# *Computer Vision*

Course 12

## Bibliography

**Minaee, Shervin, Yuri Y. Boykov, Fatih Porikli, Antonio J. Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. "Image segmentation using deep learning: A survey." *IEEE transactions on pattern analysis and machine intelligence* (2021).**

**Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. (2022). A survey of modern deep learning based object detection models. Digital Signal Processing, 103514.**

**Cheng, J., Li, H., Li, D., Hua, S., & Sheng, V. S. (2023). A survey on image semantic segmentation using deep learning techniques.**

**https://www.v7labs.com/blog/image-segmentation-guide**

## Deep Learning Image Segmentation

- essential component of many visual understanding systems,
- it involves partitioning images (or video frames) into multiple segments and objects
- applications: medical image analysis (e.g., tumor boundary extraction and measurement of tissue volumes), autonomous vehicles (e.g., navigable surface and pedestrian detection), video surveillance, augmented reality …
- Image segmentation can be formulated as:

1. the problem of classifying pixels with semantic labels (semantic segmentation), - performs pixel-level labeling with a set of object categories (e.g., human, car, tree,…)
2. partitioning of individual objects (instance segmentation) - extends the scope of semantic segmentation by detecting and delineating each object of interest in the image (e.g., individual people).
3. both (panoptic segmentation).

# *Computer Vision*
## *Course 12*

1) Fully convolutional networks

2) Convolutional models with graphical models

3) Encoder-decoder based models

4) Multiscale and pyramid network based models

5) R-CNN based models (for instance segmentation)

6) Dilated convolutional models and DeepLab family

7) Recurrent neural network based models

8) Attention-based models

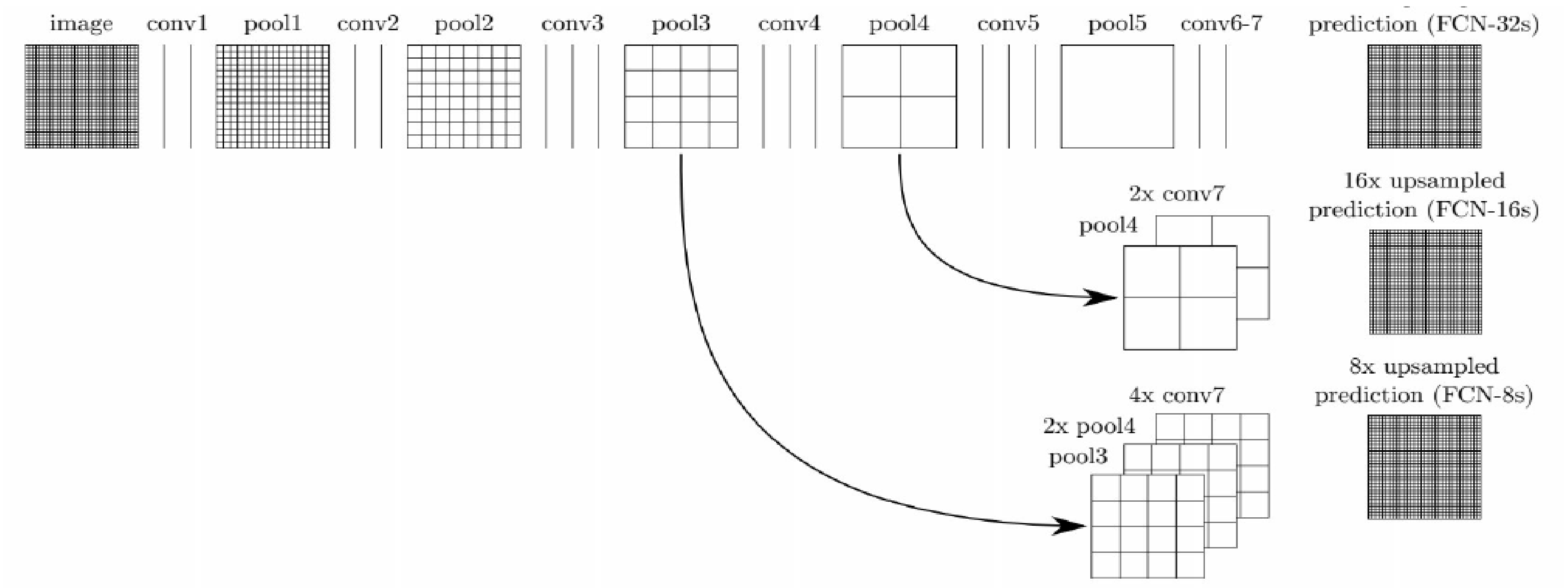
9) Generative models and adversarial training

10) Convolutional models with active contour models

11) Segment anything models (https://segment-anything.com/)

# Fully Convolutional Models

- A Fully Convolutional Network (FCNs), a milestone in DL-based semantic image segmentation includes only convolutional layers,
- outputs a segmentation map whose size is the same as that of the input image.
- To handle arbitrarily-sized images, the authors modified existing CNN architectures, such as VGG16 and GoogLeNet, by removing all fully-connected layers such that the model outputs a spatial segmentation map instead of classification scores.
- FCN learns to make pixel-accurate predictions.
- Through the use of skip connections in which feature maps from the final layers of the model are up-sampled and fused with feature maps of earlier layers, the model combines semantic information (from deep, coarse layers) and appearance information (from shallow, fine layers) in order to produce accurate and detailed segmentations.



Skip connections combine coarse and fine information.

FCNs have been applied to a variety of segmentation problems:

brain tumor segmentation ,
instance aware semantic segmentation,
skin lesion segmentation
iris segmentation [34].

the conventional FCN model has some limitations

- it is too computationally expensive for real-time inference,
- it does not account for global context information in an efficient manner, and
- it is not easily generalizable to 3D images.

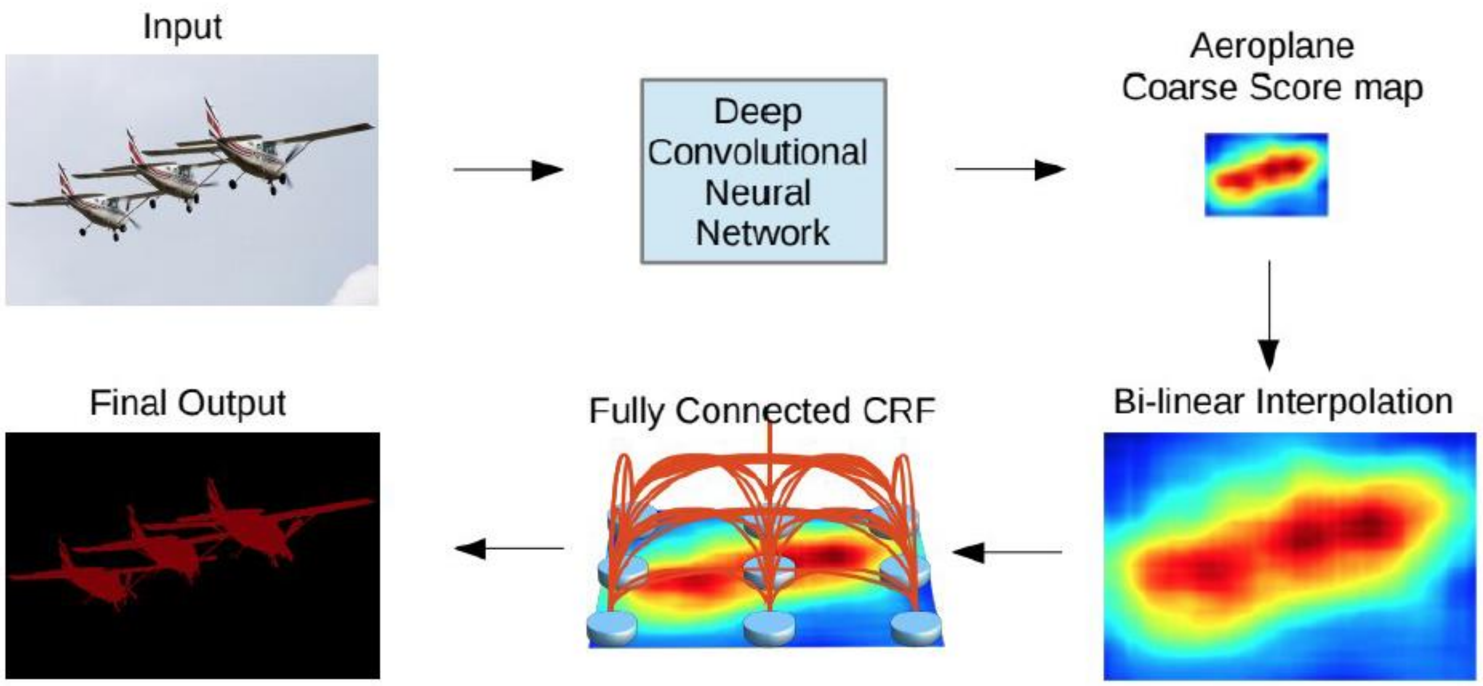
## CNNs With Graphical Models

The FCN ignores potentially useful scene-level semantic context.

To exploit more context, several approaches incorporate into DL architectures probabilistic graphical models, such as Conditional Random Fields (CRFs) and Markov Random Fields (MRFs).

Responses from the later layers of deep CNNs are not sufficiently well localized for accurate object segmentation. Solution: a semantic segmentation algorithm that combines CNNs and fully-connected CRFs or a MRF.

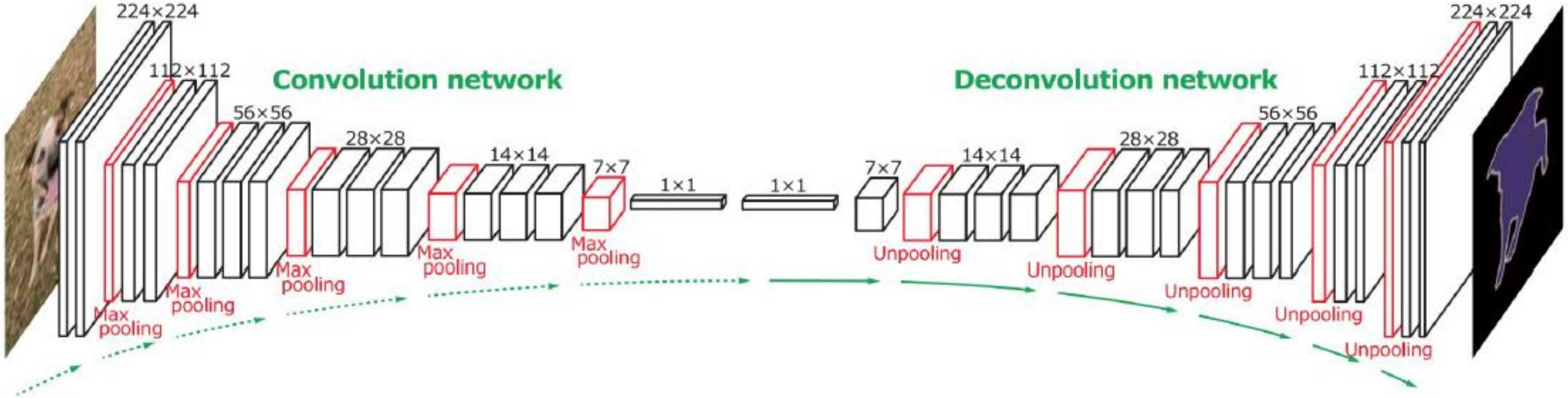
# Computer Vision
Course 12



A CNN+CRF model

# General Image Segmentation

DeConvNet introduces semantic segmentation based on deconvolution and unpooling layers. Their model, consists of two parts, an encoder using convolutional layers adopted from the VGG 16-layer network and a multilayer deconvolutional network that inputs the feature vector and generates a map of pixel-accurate class probabilities. The latter comprises deconvolution and unpooling layers, which identify pixel-wise class labels and predict segmentation masks.

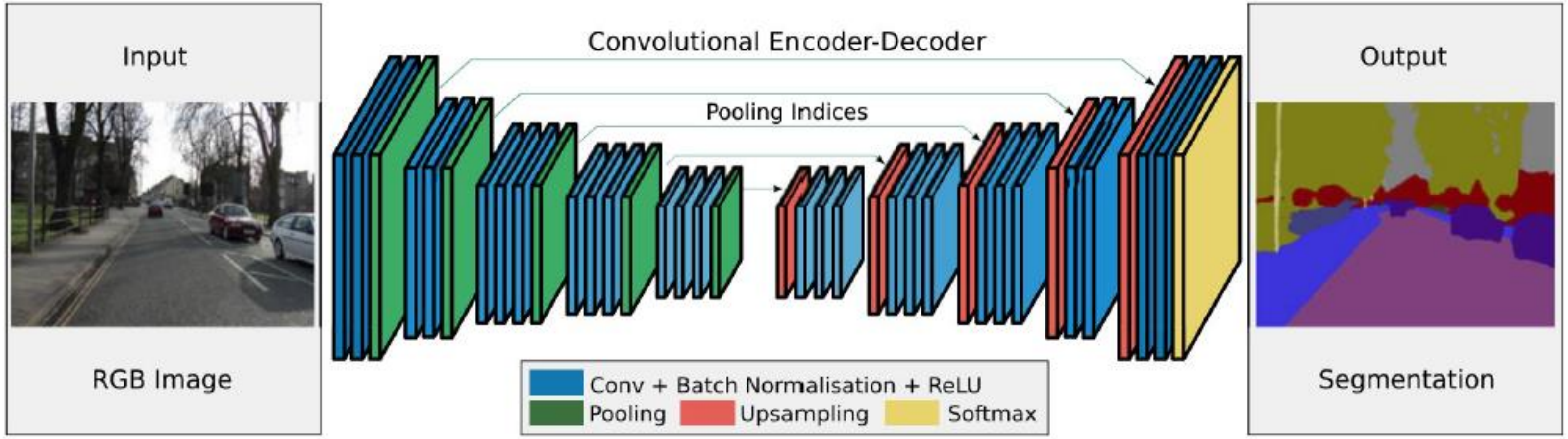


Deconvolutional semantic segmentation

**SegNet**, a fully convolutional encoder-decoder architecture for image segmentation is similar to the deconvolution network.

SegNet consists of an encoder network, which is topologically identical to the 13 convolutional layers of the VGG16 network, and a corresponding decoder network followed by a pixel-wise classification layer.

The main novelty of SegNet is in the way the decoder up-samples its lower-resolution input feature map(s); specifically, using pooling indices computed in the max-pooling step of the corresponding encoder to perform nonlinear up-sampling.



The SegNet model.

**HRNet** addresses the limitation of encoder-decoder based models is the loss of fine-grained image information, due to the loss of resolution through the encoding process.

HRNet maintains high-resolution representations through the encoding process by connecting the high-to-low resolution convolution streams in parallel and repeatedly exchanging the information across resolutions.

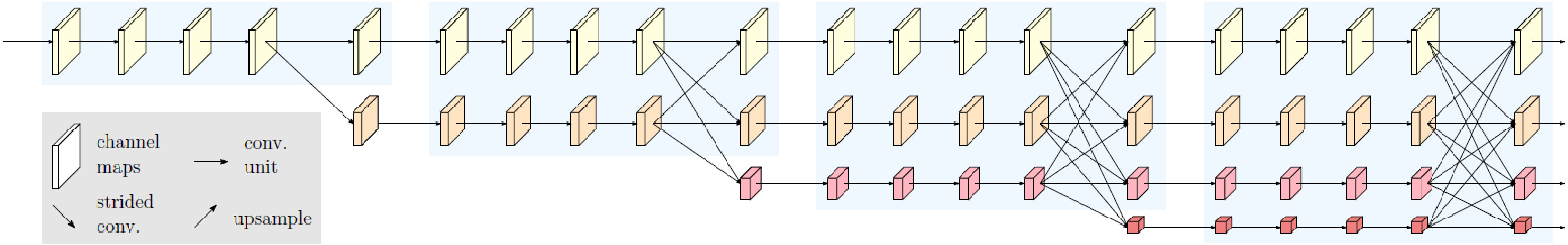
There are four stages:

- the 1st stage consists of high resolution convolutions,

- the 2nd/3rd/4th stage repeats 2-resolution/3-resolution/4-resolution blocks.

Several recent semantic segmentation models use HRNet as a backbone.

# Computer Vision
# Course 12



The HRNet architecture.

## Medical and Biomedical Image Segmentation

Several models inspired by FCNs and encoder-decoder networks were initially developed for medical/biomedical

image segmentation, but are now also being used outside the medical domain.

**U-Net** was designed for efficiently segmenting biological microscopy images.

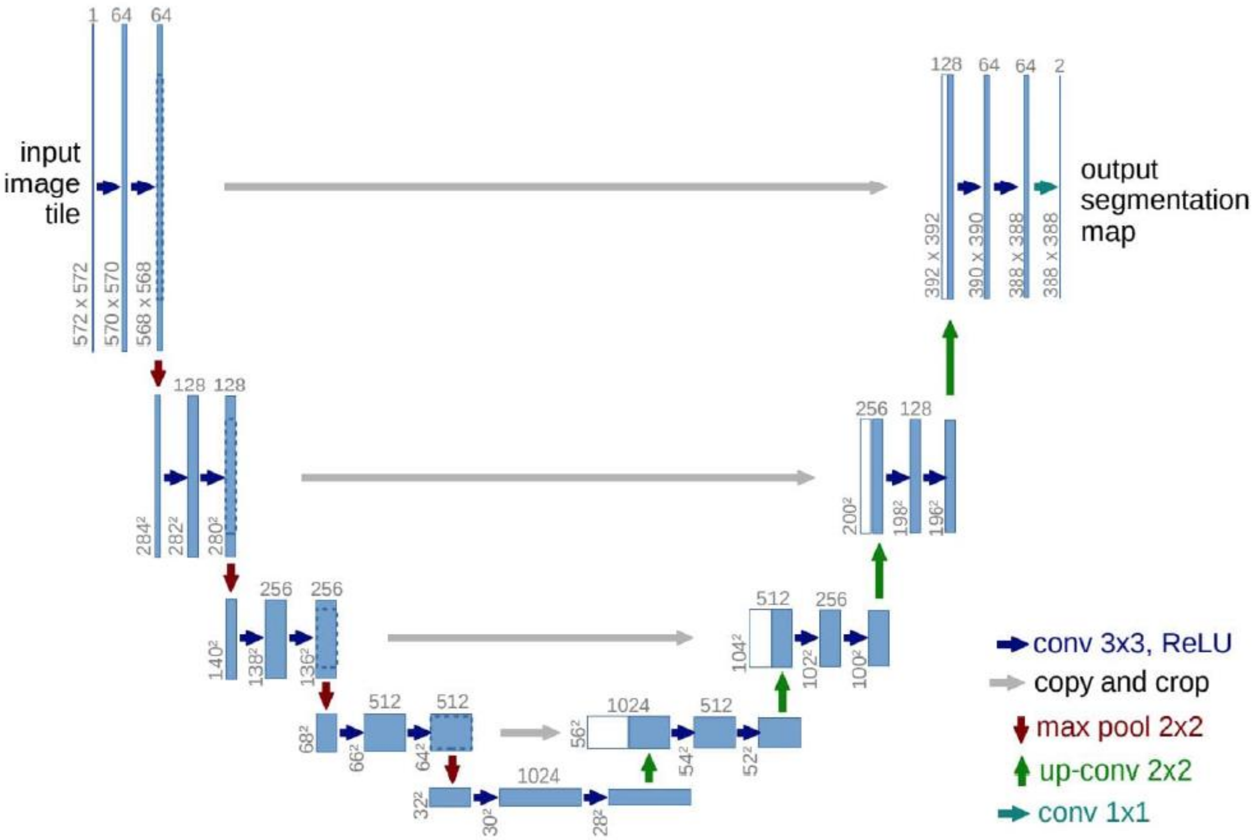
The U-Net architecture comprises two parts, a contracting path to capture context, and a symmetric expanding path that enables precise localization.

The U-Net training strategy relies on the use of data augmentation to learn effectively from very few annotated images.

It was trained on 30 transmitted light microscopy images, and it won the ISBI cell tracking challenge 2015 by a large  margin.

Various extensions of U-Net have been developed for different kinds of images and problem domains (road segmentation, 3D images)

# Computer Vision
## Course 12



The U-Net model.

**V-Net** was introduced for 3D medical image segmentation, and is based on the FCN model. A new loss function based on the Dice coefficient was used, enabling the model to deal with situations in which there is a strong imbalance between the number of voxels in the foreground and background.

The network was trained end-to-end on MRI images of the prostate and learns to predict segmentation for the whole volume at once.

# Multiscale and Pyramid Network Based Models

**Feature Pyramid Network (FPN)** was developed for object detection but was also applied to segmentation.
The inherent multiscale, pyramidal hierarchy of deep CNNs was used to construct feature pyramids with marginal extra cost. To merge low and high resolution features, the FPN is composed of a bottom-up pathway, a top-down pathway and lateral connections. The concatenated feature maps are then processed by a 3x3 convolution to produce the output of each stage. Finally, each stage of the top-down pathway generates a prediction to detect an object.
For image segmentation, two multilayer perceptrons (MLPs) are employed to generate the masks.
**Pyramid Scene Parsing Network (PSPN),** a multiscale network to better learn the global context representation of a scene. Multiple patterns are extracted from the input image using a residual network (ResNet) as a feature extractor, with a dilated network. These feature maps are then fed into a pyramid pooling module to distinguish patterns of different scales. They are pooled at four different scales, each one corresponding to a pyramid level, and processed by a 11 convolutional layer to reduce their dimensions. The outputs of the pyramid levels are up-sampled and concatenated with the initial feature maps to capture both local and global context information.
Finally, a convolutional layer is used to generate the pixelwise predictions.

## R-CNN Based Models

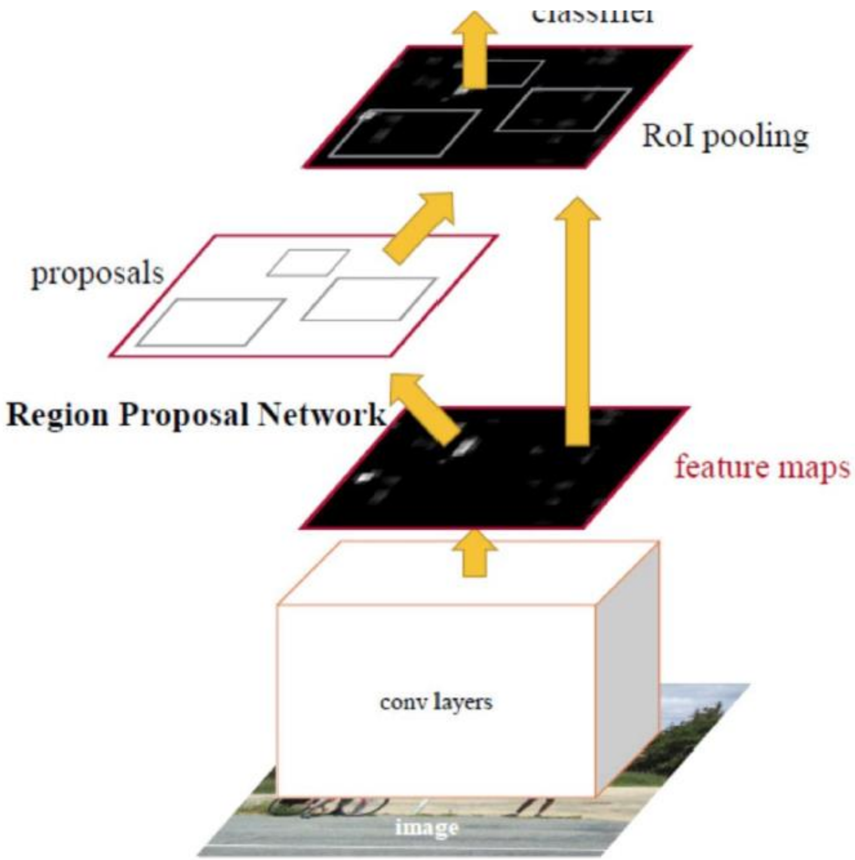
The Regional CNN (R-CNN) and its extensions have proven successful in object detection applications.

**Faster R-CNN** architecture uses a region proposal network (RPN) that proposes bounding box candidates.

The RPN extracts a Region of Interest (RoI), and an RoIPool layer computes features from these proposals to infer the bounding box coordinates and class of the object.

Some extensions of R-CNN have been used to address the instance segmentation problem;

# Computer Vision
Course 12



. Faster R-CNN architecture.

**Mask R-CNN** efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance.

It is a Faster R-CNN with 3 output branches

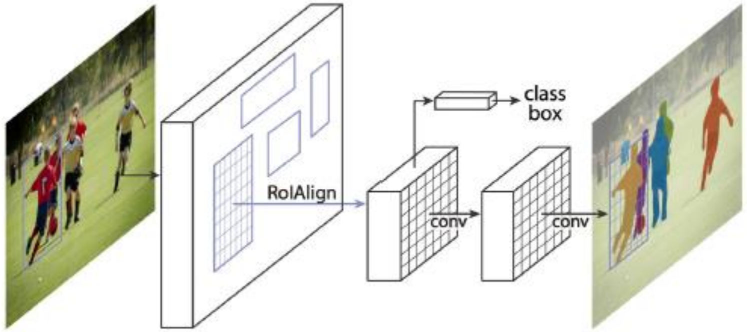
-the first computes the bounding box coordinates,

-the second computes the associated classes, and

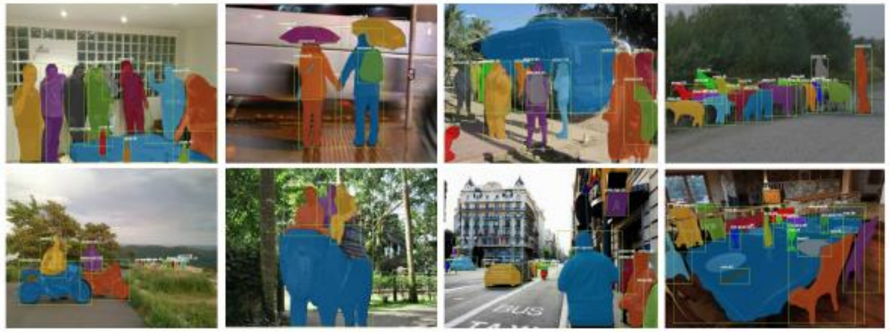
-the third computes the binary mask to segment the object.

The Mask R-CNN loss function combines the losses of the bounding box coordinates, The predicted class, and the segmentation mask, and trains all of them jointly.

# Computer Vision
Course 12



Mask R-CNN architecture.

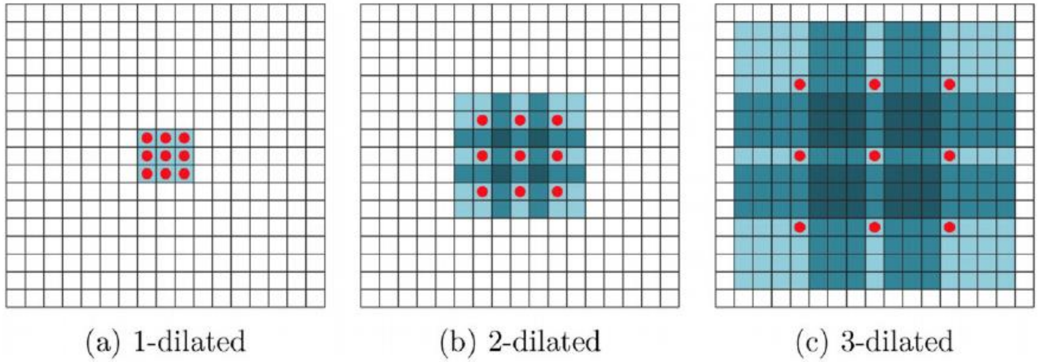


Mask R-CNN instance segmentation results.

## Dilated Convolutional Models

Dilated (a.k.a. “atrous”) convolution introduces to convolutional layers another parameter, the dilation rate.

For example, a 3x3 kernel with a dilation rate of 2 will have the same size receptive field as a 5x5 kernel while using only 9 parameters, thus enlarging the receptive field with no increase in computational cost.


(a) 1-dilated (b) 2-dilated (c) 3-dilated

Dilated convolution. A $3 \times 3$ kernel at different dilation rates.

Dilated convolutions have been popular in the field of real-time segmentation.

Some of the most important include:

- the DeepLab family
- multiscale context aggregation
- Dense Ups-ampling Convolution and
- Hybrid Dilated Convolution (DUC-HDC),
- densely connected Atrous Spatial Pyramid Pooling (DenseASPP) [79],
- Efficient Network (ENet)

## Recurrent Neural Networks (RNNs) and the LSTM

RNNs are commonly used to process sequential data, such as speech, text, videos, and time-series.

RNNs are typically problematic for long sequences as they cannot capture long-term dependencies in many real-world applications and often suffer from gradient vanishing or exploding problems.

However, a type of RNN known as the Long Short-Term Memory (LSTM) is designed to avoid these issues.

The LSTM architecture includes three gates (input gate, output gate, and forget gate) that regulate the flow of information into and out of a memory cell that stores values over arbitrary time intervals

## RNN Based Models

RNNs are useful in modeling the short/long term dependencies among pixels to (potentially) improve the estimation of the segmentation map.

Using RNNs, pixels may be linked together and processed sequentially to model global contexts and improve semantic segmentation.

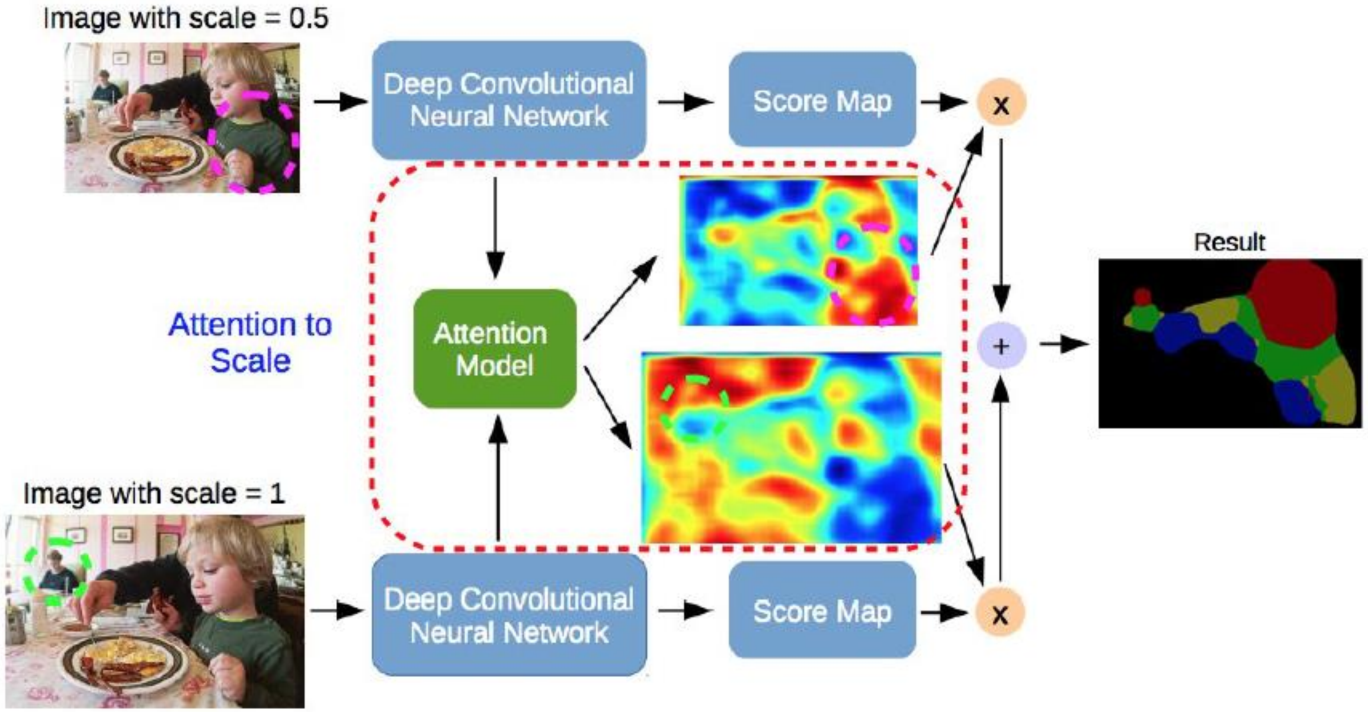
ReSeg is mainly based on ReNet, which was developed for image classification.

Each ReNet layer is composed of four RNNs that sweep the image horizontally and vertically in both directions, encoding patches/activations, and providing relevant global information.

To perform image segmentation with the ReSeg model, ReNet layers are stacked atop pretrained VGG-16 convolutional layers, which extract generic local features, and are then followed by up-sampling layers to recover the original image resolution in the final predictions.

A drawback of RNN-based models is that they will generally be slower than their CNN counterparts as their sequential nature is not amenable to parallelization

# Computer Vision
Course 12



Attention-based semantic segmentation model.

## Attention-Based Models

Attention mechanism can learn to softly weight multiscale features at each pixel location.

They adapt a powerful semantic segmentation model and jointly train it with multiscale images and the attention model

The model assigns large weights to the person (green dashed circle) in the background for features from scale 1.0 as well as on the large child (magenta dashed circle) for features from scale 0.5.

The attention mechanism enables the model to assess the importance of features at different positions and scales, and it outperforms average and max pooling

# *Computer Vision*
*Course 12*

**Reverse Attention Network** (RAN) proposes an architecture for semantic segmentation that also applies reverse

The RAN network performs the direct and reverse-attention learning processes simultaneously.

**Pyramid Attention Network** for semantic segmentation, exploits global contextual information for semantic segmentation

**OCNet** employs an object context pooling inspired by self-attention mechanism,

**ResNeSt: Split-Attention Networks**,

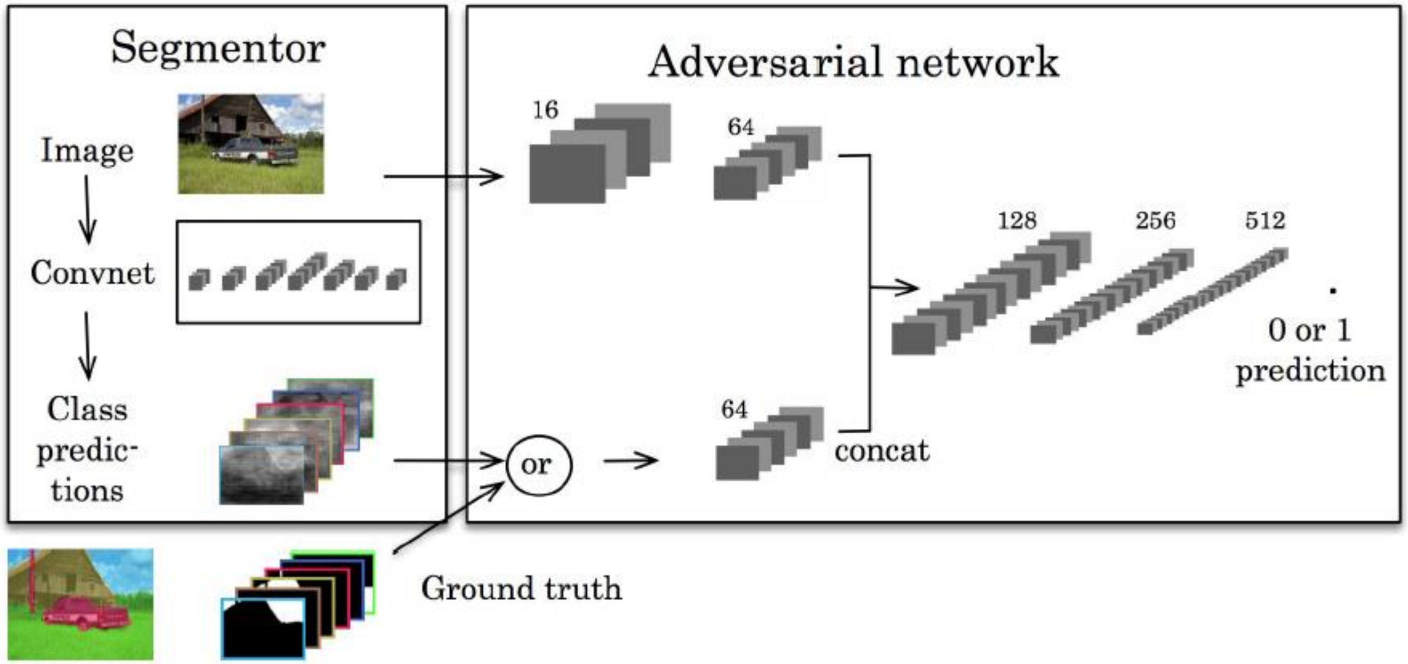
**Height-driven Attention Networks**,

**Expectation-Maximization Attention (EMANet)**,

**Criss-Cross Attention Network (CCNet)**

**Discriminative Feature Network (DFN)**

# Generative Models and Adversarial Training



The GAN for semantic segmentation.

This idea uses an adversarial training approach for semantic segmentation in which they trained a convolutional semantic segmentation network, along with an adversarial network that discriminates between ground-truth segmentation maps and those generated by the segmentation network.

Other approaches:

- semi-weakly supervised semantic segmentation using GANs. This model consists of a generator network providing extra training examples to a multiclass classifier, acting as discriminator in the GAN framework, that assigns sample a label from the possible label classes or marks it as a fake sample (extra class).

- an FCN discriminator to differentiate the predicted probability maps from the ground truth segmentation distribution, considering the spatial resolution.

- an FCN as the segmentor to generate segmentation label maps, and proposed a novel adversarial critic network with multi-scale L1 loss function to force the critic and segmentor to learn both global and local features that capture long and short range spatial relationships between pixels.

**CNN Models With Active Contour Models**

This type of approach explore the links between FCNs and Active Contour Models (ACMs)

One approach is to formulate new loss functions that are inspired by ACM principles.

- a supervised loss layer that incorporates the area and size information of the predicted masks during training of an FCN and tackled the problem of ventricle segmentation in cardiac MRI. Similarly,

- unsupervised loss function based on morphological active contours without edges for microvascular image segmentation.

A different approach initially sought to utilize the ACM merely as a post-processor of the output of an FCN and several efforts attempted modest co-learning by pre-training the FCN.

- Deep Active Contours , Deep Active Lesion Segmentation (DALS), Deep Structured Active Contours (DSAC), Trainable Deep Active Contours (TDAC)

# DATASETS

Datasets can be grouped into 3 categories

2D (pixel) images,

2.5D RGB-D (color+depth) images, and

3D (voxel) images.

**Data augmentation** is often used to increase the number of labeled samples, especially for small datasets such as those in the medical imaging domain. A set of transformations

is applied either in the data space, or feature space, or both (i.e., both the image and the segmentation map).

Typical transformations:  translation, reflection, rotation, warping, scaling, color space shifting, cropping, and projections onto principal components.

Data augmentation  can lead to faster convergence, decreasing the chance of over-fitting, and enhancing generalization. For some small datasets, data augmentation has been shown to boost model performance by more than 20%.

## 2D Image Datasets

- PASCAL Visual Object Classes (VOC)
- PASCAL Context
- Microsoft Common Objects in Context (MS COCO)
- Cityscapes
- ADE20K /MIT Scene Parsing (SceneParse150)
- SiftFlow
- Stanford Background
- Berkeley Segmentation Dataset (BSD)
- Youtube-Objects
- CamVid
- KITTI, …

## 2.5D Image Datasets

- NYU-Depth V2
- SUN-3D
- SUN RGB-D
- ScanNet
- Stanford 2D-3D
- UW RGB-D Object

## 2.5D Image Datasets

- NYU-Depth V2
- SUN-3D
- SUN RGB-D
- ScanNet
- Stanford 2D-3D
- UW RGB-D Object

## Metrics for Image Segmentation Models

- **Pixel accuracy** is the ratio of properly classified pixels divided by the total number of pixels.
- **Mean Pixel Accuracy (MPA)** is an extension of PA, in which the ratio of correct pixels is computed in a per-class manner and then averaged over the total number of classes
- **Intersection over Union (IoU)**, or the **Jaccard Index**, is defined as the area of intersection between the predicted segmentation map A and the ground truth map B, divided by the area of the union between the two maps, and ranges between 0 and 1
- **Mean-IoU** is defined as the average IoU over all classes
- **Precision / Recall / F1 score**
- **Dice coefficient**, commonly used in medical image analysis, can be defined as twice the overlap area of the predicted and ground-truth maps divided by the total number of pixels

# Computer Vision
Course 12

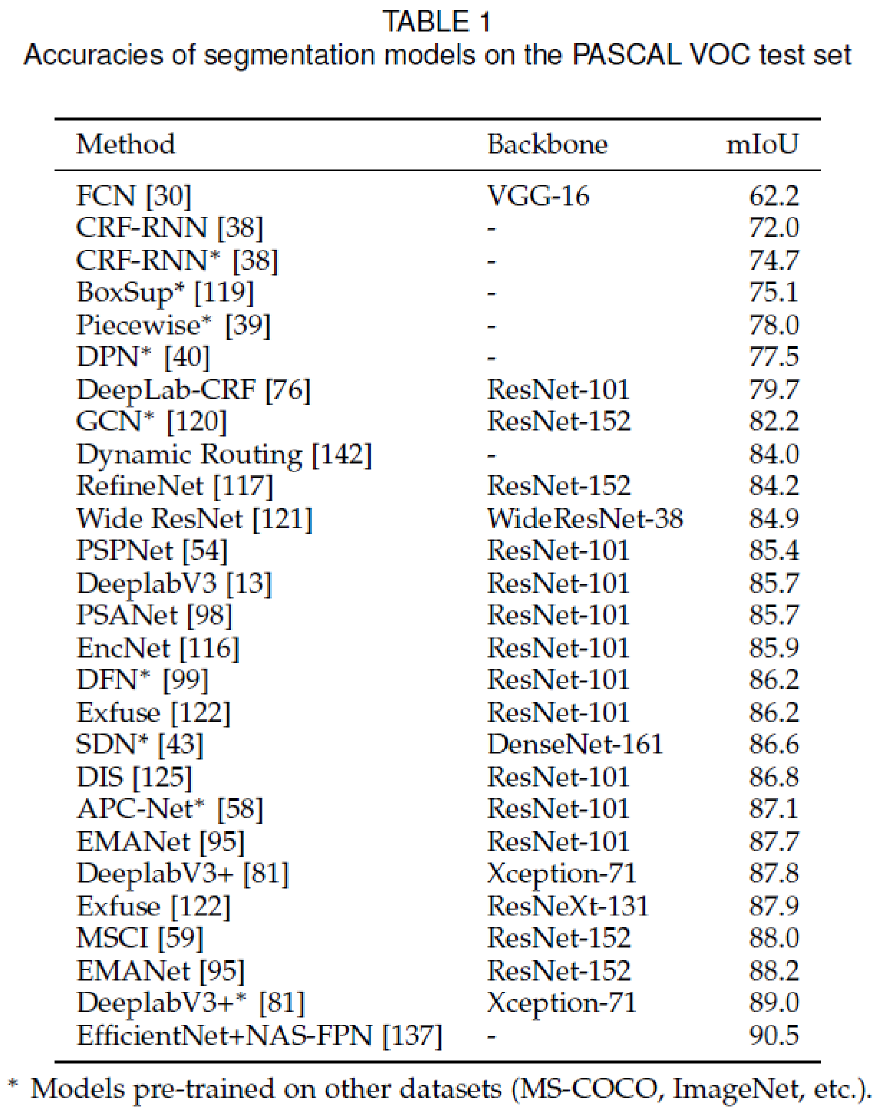


TABLE 1
Accuracies of segmentation models on the PASCAL VOC test set

| Method | Backbone | mIoU |
|---|---|---|
| FCN [30] | VGG-16 | 62.2 |
| CRF-RNN [38] | - | 72.0 |
| CRF-RNN* [38] | - | 74.7 |
| BoxSup* [119] | - | 75.1 |
| Piecewise* [39] | - | 78.0 |
| DPN* [40] | - | 77.5 |
| DeepLab-CRF [76] | ResNet-101 | 79.7 |
| GCN* [120] | ResNet-152 | 82.2 |
| Dynamic Routing [142] | - | 84.0 |
| RefineNet [117] | ResNet-152 | 84.2 |
| Wide ResNet [121] | WideResNet-38 | 84.9 |
| PSPNet [54] | ResNet-101 | 85.4 |
| DeeplabV3 [13] | ResNet-101 | 85.7 |
| PSANet [98] | ResNet-101 | 85.7 |
| EncNet [116] | ResNet-101 | 85.9 |
| DFN* [99] | ResNet-101 | 86.2 |
| Exfuse [122] | ResNet-101 | 86.2 |
| SDN* [43] | DenseNet-161 | 86.6 |
| DIS [125] | ResNet-101 | 86.8 |
| APC-Net* [58] | ResNet-101 | 87.1 |
| EMANet [95] | ResNet-101 | 87.7 |
| DeeplabV3+ [81] | Xception-71 | 87.8 |
| Exfuse [122] | ResNeXt-131 | 87.9 |
| MSCI [59] | ResNet-152 | 88.0 |
| EMANet [95] | ResNet-152 | 88.2 |
| DeeplabV3+* [81] | Xception-71 | 89.0 |
| EfficientNet+NAS-FPN [137] | - | 90.5 |

* Models pre-trained on other datasets (MS-COCO, ImageNet, etc.).

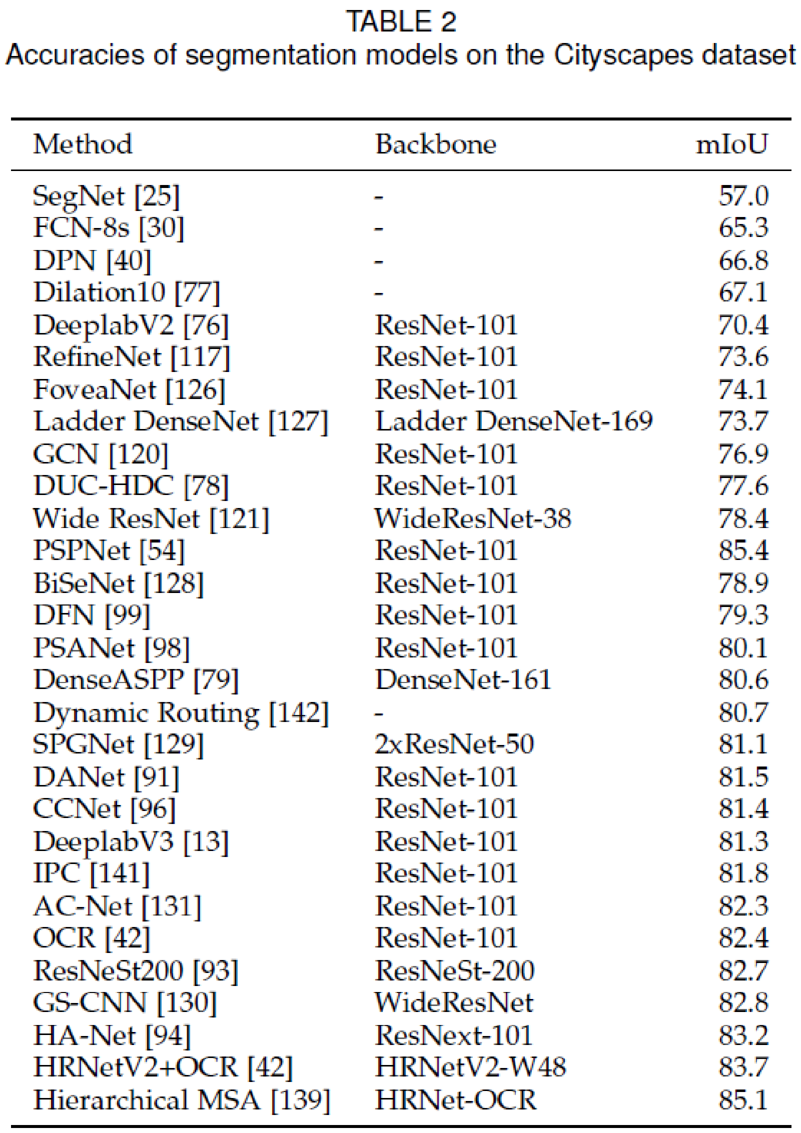
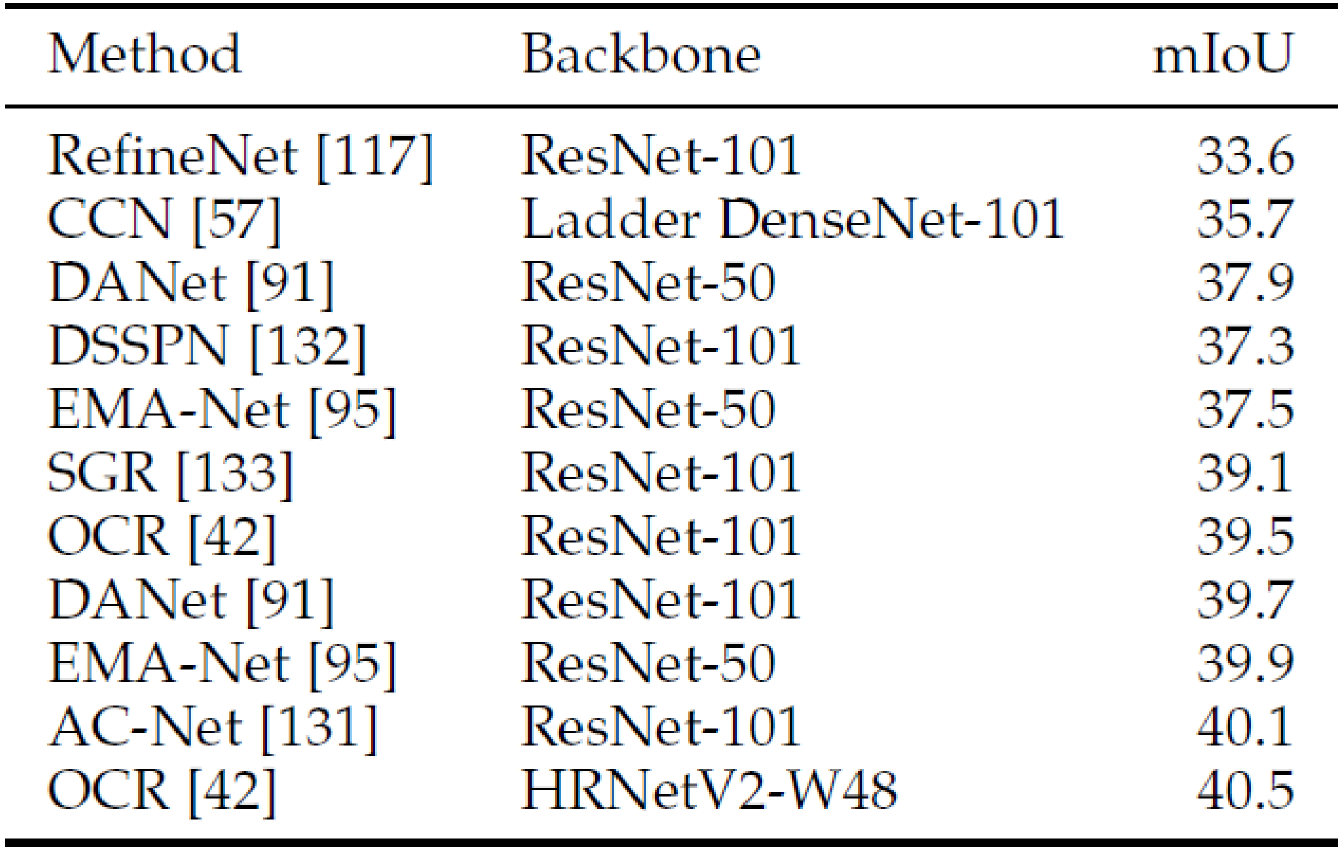


TABLE 2
Accuracies of segmentation models on the Cityscapes dataset

| Method | Backbone | mIoU |
|---|---|---|
| SegNet [25] | - | 57.0 |
| FCN-8s [30] | - | 65.3 |
| DPN [40] | - | 66.8 |
| Dilation10 [77] | - | 67.1 |
| DeeplabV2 [76] | ResNet-101 | 70.4 |
| RefineNet [117] | ResNet-101 | 73.6 |
| FoveaNet [126] | ResNet-101 | 74.1 |
| Ladder DenseNet [127] | Ladder DenseNet-169 | 73.7 |
| GCN [120] | ResNet-101 | 76.9 |
| DUC-HDC [78] | ResNet-101 | 77.6 |
| Wide ResNet [121] | WideResNet-38 | 78.4 |
| PSPNet [54] | ResNet-101 | 85.4 |
| BiSeNet [128] | ResNet-101 | 78.9 |
| DFN [99] | ResNet-101 | 79.3 |
| PSANet [98] | ResNet-101 | 80.1 |
| DenseASPP [79] | DenseNet-161 | 80.6 |
| Dynamic Routing [142] | - | 80.7 |
| SPGNet [129] | 2xResNet-50 | 81.1 |
| DANet [91] | ResNet-101 | 81.5 |
| CCNet [96] | ResNet-101 | 81.4 |
| DeeplabV3 [13] | ResNet-101 | 81.3 |
| IPC [141] | ResNet-101 | 81.8 |
| AC-Net [131] | ResNet-101 | 82.3 |
| OCR [42] | ResNet-101 | 82.4 |
| ResNeSt200 [93] | ResNeSt-200 | 82.7 |
| GS-CNN [130] | WideResNet | 82.8 |
| HA-Net [94] | ResNext-101 | 83.2 |
| HRNetV2+OCR [42] | HRNetV2-W48 | 83.7 |
| Hierarchical MSA [139] | HRNet-OCR | 85.1 |

TABLE 3
Accuracies of segmentation models on the MS COCO stuff dataset



| Method | Backbone | mIoU |
|---|---|---|
| RefineNet [117] | ResNet-101 | 33.6 |
| CCN [57] | Ladder DenseNet-101 | 35.7 |
| DANet [91] | ResNet-50 | 37.9 |
| DSSPN [132] | ResNet-101 | 37.3 |
| EMA-Net [95] | ResNet-50 | 37.5 |
| SGR [133] | ResNet-101 | 39.1 |
| OCR [42] | ResNet-101 | 39.5 |
| DANet [91] | ResNet-101 | 39.7 |
| EMA-Net [95] | ResNet-50 | 39.9 |
| AC-Net [131] | ResNet-101 | 40.1 |
| OCR [42] | HRNetV2-W48 | 40.5 |

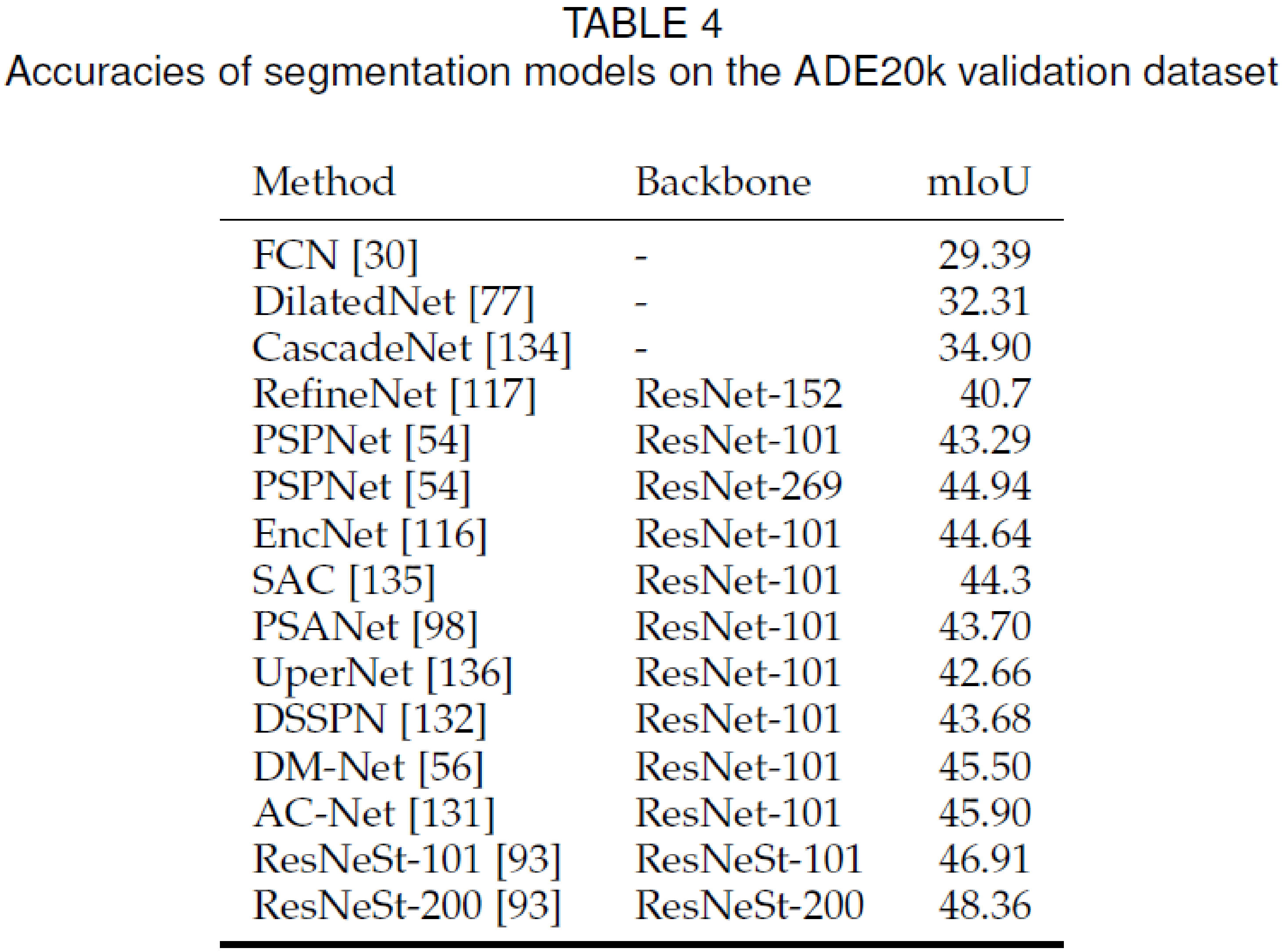


TABLE 4
Accuracies of segmentation models on the ADE20k validation dataset

| Method | Backbone | mIoU |
|---|---|---|
| FCN [30] | - | 29.39 |
| DilatedNet [77] | - | 32.31 |
| CascadeNet [134] | - | 34.90 |
| RefineNet [117] | ResNet-152 | 40.7 |
| PSPNet [54] | ResNet-101 | 43.29 |
| PSPNet [54] | ResNet-269 | 44.94 |
| EncNet [116] | ResNet-101 | 44.64 |
| SAC [135] | ResNet-101 | 44.3 |
| PSANet [98] | ResNet-101 | 43.70 |
| UperNet [136] | ResNet-101 | 42.66 |
| DSSPN [132] | ResNet-101 | 43.68 |
| DM-Net [56] | ResNet-101 | 45.50 |
| AC-Net [131] | ResNet-101 | 45.90 |
| ResNeSt-101 [93] | ResNeSt-101 | 46.91 |
| ResNeSt-200 [93] | ResNeSt-200 | 48.36 |

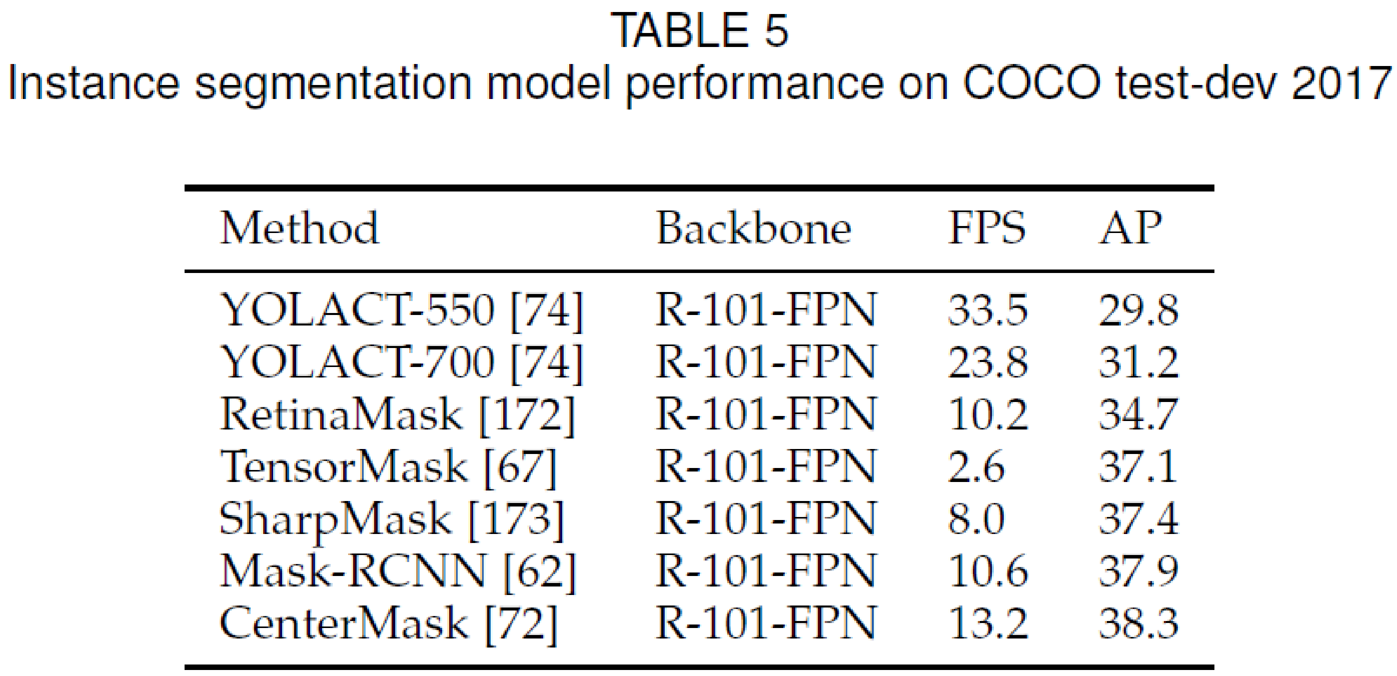


TABLE 5
Instance segmentation model performance on COCO test-dev 2017

| Method | Backbone | FPS | AP |
|---|---|---|---|
| YOLACT-550 [74] | R-101-FPN | 33.5 | 29.8 |
| YOLACT-700 [74] | R-101-FPN | 23.8 | 31.2 |
| RetinaMask [172] | R-101-FPN | 10.2 | 34.7 |
| TensorMask [67] | R-101-FPN | 2.6 | 37.1 |
| SharpMask [173] | R-101-FPN | 8.0 | 37.4 |
| Mask-RCNN [62] | R-101-FPN | 10.6 | 37.9 |
| CenterMask [72] | R-101-FPN | 13.2 | 38.3 |

Computer Vision
Course 12

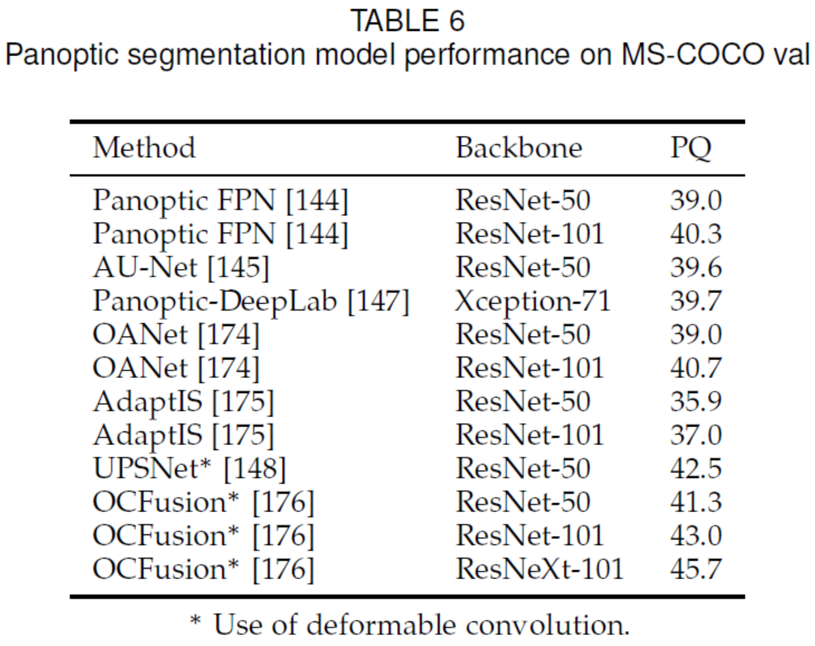


TABLE 6
Panoptic segmentation model performance on MS-COCO val

| Method | Backbone | PQ |
|---|---|---|
| Panoptic FPN [144] | ResNet-50 | 39.0 |
| Panoptic FPN [144] | ResNet-101 | 40.3 |
| AU-Net [145] | ResNet-50 | 39.6 |
| Panoptic-DeepLab [147] | Xception-71 | 39.7 |
| OANet [174] | ResNet-50 | 39.0 |
| OANet [174] | ResNet-101 | 40.7 |
| AdaptIS [175] | ResNet-50 | 35.9 |
| AdaptIS [175] | ResNet-101 | 37.0 |
| UPSNet* [148] | ResNet-50 | 42.5 |
| OCFusion* [176] | ResNet-50 | 41.3 |
| OCFusion* [176] | ResNet-101 | 43.0 |
| OCFusion* [176] | ResNeXt-101 | 45.7 |

* Use of deformable convolution.

*Computer Vision*
*Course 12*

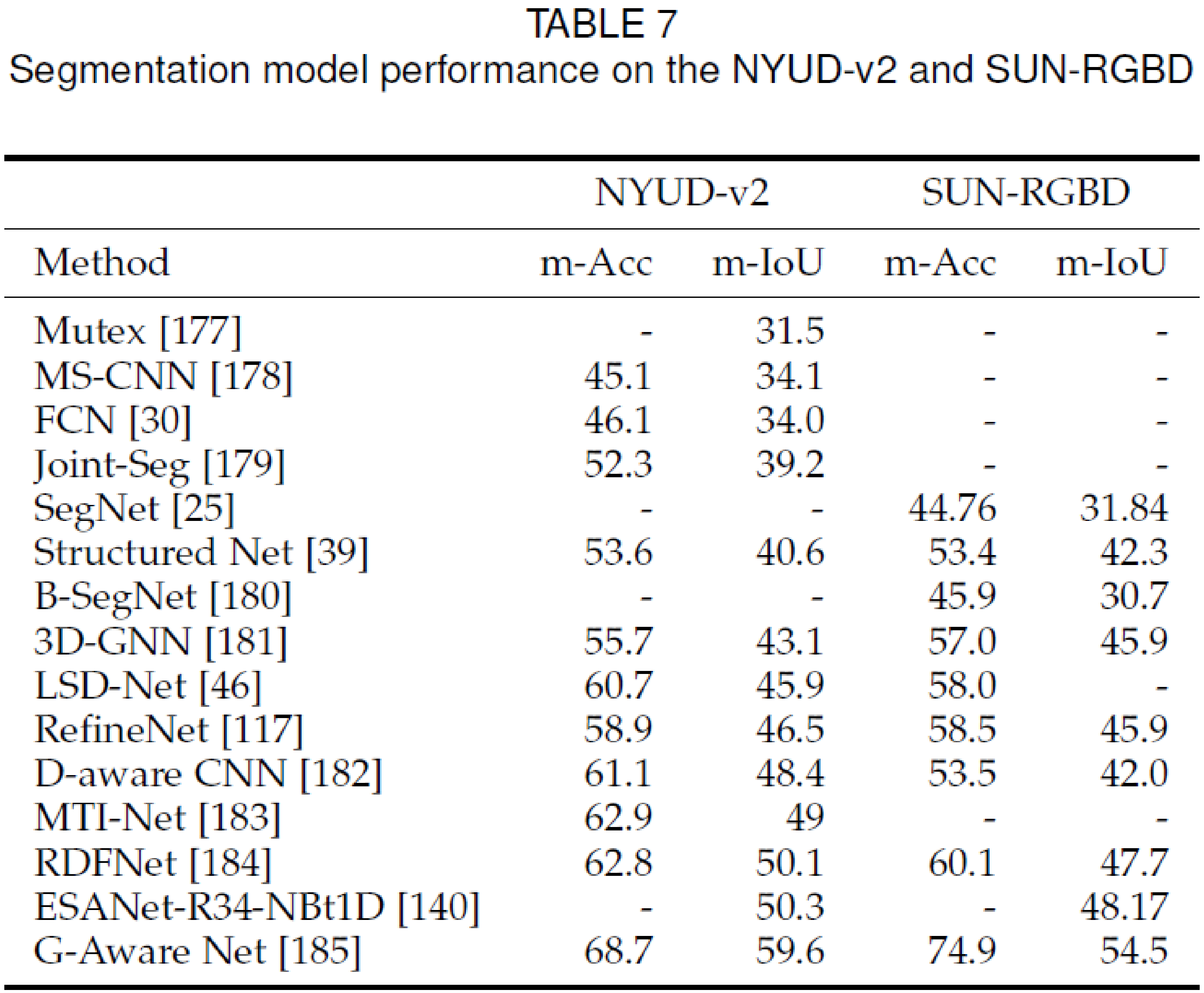
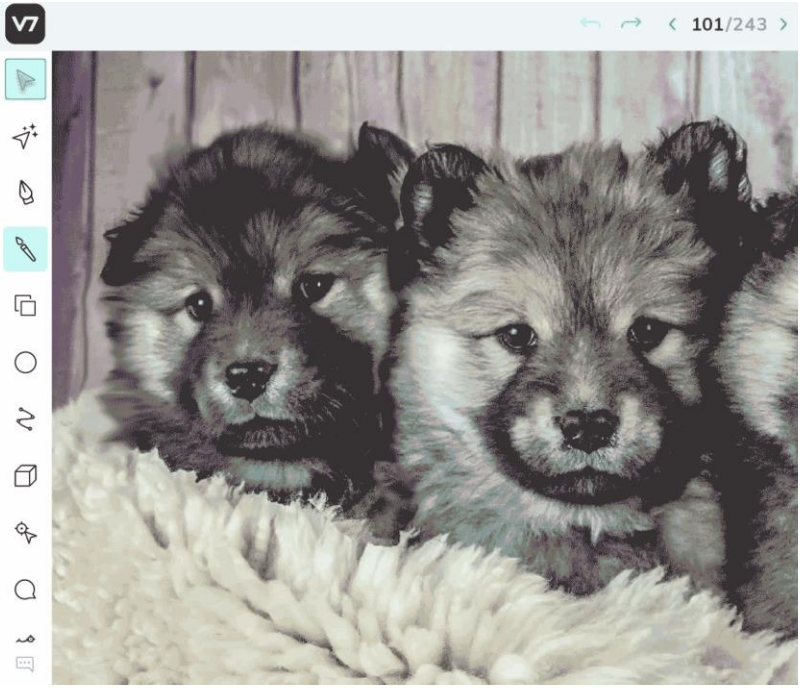


TABLE 7
Segmentation model performance on the NYUD-v2 and SUN-RGBD

| | NYUD-v2 | | SUN-RGBD | |
|---|---|---|---|---|
| Method | m-Acc | m-IoU | m-Acc | m-IoU |
| Mutex [177] | - | 31.5 | - | - |
| MS-CNN [178] | 45.1 | 34.1 | - | - |
| FCN [30] | 46.1 | 34.0 | - | - |
| Joint-Seg [179] | 52.3 | 39.2 | - | - |
| SegNet [25] | - | - | 44.76 | 31.84 |
| Structured Net [39] | 53.6 | 40.6 | 53.4 | 42.3 |
| B-SegNet [180] | - | - | 45.9 | 30.7 |
| 3D-GNN [181] | 55.7 | 43.1 | 57.0 | 45.9 |
| LSD-Net [46] | 60.7 | 45.9 | 58.0 | - |
| RefineNet [117] | 58.9 | 46.5 | 58.5 | 45.9 |
| D-aware CNN [182] | 61.1 | 48.4 | 53.5 | 42.0 |
| MTI-Net [183] | 62.9 | 49 | - | - |
| RDFNet [184] | 62.8 | 50.1 | 60.1 | 47.7 |
| ESANet-R34-NBt1D [140] | - | 50.3 | - | 48.17 |
| G-Aware Net [185] | 68.7 | 59.6 | 74.9 | 54.5 |

## Deep Learning Object Detection

- localization and classification of objects contained in an image or video
- object detection comes down to drawing bounding boxes around detected objects which allow us to locate them in a given scene (or how they move through it).



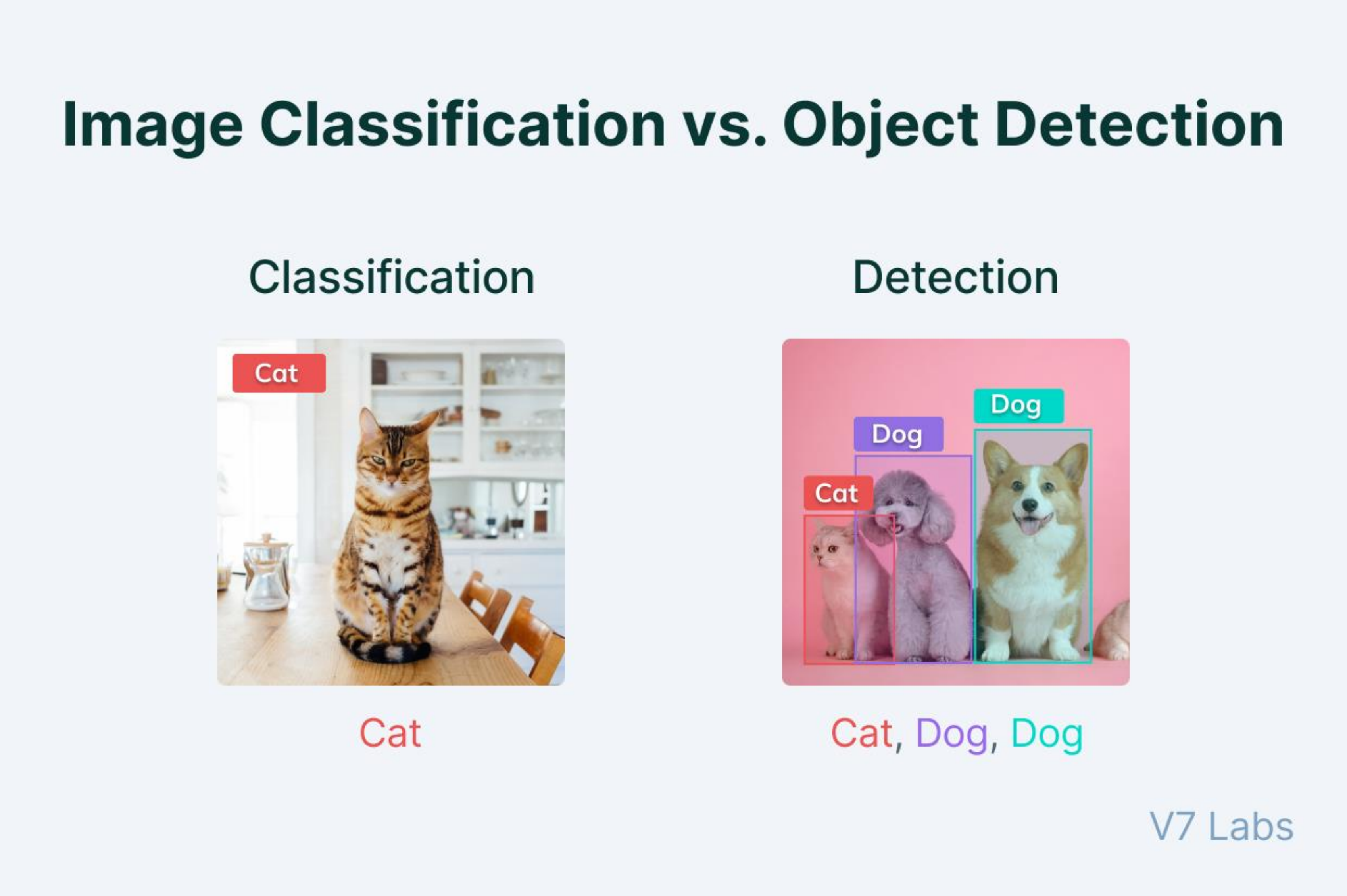
## Object detection vs. image classification

- Image classification sends a whole image through a classifier (such as a deep neural network) for it to place a tag on it. Classifiers take into consideration the whole image but don’t tell you where the tag appears in the image.

- Object detection is slightly more advanced, as it creates a bounding box around the classified object.

*Computer Vision*

*Course 12*



## Image Classification vs. Object Detection

Classification

Detection

Cat

Cat, Dog, Dog

V7 Labs

- Classification is a better option for tags that don’t really have physical boundaries, such as “blurry” or “sunny”.

- However, object detection systems will almost always outperform classification networks in spotting objects that do have a material presence, such as a cat.

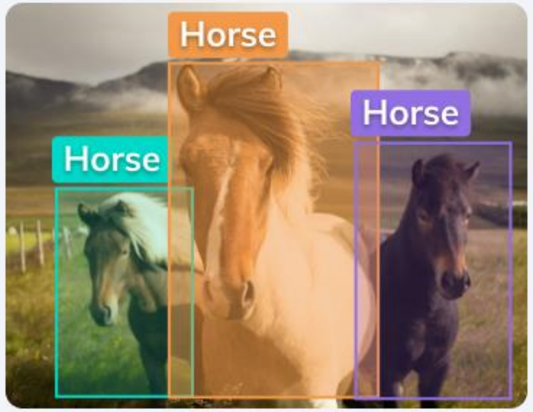
## Object detection vs image segmentation

- Image segmentation is the process of defining which pixels of an object class are found in an image.
- Semantic image segmentation will mark all pixels belonging to that tag, but won’t define the boundaries of each object.
- Object detection instead will not segment the object, but will clearly define the location of each individual object instance with a box.
- Combining semantic segmentation with object detection leads to instance segmentation, which first detects the object instances, and then segments each within the detected boxes (known in this case as regions of interest).

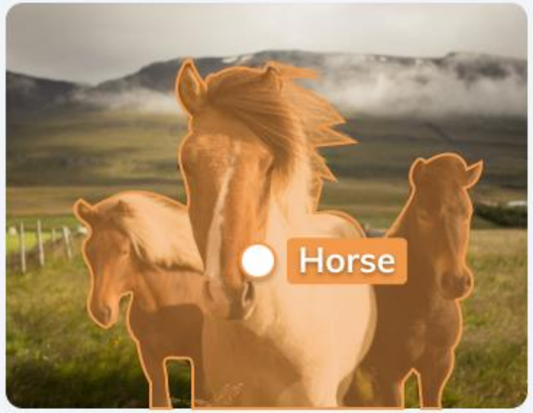
Computer Vision

Course 12

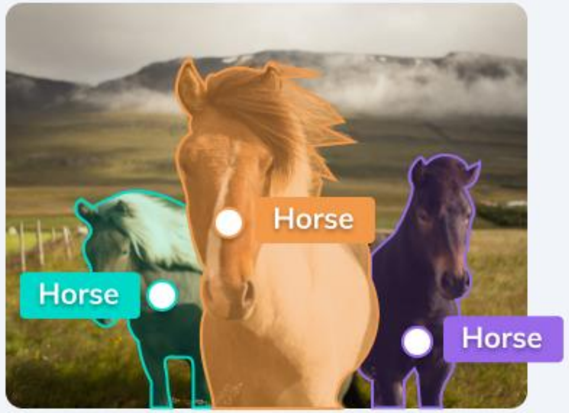
## Object Detection + Semantic Segmentation = Instance Segmentation



Object detection



Semantic Segmentation



Instance Segmentation

V7 Labs

Object detection is very good at:

- Detecting objects that take up between 2% and 60% of an image’s area.
- Detecting objects with clear boundaries.
- Detecting clusters of objects as 1 item.
- Localizing objects at high speed (>15fps)

## *Computer Vision*
*Course 12*

- Objects that are elongated— use Instance Segmentation.

  Long and thin items such as a pencil will occupy less than 10% of a box’s area when detected.

  This biases model towards background pixels rather than the object itself.

- Objects that have no physical presence—use classification

  Things in an image such as the tag “sunny”, “bright”, or “skewed” are best identified by image classification techniques—letting a network take the image and figure out which feature correlate to these tags.

- Objects that have no clear boundaries at different angles—use semantic segmentation

  The sky, ground, or vegetation in aerial images don’t really have a defined set of boundaries. Semantic segmentation is more efficient at showing pixels that belong to these classes. Object detection will find the “sky” as an object, but it takes it more time

- Objects that are often occluded—use Instance Segmentation if possible

  Occlusion is handled far better in two-stage detection networks than one-shot approaches. Instance segmentation models will do a better job at understanding and segmenting occluded objects than mere bounding-box detectors.

- Before deep learning :

  Viola-jones object detection technique,

  Scale-Invariant Feature Transforms (SIFT),

  Histogram of Oriented Gradients (HOG)

  …

- These would detect a number of common features across the image, and classify their clusters using logistic regression, color histograms, or random forests.

- Deep learning-based approaches use neural network architectures like

  RetinaNet,

  YOLO (You Only Look Once),

  CenterNet,

  SSD (Single Shot Multibox detector),

  Region proposals (R-CNN, Fast-RCNN, Faster RCNN, Cascade R-CNN)

for feature detection of the object, and then identification into labels.

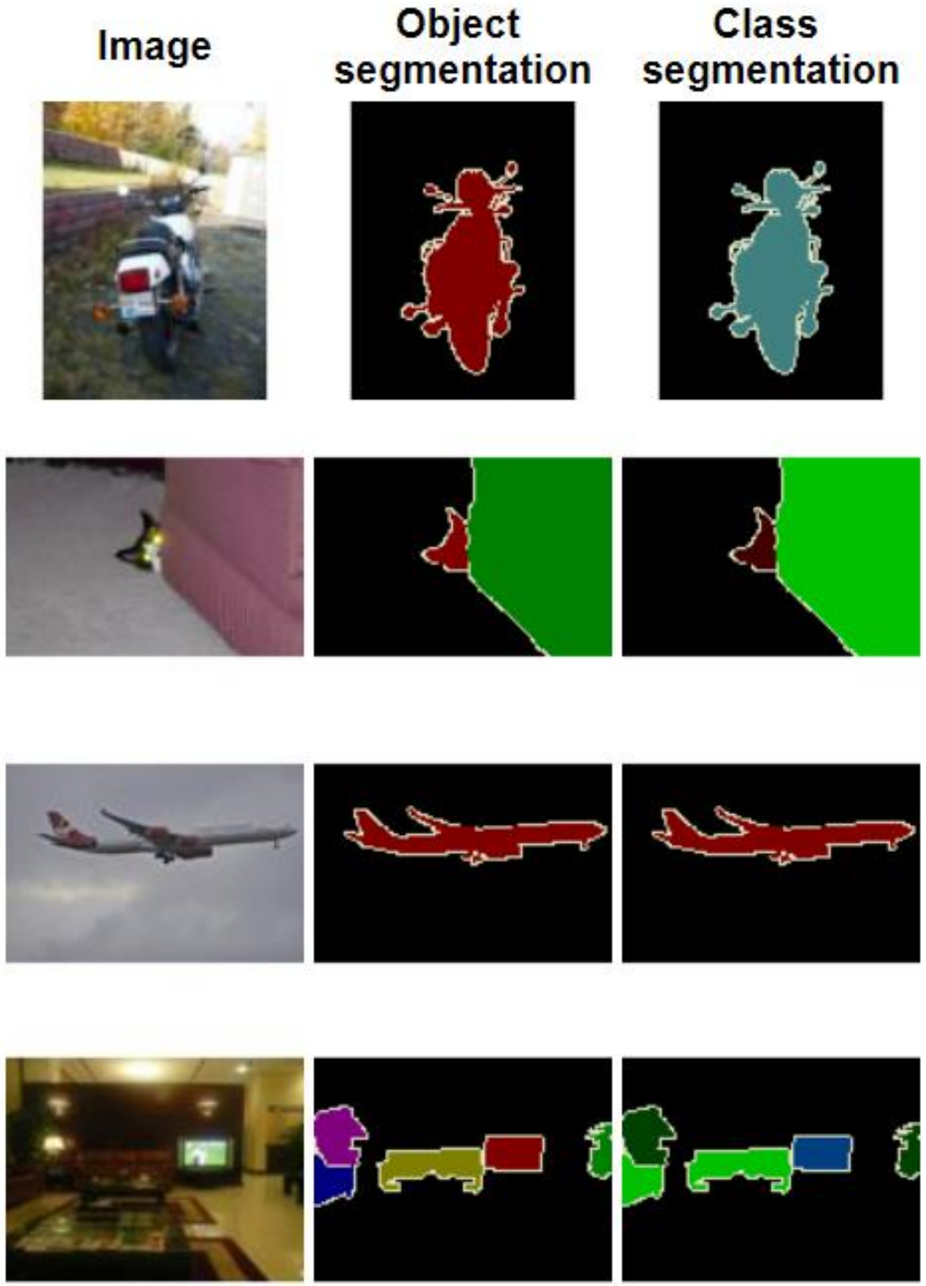
## Datasets and Metrics

- PASCAL Visual Object Classes (PASCAL VOC)

  - more than 11000 images compose the train and validation datasets

  - 1000 images are dedicated to the test dataset.

- The segmentation challenge is evaluated using the mean Intersection over Union (mIoU) metric. The Intersection over Union (IoU) is a metric also used in object detection to evaluate the relevance of the predicted locations. The IoU is the ratio between the area of overlap and the area of union between the ground truth and the predicted areas. The mIoU is the average between the IoU of the segmented objects over all the images of the test dataset.



Examples of the 2012 PASCAL VOC dataset for image segmentation. Source: [http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html](http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html)

## ***PASCAL-Context***

The PASCAL-Context dataset (2014) is an extension of the 2010 PASCAL VOC.

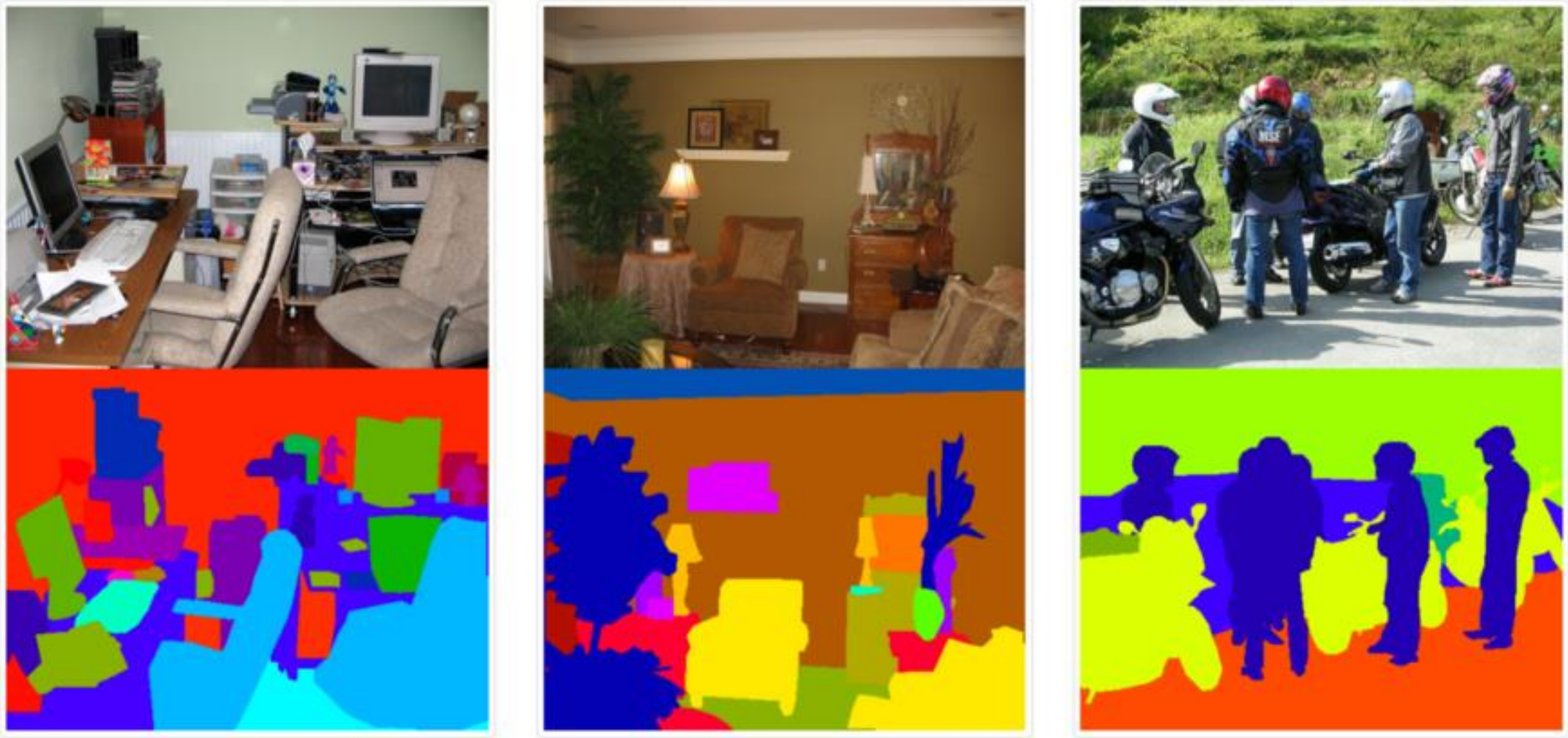
- 10000 images for training, 10000 for validation and 10000 for testing.

The entire scene is segmented providing more than 400 categories.

The images have been annotated during three months by six in-house annotators.

The official evaluation metric of the PASCAL-Context challenge is the mIoU.

Several other metrics are published by researches as the pixel Accuracy (pixAcc).



Example of the PASCAL-Context dataset.

Source: [https://cs.stanford.edu/~roozbeh/pascal-context/](https://cs.stanford.edu/~roozbeh/pascal-context/)

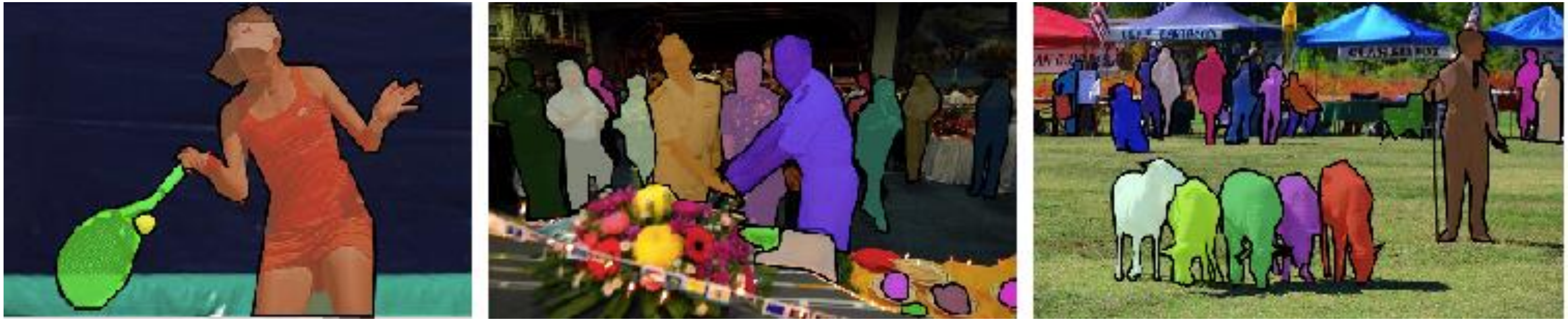
## ***Common Objects in COntext (COCO)***

There are two COCO challenges (in 2017 and 2018) for image semantic segmentation ("object detection" and "stuff segmentation"). The "object detection" task consists in segmenting and categorizing objects into 80 categories. The "stuff segmentation" task uses data with large segmented part of the images (sky, wall, grass), they contain almost the entire visual information. In this blog post, only the results of the "object detection" task will be compared because too few of the quoted research papers have published results on the "stuff segmentation" task.

The COCO dataset for object segmentation is composed of more than 200.000 images with over 500.000 object instance segmented.

It contains a training dataset, a validation dataset, a test dataset for researchers (test-dev) and a test dataset for the challenge (test-challenge). The annotations of both test datasets are not available.

These datasets contain 80 categories and only the corresponding objects are segmented. This challenge uses the same metrics than the object detection challenge: the Average Precision (AP) and the Average Recall (AR) both using the Intersection over Union (IoU).



Example of the COCO dataset for object segmentation.
Source: [http://cocodataset.org/](http://cocodataset.org/)

## *Cityscapes*

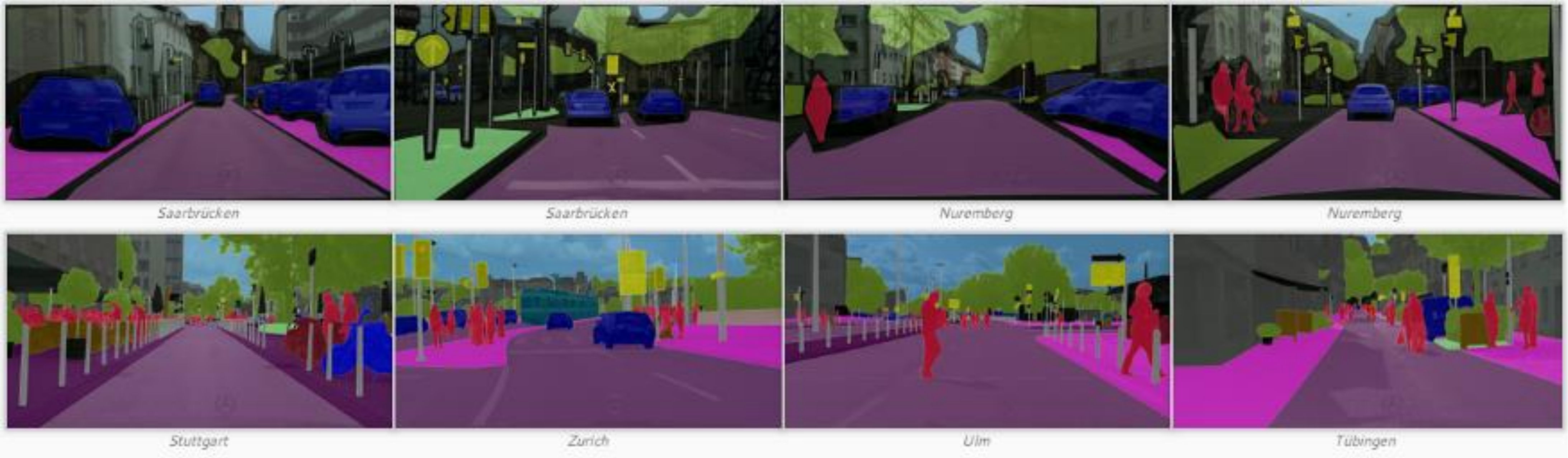
The Cityscapes dataset has been released in 2016 and consists in complex segmented urban scenes from 50 cities.

It is composed of 23.500 images for training and validation (fine and coarse annotations) and 1500 images for testing (only fine annotation).

The images are fully segmented such as the PASCAL-Context dataset with 29 classes (within 8 super categories: flat, human, vehicle, construction, object, nature, sky, void).

It is often used to evaluate semantic segmentation models because of its complexity.

It is also well known for its similarity with real urban scenes for autonomous driving applications. The performances of semantic segmentation models are computed using the mIoU metric such as the PASCAL datasets.



Examples of the Cityscapes dataset. Top: coarse annotations. Bottom: fine annotation.

Source: https://www.cityscapes-dataset.com/

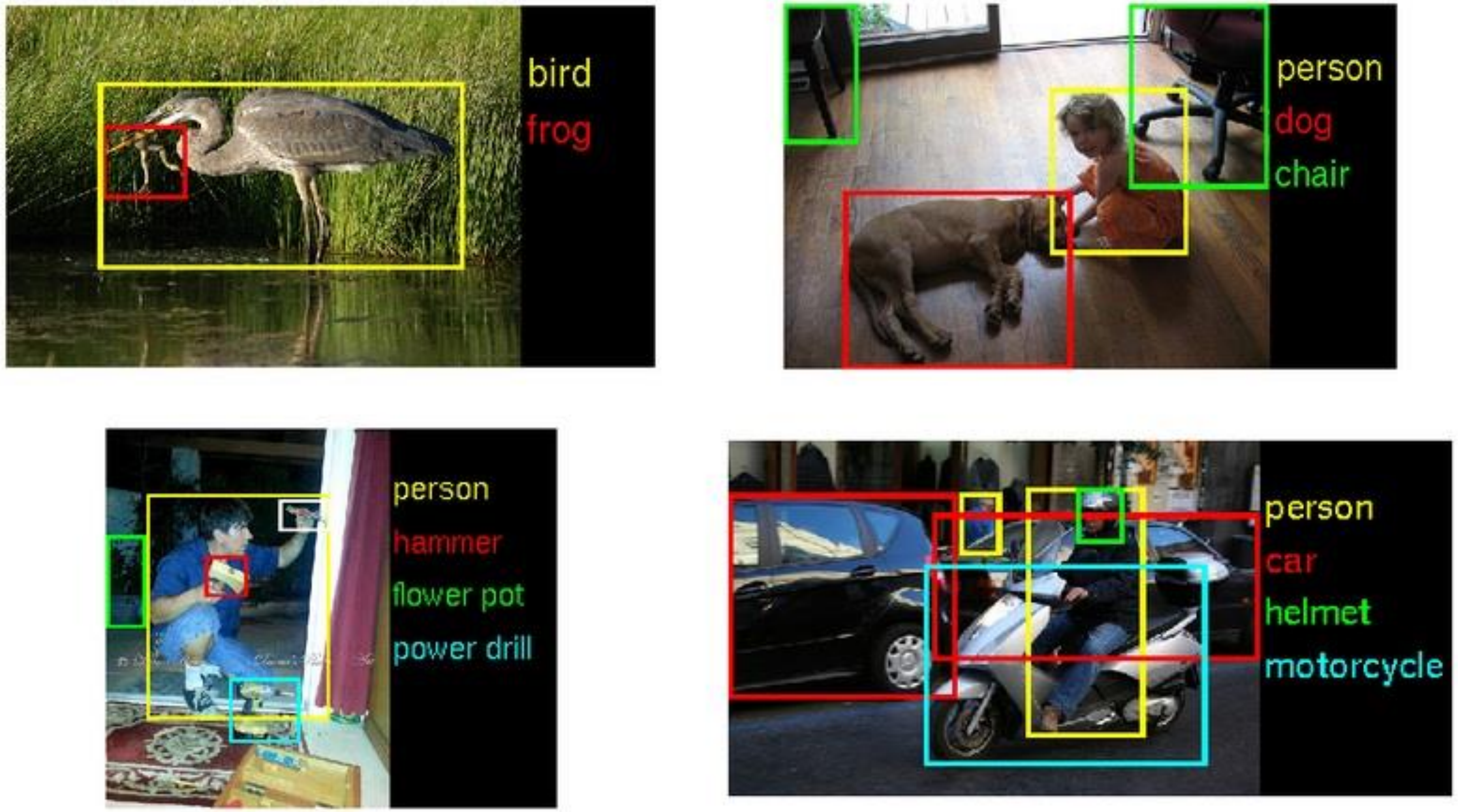
## ILSVRC

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) was an annual challenge running from 2010 to 2017 and became a benchmark for evaluating algorithm performance.

The dataset size was scaled up to more than a million images consisting of 1000 object classification classes. 200 of these classes were hand-picked for object detection task, constitute of more than 500.000 images. Various sources including ImageNet and Flikr, were used to construct detection dataset.

# Computer Vision
## Course 12

Example ILSVRC2014 images:

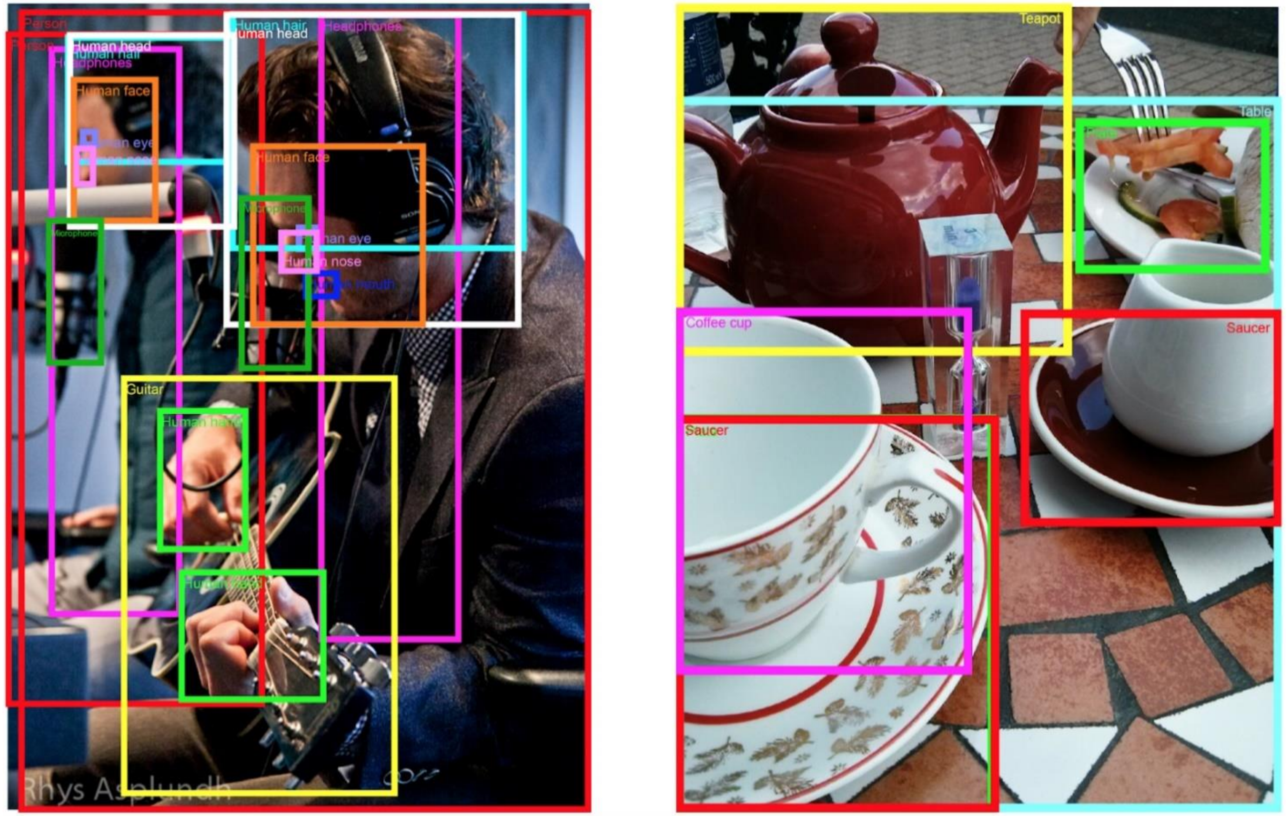


## Open Image

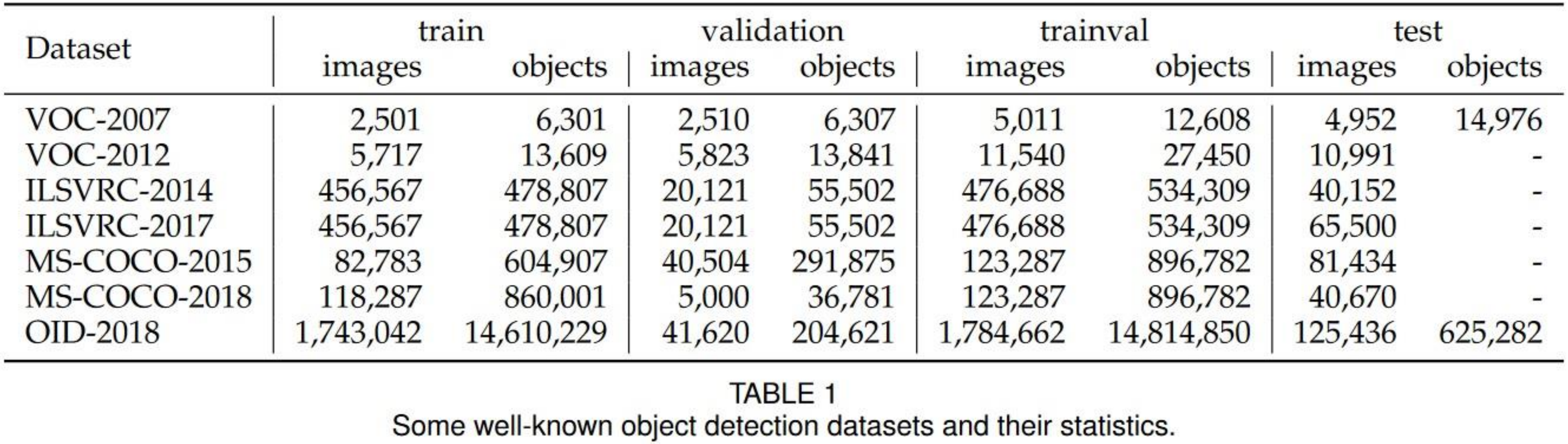
Google’s Open Images dataset is composed of 9.2 million images, annotated with image-level labels, object bounding boxes, and segmentation masks, among others.

It was launched in 2017 and has since received six updates. Open Images has 16 million bounding boxes for 600 categories on 1.9 million images for detection, making it the largest dataset for object localization.

Its creators took extra care to choose interesting, complex and diverse images, having 8.3 object categories per image. Several changes were made to the AP introduced in Pascal VOC like ignoring unannotated class, detection requirement for class and its subclass.



Open Image samples



| Dataset | train |  | validation |  | trainval |  | test |  |
|---|---|---|---|---|---|---|---|---|
|  | images | objects | images | objects | images | objects | images | objects |
| VOC-2007 | 2,501 | 6,301 | 2,510 | 6,307 | 5,011 | 12,608 | 4,952 | 14,976 |
| VOC-2012 | 5,717 | 13,609 | 5,823 | 13,841 | 11,540 | 27,450 | 10,991 | - |
| ILSVRC-2014 | 456,567 | 478,807 | 20,121 | 55,502 | 476,688 | 534,309 | 40,152 | - |
| ILSVRC-2017 | 456,567 | 478,807 | 20,121 | 55,502 | 476,688 | 534,309 | 65,500 | - |
| MS-COCO-2015 | 82,783 | 604,907 | 40,504 | 291,875 | 123,287 | 896,782 | 81,434 | - |
| MS-COCO-2018 | 118,287 | 860,001 | 5,000 | 36,781 | 123,287 | 896,782 | 40,670 | - |
| OID-2018 | 1,743,042 | 14,610,229 | 41,620 | 204,621 | 1,784,662 | 14,814,850 | 125,436 | 625,282 |

TABLE 1
Some well-known object detection datasets and their statistics.

ZOU, Zhengxia, et al. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.

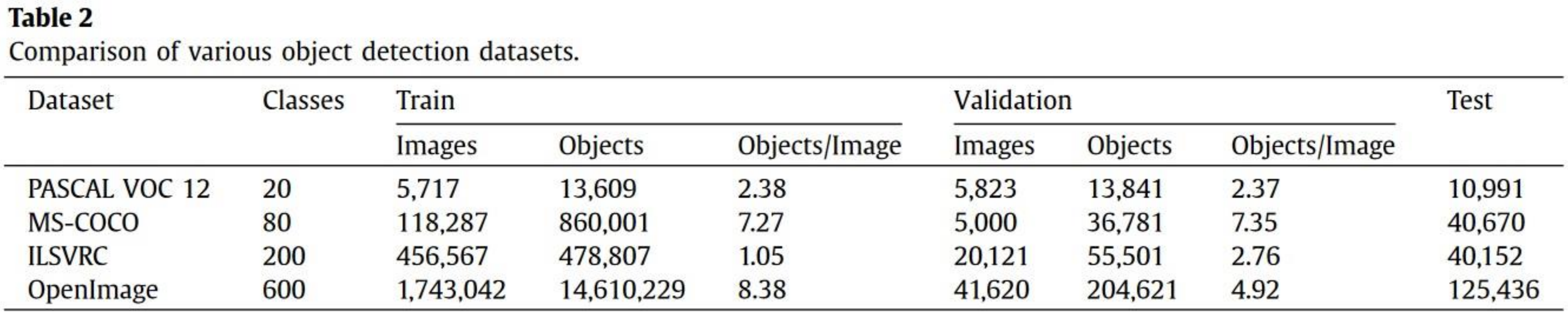


**Table 2**
Comparison of various object detection datasets.

| Dataset | Classes | Train | | | Validation | | | Test |
|---|---|---|---|---|---|---|---|---|
| | | Images | Objects | Objects/Image | Images | Objects | Objects/Image | |
| PASCAL VOC 12 | 20 | 5,717 | 13,609 | 2.38 | 5,823 | 13,841 | 2.37 | 10,991 |
| MS-COCO | 80 | 118,287 | 860,001 | 7.27 | 5,000 | 36,781 | 7.35 | 40,670 |
| ILSVRC | 200 | 456,567 | 478,807 | 1.05 | 20,121 | 55,501 | 2.76 | 40,152 |
| OpenImage | 600 | 1,743,042 | 14,610,229 | 8.38 | 41,620 | 204,621 | 4.92 | 125,436 |

ZAIDI, Syed Sahil Abbas, et al. A survey of modern deep learning  based object detection models. *Digital Signal Processing*, 2022, 103514.

## Backbone architectures

Are one of the most important component of the object detector.

These networks extract feature from the input image used by the model.

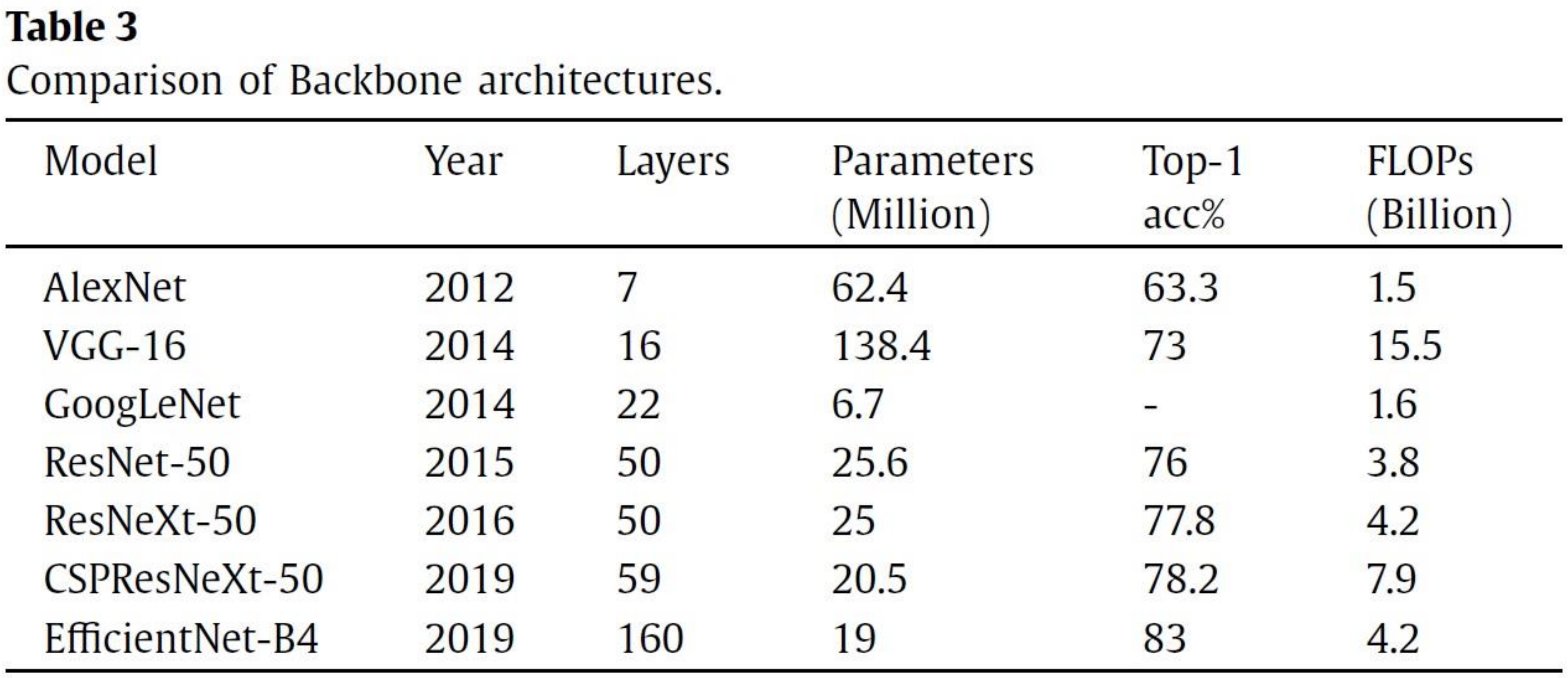


**Table 3**
Comparison of Backbone architectures.

| Model | Year | Layers | Parameters (Million) | Top-1 acc% | FLOPs (Billion) |
|---|---|---|---|---|---|
| AlexNet | 2012 | 7 | 62.4 | 63.3 | 1.5 |
| VGG-16 | 2014 | 16 | 138.4 | 73 | 15.5 |
| GoogLeNet | 2014 | 22 | 6.7 | - | 1.6 |
| ResNet-50 | 2015 | 50 | 25.6 | 76 | 3.8 |
| ResNeXt-50 | 2016 | 50 | 25 | 77.8 | 4.2 |
| CSPResNeXt-50 | 2019 | 59 | 20.5 | 78.2 | 7.9 |
| EfficientNet-B4 | 2019 | 160 | 19 | 83 | 4.2 |

Object detection generally is categorized into :

- Two-stage object detectors
- Single-stage object detectors
- Transformer-based detectors

- A network which has a separate module to generate region proposals is termed as a two-stage detector. These models try to find an arbitrary number of objects proposals in an image during the first stage and then classify and localize them in the second. They generally take longer to generate proposals, have complicated architecture and lacks global context.
- Single-stage detectors classify and localize semantic objects in a single shot using dense sampling. They use predefined boxes/keypoints of various scale and aspect ratio to localize objects. It edges two-stage detectors in real-time performance and simpler design.
- Transformers were initially introduced for NLP, their general-purpose nature helped in migrating to vision tasks. Although transformer based detectors have achieved SOTA in many vision tasks, they require comparatively more data to train.

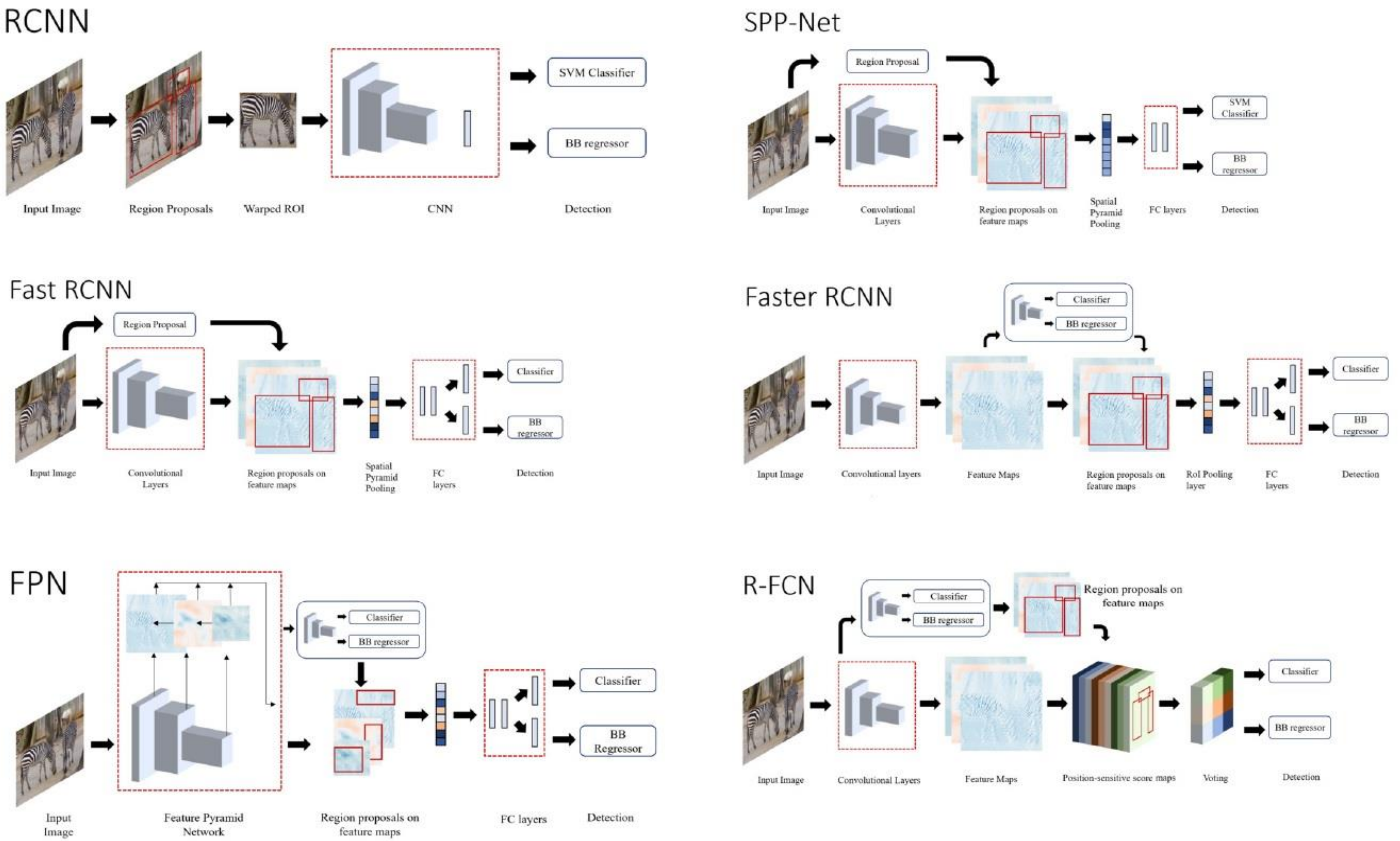


**Fig. 8.** Illustration of the internal architecture of different two stage object detectors. Features created using: https://poloclub.github.io/cnn-explainer/.

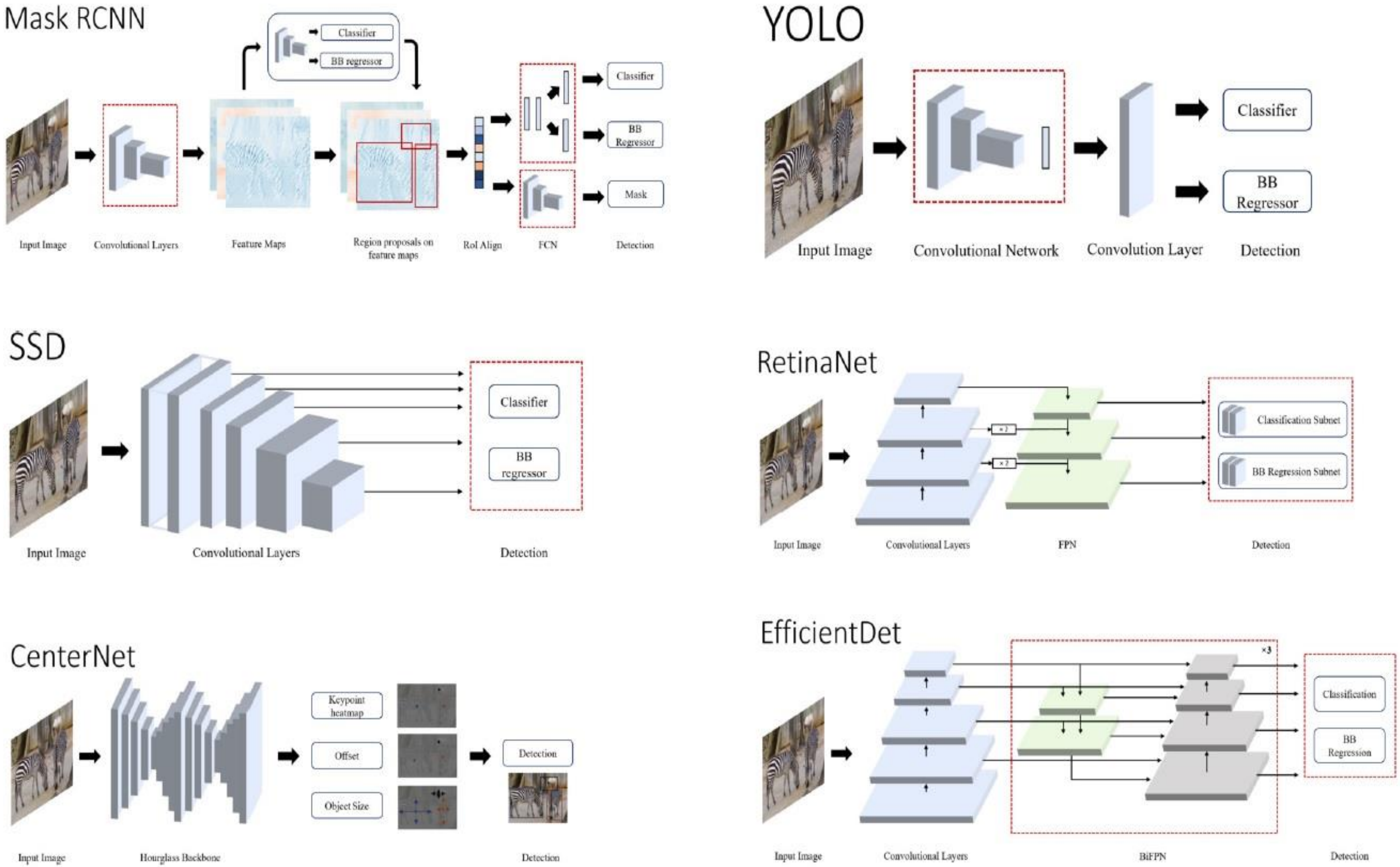


**Fig. 9.** Illustration of the internal architecture of different two and single stage object detectors. Features created using: https://poloclub.github.io/cnn-explainer/.

# Two-stage object detectors

## *R-CNN*

The Region-based Convolutional Neural Network (R-CNN) use a class agnostic region proposal module with CNNs to convert detection into classification and localization problem. A mean-subtracted input image is first passed through the region proposal module, which produces 2000 object candidates. This module find parts of the image which has a higher probability of finding an object using Selective Search. These candidates are then warped and propagated through a CNN network, which extracts a 4096-dimension feature vector for each proposal.

The feature vectors are then passed to the trained, class-specific Support Vector Machines (SVMs) to obtain confidence scores. Non-maximum suppression (NMS) is applied to the scored regions, based on its IoU and class. Once the class has been identified, the algorithm predicts its bounding box using a trained bounding-box regressor, which predicts four parameters i.e., center coordinates of box along with its width and height.

## *SPP-net*

One uses Spatial Pyramid Pooling (SPP) layer [to process image of arbitrary size or aspect ratio. Only the fully connected part of the CNN required a fixed input. SPP-net  merely shifted the convolution layers of CNN before the region proposal module and added a pooling layer, thereby making the network independent of size/aspect ratio and reducing the computations. The selective search algorithm is used to generate candidate windows.

Feature maps are obtained by passing the input image through the convolution layers of a ZF-5 network.  The candidate windows are then mapped on to the feature maps, which are subsequently converted into fixed length representations by spatial bins of a pyramidal pooling layer.

This vector is passed to the fully connected layer and ultimately, to SVM classifiers to predict class and score.  SPP-net has a post processing layer to improve localization by bounding box regression.

SPP-Net is considerably faster than the R-CNN model with comparable accuracy. It can process images of any shape/aspect ratio and thus, avoid object deformation due to input warping.

Due to the similar R-CNN architecture, multistage training, this network remains  computationally expensive and still has long training time.

# Fast R-CNN

One of the major issues with R-CNN/SPP-Net was the need to train multiple systems separately.

Fast R-CNN solved this by creating a single end-to-end trainable system. The network takes as input an image and its object proposals. The image is passed through a set of convolution layers to obtain feature maps and the object proposals are mapped to it. The pyramidal structure was replaced by a single spatial bin, called RoI pooling layer. This layer is connected to 2 fully connected layers and then branches out into a *N+1*-class Soft-Max layer and a bounding box regressor layer, which has a fully connected layer as well. The model also changed the loss function of bounding box regressor from L2 to smooth L1 to improve performance and introduced a multi-task loss to better train the network.

Fast R-CNN was introduced as an improvement in speed (146x) on R-CNN while the increase in accuracy was supplementary. It simplified training procedure, removed pyramidal pooling and presented a new loss function. The object detector, without the region proposal network, reported near real time speed with considerable accuracy.

## *Faster R-CNN*

Fast R-CNN has a  region proposal generation that is rather slow (2 sec per image compared to 0.2 sec per image).

The solution is a fully convoluted network as a region proposal network (RPN) that takes an arbitrary input image and outputs a set of candidate windows. Each such window has an associated *objectness score* which determines likelihood of an object. RPN introduces Anchor boxes. It used multiple bounding boxes of different aspect ratios and regressed over them to localize the object. The input image is first passed through the CNN to obtain a set of feature maps. These are forwarded to the RPN, which produces bounding boxes and their classification. Selected proposals are then mapped back to the feature maps obtained from the previous CNN layer in the RoI pooling layer, and ultimately fed to the fully connected layer. The result is then sent to classifier and bounding box regressor. Faster R-CNN is essentially Fast R-CNN with RPN as region proposal module.

Faster R-CNN improved the detection accuracy over the previous state-of-art by more than 3% and decreased inference time by an order of magnitude. It fixed the bottleneck of slow region proposal and ran in near real time at 5 frames per second. An-other advantage of having a CNN in region proposal was that it could learn to produce better proposals and thereby increase accuracy.

## Mask R-CNN

Mask R-CNN extends the Faster R-CNN by adding another branch in parallel for pixel-level object instance segmentation. The branch is a fully connected network applied on the RoIs to classify each pixel into segments with little overall computation cost. It uses similar basic Faster R-CNN architecture for object proposal, but adds a mask head parallel to classification and bounding box regressor head. One major difference is the use of RoIAlign layer, instead of RoIPool layer, to avoid pixel level misalignment due to spatial quantization. As backbone architecture ResNeXt-101 is employed  with the feature Pyramid Network (FPN) for better accuracy and speed. The loss function of Faster R-CNN is updated with the mask loss and uses 5 anchor boxes with 3 aspect ratio, similar to FPN. Overall training of Mask R-CNN is similar to faster R-CNN.

Mask R-CNN performed better than the existing state of the art single-model architectures, added an extra functionality of instance segmentation with little overhead computations. It is simple to train, flexible and generalizes well in applications like keypoint detection, human pose estimation, etc. However, it is still below the real time performance (>30 fps).

## DetectoRS

Many contemporary two stage detectors use the mechanism of looking and thinking twice i.e. calculating object proposals first and using them to extract features to detect objects. DetectoRS applies this mechanism at both macro and micro level of the network. At macro level, they propose Recursive Feature Pyramid (RFP), formed by stacking multiple feature pyramid network (FPN) with extra feedback connection from the top-down level path in FPN to the bottom-up layer. The output of the FPN is processed by the Atrous Spatial Pyramid Pooling layer (ASPP) before passing it to the next FPN layer. A Fusion module is used to combine FPN outputs from different modules by creating an attention map. At micro level, Switchable Atrous Convolution (SAC) to regulate the dilation rate of convolution. An average pooling layer with *5x5*filter and a *1x1* convolution is used as a switch function to decide the rate of atrous convolution, helping the backbone detect objects at various scales on the fly.

The combination of these two techniques, Recursive Feature Pyramid and Switchable Atrous Convolution results in DetectoRS (ResNext-101 as backbone).

DetectoRS combined multiple systems to improve performance of the detector and sets the state-of-the-art for the two stage detectors. Its RFP and SAC modules generalized well and can be used in other detection models. However, it is not suitable for real time detections as it can only process about 4 frames per second.

*Computer Vision*
*Course 12*

## Single stage detectors

### *YOLO* (You Only Look Once )

Two stage detectors solve the object detection as a classification problem, a module presents candidates which the network classifies as either an object or background.

YOLO reframed it as a regression problem, directly predicting the image pixels as objects and its bounding box attributes.

In YOLO, the input image is divided into a *S x S* grid and the cell where the object's center falls is responsible for detecting it. A grid cell predicts multiple bounding boxes, and each prediction array consists of 5 elements: center of bounding box – x and y, dimensions of the box – w and h, and the confidence score.

YOLO was inspired from the GoogLeNet model for image classification, which uses cascaded modules of smaller convolution networks. It is pre-trained on ImageNet data till the model achieves high accuracy and consequently modified by adding randomly initialized convolution and fully connected layers.

At training time, the grid cells predict only one class as the network converges better, but it can be increased during the inference time. Multitask loss, i.e. combined loss of all predicted components, is used to optimize the model. Non maximum suppression (NMS) removes the class-specific multiple detections.

YOLO surpassed its contemporary single stage real time models by a huge margin in both accuracy and speed. There are seven versions of YOLO, each one better than the previous one.

# SSD

Single Shot MultiBox Detector (SSD) [was the first single stage detector that matched accuracy of contemporary two stage detectors like Faster R-CNN, while maintaining real time speed.

SSD was built on VGG-16, with additional auxiliary structures to improve performance.

SSD detects smaller objects earlier in the network when the image features are not too crude, while the deeper layers are responsible for offset of the default boxes and aspect ratios.

During training, SSD match each ground truth box to the default box with the best Jaccard overlap and train the network accordingly. It also uses hard negative mining and heavy data augmentation.

Even though SSD was significantly faster and more accurate than both state-of-the-art networks like YOLO and Faster R-CNN, it had difficulty in detecting small objects. This issue was later solved by using better backbone architectures like ResNet and implementing other small fixes.

## RetinaNet

The reason single stage detectors lag is the “extreme foreground-background class imbalance”. They proposed a reshaped cross entropy loss, called Focal loss as the means to remedy the imbalance. Focal loss parameter reduces the loss contribution from easy examples.

RetinaNet predicts objects by dense sampling of the input image in location, scale and aspect ratio. It uses ResNet augmented by Feature Pyramid Network (FPN) as the backbone and two similar subnets -classification and bounding box regressor. Each layer from the FPN is passed to the subnets, enabling it to detect objects as various scales. The classification subnet predicts the object score for each location while the box regression subnet regresses the offset for each anchor to the ground truth. Both subnets are small FCN and share parameters across the individual networks. Unlike most previous works, a class-agnostic bounding box regressor was employed and found to be equally effective.

RetinaNet is simple to train, converges faster and easy to implement. It achieved better performance in accuracy and run time than the two stage detectors. RetinaNet also pushed the envelope in advancing the ways object detectors are optimized by the introduction of a new loss function.

# CenterNet

In CenterNet the objects are modelled as points, instead of the conventional bounding box representation.

CenterNet predicts the object as a single point at the center of the bounding box. The input image is passed through the FCN that generates a heatmap, whose peaks correspond to center of detected object.

It uses a ImageNet pretrained stacked Hourglass-101 as the feature extractor network and has 3 heads – heatmap head to determine the object center, dimension head to estimate size of object and offset head to correct offset of object point. Multitask loss of all three heads is back propagated to feature extractor while training. During inference, the output from the offset head is used to determine the object point and finally a box is generated.

CenterNet is more accurate and has lesser inference time than its predecessors. It has high precision for multiple tasks like 3D object detection, keypoint estimation, pose, instance segmentation, orientation detection and others.

# Transformer based detectors

Transformers have had a profound impact in the Natural Language Processing (NLP) domain since its inception. Transformers use the attention model to establish dependencies among the elements of the sequence and can attend to longer context than other sequential architectures. The success of transformers in NLP sparked interest in its application in computer vision.

A transformer is a deep learning model that adopts the mechanism of self-attention, differentially weighting the significance of each part of the input data.

# *DeTR*

Detection Transformer or DeTR uses a combination of CNN and transformer to create an end-to-end trainable detector and removed any hand-crafted modules, like non-maximum suppression or anchor generation.

The input image is first passed through a CNN backbone network to obtain image features and then through a set of transformers. Final predictions, a bipartite output of class and bounding box, are obtained by passing transformer output through feed-forward networks.

DeTR used ResNets as backbone network. The transformer encoder takes image features, along with the position encodings as input and directs result to the decoder. The decoder manipulates *N* input embeddings, called object queries, to generate output. Object queries are position encodings that are initialized as zeros but learnt during training.

Multi-head attentions in the decoder modifies these object queries with encoder embeddings to generate results, which are ultimately passed through multi-layer perceptrons to predict class and bounding boxes. DeTR uses bipartite matching loss to find the optimal one-to-one matching between detector output and padded ground truth.

DeTr is a simple, general-purpose transformer-based detector which is competitive with the CNN based detectors. However, its performance on small objects leaves something to be desired.

## *Swin Transformer*

Swin Transformer seeks to provide a transformer-based backbone for computer vision tasks. It splits the input images in multiple, non-overlapping patches and converts them into embeddings. Numerous Swin Transformer blocks are then applied to the patches in 4 stages, with each successive stage reducing the number of patches to maintain hierarchical representation.

The Swin Transformer block is composed of local multi-headed self-attention (MSA) modules, based on alternating shifted patch window in successive blocks. Computation complexity becomes linear with image size in local self-attention while shifted window enables cross-window connection. [

Swin Transformer achieved the state-of-the-art on MS COCO dataset, but utilises comparatively higher parameters than convolutional models.

Transformers present a paradigm shift from the CNN based neural networks. While its application in vision is still in a nascent stage, its potential to replace convolution from these tasks is very real.

## Lightweight networks

A new branch of research has shaped up in recent years, aimed at designing small and efficient networks for resource constrained environments as is common in Internet of Things (IoT) deployments. This trend has influenced the design of potent object detectors too. It is seen that although a large number of object detectors achieve excellent accuracy and perform inference in real-time, a majority of these models require excessive computing resources and therefore cannot be deployed on edge devices.

Many different approaches have shown exciting results in the past. Utilization of efficient components and compression techniques like pruning, quantization, hashing, etc. have improved the efficiency of deep learning models. Use of trained large network to train smaller models, called distillation, has also shown interesting results.

Some prominent examples of efficient neural network design for achieving high performance on edge devices are:  SqueezeNet, MobileNets, ShuffleNets, PeleeNet, MnasNet, Once-for-all (OFA)