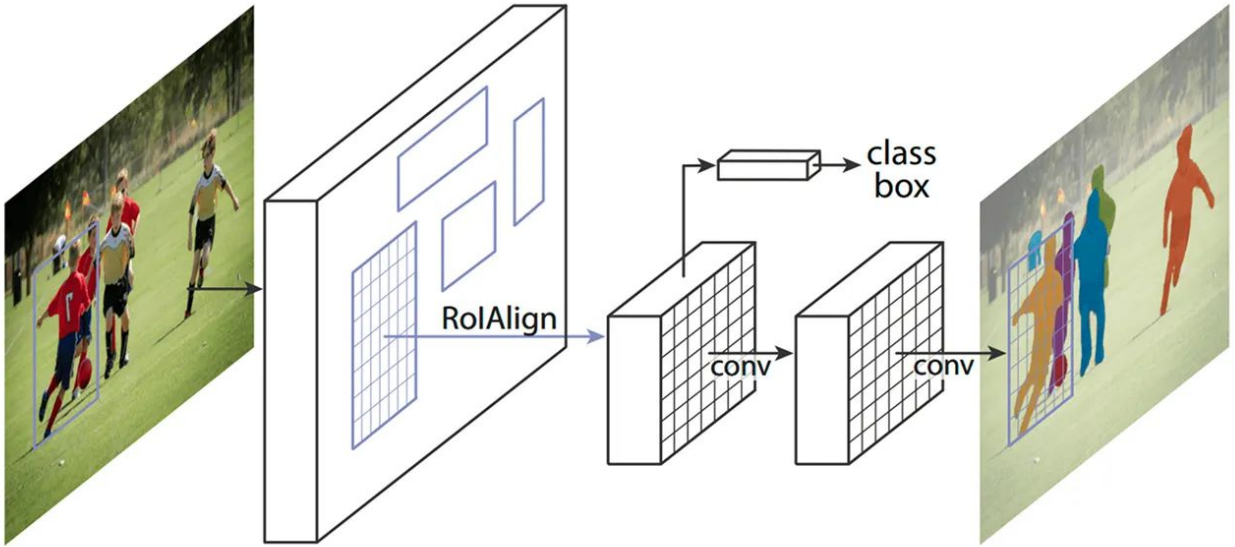


Figure 1. The **Mask R-CNN** framework for instance segmentation.

[Source: https://arxiv.org/pdf/1703.06870 2018]

Instance Segmentation w/ Mask R-CNN

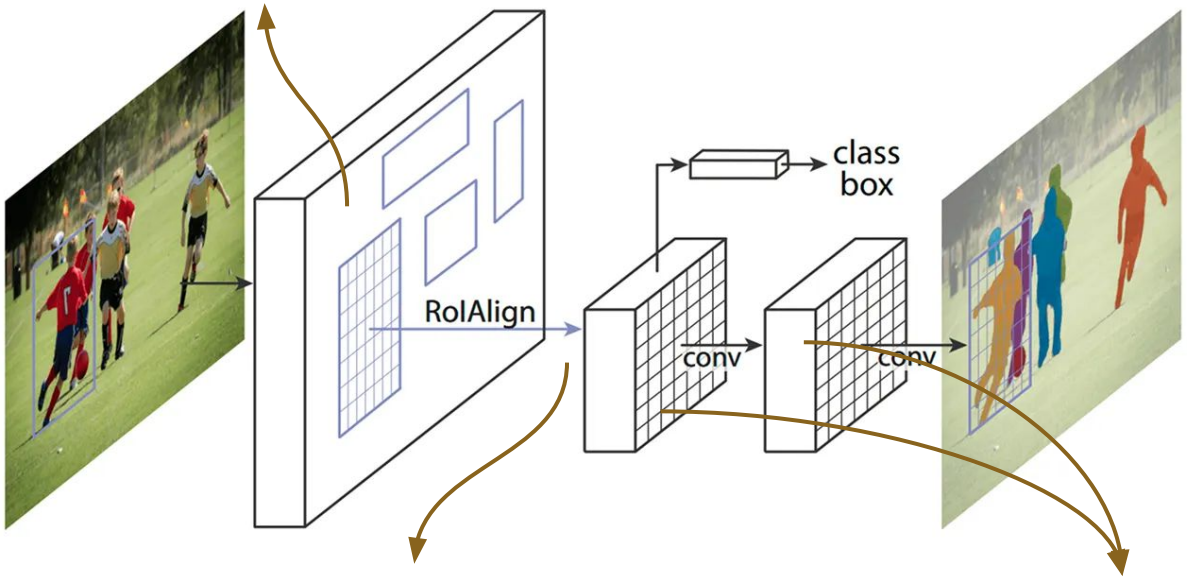


[Source: https://arxiv.org/pdf/1703.06870 2018]

Instance Segmentation w/ Mask R-CNN

The backbone extracts features

[Source: https://arxiv.org/pdf/1703.06870 2018]



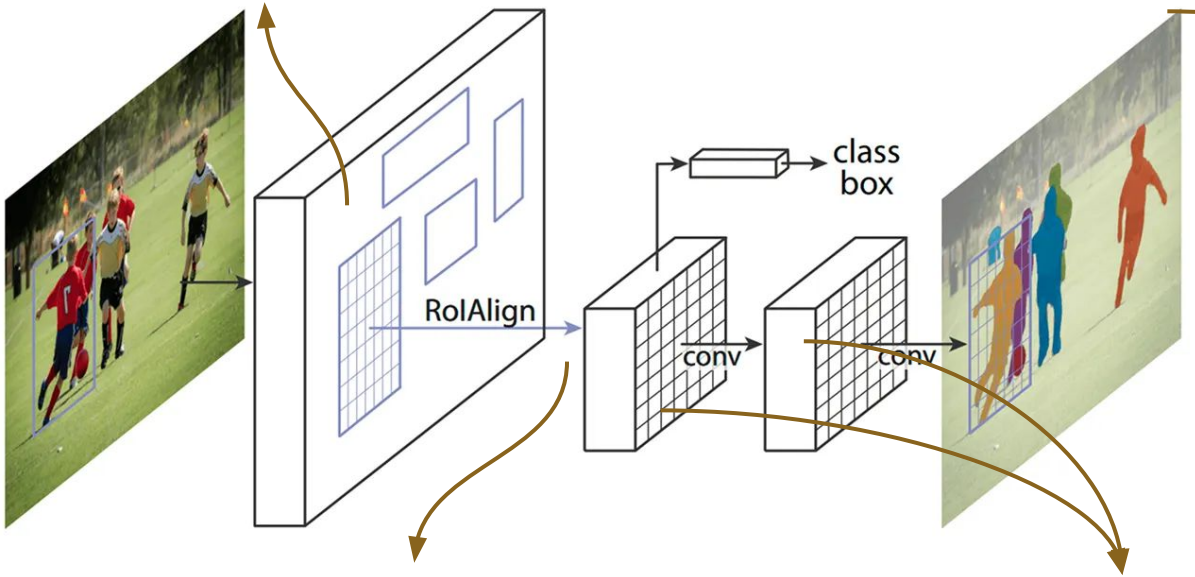
Region of Interest Align : crops and aligns the feature region for each detected object precisely

Two conv layers process each region.

Instance Segmentation w/ Mask R-CNN

The backbone extracts features

[Source: https://arxiv.org/pdf/1703.06870 2018]



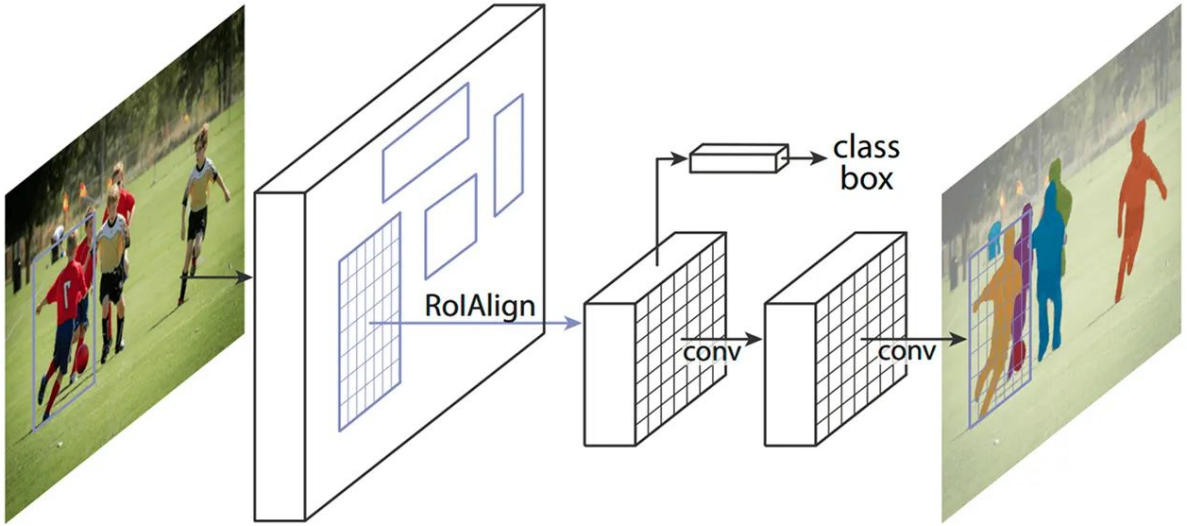
Two outputs: class and box at the top, pixel mask at the bottom

Region of Interest Align : crops and aligns the feature region for each detected object precisely

Two conv layers process each region.

Instance Segmentation w/ Mask R-CNN

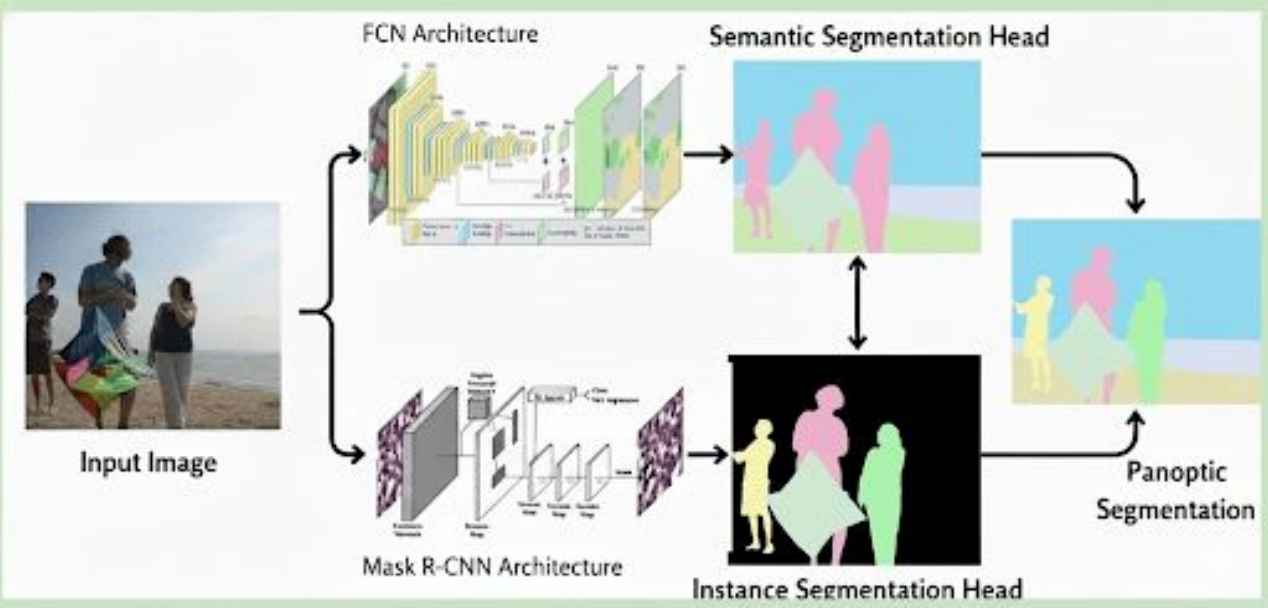
[Source: https://arxiv.org/pdf/1703.06870 2018]



Problems :

- **Still needs large annotated datasets, every mask drawn by hand**
- **Closed vocabulary, only detects the classes it was trained on, nothing else**

Panoptic Segmentation



[Source: https://www.lightly.ai/blog/panoptic-segmentation]

Semantic; every pixel gets a class.
Instance; every object gets a mask.
Panoptic combines both.

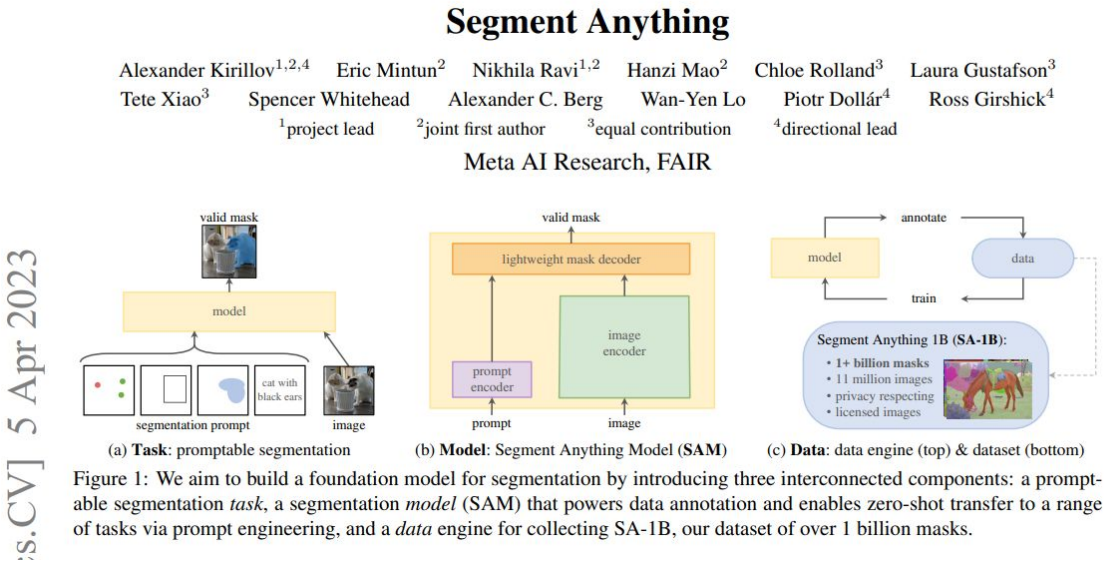
Limitations

Every model we saw is a specialist. Change the domain, start over !

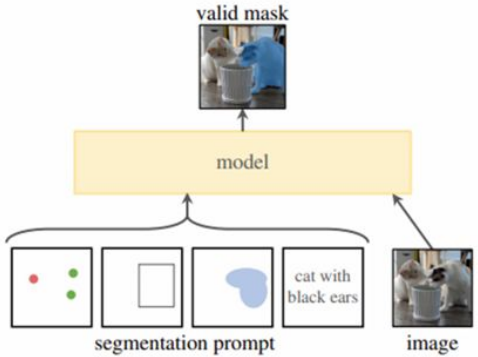
Limitations

Every model we saw is a specialist. Change the domain, start over !

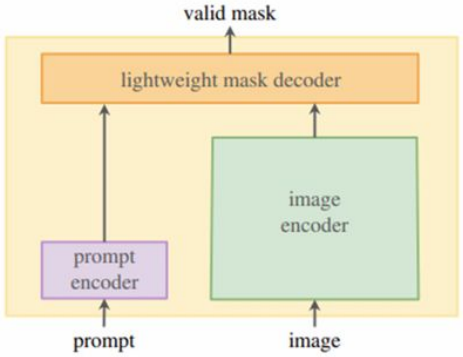
Segment Anything Model (SAM): the first foundation model for promptable segmentation.



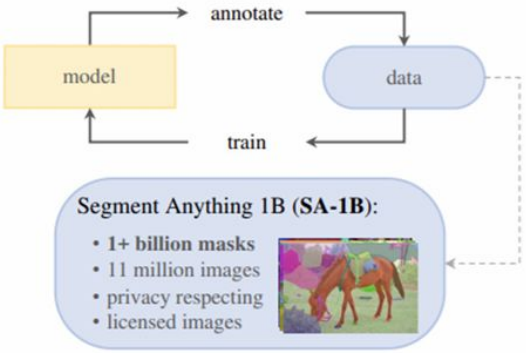
SAM is built with three interconnected components: A task, an model, and a data engine.



(a) **Task**: promptable segmentation



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

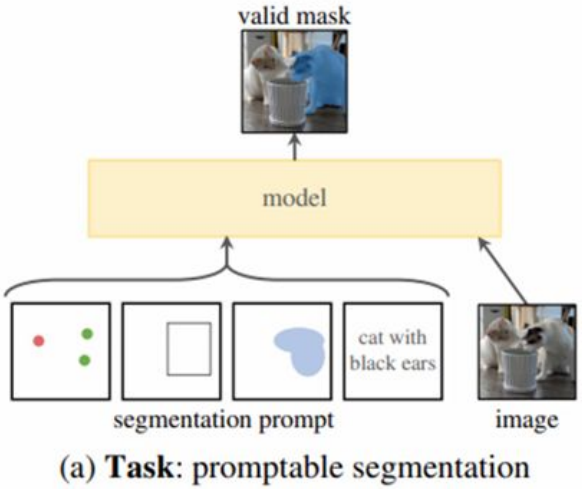


(c) **Data**: data engine (top) & dataset (bottom)

[Slide Credits: Qin Liu]

SAM: Promptable Segmentation

Given an image and a prompt, return a valid mask



(a) **Task**: promptable segmentation

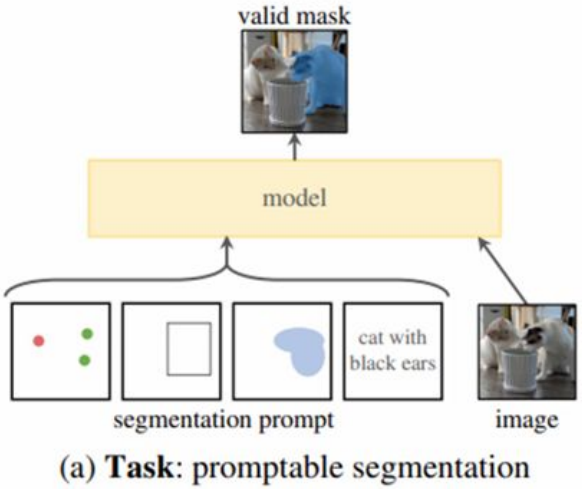
[Slide Credits: Qin Liu]

SAM: Promptable Segmentation

Given an image and a prompt, return a valid mask

SAM accepts two kinds of prompts :

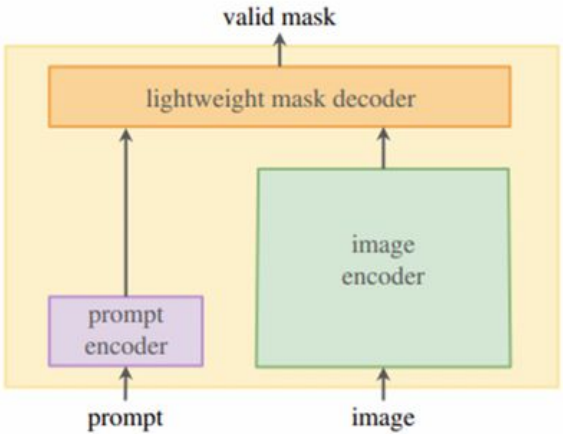
- Sparse: a click, a box, a text description. Minimal information, maximum flexibility
- Dense: a rough mask as prior. The user already has an approximation and wants SAM to refine it



(a) **Task**: promptable segmentation

[Slide Credits: Qin Liu]

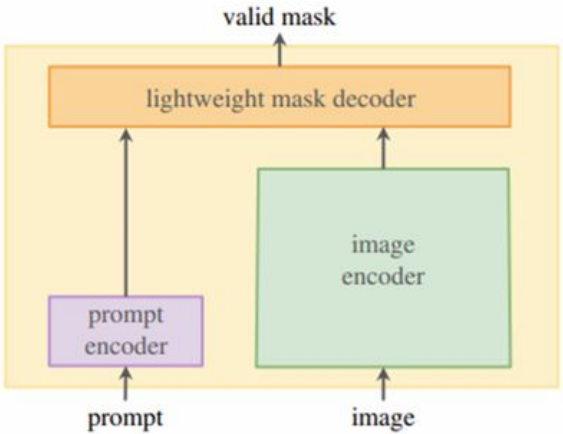
SAM: Model



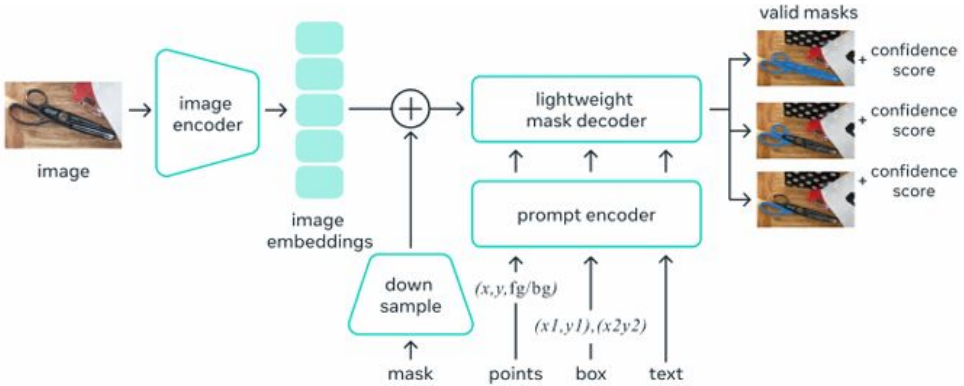
(b) **Model**: Segment Anything Model (**SAM**)

[Slide Credits: Qin Liu]

SAM: Model

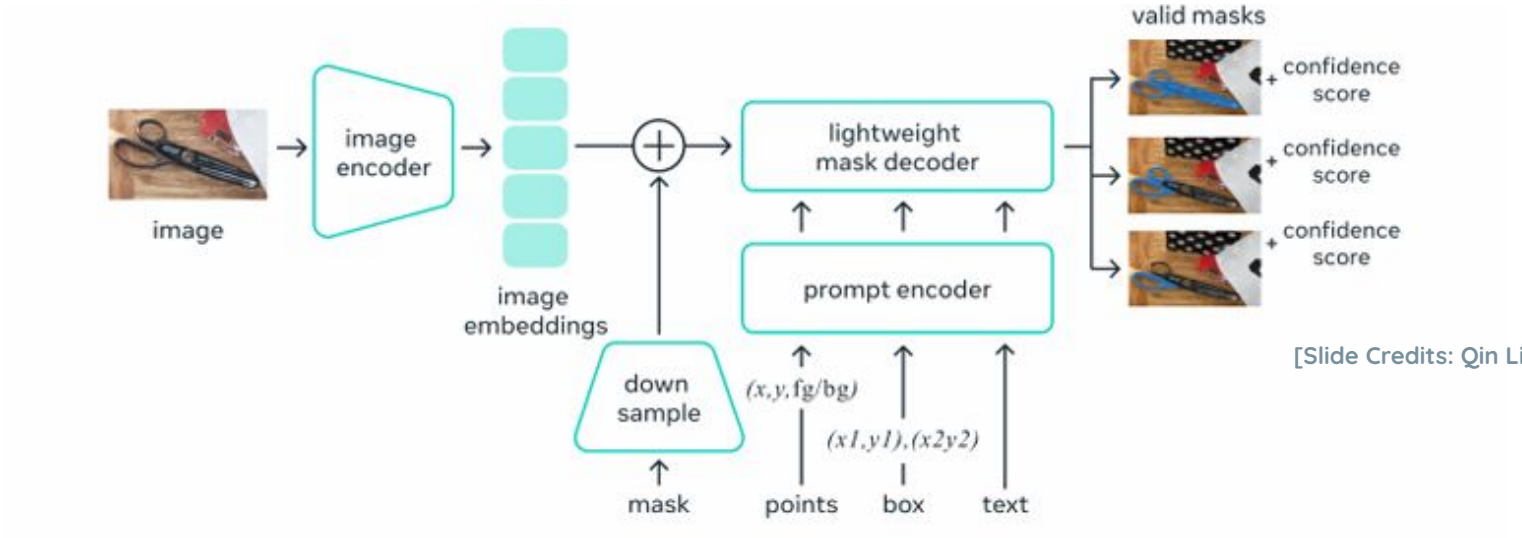


(b) **Model**: Segment Anything Model (**SAM**)



[Slide Credits: Qin Liu]

SAM: Model



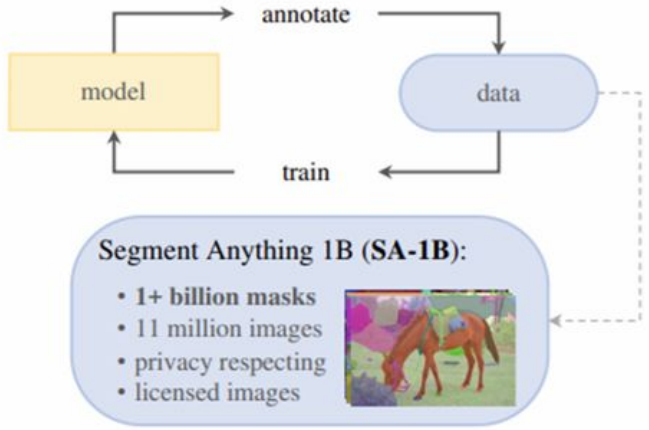
[Slide Credits: Qin Liu]

- A heavyweight image encoder outputs an image embedding.
- A lightweight prompt encoder converts the prompt into an embedding
- A mask decoder fuses image and prompt embeddings to produce masks

SAM: Data

You cannot train a foundation model on existing segmentation datasets.

COCO has 1.5 million masks.
ADE20k has a few hundred thousand



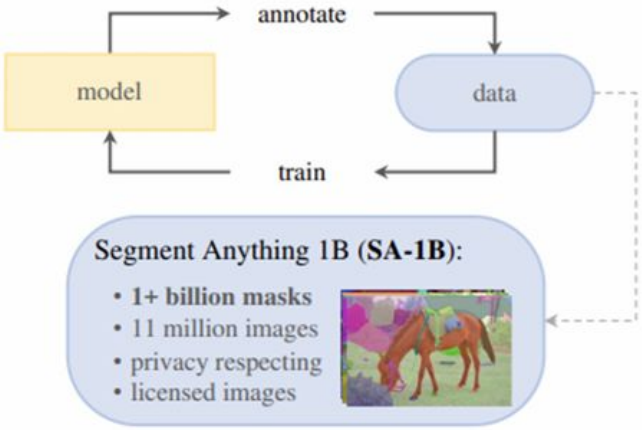
(c) **Data**: data engine (top) & dataset (bottom)

SAM: Data

You cannot train a foundation model on existing segmentation datasets.

COCO has ~860K masks.
ADE20k has a few hundred thousand

SAM needed something much larger. So Meta built the data engine, a system where SAM itself generates its own training data.

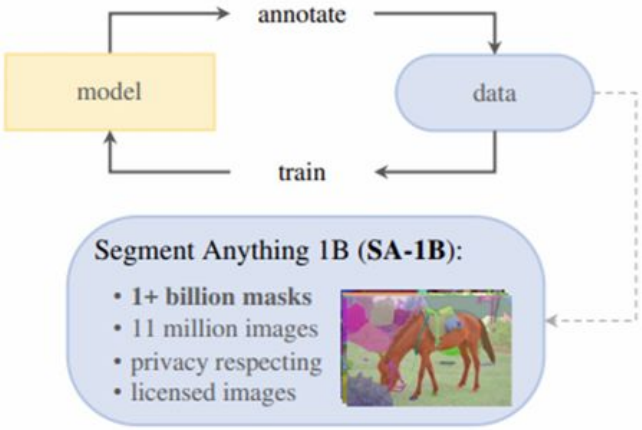


(c) **Data**: data engine (top) & dataset (bottom)

SAM: Data

Three phases :

- **Phase 1:** Human assisted Annotators use SAM to label faster. They correct, SAM proposes.



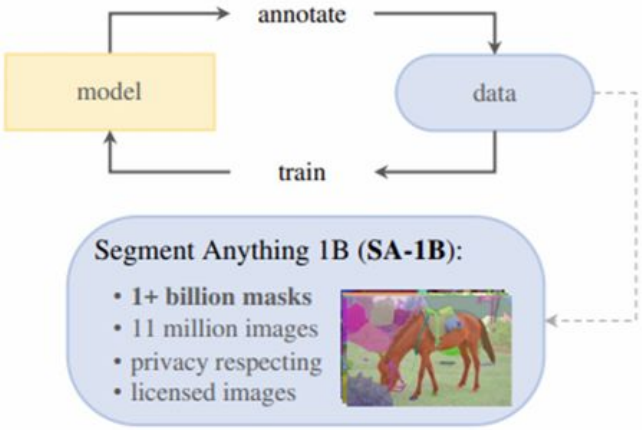
(c) **Data**: data engine (top) & dataset (bottom)

SAM: Data

Three phases :

- **Phase 1:** Human assisted Annotators use SAM to label faster. They correct, SAM proposes.

- **Phase 2:** Semi-automatic SAM proposes confident masks. Humans only fill the gaps.

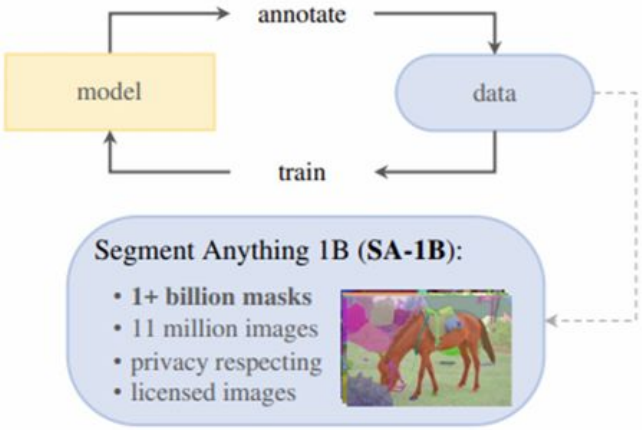


(c) **Data**: data engine (top) & dataset (bottom)

SAM: Data

Three phases :

- Phase 1: Human assisted Annotators use SAM to label faster. They correct, SAM proposes.
- Phase 2: Semi-automatic SAM proposes confident masks. Humans only fill the gaps.
- Phase 3: Fully automatic SAM runs alone. No human at all.



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

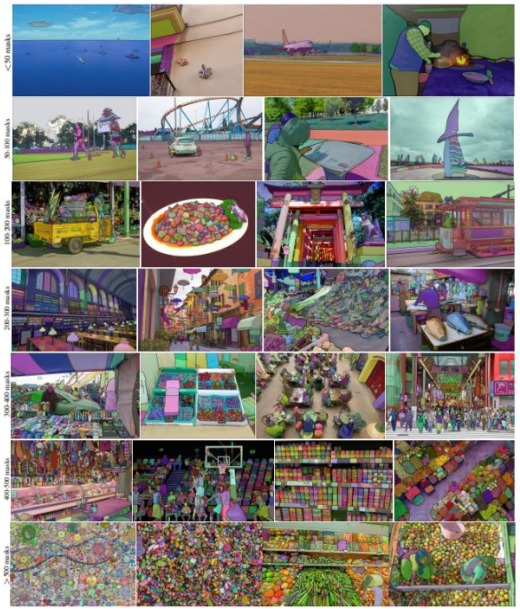
SAM: Data

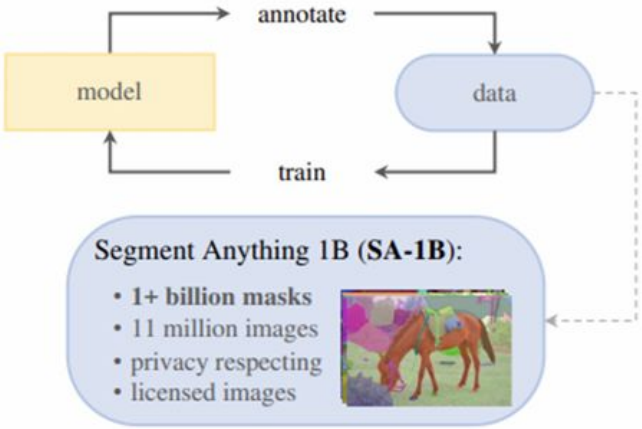
The result:

SA-1B Dataset

11 million images

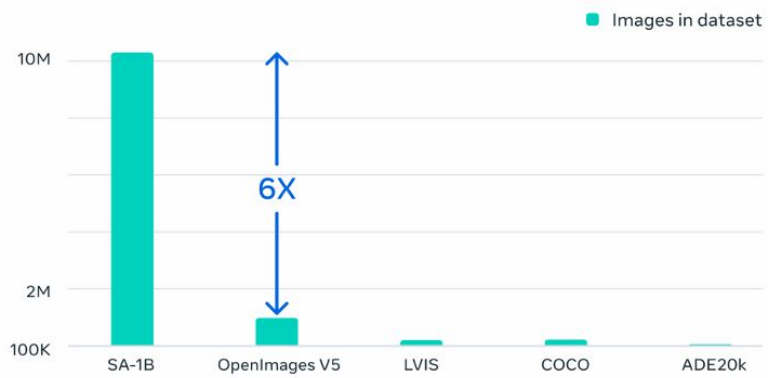
1.1 billion masks

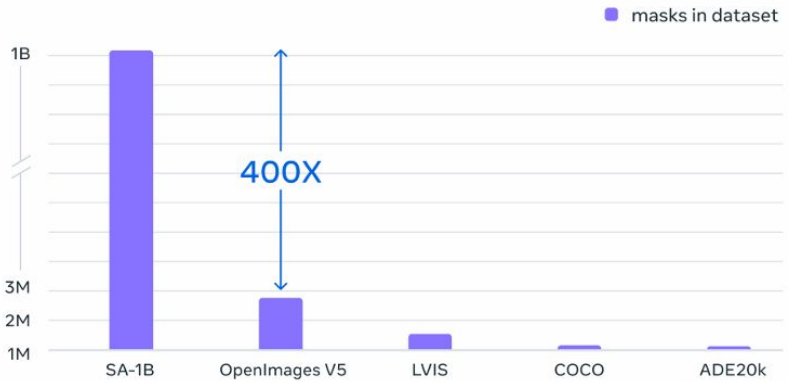




(c) **Data**: data engine (top) & dataset (bottom)

SAM: Data





400x more masks than any prior segmentation dataset

SAM: Evolution

SAM (2023)

- First foundation model for promptable segmentation
- Image encoder ViT-H, prompt encoder, mask decoder
- SA-1B: 11M images, 1.1B masks
- Images only

SAM: Evolution

SAM (2023)

- First foundation model for promptable segmentation
- Image encoder ViT-H, prompt encoder, mask decoder
- SA-1B: 11M images, 1.1B masks
- Images only

SAM2 (2024)

- Extends SAM to video
- Adds a memory bank, stores information from previous frames
- Faster

SAM: Evolution

SAM (2023)

- First foundation model for promptable segmentation
- Image encoder ViT-H, prompt encoder, mask decoder
- SA-1B: 11M images, 1.1B masks
- Images only

SAM2.1 (2024)

- Improved model checkpoints
- Better performance on complex scenes
- Same architecture, better training

SAM: Evolution

SAM (2023)

- First foundation model for promptable segmentation
- Image encoder ViT-H, prompt encoder, mask decoder
- SA-1B: 11M images, 1.1B masks
- Images only

SAM3 (2025)

- Prompt with an example image → finds all similar objects

SAM: Evolution

SAM (2023)

- First foundation model for promptable segmentation
- Image encoder ViT-H, prompt encoder, mask decoder
- SA-1B: 11M images, 1.1B masks
- Images only

SAM3.1 (March 2026)

- Optimized for real-time video processing
- Faster, more accessible

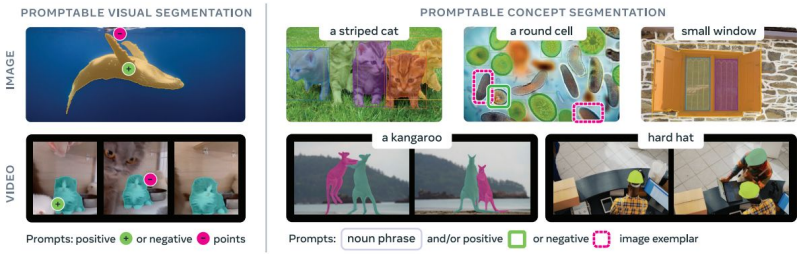


Figure 1 SAM 3 improves over SAM 2 on promptable *visual* segmentation with clicks (left) and introduces the new promptable *concept* segmentation capability (right). Users can segment all instances of a visual concept specified by a short noun phrase, image exemplars (positive or negative), or a combination of both.

Evaluate the course!

References I

Books

• Computer Vision: Models, Learning, and Inference (2012). https://amzn.to/2rxrdOF
• Programming Computer Vision with Python (2012). https://amzn.to/2QKTaAL
• Multiple View Geometry in Computer Vision (2004). https://amzn.to/2LfHLE8
• Computer Vision: Algorithms and Applications (2010). https://amzn.to/2LcIt4J
• Computer Vision: A Modern Approach (2002). https://amzn.to/2rv7AqJ
• Deep Learning for Computer Vision : Image Classification, Object Detection and Face Recognition in Python(2019). https://machinelearningmastery.com/deep-learning-for-computer-vision/
• Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow — Aurélien Géron (2019) https://amzn.to/3uZbN6Z
• Hands-On Large Language Models - Jay Alammar & Grootendorst

Articles & Online Resources

• Krizhevsky, Sutskever, Hinton (2012) — ImageNet Classification with Deep CNNs (AlexNet — deep learning breakthrough)
• Improved Texture Networks: Maximizing Quality and Diversity in Feed-forward Stylization and Texture Synthesis, 2017
• "Computer vision," Wikipedia. https://en.wikipedia.org/wiki/Computer_vision
• "Machine vision," Wikipedia. https://en.wikipedia.org/wiki/Machine_vision
• "Digital image processing," Wikipedia. https://en.wikipedia.org/wiki/Digital_image_processing

References II

- "Computer vision," Wikipedia. https://en.wikipedia.org/wiki/Computer_vision
- https://medium.com/data-reply-it-datatech
- https://huggingface.co/blog/RDTvlokip/quand-l-ia-apprend-de-l-experience-comme-toi
- Zhuang et al., "A Comprehensive Survey on Transfer Learning," IEEE, 2020.
- https://jalammar.github.io/illustrated-transformer/
- https://sh-tsang.medium.com/review-vision-transformer-vit-406568603de0
- Laurier, Linda. (2025). Vision Transformers: From Patch-Based Processing to Visual Intelligence.
- "Training Convolutional Neural Networks: What Is Machine Learning" Analog Devices. https://www.analog.com/en/resources/analog-dialogue/articles/training-convolutional-neural-networks-what-is-machine-learning-part-2.html
- Roth, Peter M.. (2011). On-line Conservative Learning.