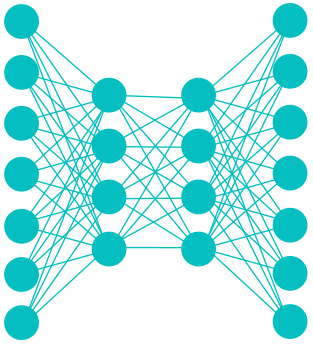
Lecture Notes for

# Neural Networks and Machine Learning

## Fully Convolutional Learning
## Instance Segmentation
## And Summary

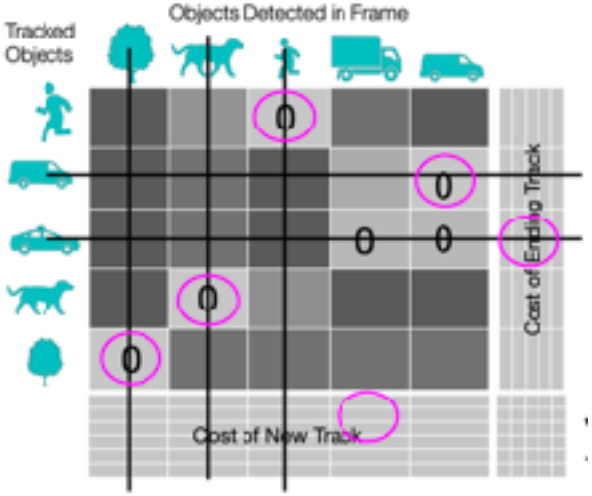


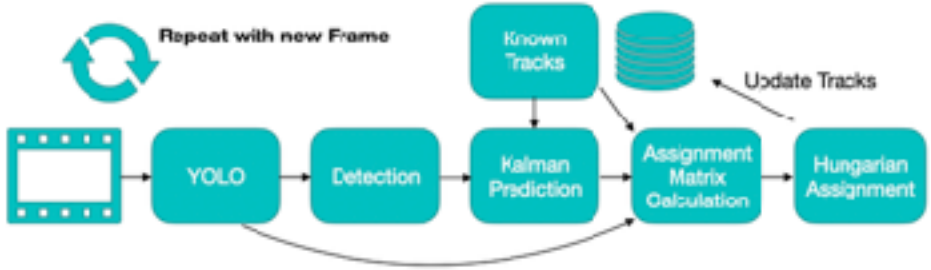
# Logistics and Agenda

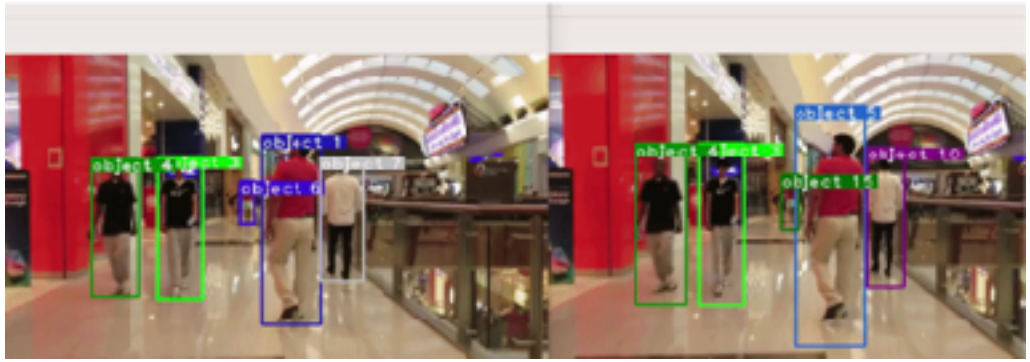
- Logistics
  - Grading update
- Agenda
  - Fully Convolutional Learning
    - Semantic Segmentation (last last time)
    - Object Detection (last time)
    - Instance Segmentation (this time)
    - Wrap Up (this time)

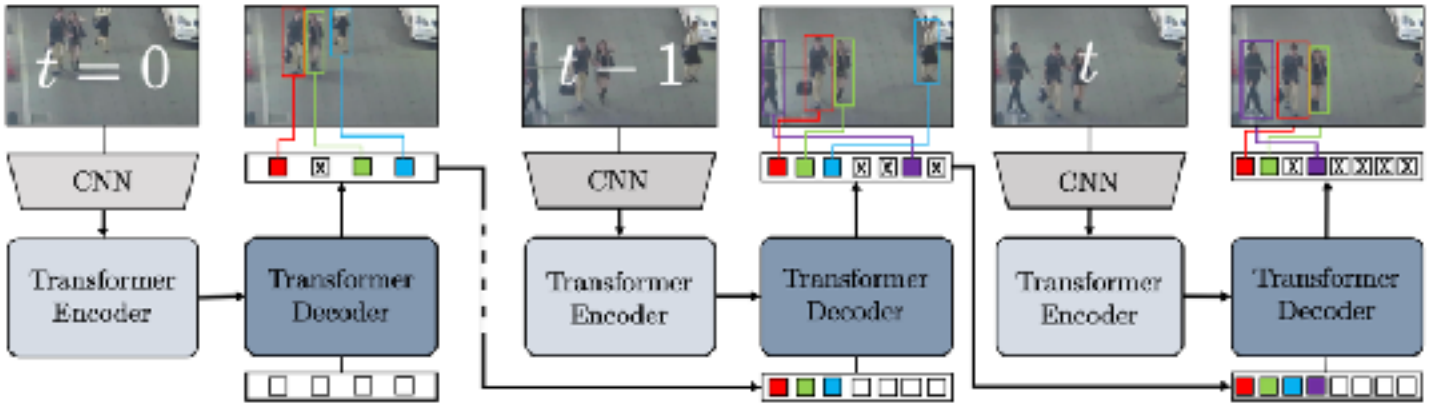


# Last Time: Tracking



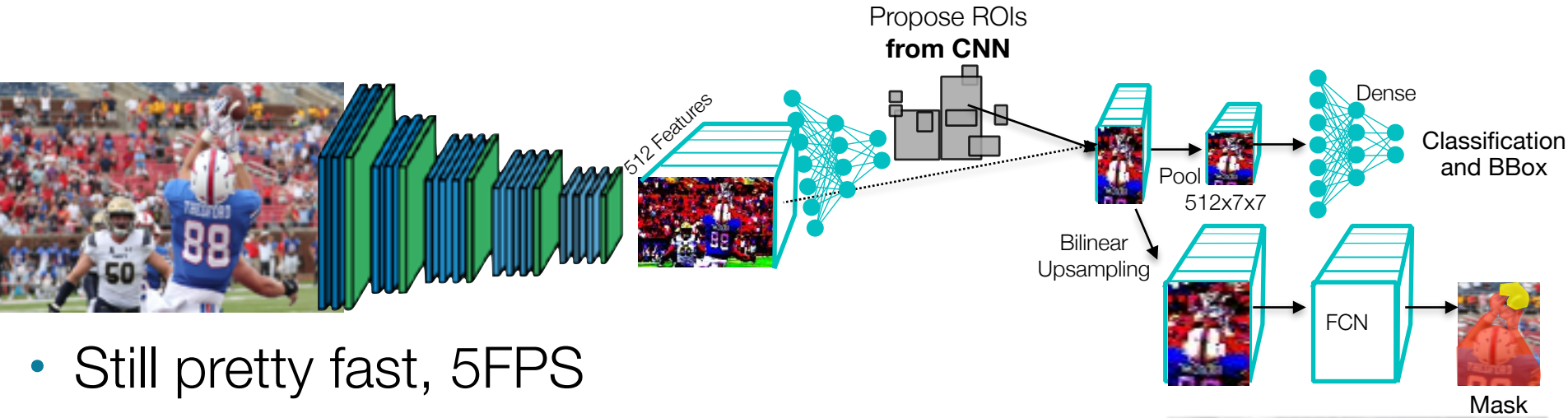



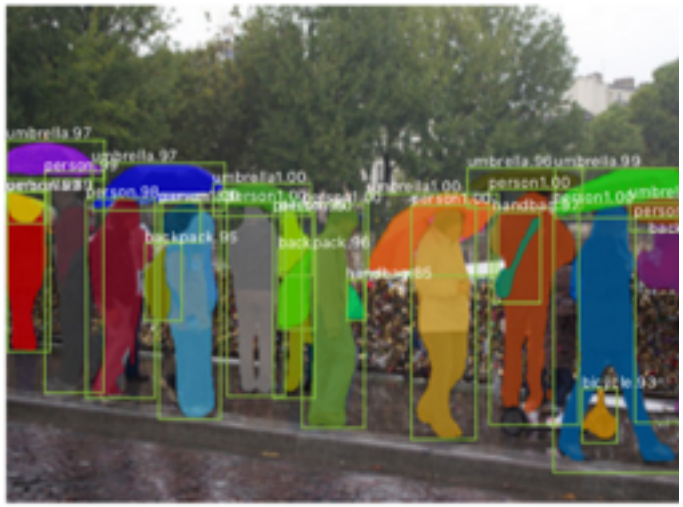


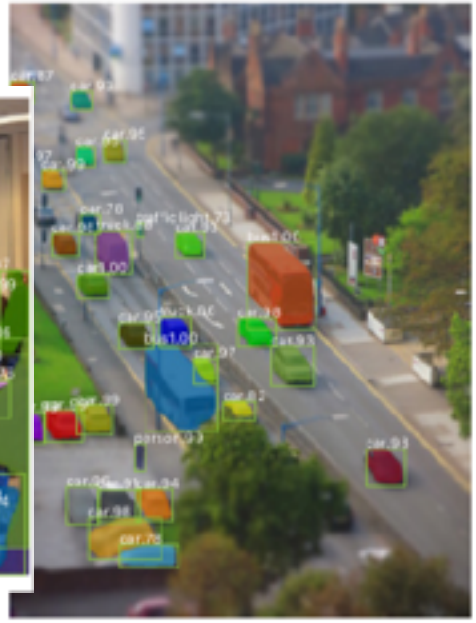
# Instance Segmentation

# 2018: Mask R-CNN



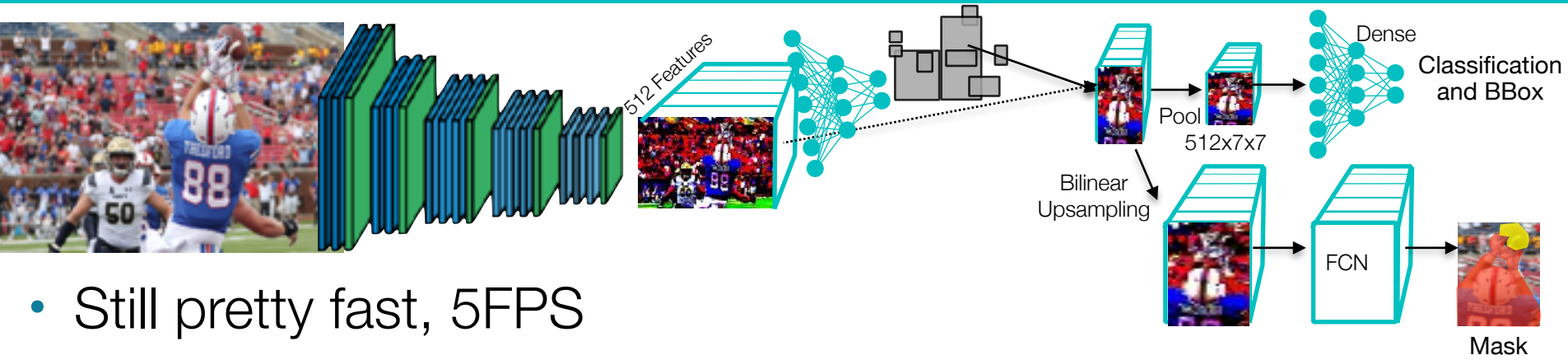

- Still pretty fast, 5FPS
- Excellent results





Ren et al. Mask R-CNN, 2018

# 2018: Mask R-CNN



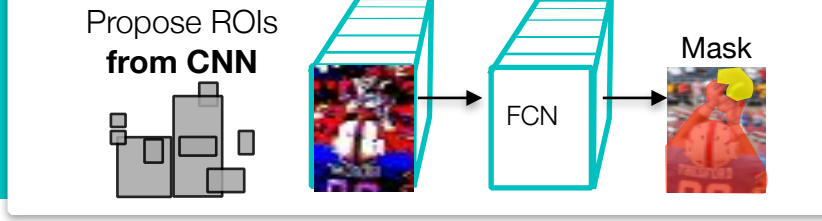

- Still pretty fast, 5FPS
- Excellent results

An Excellent, well documented Implementation here:
https://github.com/matterport/Mask_RCNN

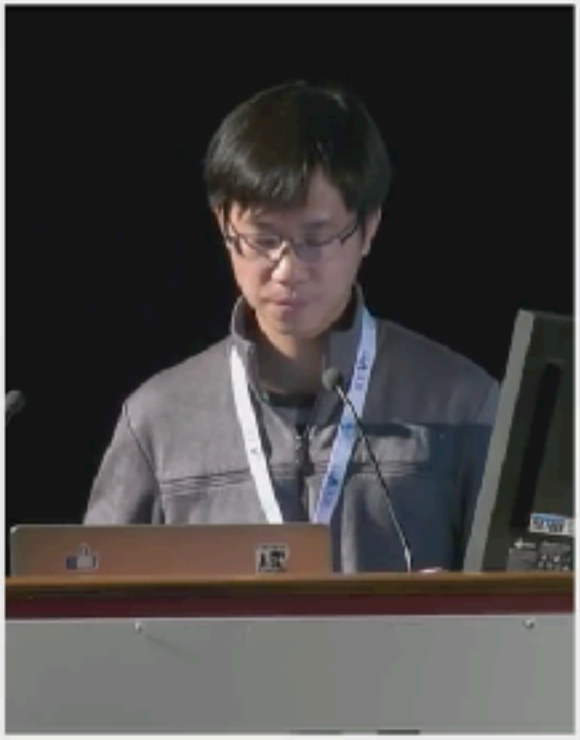
```
# Use shape of first image. Images in a batch must have the same size.
image_shape = parse_image_meta_graph(image_meta)['image_shape'][0]
# Equation 1 in the Feature Pyramid Networks paper. Account for
# the fact that our coordinates are normalized here.
# e.g. a 224x224 ROI (in pixels) maps to P4
image_area = tf.cast(image_shape[0] * image_shape[1], tf.float32)
roi_level = log2_graph(tf.sqrt(h * w) / (224.0 / tf.sqrt(image_area)))
roi_level = tf.minimum(5, tf.maximum(
```

Ren et al. Mask R-CNN, 2018

# 2018: Mask R-CNN

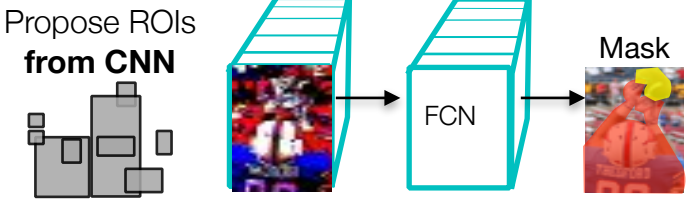



Can also provide **key point detection** from same FCN features (not real time, post processed)




ICCV17

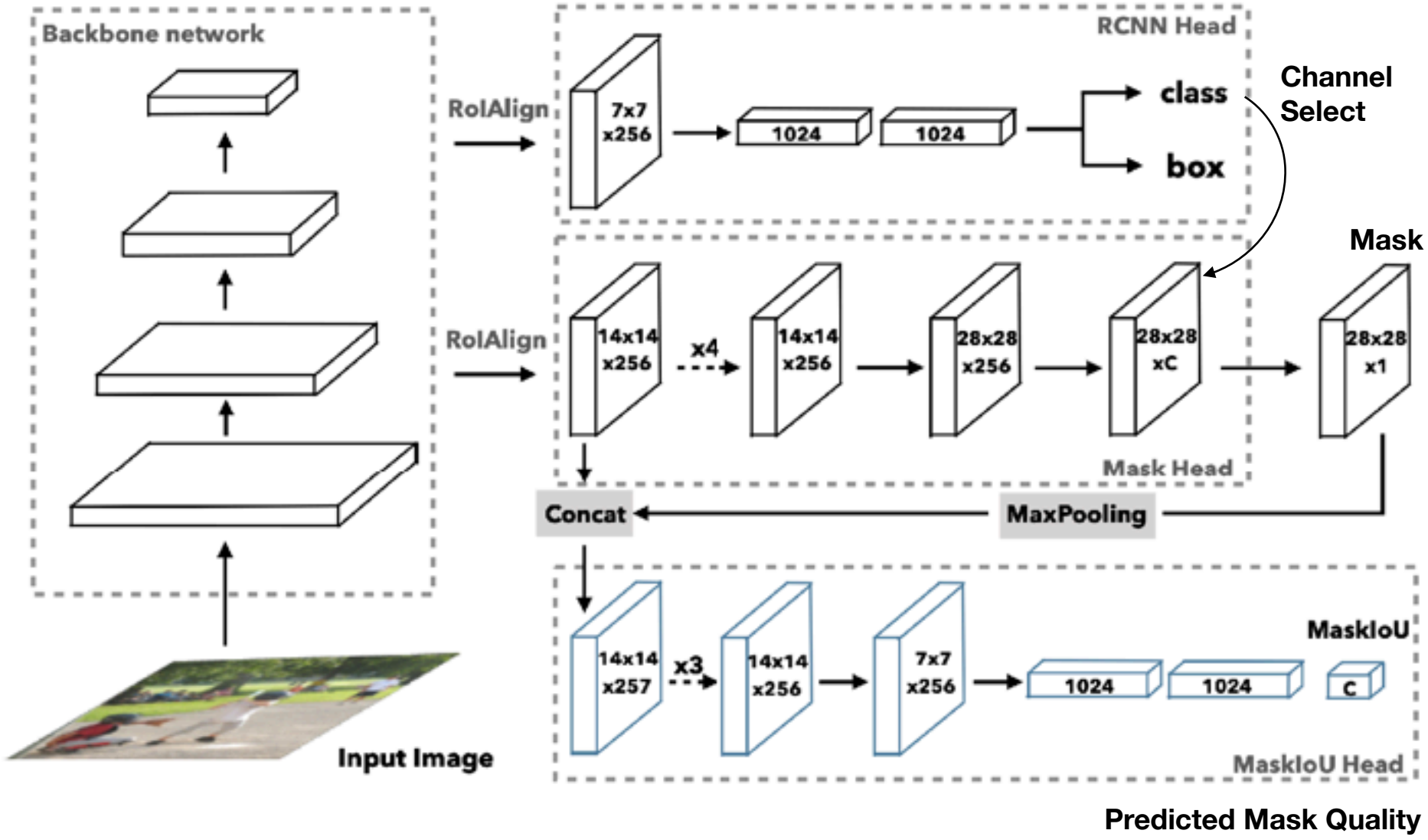
# 2018: Mask R-CNN




- Real time, Mask R-CNN


https://www.youtube.com/watch?v=nEug0-pD0Ms

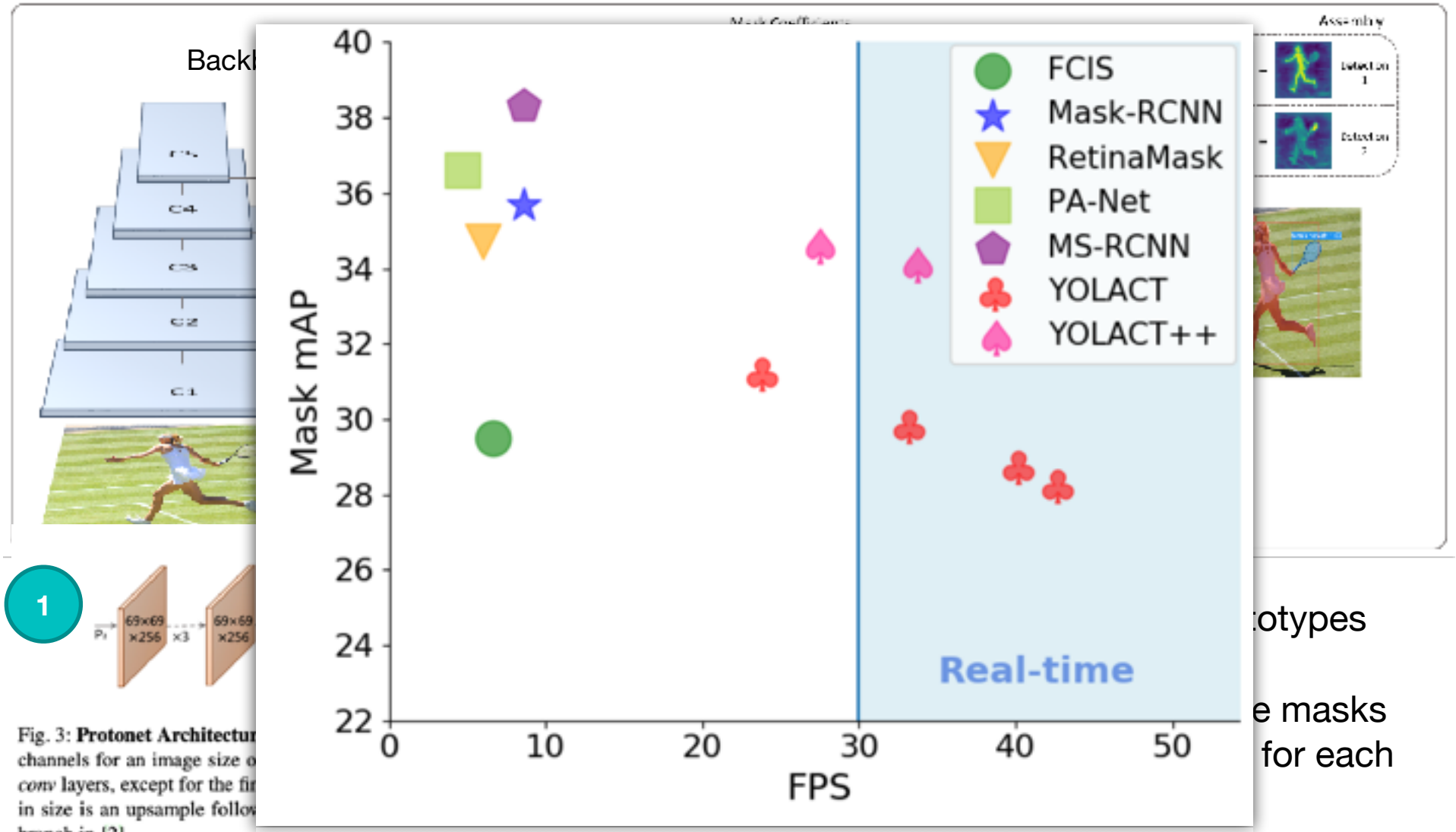
# March 2019: Mask Scoring RCNN, MS-RCNN




Zhaojin Huang *et al.* Mask Scoring RCNN. arXiv pre-print, 2019

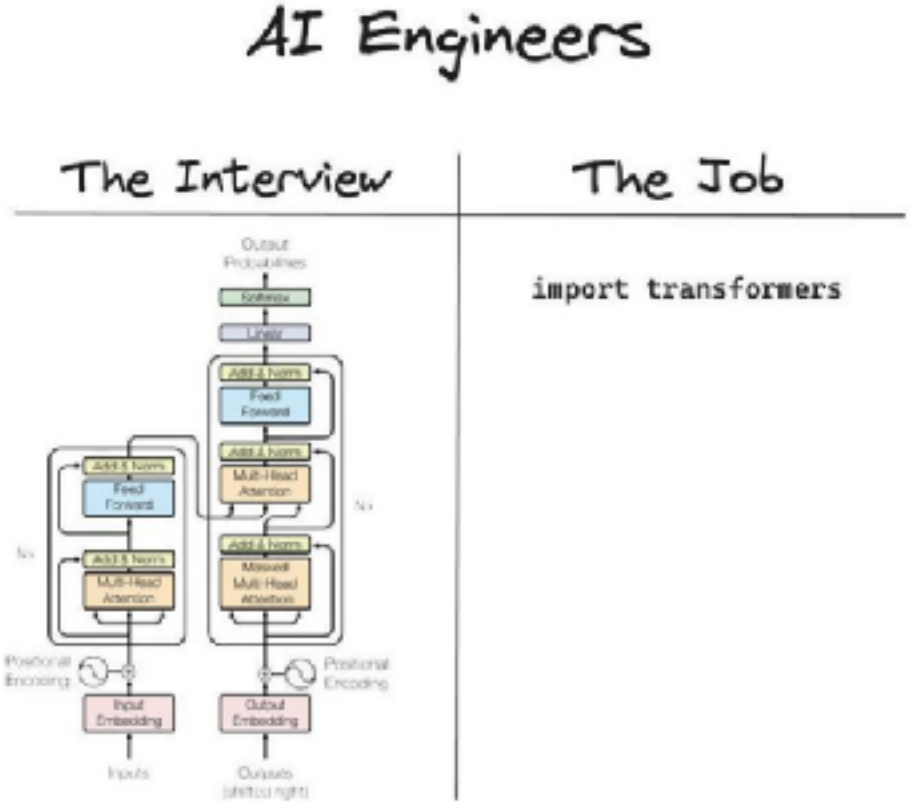


# Early 2020: YOLACT++

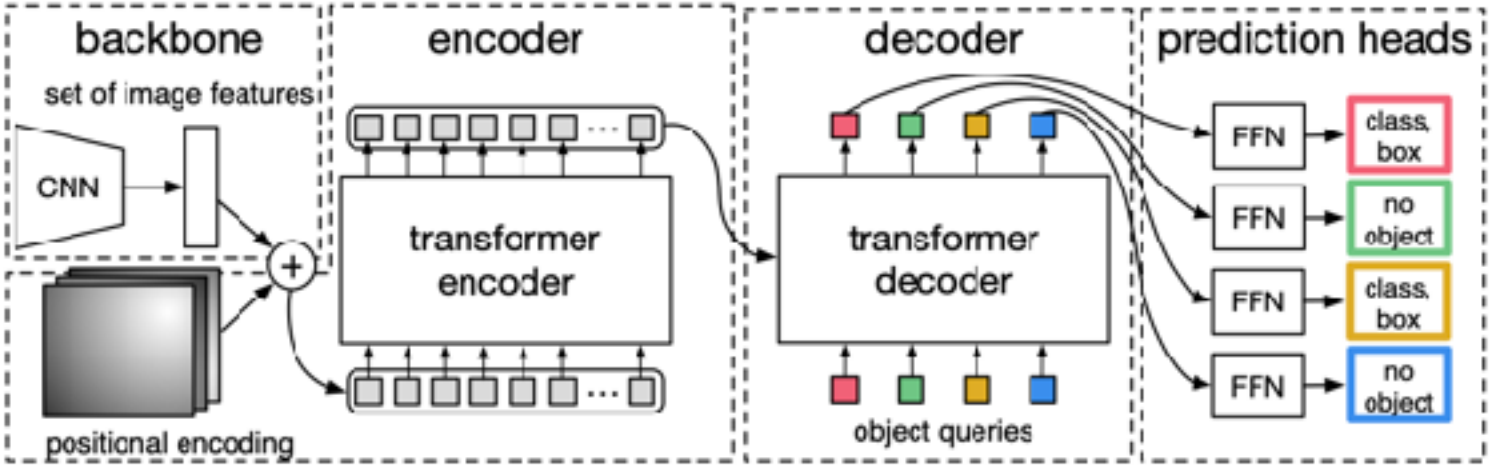


# YOLACT++ Example Video (Real time)

# ViT for Detection and Segmentation



# Object Detection with Transformers



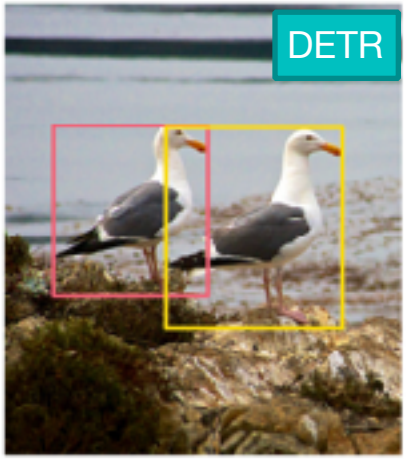
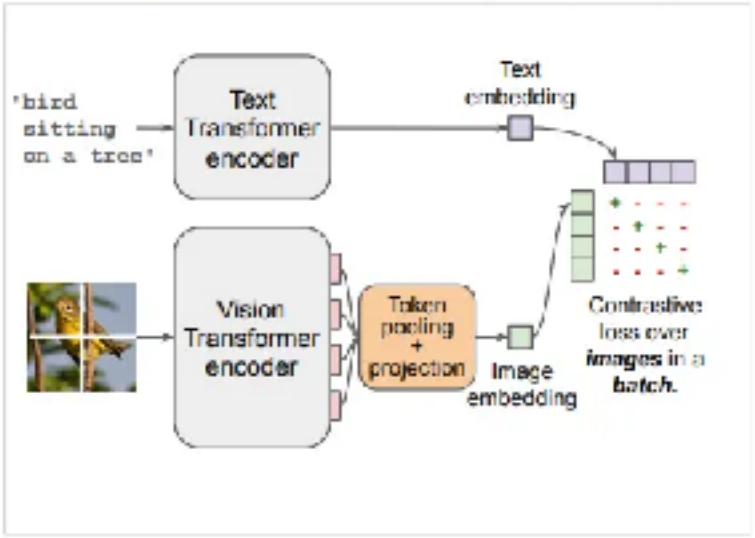
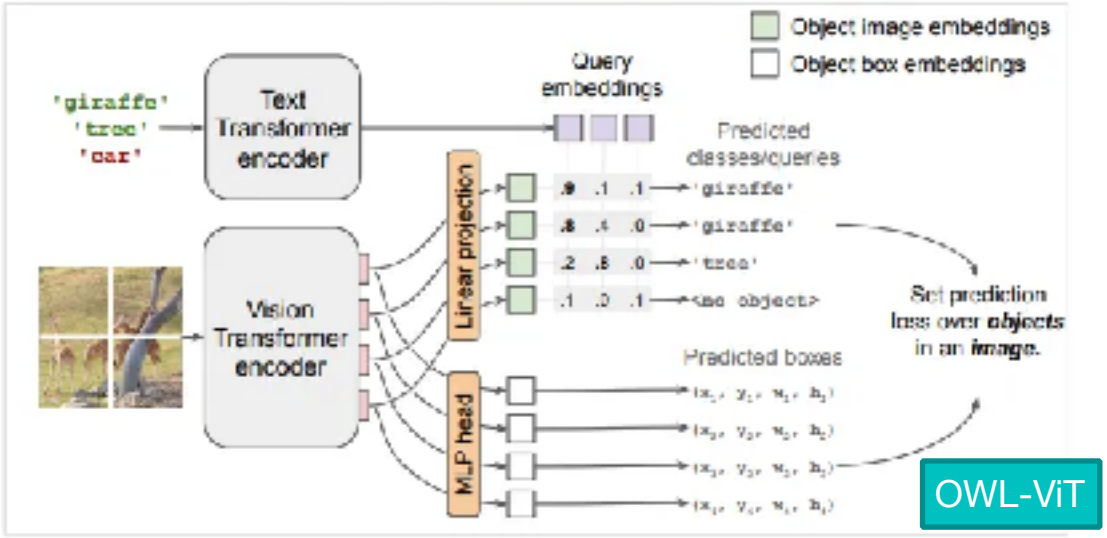




Image-level contrastive pre-training




Transfer to open-vocabulary detection



# Extending ViT to Masks

## Method

https://github.com/czczup/ViT-Adapter?tab=readme-ov-file

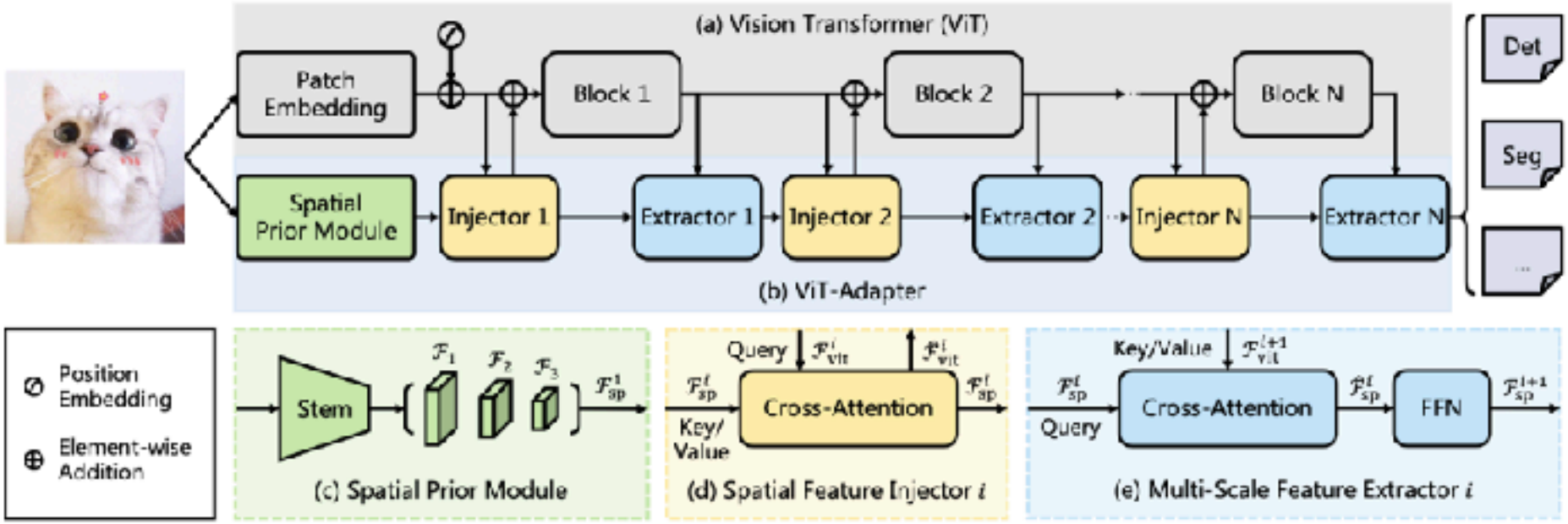
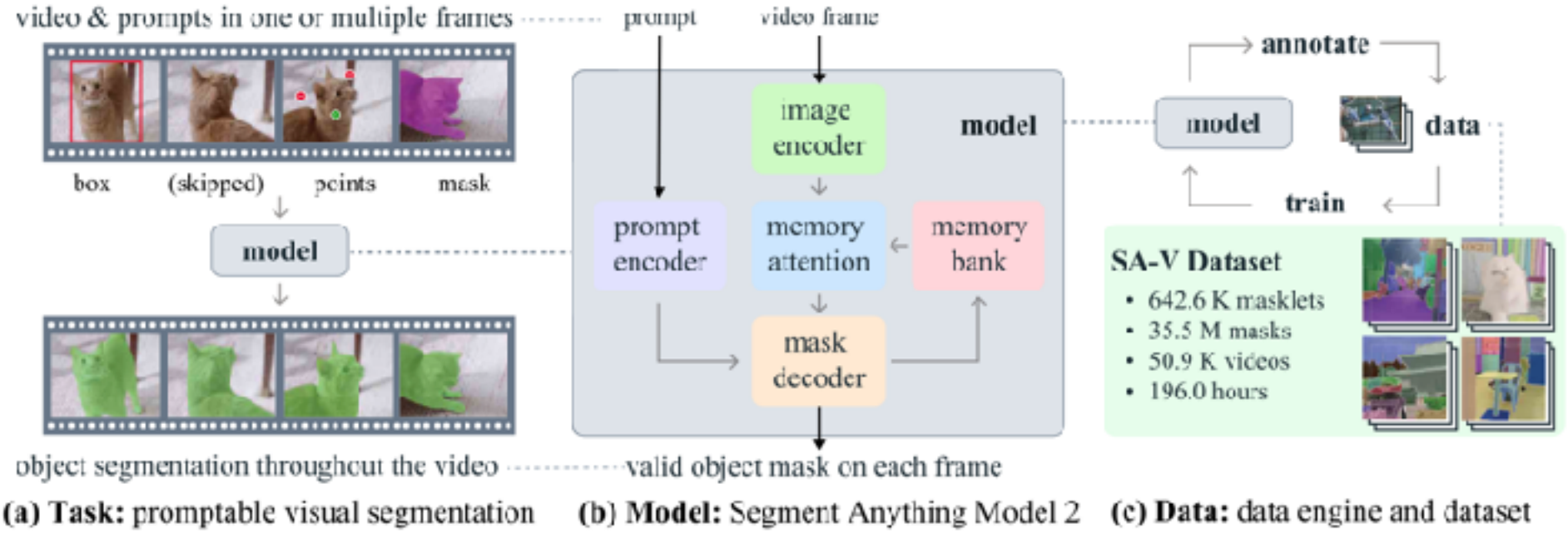


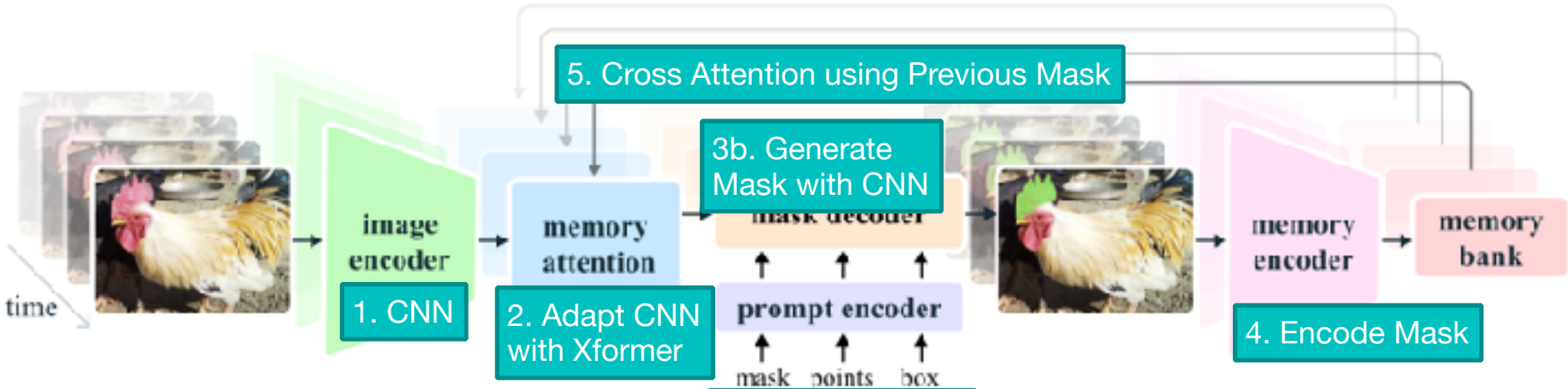

Figure 4: **Overall architecture of ViT-Adapter.** (a) The ViT, whose encoder layers are divided into $N$ (usually $N = 4$) equal blocks for feature interaction. (b) Our ViT-Adapter, which contains three key designs, including (c) a spatial prior module for modeling local spatial contexts from the input image, (d) a spatial feature injector for introducing spatial priors into the ViT, and (e) a multi-scale feature extractor for reconstructing multi-scale features from the single-scale features of ViT.




We verify ViT-Adapter on multiple dense prediction tasks, including object detection, instance segmentation, and semantic segmentation. Notably, without using extra detection data, our ViT-Adapter-L yields state-of-the-art 60.9 box AP and 53.0 mask AP on COCO test-dev.

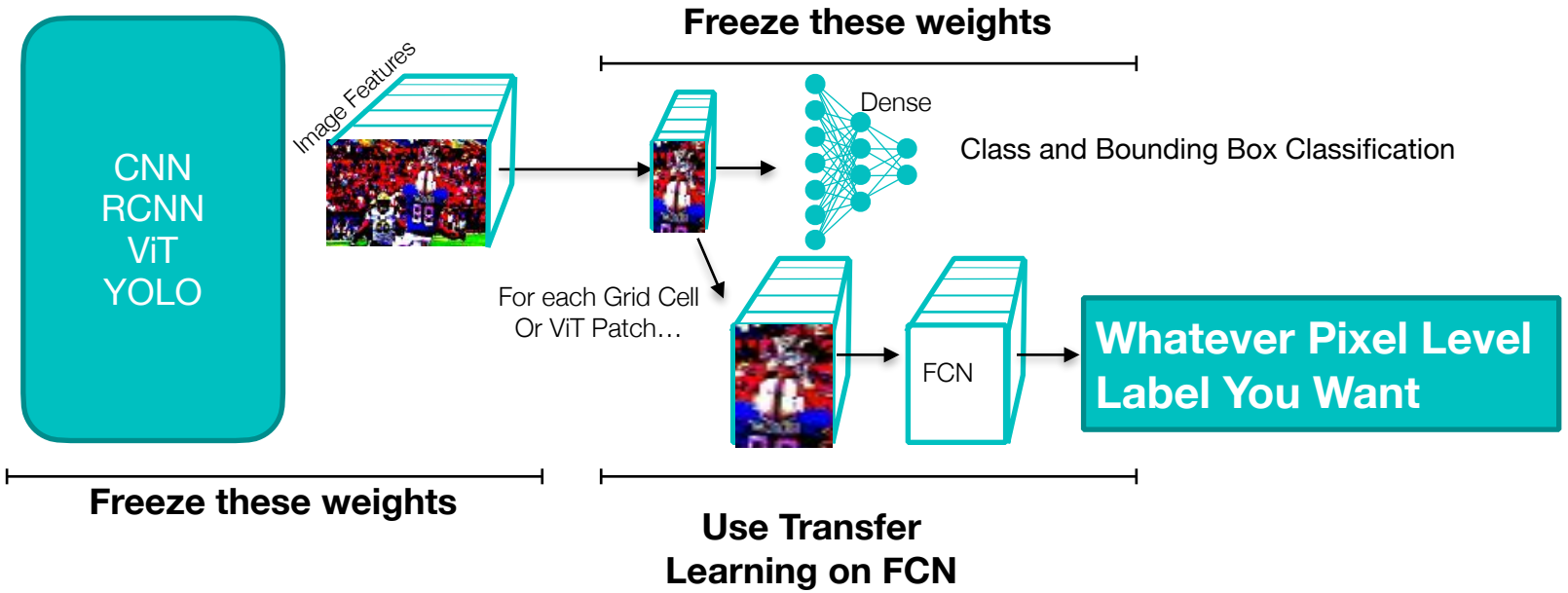
# Segment Anything (already covered!)



# Expanding Masking

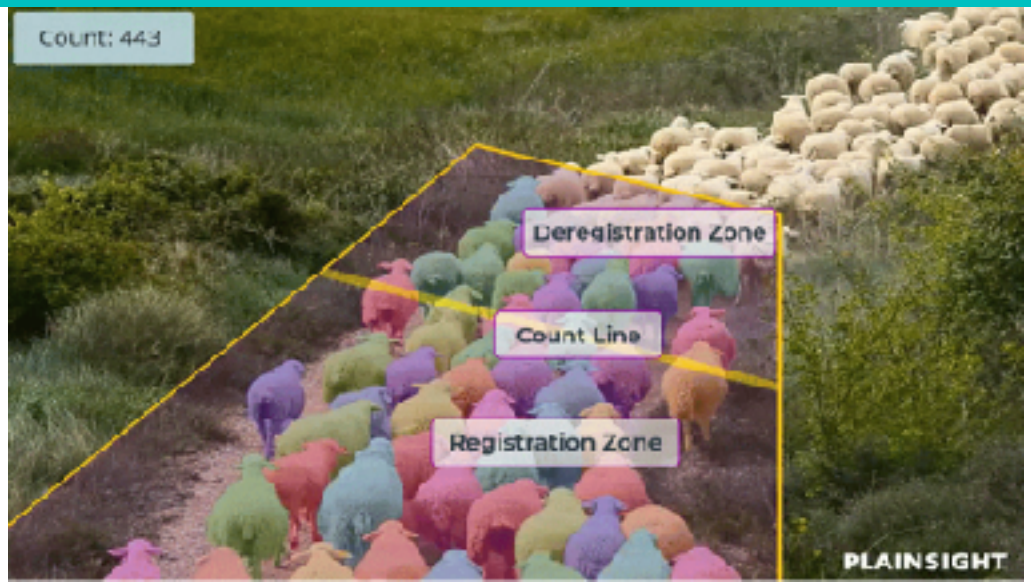
- **Key insight**: features that can be used for getting mask of object, are good at doing other things:
  - Like human pose estimation
  - …Depth processing *and more*
- Just connect FCN to model features and learn any label



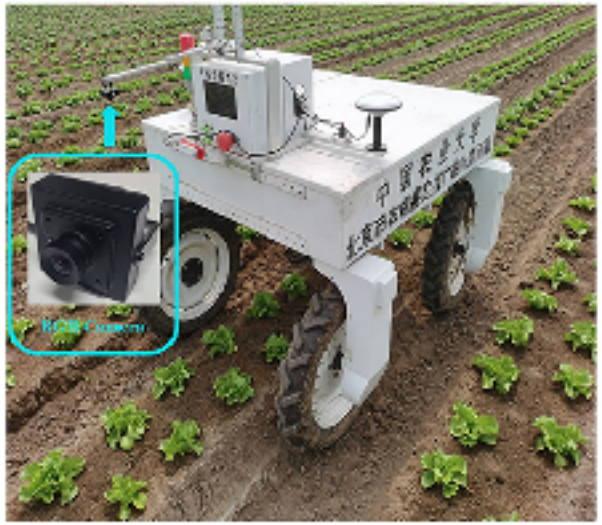


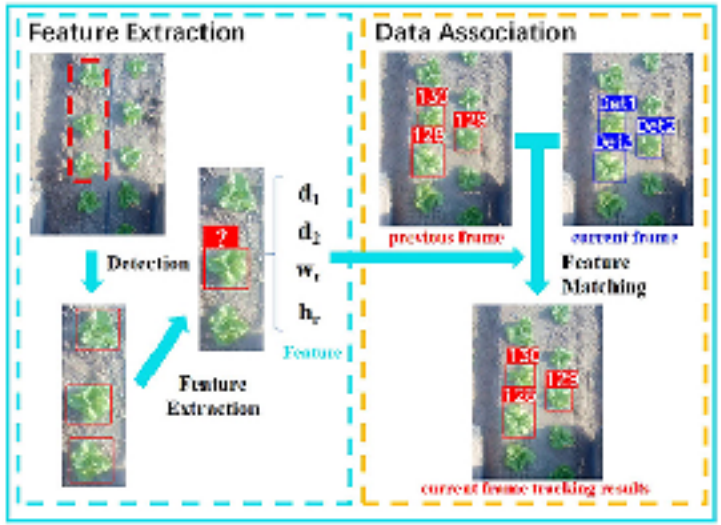

# Applications of Object Detection and Instance Segmentation

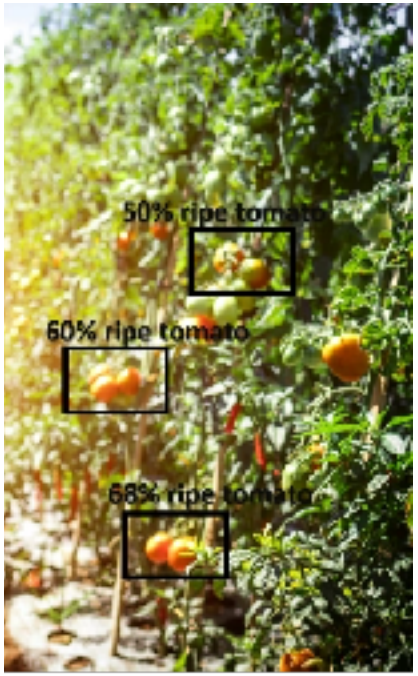
# Farming, Ranching, and Agriculture









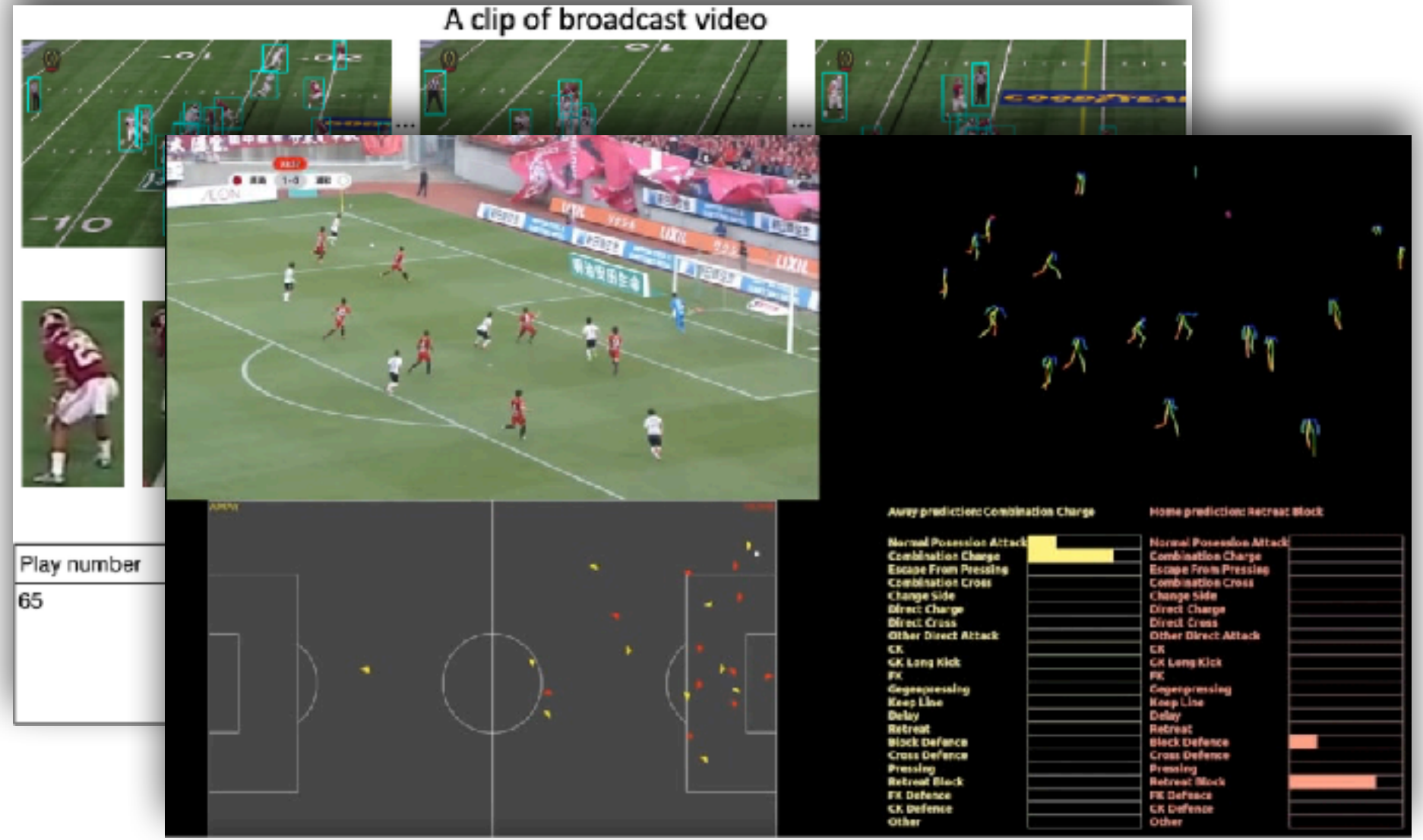



# Operationalizing Masks: Ripeness Detection

**Fruit Sorting: YOLO at 130FPS, ripe versus not ripe**



# Sports Automation and Detection



# Expanding Masks

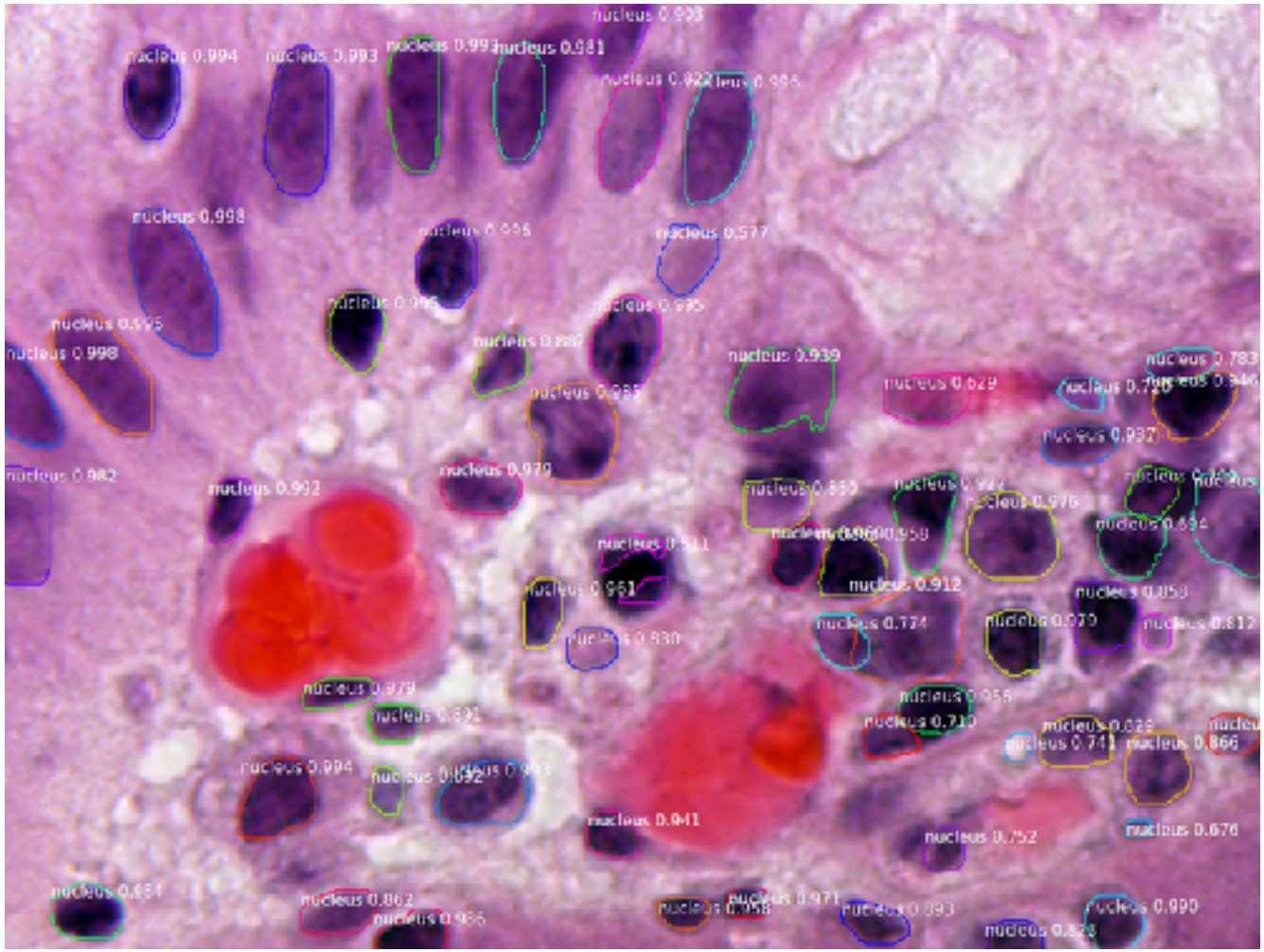
**3D Building Reconstruction (mask becomes 3D point cloud)**

https://github.com/matterport/Mask_RCNN



# Retraining Masks

## Segmenting Nuclei

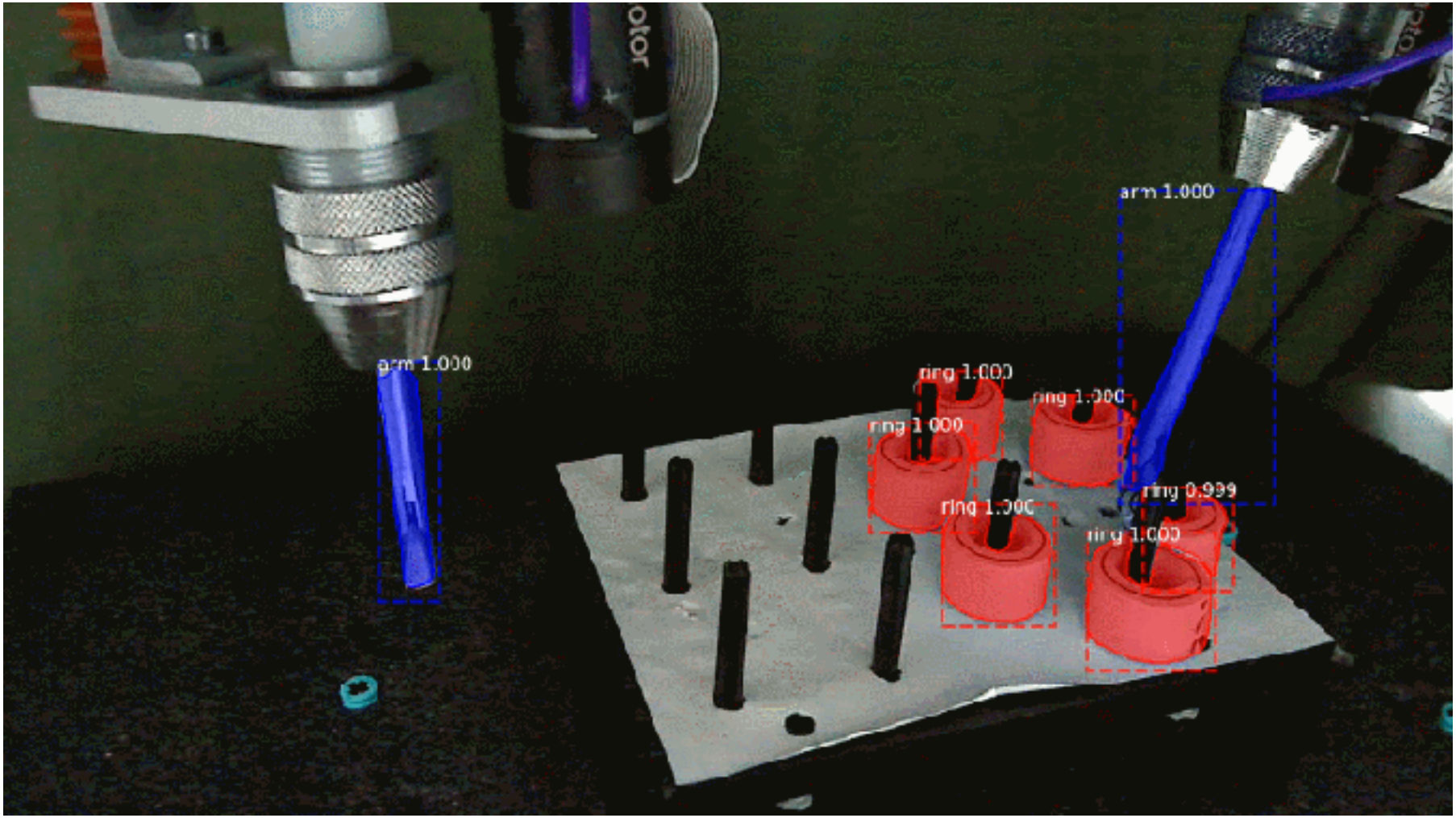


https://github.com/matterport/Mask_RCNN



# Repurposing for Robotics

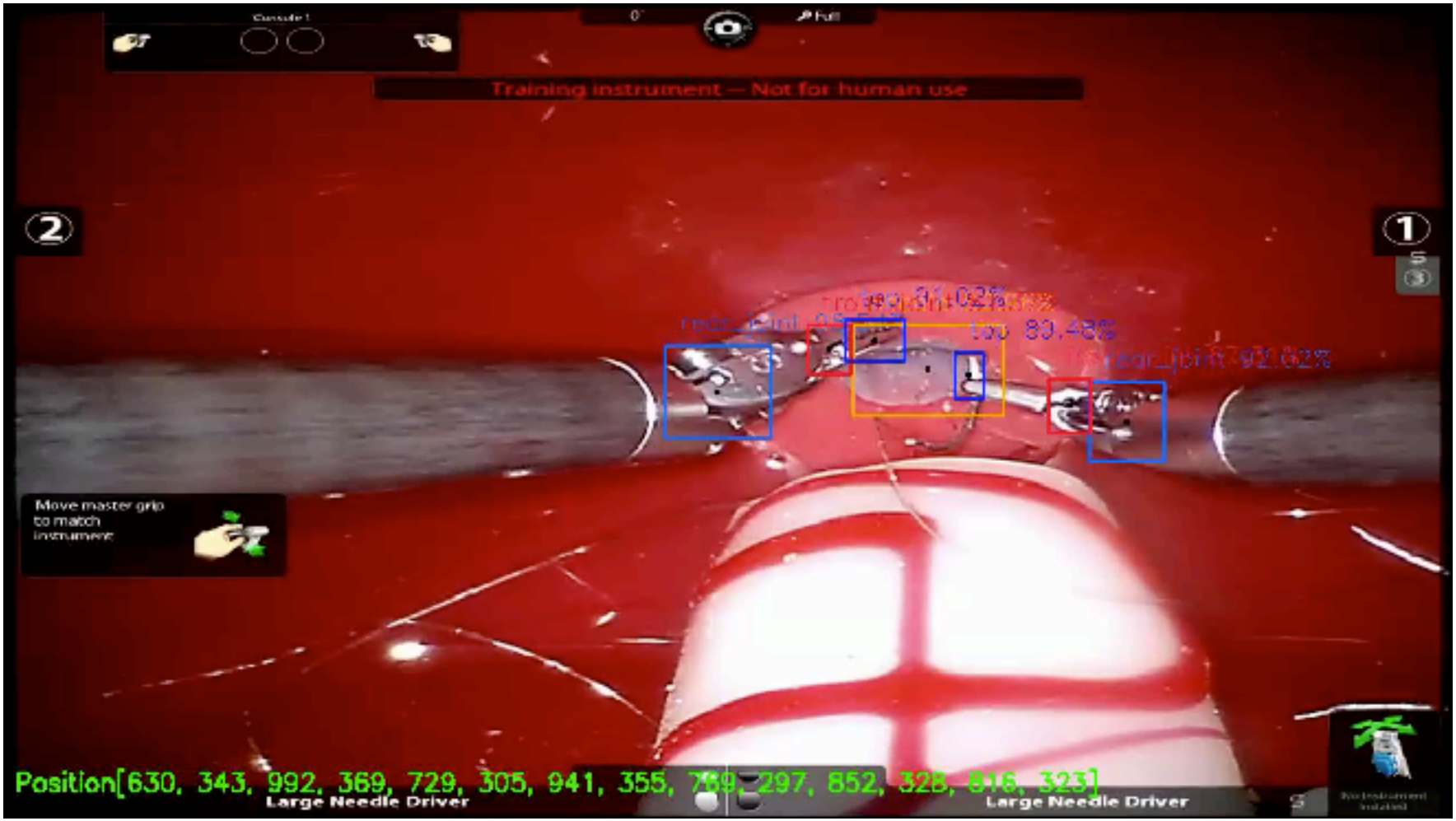
**Robotic Movement for various applications**




https://github.com/matterport/Mask_RCNN



# Robotic Surgery Assessment



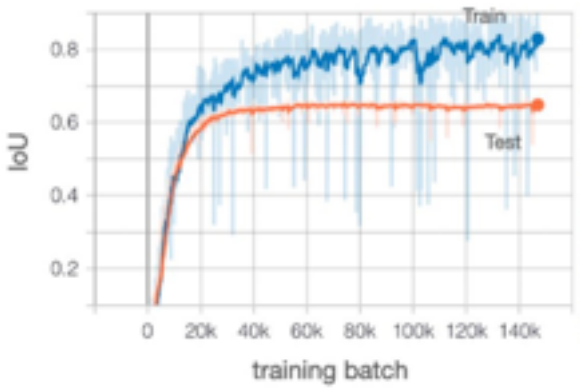

X. Qu, M. El-Saied, J. Gahan, R. Steinberg, and E.C. Larson (2019). "Machine Learning using a Multi-task Convolutional Neural Networks Can Accurately Provide Robotic Skills Assessment." 2019 World Congress of Endourology.

Y. Wang, J. Dai, T. Morgan, M. Elsaid, A. Garbens, X. Qu, R. Steinberg, J. Gahan, and E.C. Larson (2021). "Evaluating Robotic-Assisted Surgery Training Videos with Multi-task Convolutional Neural Networks." Journal of Robotic Surgery (JORS), 2021. Doi: 10.1007/s11701-021-01316-2



# Real Surgery Assessment

- Extended to real surgery also
- But not showing any pictures here




Journal of Robotic Surgery
https://doi.org/10.1007/s11701-023-01657-0

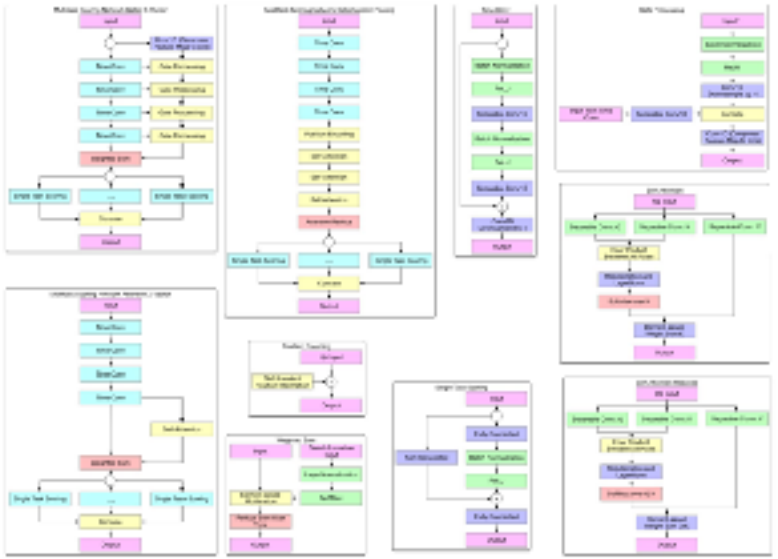
RESEARCH

## Evaluating robotic-assisted partial nephrectomy surgeons with fully convolutional segmentation and multi-task attention networks

Yihao Wang[1] · Zhongjie Wu[1] · Jessica Dai[2] · Tara N. Morgan[2] · Alaina Garbens[2] · Hal Kominsky[2] · Jeffrey Gahan[2] · Eric C. Larson[1]

| Model | Loss | | Attention | | | GEARS | OSATS | Task |
|---|---|---|---|---|---|---|---|---|
| | $\kappa$ | CE | WG | DP | SA | Acc. | Acc. | Acc. |
| Rater–Rater | – | – | – | – | – | 0.71 | 0.75 | N/A |
| WG-$\kappa$ | ✓ | – | ✓ | – | – | 0.35 | 0.32 | 0.80 |
| DP-$\kappa$ | ✓ | – | – | ✓ | – | 0.46 | 0.51 | 0.56 |
| SA-$\kappa$ | ✓ | – | – | – | ✓ | 0.45 | 0.55 | 0.91 |
| WG-CE | – | ✓ | ✓ | – | – | 0.57 | 0.59 | 0.81 |
| DP-CE | – | ✓ | – | ✓ | – | 0.52 | 0.60 | 0.68 |
| SA-CE | – | ✓ | – | – | ✓ | **0.60** | **0.63** | **0.93** |
| SA-CE | Bootstrapping Aggregation | | | | | $0.59 \pm 0.13$ | $0.62 \pm 0.12$ | $0.75 \pm 0.22$ |

Bold in the table indicates the best performing model in each category





# Learning Depth and 3D Shapes

**Mesh R-CNN, Facebook AI January 2020**


https://ai.facebook.com/blog/pushing-state-of-the-art-in-3d-content-understanding/



# In summary

- **Semantic segmentation** through FCN is active research area
  - DeepLabV3+ or GSCNN are excellent choices, but have performance issues
  - EVA: state of the art vanilla ViT for many tasks (1B params)
- **Object detection** is excellent, ready for use in industry (Apple's ObjectDetector uses YOLO variant)
  - Already deployed in a many of Apps
  - At 120+ FPS, supports tracking applications and AR
  - Can backoff to CPU only at about 15+ FPS (on phone)
- **Instance Segmentation** is ready for deployment in a number of areas, and is now better than realtime with good performance
  - Mask-RCNN, YOLACT,
  - SAM is a great choice, if visual prompts are possible

https://arxiv.org/pdf/2211.07636v2.pdf https://github.com/baaivision/EVA/tree/master/EVA-01



Lecture Notes for

# Neural Networks and Machine Learning

FCN: Tracking

**Next Time:**

Stable Diffusion