Intro to Image Segmentation

Navigating the World of Visual Decomposition

Presented By:

M. Junaid Ali Asif Raja

S. Zoraiz Ali Sherazi

TABLE OF CONTENTS

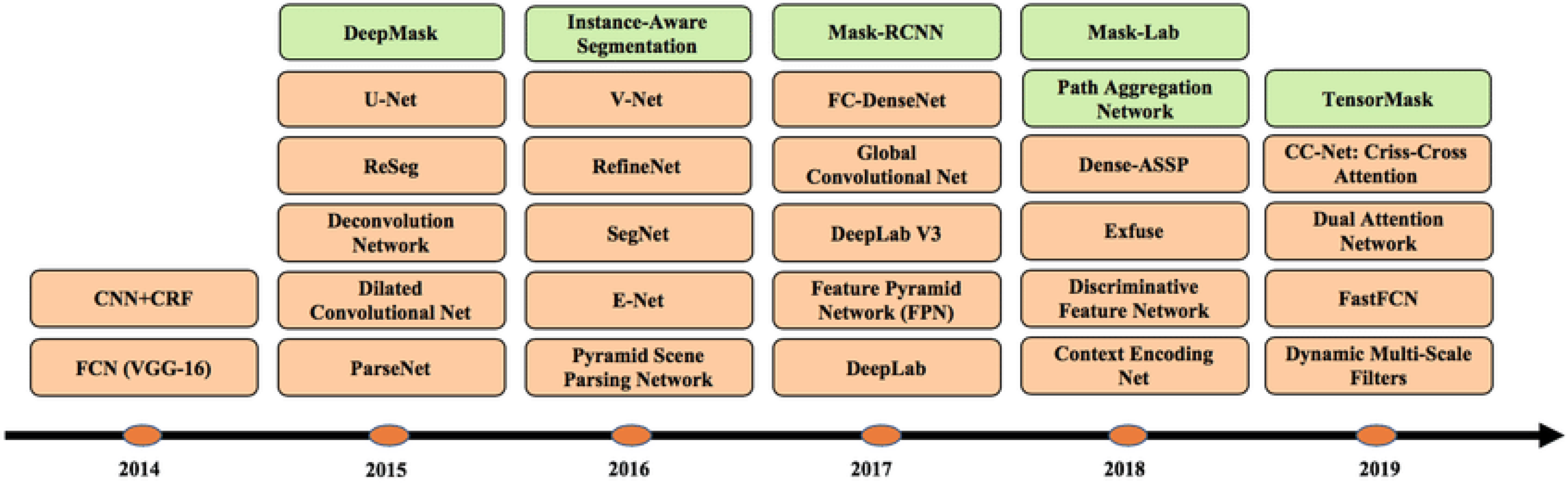
Introduction

- Image segmentation divides an image into distinct, meaningful regions or segments. The task involves assigning a label or category to each pixel based on visual characteristics.
- Pixels with similar visual features are grouped together during the segmentation process.
- The primary goal is to enhance computer understanding of different objects or parts within an image.
- This process facilitates more advanced analysis and recognition tasks in computer vision.
- In simpler terms, image segmentation is akin to breaking down an image into digital puzzle pieces, aiding machines in recognizing and understanding the image content.

History

- Developed clustering methods in the 1990s, such as k-means, to segment images based on pixel similarities.
- Graph-based approaches gained traction in the 2000s, with algorithms like the normalized cuts algorithm focusing on representing images as graphs and identifying optimal cuts for segmentation.
- Machine learning techniques, particularly Support Vector Machines (SVMs) and Random Forests, were introduced in the 2000s, leveraging learning from labeled training data for image segmentation tasks.
- The deep learning revolution in the 2010s marked a significant shift in image segmentation, especially with the introduction of convolutional neural networks (CNNs).
- The U-Net architecture (2015) played a pivotal role in semantic segmentation tasks, providing a robust framework for pixel-level classification.
- Later in the 2010s, models like Mask R-CNN (2017) improved instance segmentation, allowing for the identification and differentiation of individual objects within the same class..

History...(contd)



Types of Segmentation

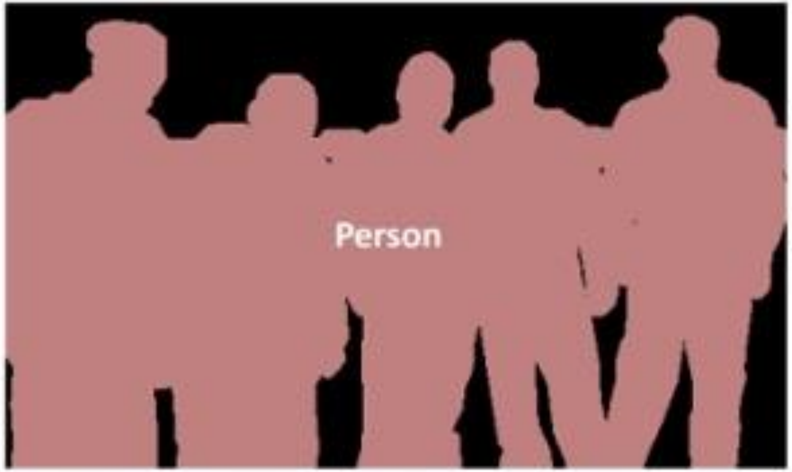


Image 1

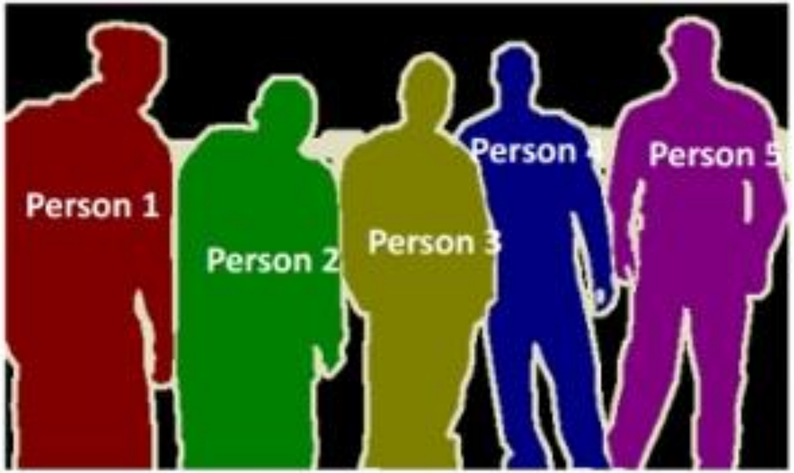
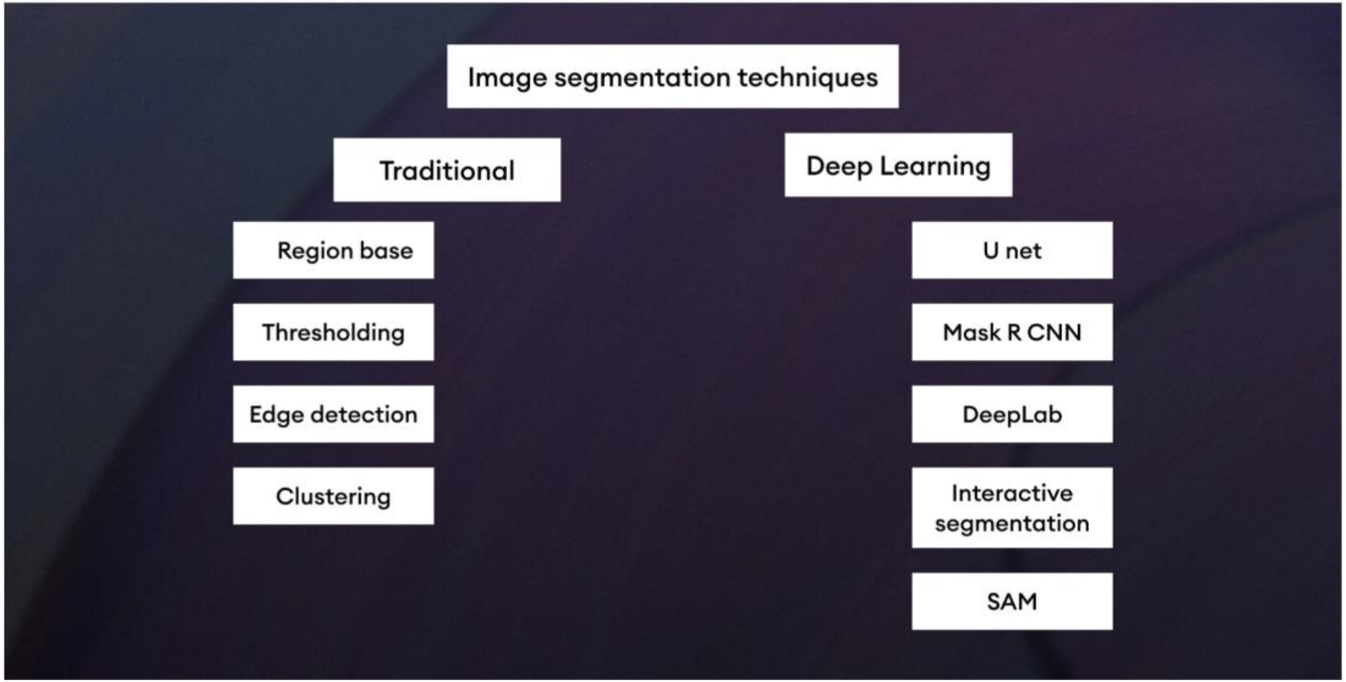


Image 2

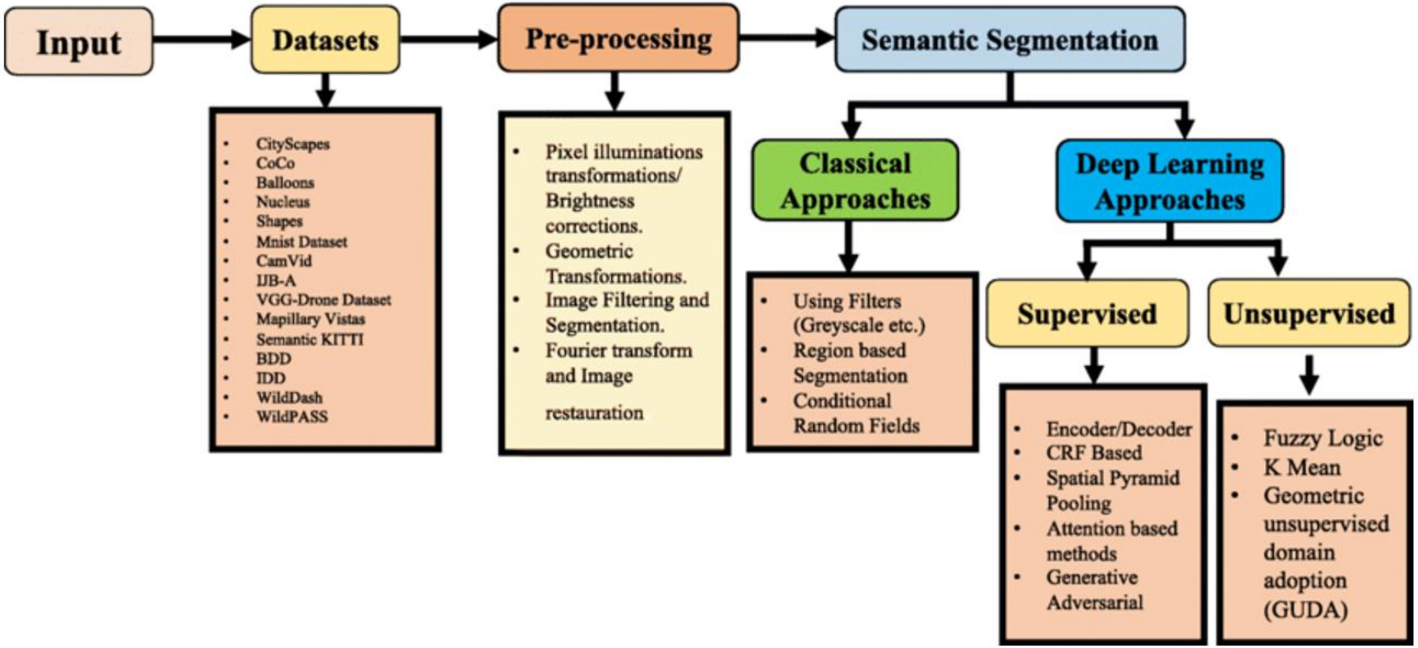
Types of Segmentation

Aspect	Semantic Segmentation	Instance Segmentation
Objective	Classify each pixel into predefined categories.	Identify and differentiate individual instances of objects.
Output	Assigns a single label to each pixel in the image.	Labels each pixel and distinguishes instances within the same class.
Use Case	Provides a holistic understanding of the scene.	Useful for tasks requiring precise object delineation, such as in robotics and autonomous vehicles.
Application	Scene understanding, image categorization.	Object detection, tracking, and autonomous systems.
Output Example		
Algorithm Examples	U-Net, DeepLab, FCN.	Mask R-CNN, YOLACT.
Pixel Labeling Consistency	Consistent labeling for pixels of the same class.	Distinguishes individual instances within the same class.
Challenges	May struggle with distinguishing instances of the same class.	More computationally demanding due to the need for precise instance delineation.
Useful in	Image understanding, general segmentation tasks.	Robotics, autonomous vehicles, where object separation is crucial.

Segmentation Techniques



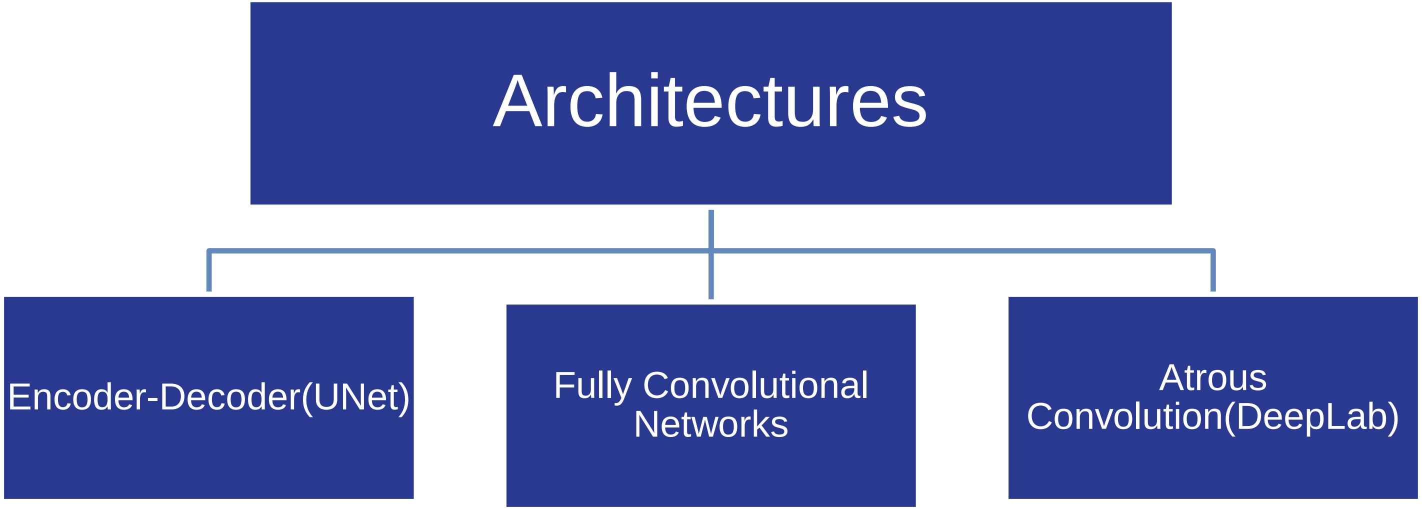
Semantic Segmentaion



Dataset Creation

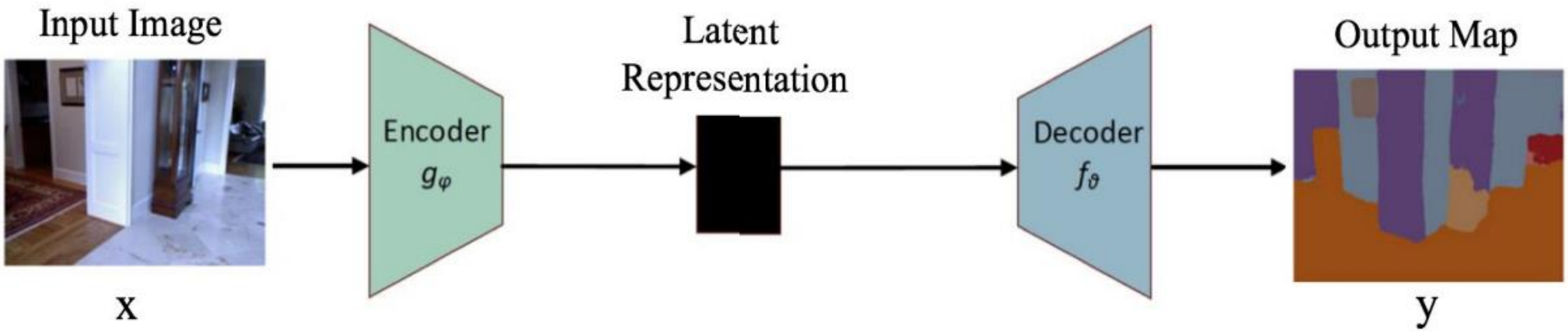
https://roboflow.com/

Models & Architectures



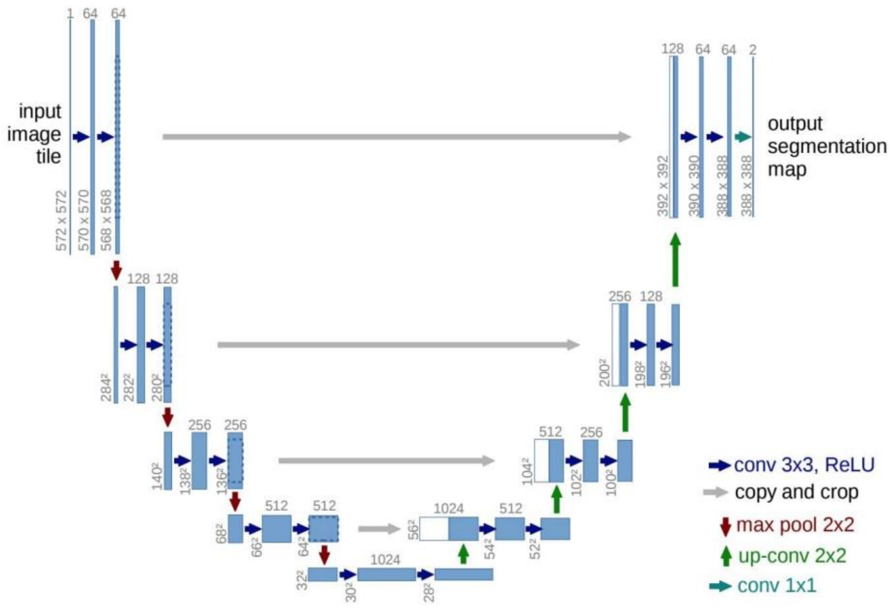
Encoder-Decoder Architecture

- The encoder comprises down-sampling layers, such as convolutional and pooling layers.
- These layers capture hierarchical and abstract features from the input image.
- The decoder involves up-sampling layers to generate the final segmentation map
- Latent representation refers to the learned, compressed, and abstracted form of data that captures essential features and patterns.



Encoder-Decoder Architecture(U-Net)

- U-Net was introduced in 2015 by Ronneberger et al., primarily for biomedical image segmentation. It follows an encoder-decoder structure with skip connections.
- Skip connections concatenate feature maps from corresponding encoder layers to preserve spatial information.
- They assist in overcoming the vanishing gradient problem and aid in precise localization.



Encoder-Decoder Architecture(U-Net)

Mathematics:

•**Convolution Operation:**

- Y = X * W, where Y is the output tensor.

•**Rectified Linear Unit (ReLU):**

- Activation Function: f(x) = max(0, x), where x is the input.

•**Skip Connection Concatenation:**

- Concatenation Operation: Y = concat(X_1, X_2), where X_1 and X_2 are feature maps from different layers.

•**Loss Function:**

- Commonly used loss functions for segmentation tasks include the Dice loss or cross-entropy loss, depending on the specific requirements of the task.

(1) BCE Loss represent as

$$BCELoss\left(y,\overline{y}\right) = -\left(ylog\left(\overline{y}\right) + \left(1-y\right)log\left(1-\overline{y}\right)\right) \quad (2)$$

Here y is the actual value and ȳ is the predicted value

(2) Focal Loss represent as

$$FocalLoss\left(p_t\right) = -\alpha_t\left(1-p_t\right)^{\gamma}log\left(p_t\right) \quad (3)$$

Here γ is always greater than zero but when γ is equal to 1 then it works like a Cross Entropy function, the range of α is between 0 to 1, it is treated as a hyperparameter.

(3) Dice Loss represent as

$$DiceLoss\left(y,\overline{p}\right) = 1 - \left(\left(2y\overline{p}+1\right) \div \left(y+\overline{p}+1\right)\right) \quad (4)$$

Here 1 is added to ensure that function is not become undefined in edge case such as y = $\overline{p} = 0$

Our proposed loss function represent as

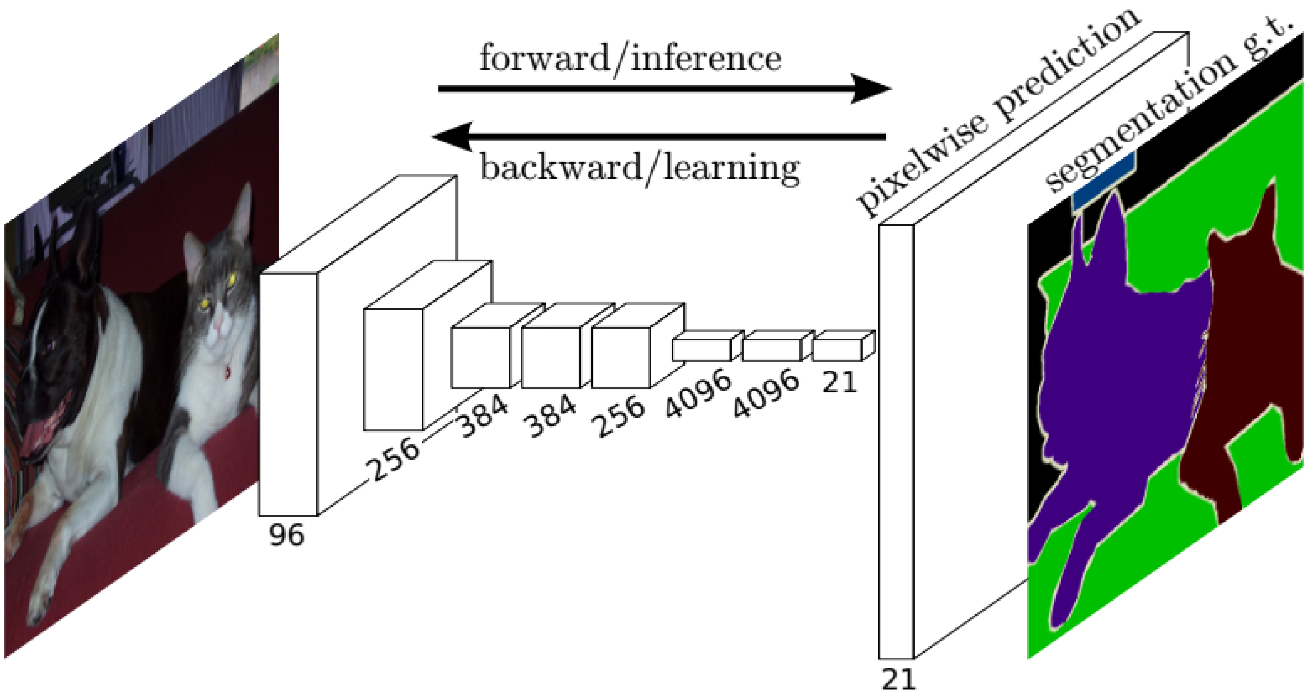
$$MultiLoss = \left(BCELoss + FocalLoss\right) + DiceLoss \quad (5)$$

Implementation in Code

Fully Convolutional Network

- Fully Convolutional Networks (FCNs) are a type of neural network designed for semantic segmentation tasks, where the goal is to predict pixel-wise class labels for an input image.
- FCNs replace fully connected layers with convolutional layers, allowing them to accept input images of varying sizes.
- Many FCN architectures incorporate skip connections, which concatenate feature maps from earlier layers to preserve spatial information during the upsampling process.
- FCNs use transposed convolutional layers (also known as deconvolution or fractionally strided convolution) for upsampling feature maps.
- The transposed convolution operation can be represented as $Y = X * W + b$, where X is the input, W is the transposed convolutional kernel, and b is the bias term.

Fully Convolutional Network



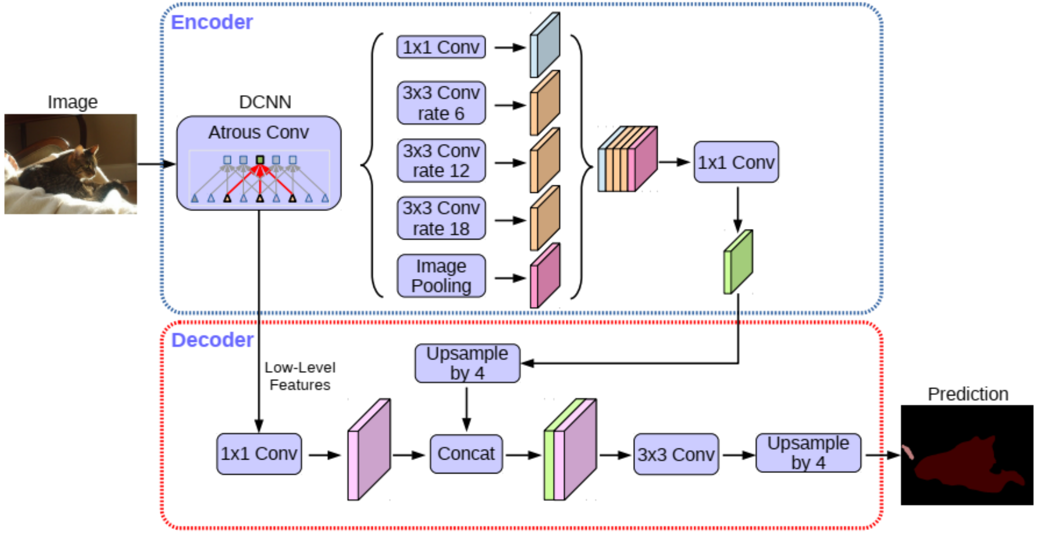
DeepLab with Atrous Convolution

- DeepLab integrates atrous (dilated) convolutions for semantic segmentation.

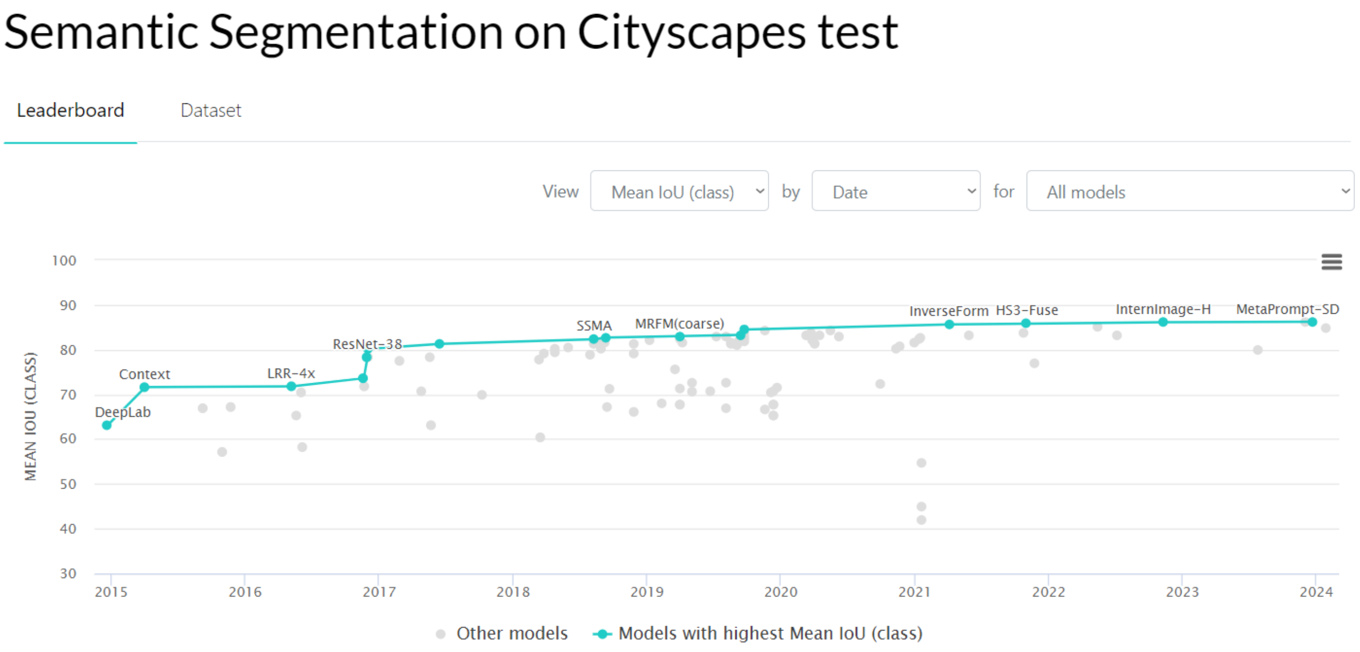
Atrous Convolution:

  - Atrous convolution introduces gaps in kernels, enhancing the receptive field without a parameter surge.
  - Mathematically, $Y = X * W$, where Y is the output tensor, X is the input, and W is the atrous convolutional kernel.

- **Multi-Scale Context:**
  - Atrous convolutions at various rates capture contextual information at different scales.
  - Beneficial for handling objects and structures of varying sizes.



Results



Semantic Segmentation on Cityscapes test
Leaderboard
Dataset
View
Mean IoU (class)
by
Date
for
All models
MEAN IOU (CLASS)
100
90
80
70
60
50
40
30
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
DeepLab
Context
LRR-4x
ResNet-38
SSMA
MRFM(coarse)
InverseForm
HS3-Fuse
InternImage-H
MetaPrompt-SD
Other models
Models with highest Mean IoU (class)

Latest Trends

- **Transformer-Based Architectures:**
  - Adoption of transformer-based models for image segmentation, leveraging their ability to capture long-range dependencies in images.
- **Self-Supervised Learning:**
  - Growing trend in self-supervised learning approaches, enabling segmentation models to learn from unlabeled data and reduce dependency on extensive labeled datasets.
- **Real-Time Segmentation:**
  - Increased emphasis on efficient segmentation models suitable for real-time applications, catering to the demands of robotics, autonomous vehicles, and augmented reality.
- **Domain Adaptation and Transfer Learning:**
  - Continued exploration of domain adaptation and transfer learning techniques, enhancing segmentation model performance across diverse datasets and domains.

Latest Trends...(contd)

Segment Anything Model (SAM):

SAM, short for Segment Anything Model, is an advanced and versatile segmentation model designed to address complex challenges in computer vision tasks.

Key Features:

Versatility:

- SAM exhibits exceptional versatility, demonstrating effectiveness in segmenting a diverse array of objects and structures within images.

Innovation in Approach:

- SAM employs innovative techniques, pushing the boundaries of traditional segmentation methods. It introduces novel approaches to handle intricate visual scenarios.

User-Friendly Interaction:

- SAM is user-friendly, featuring interactive and allows for user-guided segmentation capabilities. This allows users to actively participate in refining segmentation results based on specific requirements.

High Performance:

- SAM achieves high-performance levels, particularly in accurately delineating object boundaries. Its precision makes it well-suited for tasks demanding meticulous segmentation.

Challenges

Segmentation tasks face various challenges impacting accuracy and applicability.

1. **Ambiguity and Overlapping Objects:**
   Handling overlapped and ambiguously bounded objects requires advanced techniques.
2. **Lack of Annotated Data:**
   Limited annotated data impedes training robust segmentation models.
3. **Scale Variation:**
   Dealing with objects at different scales demands models capturing fine details and global context.
4. **Real-Time Processing:**
   Achieving real-time segmentation demands efficient models without compromising accuracy.
5. **Semantic Understanding:**
   Gaining a deep semantic understanding remains a challenge, requiring comprehension of object relationships.
6. **Adaptability to Diverse Domains:**
   Models must adapt to diverse domains with varying conditions and object types.

Case Study: Pin Detection

THANK YOU

Any questions?