Enhancing Image Segmentation: A Comparative Study

Aaryan, Aayush Kumar, Darpan Gaur, Varun Gupta
Indian Institute of Technology Hyderabad
{co21btech11001, co21btech11002, co21btech11004, cs21btech11060}@iith.ac.in

Abstract

Image segmentation is a crucial task in computer vision with significant applications in various fields such as autonomous driving, medical imaging, and object detection. Accurate segmentation allows for precise identification and localization of objects within an image, which is essential for tasks that require detailed scene understanding. In this project, we aim to compare the performance of different image segmentation models, including Convolutional Neural Networks (CNNs), Graph Convolutional Networks (GraphCNNs), and Transformers. Our goal is to evaluate their effectiveness and identify the strengths and weaknesses of each approach in the context of image segmentation. This literature survey was greatly inspired by [11].

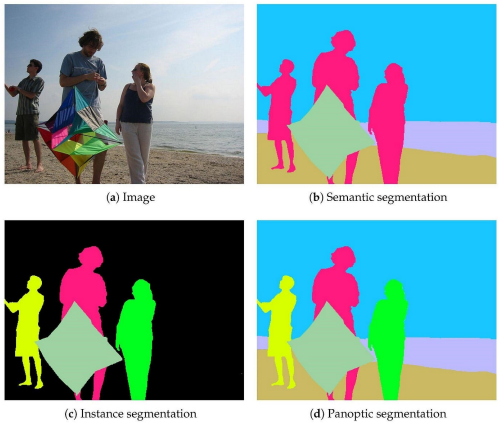

Figure 1. An Example showing the three types of segmentation tasks: Semantic Segmentation, Instance Segmentation, and Panoptic Segmentation. From [7]

.

1. Introduction

In Image Segmentation, there are three common types of segmentation tasks: *Semantic Segmentation*, *Instance Segmentation*, and *Panoptic Segmentation*. In Semantic Segmentation, the goal is to classify each pixel in the image into a predefined set of categories (such as ”car”, ”road”, ”tree”, ”person”, etc.). All pixels that belong to the same category are assigned the same label. Instance Segmentation identifies individual objects within the image and assigns a unique label to each object i.e., it distinguishes between different instances of the same category (such as two cars will be assigned different labels). Panoptic Segmentation combines both semantic and instance segmentation. It aims to provide a complete understanding of the scene by identifying both object instances and background areas. In this work, we focus on the task of Semantic Segmentation.

2. Problem Statement

This project aims to conduct a comprehensive comparative analysis of CNN, Transformer, and GNN-based approaches for image segmentation. These models will be evaluated using the dataset mentioned in the dataset section. By leveraging the strengths of these architectures, we seek to identify their respective advantages and shortcomings. In addition to evaluating these architectures, we will implement a series of experiments designed to improve segmentation outcomes. Also, use the above analysis to the ongoing Kaggle competition focused on segmenting vasculature in 3D scans of human kidneys [1].

3. Related Work

3.1. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have become a standard approach for image segmentation tasks. This section discusses three important CNN architectures for segmentation: FCN, UNet++ and DeepLabV3+, providing a mathematical overview based on the foundational principles and improvements proposed in their respective works.

3.1.1 Fully Convolutional Networks (FCN)

The Fully Convolutional Network (FCN) [8] was one of the pioneering architectures for dense predictions in seman-

tic segmentation. Unlike traditional CNNs, FCNs replace fully connected layers with convolutional layers, enabling the model to predict a segmentation mask with spatial dimensions corresponding to the input image. The core idea is to use convolutional and pooling layers to produce feature maps at various scales, followed by a deconvolution (upsampling) layer to restore the original image resolution.

The output of an FCN can be expressed as:

$$\hat{y} = f(W * X + b)$$

where W represents the learned convolutional weights, X is the input image, b is the bias term, and f is the activation function. Using convolutional layers throughout ensures the model can handle inputs of arbitrary sizes, and the final upsampling layer reconstructs the pixel-wise segmentation map.

3.1.2 UNet++

UNet++ [13] improves upon the original U-Net architecture by introducing nested and dense skip connections between the encoder and decoder paths, bridging the semantic gap between feature maps at different scales. This refinement reduces the complexity of the decoder and improves the network's ability to capture fine-grained details, particularly in medical imaging tasks.

The main innovation in UNet++ is the re-designed skip connections:

$$F_{i,j} = H_{i,j}\left(F_{i-1,j} + \sum_{k=0}^{j-1} H_{i,k}\right)$$

where $F_{i,j}$ represents the feature map at the i-th depth and j-th layer, and $H_{i,j}$ denotes the convolutional operations. The nested architecture allows for the gradual refinement of feature maps at each level, making it more effective in learning from complex and hierarchical data.

3.1.3 DeepLabV3+

DeepLabV3+ [4] builds on DeepLabV3 by integrating an encoder-decoder structure and employing atrous (dilated) convolutions to capture multi-scale context efficiently. Atrous convolution introduces the dilation rate, r, which controls the spacing between kernel elements. This allows the network to expand the receptive field without increasing the number of parameters.

The atrous convolution operation is defined as:

$$y[i] = \sum_k x[i + r \cdot k] \cdot w[k]$$

where x is the input, w represents the filter weights, and r is the dilation rate. DeepLabV3+ also includes an Atrous Spatial Pyramid Pooling (ASPP) module that applies multiple atrous convolutions with different rates, allowing the model to capture information at multiple scales. The decoder then upsamples the feature maps to generate the final segmentation mask.

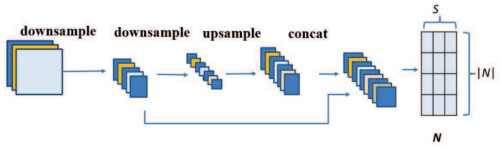

Figure 2. Graph-FCN node initialization. From [9].

3.2. Graph Neural Networks: Graph-FCN and CNN-G

Deep learning has made significant advancements in semantic segmentation by classifying pixels in images. However, during high-level feature extraction, it often ignores the local spatial information, which is important for accurate segmentation. Graph-based models address this limitation by including the missing local context.

3.2.1 Graph-FCN

Graph-FCN combines graph convolutional networks (GCNs) with fully convolutional networks (FCNs) to capture local spatial relationships in images. Initially, a convolutional network converts the image grid data into a graph structure, transforming the semantic segmentation task into a graph node classification problem. The graph convolutional network is then applied to classify the nodes in this graph, effectively solving the segmentation challenge.

FCN-16s generates feature maps, and node annotations are initialized by combining upsampled feature vectors and node locations. Labels for nodes are obtained by pooling the raw label image during training. The process is shown in Figure 2.

In the graph model, edges are represented by an adjacency matrix, where each node is connected to its nearest l nodes. These connections allow node annotations to transfer through the edges in the graph neural network.

FCN-16s handles node classification and graph model initialization on a small feature map, while a 2-layer GCN classifies the nodes within the graph. Cross-entropy loss is computed for both outputs, and like FCN-16s, Graph-FCN is trained end-to-end. The output of the prediction process is the output of the convolutional network. The graph model is only used during the training.

The structure of the Graph-FCN model is shown in Figure 3.

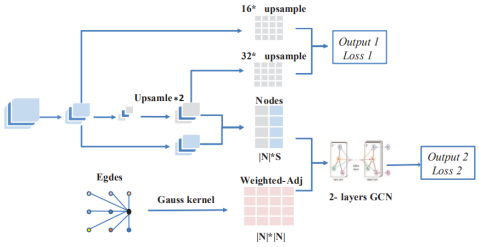

Figure 3. Graph-FCN model structure. From [9].

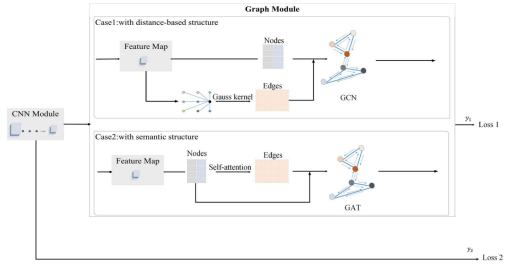

Figure 4. CNN-G model structure. From [10].

3.2.2 CNN-G

CNN-G builds on the Graph-FCN model by incorporating a graph-based approach with a convolutional neural network. Two types of structure models are used in CNN-G: distance-based and semantic-based, solved by GCN and GAT, respectively. The distance-based model captures the spatial relationships between nodes, while the semantic-based model focuses on the semantic relationships.

Using a graph attention network (GAT) enables flexible feature extraction across various receptive fields. This approach allows the model to integrate both structure learning and feature extraction.

Distance-Based Structure: Based on the assumption that the closer nodes are more correlated, the Gauss kernel function is used to generate the weighted edges.

The adjacent matrix $A = [e_{ij}]_{|N|\times|N|}$ is used to represent the edge set:

$$e_{ij} = \begin{cases} \exp\left(-\frac{\|p_i - p_j\|^2}{\sigma^2}\right), & \text{an edge between } n_i \text{ and } n_j \\ 0, & \text{otherwise} \end{cases}.$$

To simplify the calculation, we make the nodes connect to the l closest nodes.

Semantic-based model: The initial attention coefficient c_{ij} is used to measure the correlation between two nodes. The features of nodes with the same category will be more similar, the attention coefficient between each other will be larger, and the similarity is gradually strengthened in the iteration. A linear transformation is taken to map the concatenation of two nodes' features into a real number:

$$c_{ij} = a\left(n'_i || n'_j\right)$$

The final attention coefficient is obtained through the normalization of all neighborhood nodes:

$$e_{ij} = \frac{\exp(c_{ij})}{\sum_{k\in neib_i} \exp(c_{ik})}.$$

When the graph structure is unknown, the matrix $A_{\text{att}} = [e_{ij}]_{|N|\times|N|}$ can be taken as the adjacency matrix. In this case, the edge set is generated by the calculation of the attention coefficients.

In each case, the model generates two outputs, $y1$ and $y2$, which corresponds to two losses, $loss1$ and $loss2$. Both share the convolutional layer's extracted features. The final prediction is based on $y1$. Similar to Graph-FCN, the graph model is only used during training.

The structure of the CNN-G model is shown in Figure 4.

3.3. Transformer based approaches

3.3.1 TransUnet

TransUNet [3] has emerged as a robust framework for medical image segmentation by combining the strengths of U-Net and Transformer architectures.

TransUNet addresses these limitations by introducing a hybrid architecture that leverages both CNNs and Transformers. The encoder consists of a CNN followed by a Transformer module. The CNN is used to extract feature maps from the input images, which are then tokenized into image patches and fed into the Transformer. This CNN-Transformer hybrid allows TransUNet to capture detailed spatial features (via CNN) and global context (via the Transformer) simultaneously.

In the decoder, a cascaded upsampling module (CUP) is employed, consisting of multiple upsampling blocks with convolutional layers and ReLU activation. These upsampled features are combined with the high-resolution CNN feature maps from the encoder through skip connections, similar to the U-Net structure, enabling precise localization. This U-Net-like design ensures that TransUNet can recover lost spatial detail while preserving the high-level semantic understanding captured by the Transformer.

3.3.2 Seg Former

SegFormer [12] framework consists of two main modules: A hierarchical Transformer encoder to extract coarse and fine features; and a lightweight All-MLP decoder to directly fuse these multi-level features and predict the semantic segmentation mask.

- **Hierarchial Transformer Encoder**: designed to generate multi-scale features by progressively reducing the spatial resolution through a series of Mix Transformer (MiT) blocks. This encoder eliminates the need for positional encodings, making it adaptable to varying image sizes. Each MiT block employs overlapping patch embeddings and an efficient self-attention mechanism that reduces computational complexity, allowing it to handle high-resolution inputs. By producing features at multiple resolutions—1/4, 1/8, 1/16, and 1/32 of the original size—it captures both local and global contexts for improved segmentation of complex objects.
- **Lightweight All-MLP decoder**: It consists solely of MLP layers, which reduces computational overhead. The decoder first unifies the channel dimensions of the multi-level features, upsampling them to a common resolution before concatenation. This process enables the decoder to leverage fine-grained details and contextual information, ultimately generating a precise semantic segmentation mask. The lightweight design of the MLP decoder ensures high accuracy while maintaining low latency, making SegFormer suitable for real-time applications.

3.3.3 Mask Former

Mask Former [5] architecture consists of three main components:

- **Pixel-Level Module**: The pixel-level module extracts per-pixel embeddings from the image features. A backbone network (e.g., ResNet or Swin Transformer) first processes the input image to obtain low-resolution feature maps. These feature maps are then gradually upsampled using a lightweight pixel decoder based on the Feature Pyramid Network (FPN) to produce high-resolution, per-pixel embeddings.
- **Transformer Module**: The transformer module uses a standard decoder with learnable queries to compute global embeddings for each segment in the image. It takes the pixel-level features and processes them with multiple transformer decoder layers (6 by default). Each query represents a potential segment and generates a per-segment embedding, which encodes global information about the predicted mask.
- **Segmentation Module**: The segmentation module uses the segment embeddings from the transformer to produce a set of binary mask predictions and class labels. Each segment embedding is transformed into a mask embedding via a small Multi-Layer Perceptron (MLP). The mask embeddings are then combined with the per-pixel embeddings from the pixel-level module to generate binary masks using a dot-product operation. Finally, each binary mask is associated with a single global class prediction, and a classification loss is applied to these predictions.

Figure 5. An example image from the PASCAL VOC dataset. From [2]

Figure 6. An example image from the Cityscapes dataset. From [6]

4. Datasets

There are a lot of 2D image datasets available for various purposes. Below is a list of some of the most popular:

- **PASCAL Visual Object Classes (VOC)** is a dataset for object detection, segmentation, and classification. For the segmentation task, there are 21 labeled object classes and pixels are labeled as background if they do not belong to any of these classes. The dataset is divided into two sets, training and validation, with 1,464 and 1,449 images, respectively. An example image from the dataset is shown in Figure 5.
- **Cityscapes** contains a diverse set of stereo video sequences recorded in street scenes from 50 cities, with high-quality pixel-level annotation of 5,000 frames, in addition to a set of 20,000 weakly annotated frames. It includes semantic and dense pixel annotations of 30 classes, grouped into 8 categories — flat surfaces, humans, vehicles, constructions, objects, nature, sky, and void. An example image from the dataset is shown in Figure 6.

5. Evaluation Metrics

5.1. Pixel Accuracy

Pixel accuracy is the ratio of correctly classified pixels divided by the total number of pixels. For $K+1$ classes (K foreground classes and the background) pixel accuracy is defined as

$$\text{PA} = \frac{\sum_{i=0}^{K} p_{ii}}{\sum_{i=0}^{K} \sum_{j=0}^{K} p_{ij}}$$

where p_{ij} is the number of pixels of class i predicted as belonging to class j.

5.2. IoU Score

Intersection over Union (IoU) is defined as the ratio of the area of intersection between the predicted segmentation map A and the ground truth map B to the area of their union.

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|}.$$

5.3. Dice Score

Dice score is defined as twice the area of overlap between the predicted (map A) and ground-truth maps (map B), divided by the total number of pixels in both maps.

$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|}.$$

References

[1] Sennet + hoa - hacking the human vasculature in 3d. Kaggle competition Link. 1

[2] PASCAL VOC available online here. 4

[3] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation, 2021. 3

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 801–818, 2018. 2

[5] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation, 2021. 4

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016. 4

[7] Abhishek Jain. Semantic vs instance vs panoptic segmentation on medium Link. 1

[8] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1

[9] Yi Lu, Yaran Chen, Dongbin Zhao, and Jianxin Chen. Graph-fcn for image semantic segmentation, 2020. 2, 3

[10] Yi Lu, Yaran Chen, Dongbin Zhao, Bao Liu, Zhichao Lai, and Jianxin Chen. Cnn-g: Convolutional neural network combined with graph for image segmentation with theoretical analysis. *IEEE Transactions on Cognitive and Developmental Systems*, 13(3):631–644, 2021. 3

[11] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):3523–3542, 2022. 1

[12] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers, 2021. 3

[13] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. *Deep learning in medical image analysis and multimodal learning for clinical decision support*, 2019. 2