

A Brief Perspective on Deep Learning Approaches for 2D Semantic Segmentation

Shazia Sulemane ¹, Nuno Fachada ^{1,2,*} and João P. Matos-Carvalho ^{2,3}

¹ Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI), Lusófona University, 1749-024 Lisboa, Portugal; p6816@ulusofona.pt

² Center of Technology and Systems (UNINOVA-CTS) and Associated Lab of Intelligent Systems (LASI), 2829-516 Caparica, Portugal

³ LASIGE, Faculdade de Ciências, Universidade de Lisboa, 1749-016 Lisboa, Portugal; jpecarvalho@ciencias.ulisboa.pt

* Correspondence: nuno.fachada@ulusofona.pt

Abstract

Semantic segmentation is a vast field with many contributions, which can be difficult to organize and comprehend due to the amount of research available. Advancements in technology and processing power over the past decade have led to a significant increase in the number of developed models and architectures. This paper provides a brief perspective on 2D segmentation by summarizing the mechanisms of various neural network models and the tools and datasets used for their training, testing, and evaluation. Additionally, this paper discusses methods for identifying new architectures, such as Neural Architecture Search, and explores the emerging research field of continuous learning, which aims to develop models capable of learning continuously from new data.

Keywords: recurrent neural networks; transformer; convolutional neural networks; graph convolutional networks; encoder–decoder model; feature pyramid networks; generative adversarial networks; attention networks; neural architecture search; continuous learning



Academic Editor: Juvenal Rodriguez-Resendiz

Received: 15 May 2025

Revised: 1 July 2025

Accepted: 15 July 2025

Published: 18 July 2025

Citation: Sulemane, S.; Fachada, N.; Matos-Carvalho, J.P. A Brief Perspective on Deep Learning Approaches for 2D Semantic Segmentation. *Eng* **2025**, *6*, 165. <https://doi.org/10.3390/eng6070165>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Semantic segmentation is an image analysis technique that assigns a label to every pixel in an image to obtain pixel-accurate masks of all represented objects [1]. This approach improves on object detection methods that frame objects within a rectangle, which have larger errors, finding applications in fields such as medicine, autonomous driving, and video surveillance. For instance, medical imaging serves as a non-invasive diagnostic technique that produces images of the internal structures of a patient's body. These images can be analyzed using semantic segmentation to aid medical professionals in diagnosing conditions such as cancer [2], as illustrated in Figure 1.

Autonomous driving, on the other hand, involves the integration of sensors and software, including segmentation networks, into a vehicle to actively identify and segment objects, pedestrians, traffic signs, and other relevant elements. The ultimate objective is to achieve fully autonomous vehicles [3]. Similarly, video surveillance, commonly referred to as Closed-Circuit Television (CCTV), involves the real-time recording of individuals, locations, and events. The use of artificial intelligence in these systems improves the detection of individuals and suspicious activities. However, it also raises concerns regarding privacy, civil liberties, and the potential misuse of this technology [4,5].

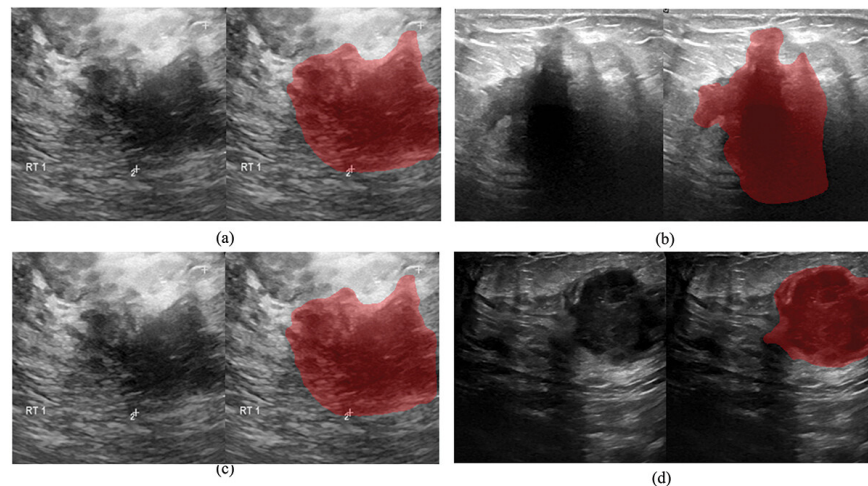


Figure 1. Example of a breast cancer segmentation result, using DeepLabV3+ with (a) DarkNet53, (b) SqueezeNet, (c) EfficientNet-b0, (d) DarkNet19. Reproduced from Deepak and Bhat [6], licensed under CC BY-NC 4.0.

Currently, machine learning algorithms and techniques form the basis of most semantic image segmentation solutions. Traditional methods, which include thresholding [7], region-growing [8], watershed segmentation [9], Conditional and Markov Random Fields (CRFs) [10], active contours [11], and edge detection methods [12], are sometimes incorporated into artificial intelligence solutions but are typically less accurate and adaptable to light changes and intraclass differences compared to deep learning methods [13]. Intraclass differences are particularly important for object identification, as objects can vary widely in shape, size, and texture.

Semantic segmentation can be improved by including instance segmentation, which differentiates multiple instances of an object in an image [14]. This technique is known as panoptic segmentation and combines semantic and instance segmentation to obtain a more comprehensive description of an image. Panoptic segmentation presents additional challenges since it requires accurate semantic and instance segmentation simultaneously [15,16], as exemplified in Figure 2.

Object detection using semantic segmentation can be computationally intensive, making it challenging to use in real-time or in video applications. Video Semantic Segmentation (VSS) introduces additional challenges because it needs to produce consistent results over time while dealing with changing lighting conditions, possible occlusions, and object movement. Furthermore, semantic segmentation requires significant time to label each image, adding to the computation time [17]. To address these challenges, researchers have been exploring semi-supervised annotation methods to train their networks more efficiently [18]. One such model is SimCLRv2 [19], which pre-trains a ResNet model with unlabeled data in a task-agnostic way, followed by training with a small amount of labeled data, and finally training with unlabeled data in a task-specific way, as depicted in Figure 3. In addition to SimCLRv2, other techniques such as pseudo-labeling [20], consistency regularization [21], and teacher–student networks [22] have been proposed for semi-supervised semantic segmentation, each aiming to reduce the need for labeled data while preserving model performance.

Despite the abundance of research on semantic segmentation and panoptic segmentation for images, there are fewer examples of networks designed for their video equivalents: VSS [23] and Video Panoptic Segmentation (VPS) [15]. VPS has been less popular, likely due to the scarcity of appropriately labeled datasets for this task, which are not as readily available as image segmentation datasets. An example of video semantic segmentation is shown in Figure 4.



Figure 2. Example of semantic, instance, and panoptic image segmentation: (a) the original image; (b) semantic segmentation, with no distinction between each individual; (c) instance segmentation, distinguishing between individuals, but no segmentation of the background; and (d) panoptic segmentation, distinguishing between individuals while performing background segmentation. Reproduced from Jung et al. [24], licensed under CC BY 4.0.

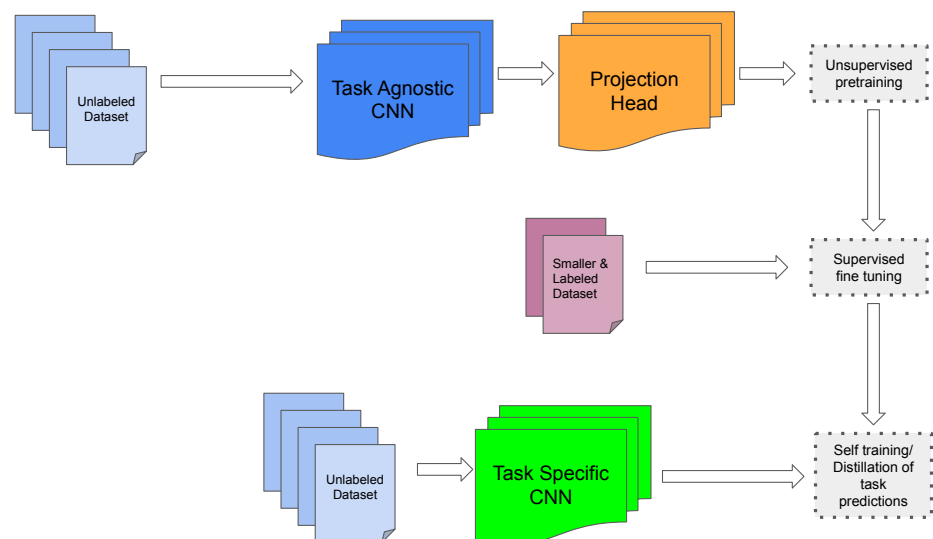


Figure 3. SimCLRv2, a model where deep semi-supervised learning is applied. The ResNet model is pre-trained with unlabelled data, then trained with a small amount of labeled data, and finally trained for a more specific task with unlabelled data.

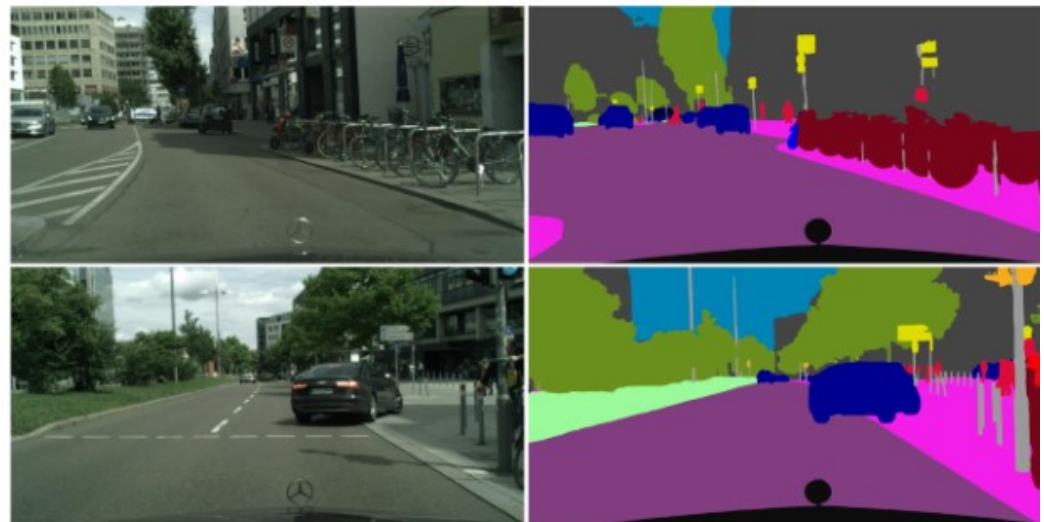


Figure 4. Ground-truth labels of objects in different video frames. Adapted from Portillo-Portillo et al. [25], licensed under CC BY 4.0.

This paper presents a brief perspective of deep learning 2D semantic segmentation techniques, discussing diverse approaches critical to this field of research. Although many recent models integrate components from multiple architectural paradigms—such as encoder–decoder structures enhanced with attention mechanisms or hybrid convolutional–transformer designs—we adopt a modular organization centered on primary architectural families for simplicity and historical continuity. In Section 2, we start by highlighting the fundamentals of Fully Convolutional Neural Networks (FCNs), introducing core concepts and analyzing various convolution types such as atrous and transposed convolution [26–28]. In Section 3, we introduce Graph Convolutional Networks (GCNs), examining their functionalities and inherent limitations. In Section 4, we introduce fundamental encoder–decoder models, offering insights into their structures, with special emphasis on U-Nets [29]; this is followed by Section 5, where Feature Pyramid Networks (FPNs) are discussed. Recurrent Neural Networks (RNNs) are introduced in Section 6. Section 7 describes Generative Adversarial Networks (GANs), and Section 8 introduces Attention-Based Networks (ABNs). We focus on Neural Architecture Search (NAS) in Section 10, followed by a detailed discussion on Continuous Learning in Section 11. Section 12 presents several important datasets of the different deep learning architectures, covering both 2D and 2.5D cases. Section 13 addresses key implementation considerations, including software frameworks, hardware constraints, and optimization techniques relevant to training segmentation models. Metrics for assessing algorithm accuracy and their respective drawbacks are discussed in Section 14. Section 15 looks into the accuracy of several state-of-the-art models on various datasets, discussing architectural reasons for performance increases over time, highlighting possible paths for additional improvement. In Section 16, several very recent trends in semantic segmentation are analysed, hinting at potential future paths in this field. This paper closes with Section 17, in which we offer some conclusions.

2. Fully Convolutional Neural Networks

Convolutional Neural Networks (CNNs) [30] are a popular type of neural network architecture, especially for tasks related to image processing. CNNs use convolutional layers to apply convolutions to input images, as demonstrated in Figure 5.

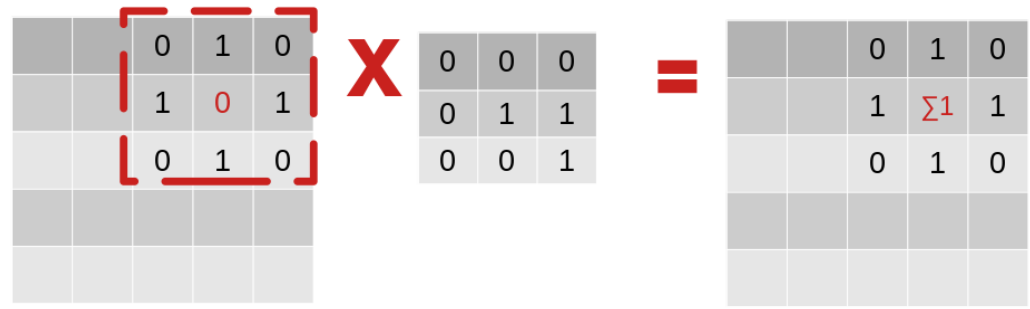


Figure 5. Example of a convolutional layer operation, where a convolutional 3×3 filter is applied to the center pixel highlighted in red. The filter sums the multiplications between the area lined in red (on the left) and the filter (center), which results in the center pixel changing value from 0 to 1.

The convolutional layer operation is defined by Equation (1), where $\mathbf{X}$ is the input feature map, $\mathbf{K}$ is the convolutional kernel, $\mathbf{Y}$ is the output feature map, f is the activation function (such as Rectified Linear Unit, or ReLU), b is the bias term, and M and N are the dimensions of the convolution kernel.

$$\mathbf{Y}_{i,j} = f \left(\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \mathbf{X}_{i+m,j+n} \cdot \mathbf{K}_{m,n} + b \right) \quad (1)$$

Note that Equation (1) assumes that the input feature map and the convolutional kernel have the same number of channels. When input and output channel dimensions differ, convolutional layers apply multiple filters, where each filter spans all input channels and produces one output channel, enabling the network to learn diverse feature representations [31].

Apart from convolutional layers, CNNs often include non-convolutional layers such as pooling layers, which simplify the feature maps and activation functions, allowing the network to model non-linear functions. The most commonly used types of pooling layer are max and average [31,32]. Equations (2) and (3) describe the operation of max and average pooling layers, respectively, where $\mathbf{X}$ is the input feature map, $\mathbf{Y}$ is the output feature map, M and N are the dimensions of the pooling kernel, and s is the stride (i.e., the distance between adjacent pooling operations). Each $s \times s$ pooling kernel slides over the input feature map, replacing each region with its maximum or average value [33].

$$\mathbf{Y}_{i,j} = \max_{m=0}^{M-1} \max_{n=0}^{N-1} \mathbf{X}_{i+s+m,j+s+n} \quad (2)$$

$$\mathbf{Y}_{i,j} = \frac{1}{M \cdot N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \mathbf{X}_{i+s+m,j+s+n} \quad (3)$$

There are other types of pooling layers. For example, in L2 pooling [34], each $s \times s$ patch of the input feature map is replaced with the L2 norm of the values in that patch. Similarly, a stochastic pooling layer [35] replaces each $s \times s$ patch of the input feature map with a randomly selected value from that patch, with the probability of each value being selected proportional to its magnitude. Finally, global pooling layers [36], such as global max or global average, operate on the entire feature map and produce a single output value, which creates lighter architectures. All of these components enable CNNs to achieve state-of-the-art performance on a variety of computer vision tasks.

FCNs [1] are a variation of CNNs that consist solely of convolutional layers. This architecture preserves spatial information by allowing the output to be the same size as the input image. However, FCNs typically have significantly more weight parameters than traditional CNNs, since the convolutional layers in FCNs have to learn more complex and

diverse features from the input data. Despite this drawback, FCNs are powerful tools for various image segmentation and dense prediction tasks.

2.1. Atrous Convolution

In addition to the standard convolutions, another technique called atrous or dilated convolution has gained popularity [26,27]. The purpose of atrous convolution is to address the issue of losing spatial information during pooling stages in a traditional convolutional network. By introducing “holes” into the convolutional kernel, as shown in Figure 6, atrous convolutions upsample the filter kernels, allowing for the recycling of pre-trained models to extract more detailed feature maps [27]. Specifically, the last pooling layers in a model can be replaced with atrous convolutional layers [37]. Dilated convolutions also increase the network’s receptive field, enabling it to learn more detailed features in the inputs without adding extra learnable parameters [38]. Dilated convolutions are commonly used in various applications, such as image segmentation [26], object detection [38], and video processing [39].

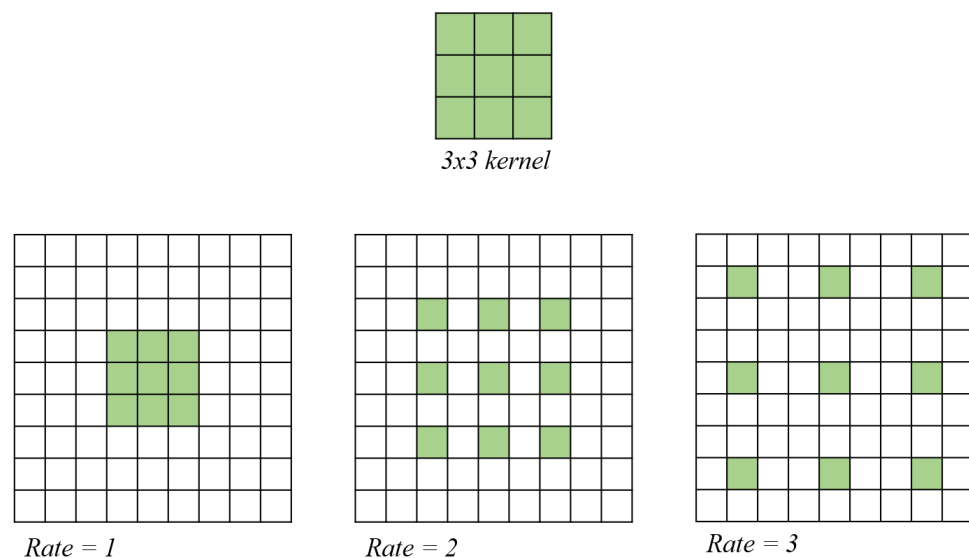


Figure 6. Atrous convolutional layer. The kernel uses the same functions as a convolutional kernel but includes a new rate parameter. This parameter determines how distant the kernel cells are from each other and performs the convolutional operation with the represented pixels in green, depending on the rate value.

2.2. Transposed Convolution

Transposed Convolution [28], also known as fractionally strided convolution or deconvolution, operates in the opposite direction of regular convolution and is typically used in the decoder layers of autoencoders or segmentation networks to increase the spatial resolution of feature maps. The purpose of transposed convolution is to recover spatial resolution by upsampling feature maps, helping to reverse the downsampling effects of pooling layers. While it can help control model complexity, the number of learnable parameters depends on how it is implemented within the network architecture [40]. This process is highlighted in Figure 7.

Despite its many advantages, one of the main problems with transposed convolution is the checkerboard problem. This arises from uneven overlap patterns during the operation, which can occur due to the interaction between kernel size, stride, and padding—even when the kernel size is divisible by the stride—leading to some output pixels being updated more frequently than others. As a result, some pixels may be updated multiple times while others are not, creating a checkerboard pattern in the output feature map [41]. This can limit

the network's capacity to recreate photo-realistic images. Several techniques have been proposed to solve this problem. One solution is to use the pixel transpose convolutional layer proposed by Gao et al. [42], which creates intermediate feature maps of the input feature maps in sequence. This adds dependencies between adjacent pixels in the final output feature map, which helps avoid the creation of checkerboard artifacts.

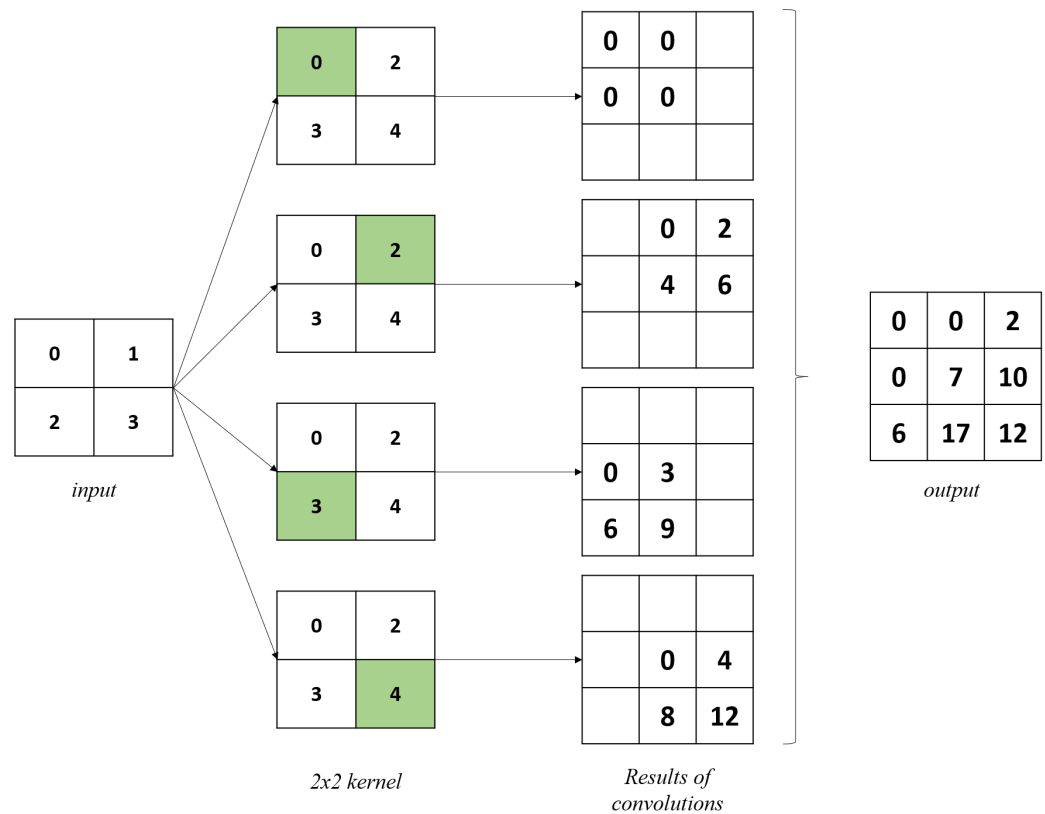


Figure 7. Transposed convolutional layer. As pictured, a transposed convolution operation multiplies each kernel cell with each pixel value in the input space and organizes it in a $(n + 1) \times (n + 1)$ space, and whichever values occupy the same output space are summed.

3. Graph Convolutional Networks

GCNs [43] are a type of CNN used for analyzing data represented in graphs. Graph data structures can capture complex relationships present in the data. However, the non-Euclidean geometry of graph-structured data presents a significant challenge for obtaining deep insights about the information it contains. Traditional graph [44–46] and network embedding methods [47,48] attempt to address this issue by studying the low-dimensional characteristics of graphs. However, these methods can suffer from shallow learning and may not capture the complex nature of the data [49].

To overcome these limitations, deep learning approaches like GCNs have been developed to enable a better understanding of graph-structured data. GCNs extend CNNs to non-Euclidean geometry, which is necessary for analyzing data represented in graphs that lack a regular structure [50]. GCNs work by applying convolutional filters to node features through message passing. In this process, each node in the graph aggregates information from its neighbors, transforms the information, and then sends it back to its neighbors [49]. This enables GCNs to learn representations that capture the underlying patterns and relationships within the graphs. One such GCN for semantic segmentation is DGCNet [51], as represented in Figure 8.

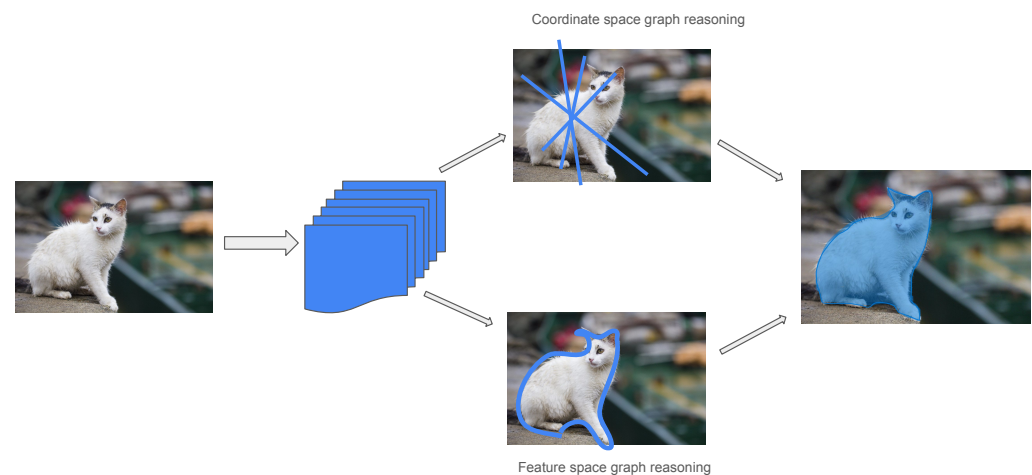


Figure 8. DGCNet, an example of a GCN created for semantic segmentation. The model consists of two GCN branches that propagate information in both spatial and channel dimensions of a convolutional feature map.

By leveraging these representations, GCNs can solve a wide range of problems in various domains, including social networks [52], biology [53], recommender systems [54], drug development [55], and text classification [56].

4. Encoder–Decoder Models and U-Nets

Encoder–decoder models (Figure 9) consist of two key components: an encoder, which applies a function of $z = f(x)$ to the input, and a decoder, which predicts the output y from z . The encoder extracts semantic information from the input, which is useful for predicting the output, such as pixel-wise masks. Auto-encoders are a special type of encoder–decoder model, where the decoder attempts to reconstruct the original input from the lower-dimensional representation given by the encoder.

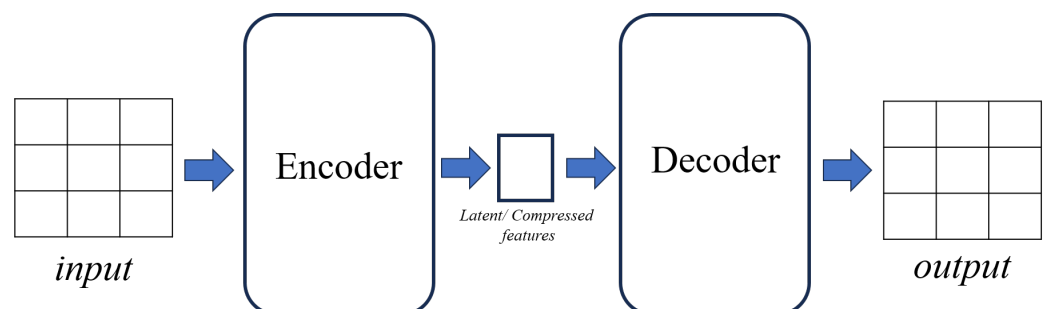


Figure 9. Encoder–decoder networks architecture. The architecture and what functions it applies to the inputs are varied. This figure describes an input image that goes through an encoder, into other possible operations of feature compression before passing through the decoder network.

U-Nets, illustrated in Figure 10, are a type of encoder–decoder architecture characterized by an encoder for feature extraction and a decoder that mirrors the encoder's structure. The decoder consists of transposed convolutions and skip connections, which link corresponding layers in the encoder and decoder to preserve spatial information.

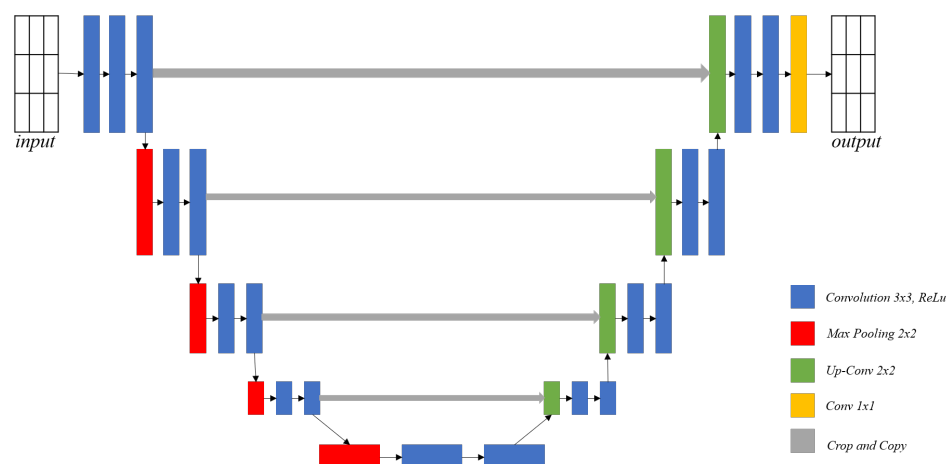


Figure 10. The U-Net architecture is composed of a decoder that behaves similarly to an auto-encoder. The input passes through several convolutional and pooling layers, and each level in the architecture sends their respective feature maps to the layer symmetrically opposed, which, through a series of up convolutions, tries to recreate the input using both of the available feature maps.

U-Nets are commonly used in medical imaging [29], displaying exceptional efficiency in segmentation and classification tasks—including cell tracking and radiography [57]—and high precision when distinguishing between lesions and organs. Figure 11 shows the use of a U-Net architecture in retinal birefringence scanning segmentation tasks. In addition, U-Nets have been studied in combination with other methods—such as attention modules and residual structures—to further enhance their performance. Attention U-Net [58], for instance, has been successful in accurately detecting smaller organs like the pancreas, while RU-Net [59] has fewer parameters, improving performance. TransUNet [60], on the other hand, integrates transformers and CNNs into the encoder to utilize medium and high-resolution feature maps, which help to maintain more fine-grained information about the images.

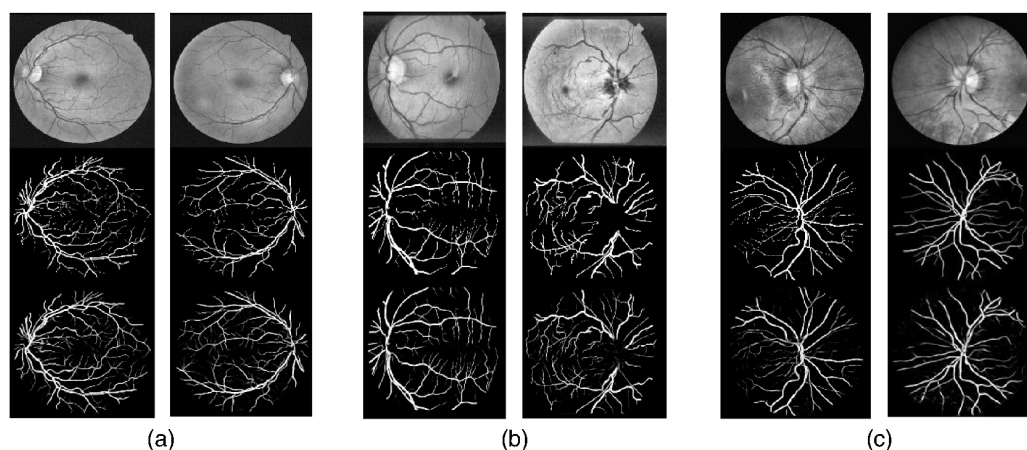


Figure 11. Examples of different output images from R2U-Net model for Retinal Birefringence Scanning (RBS) segmentation in three datasets: (a) DRIVE dataset; (b) STARE dataset; and, (c) CHASE_DB1 dataset. For all examples, the first row displays the input images, the second row depicts the ground-truth images, and the third row shows the results. Adapted from Alom et al. [61], licensed under CC BY 4.0.

5. Feature Pyramid Networks

FPNs, as shown in Figure 12, were developed to reduce the computational costs associated with pyramid representations in object detectors [62]. FPN leverages the inherent multi-scale and pyramidal hierarchy found in deep convolutional networks, enabling the

creation of a feature pyramid with significantly reduced computational demands. This approach also facilitates object recognition across a wide range of scales, thanks to the pyramid-scaled feature maps. FPN comprises the following key components [62]:

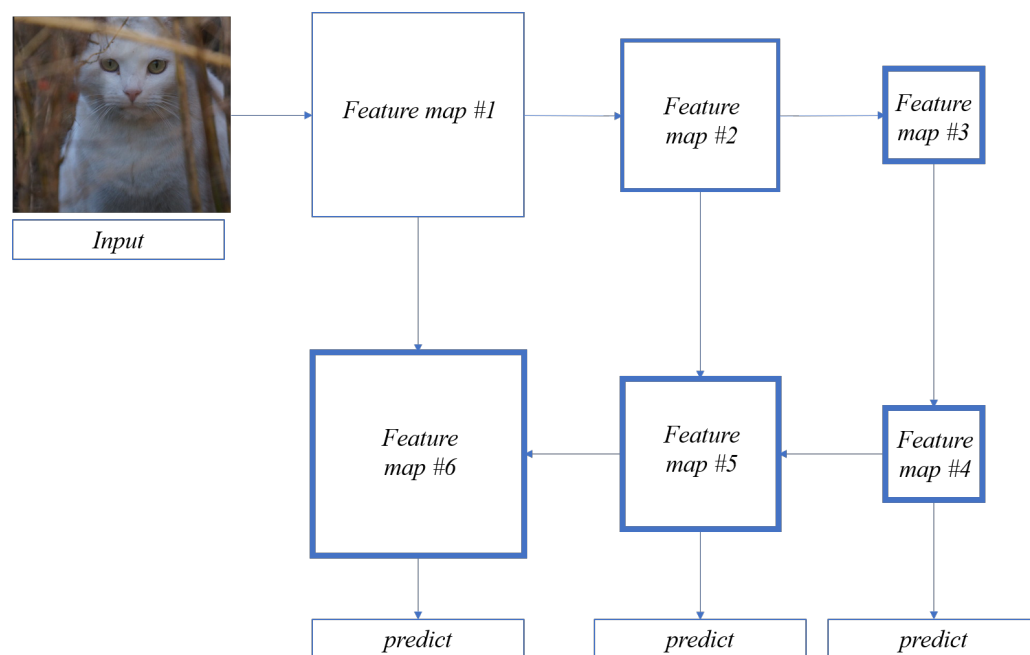


Figure 12. Feature Pyramid Network architecture. The input passes through each layer from #1 to #6 consecutively through the bottom-up pathways, which are represented horizontally, creating the respective feature maps. In this image, the stronger the features in each map, the bolder the borders of the squares. Besides this pathway, the input also travels laterally, as represented by the vertical arrows between each layer. In the final layers, the input is fed through the top-down pathway, going right to left in this image; in addition to feeding into the next layer, each stage also receives the feature maps from the lateral connections that send the superficial feature maps.

- The bottom-up pathway, responsible for the feed-forward computation of the convolutional network's backbone. This pathway aggregates feature maps from multiple scales, scaling them by a factor of 2 to establish the feature hierarchy.
- Top-down pathways, which generate higher-resolution features by upsampling feature maps with lower spatial resolutions but richer semantic content from higher pyramid levels.
- Lateral connections, which merge maps from both the bottom-up and top-down pathways, ensuring that maps with matching spatial sizes are combined. The bottom-up pathway provides detailed information about lower-level semantics and precise object locations, while the top-down pathway contributes to higher-level semantic context. This merging process involves element-wise addition, followed by a 3×3 convolution to reduce aliasing effects resulting from upsampling.

By employing these techniques, the network creates a feature pyramid with rich semantics at all levels. This feature pyramid finds applications across various domains and serves as a versatile network that can replace traditional featured image pyramids without introducing additional computational burdens, compromising speed, or increasing memory usage [63,64].

6. Recurrent Neural Networks

Processing sequential data, such as speech, videos, or text, requires the network to remember previous data. This allows the network to maintain an understanding of the

context and relationships between the inputs and classify the action or intent behind the data. RNNs [65] are commonly used when the data need this type of analysis, as they can process sequential data with a variable length. Figure 13 depicts a simplified representation of an unfolded RNN.

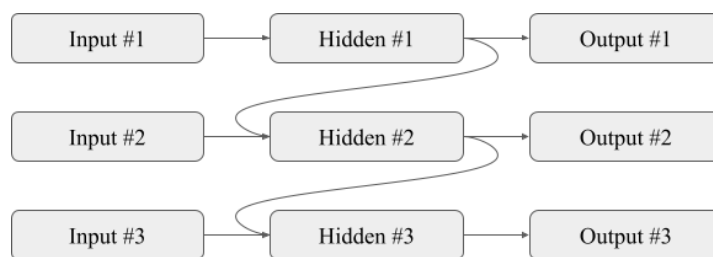


Figure 13. Simplified representation of an unfolded RNN. Each hidden layer has n hidden units that are connected through time recurrently.

RNNs work by recurrently feeding information regarding the previous input states into the following processing states, which allows the end nodes to analyze data regarding all previous inputs and their relationships with minimal loss of context. However, simple RNNs have some limitations. For example, vanishing gradients can make it difficult to learn long-term dependencies, and short-term memory makes them less effective for longer chains of data that need to be analyzed together [66]. To overcome these limitations, two specialized architectures were created: Long Short-Term Memory (LSTM) [67] and Gated Recurrent Unit (GRU) [68]. An LSTM block is shown in Figure 14.

LSTMs and GRUs can learn the relationships between different inputs of sequential data while solving the short-term memory problem using gates. These gates can learn which information to add or remove from the hidden state, allowing them to selectively keep or discard relevant information.

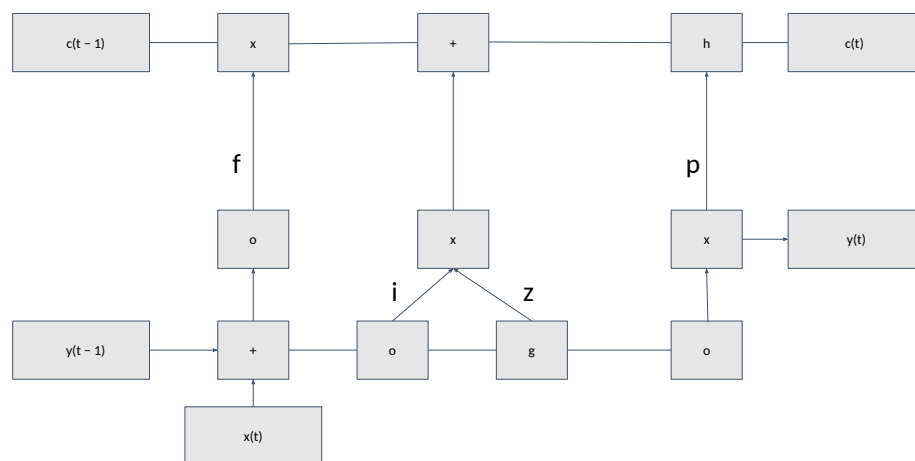


Figure 14. LSTM block, where the σ represents the sigmoid activation function and g and h represent input activation functions, usually tanh.

RNNs have been successfully used in many applications, including speech recognition [69], language modeling [70], machine translation [68], and sentiment analysis [71], to name a few. Their ability to process sequential data with a variable length makes them a powerful tool for many real-world problems.

The ReSeg model is an illustrative application of RNNs in semantic segmentation [72]. This approach incorporates ReNet layers, each consisting of four RNN modules that analyze

the image from different directions (up-down, left-right, and vice-versa). The overall network architecture combines a VGG-16 network (a CNN typically used for image recognition and classification tasks) [73], which takes input images, with ReNet layers. Subsequently, the processed information passes through upsampling and softmax layers to generate the final object mask, as depicted in Figure 15.

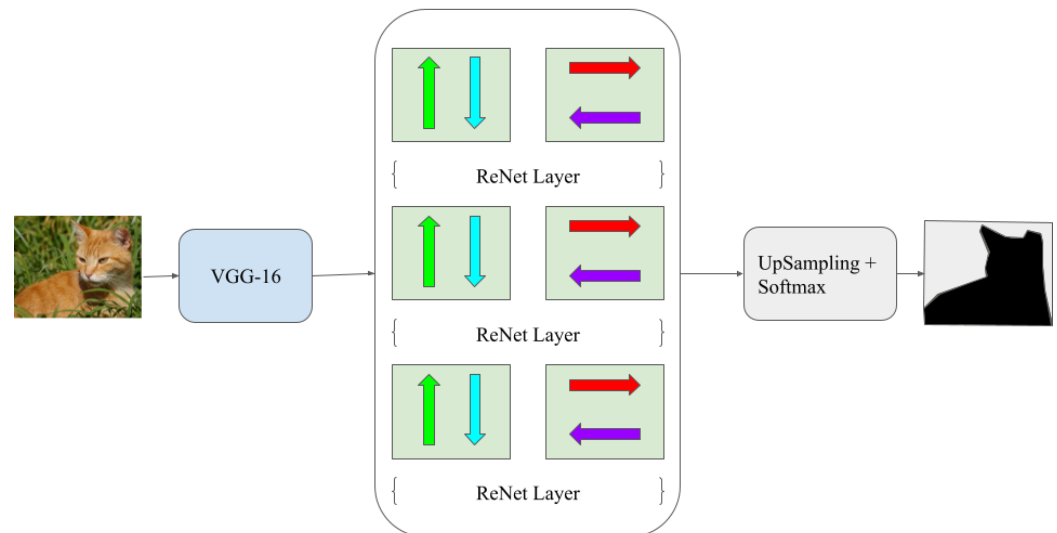


Figure 15. ReSeg network. The input image is preprocessed through a pre-trained VGG-16 network and fed through consecutive ReNet layers, followed by upsampling and non-linear softmax. The ReNet layers use two RNNs. The first RNN combs through the image pixel values in an up-down pattern, and the second does the same but reads the image sideways. Both feature maps are then fed through two more ReNet layers which behave in the same way, followed by the upsampling and softmax layers.

7. Generative Adversarial Networks

GANs [74] are a type of neural network composed of two components: a generator network G and a discriminator network D , as represented in Figure 16. The generator network creates new data samples, such as images, while the discriminator network learns to distinguish between real data and data created by the generator. The two networks are trained in a minimax game, where the generator minimizes the loss of generating realistic samples while the discriminator maximizes its ability to distinguish real from fake data.

GANs have shown great promise in various machine learning tasks, including unsupervised learning [75], semi-supervised learning [76], and reinforcement learning [77]. For example, in unsupervised learning, GANs have been used for image synthesis [78], data augmentation [79], and video prediction [80]. However, GANs also have some limitations, such as instability during training [81], mode collapse [82], and difficulty in controlling the output [83].

GANs have been applied to the problem of insufficient data for certain image segmentation problems. For example, Andreini et al. [84] created a GAN model trained to generate synthetic data of bacterial colonies in agar plates to train a CNN to segment the bacteria from the background, as illustrated in Figure 17. Besides being useful for the creation of new data from smaller datasets, GANs can also be used for optimizing the initial weights for certain networks; a case in point, Majurski et al. [85] used a GAN trained on unlabeled data, transferred the values of the resulting weights into a U-Net, and finally re-trained the U-Net with a small amount of labeled data, thus improving the overall performance compared to a baseline U-Net. This overall methodology is summarized in Figure 18.

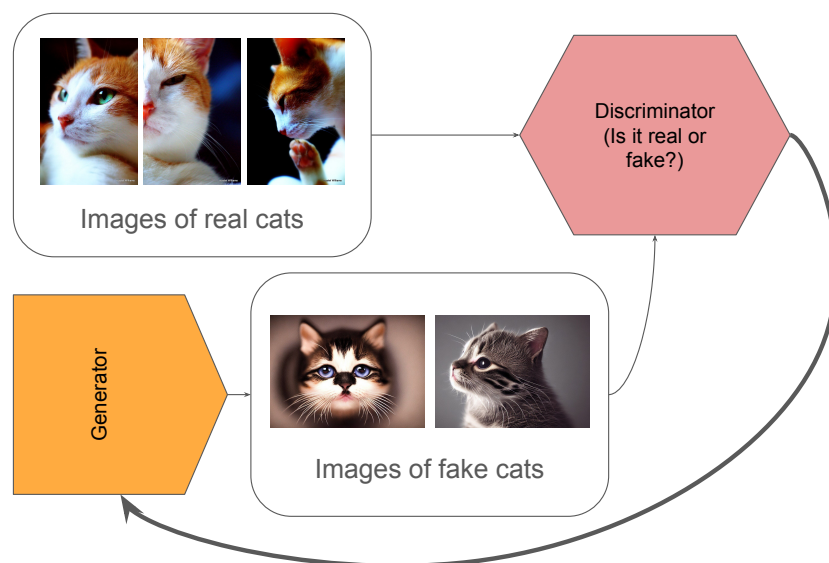


Figure 16. GAN network simplified. The network is made of two components: the discriminator and the generator. The discriminator receives two sets of inputs: real images and fake images generated by the generator, and tries to differentiate one type from the other. After this, the generator receives the results of the discriminator's test and tries to create better fake images.

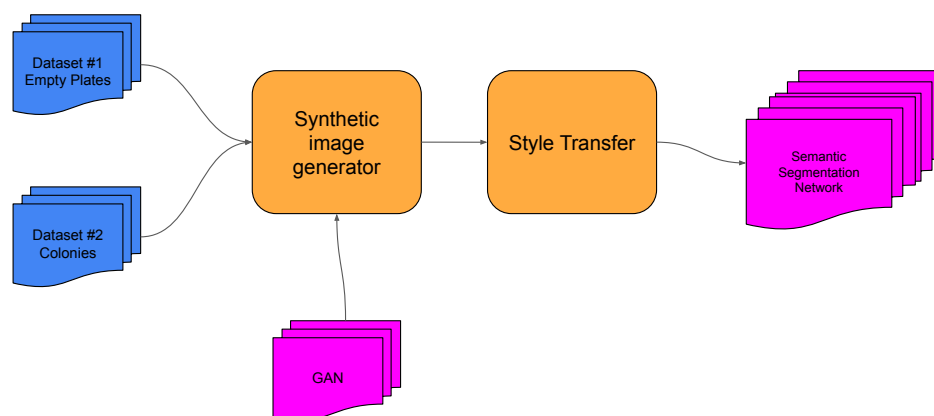


Figure 17. GAN network trained with images of bacterial colonies in agar plates to create synthetic data for a CNN to segment the bacteria from the rest of the image. This diagram refers to the method described by Bonechi et al. [84].

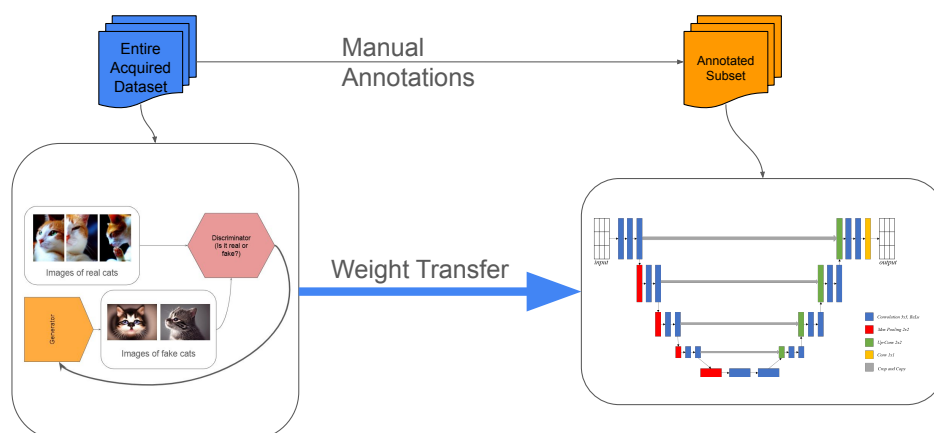


Figure 18. GAN-based transfer learning for U-Net segmentation. The GAN learns without supervision from unlabeled data, capturing different patterns and relationships. The resulting weights are then transferred to a U-Net that is trained using a small labeled dataset. This diagram demonstrates the method described by Majurski et al. [85].

8. Attention-Based Networks

Attention-based networks, or transformer networks [86], use the concept of self-attention from Natural Language Processing (NLP) [87], which assigns importance scores to each token in a sentence. This mechanism enables the transformer, as illustrated in Figure 19, to capture long-range dependencies more effectively than RNNs, which process inputs sequentially and may forget earlier inputs.

Recently, researchers have applied the same concepts and structures to computer vision tasks [88], including semantic segmentation [89]. A transformer model can process multiple inputs at once by dividing the image into smaller patches [90], which are loaded and processed simultaneously. The network receives each batch and creates several attention vectors that describe the contextual relationship between the different regions in the image [90]. Attention allows the model to selectively focus on important parts of the input, improving its ability to capture long-range dependencies and relationships [91].

Compared to convolutional architectures, transformers have several advantages for computer vision tasks, including the ability to train on larger datasets [92], better inductive bias [93], generalization to other fields, and better modeling of long-range interactions between inputs [94]. In semantic segmentation, transformers have shown promising results and may offer an alternative to traditional CNNs [88]. Due to the large size of these architectures, the network requires more data for training and may struggle to learn effectively with smaller datasets. Additionally, these architectures consume more memory.

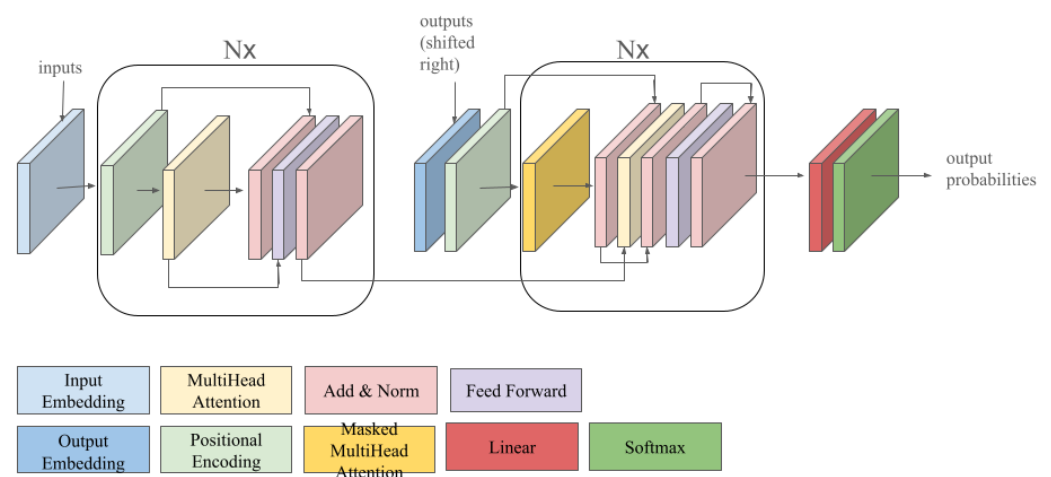


Figure 19. A simplified diagram of a transformer model architecture. This model uses a pair of encoder–decoder blocks with fully connected layers, in addition to attention layers.

9. Spatial Transformer Networks

Spatial Transformer Networks (STNs) enable the dynamic manipulation of spatial data within images. They consist of three main components: the localization network, the grid generator, and the sampler, as depicted in Figure 20.

The localization network predicts transformation parameters based on the input feature map. The grid generator creates a coordinate grid based on predicted parameters. The sampler uses a grid to sample the input feature map and generates a transformed feature map [86]. STNs can be plugged as learnable modules into CNN architectures, allowing the latter to actively transform feature maps based on the feature maps themselves, and do not require additional training supervision or modifications to the optimization process. STNs enable models to learn invariant transformations such as translation, scaling, rotation, and more general distortions, resulting in state-of-the-art performance across various benchmarks and transformation types [86].

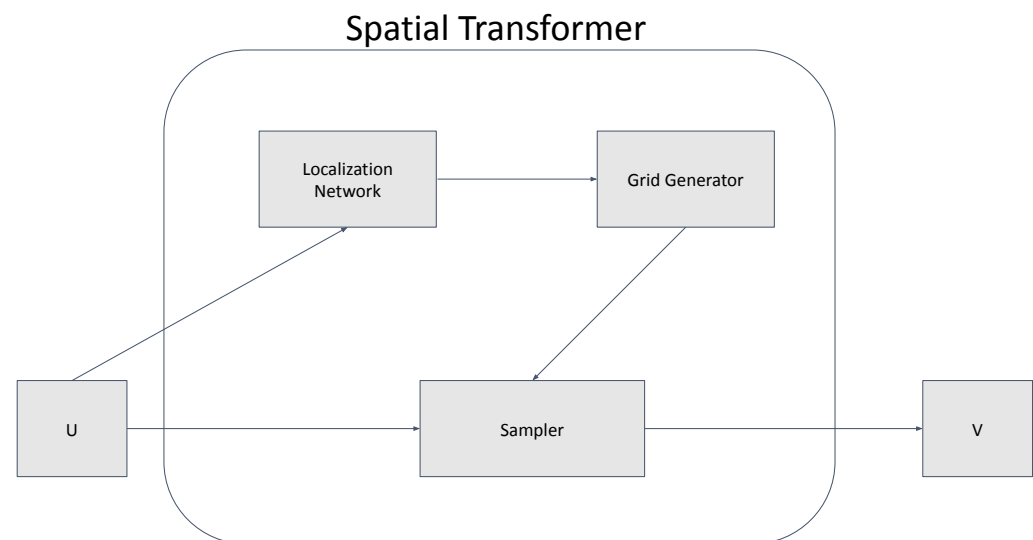


Figure 20. Architecture of a spatial transform network. Input U passes through a localization net that performs a regression operation. The grid generator creates a sampling grid that is applied over U , producing the feature map V . The central block represents the spatial transformer network, which consists of a localization net, a grid generator, and a sampler.

10. Neural Architecture Search

The preceding sections delineated solutions that address the challenges inherent to semantic segmentation [95]. Nonetheless, these solutions present a high barrier to entry, as they require the involvement of experts well-versed in the details and operations of neural networks for their creation and/or adaptation to specific problems [96]. Neural Architecture Search (NAS) aims to design architectures primarily through algorithmic processes, with minimal human intervention or oversight. Within this domain, two notable pioneers are NAS-RL and MetaQNN [97]. The use of NAS has been successful at creating new architectures that outperform human-designed ones in image classification. This relatively new approach to creating networks has already been applied to semantic segmentation with promising results.

These architectures can be derived using techniques such as Reinforcement Learning (RL) [98], heuristic algorithms [99,100] (such as evolutionary algorithms [101,102]), Bayesian optimization [103], and gradient-based search [97]. Initially, several studies focused on creating the architecture from scratch; however, the large computational overhead was a major drawback to this approach [96]. Recent works deal with this issue by keeping the backbone static and using NAS to optimize the repeatable cell patterns. Weng et al. [104] designed two cell types using NAS, which were repeatedly stacked to build a U-Net-like architecture to apply medical semantic segmentation while requiring fewer parameters than U-Net. Liu et al. [105] used a hierarchical search space that included both cell-based and architecture search to create Auto-DeepLab. Fan et al. [106] created a search space to find a self-attention unit that can capture relations in all dimensions (height, width, and channels). Despite the vast body of work within NAS research, replication proves challenging due to the variability across experiments in terms of search space, hyperparameters, strategies, and other factors.

The exploration of the search space in NAS is crucial for refining existing architectures, with many efforts dedicated to enhancing both its efficiency and speed. NAS is also able to optimize the parameters for a specific module, replicating it to make improved models [104,105,107–109]. Additionally, NAS can replicate the macrostructure of well-established architectures while searching for improved architectural components, such as block types or layer connections, rather than directly optimizing hyperparameters.

11. Continual Semantic Segmentation

The previous sections addressed several network architectures for 2D segmentation, along with their primary features, weighing their respective advantages and disadvantages; they also discussed algorithmic network construction using NAS. However, once a network is established, the main question emerges: how to effectively train it [110]. Traditionally, networks are trained using self-contained labeled image datasets. Yet, contemporary approaches require continuous learning and adaptation to newly available data while also maintaining previous knowledge [110]. This previous knowledge can be tasks learned by the network, domain knowledge for when the network is trained in one dataset, class knowledge that includes all learned classes, and certain modalities of data included during training, such as text present in the image data. Since a network can hold these different types of knowledge, continuous learning can be divided into four categories: (1) task-incremental, which retrains the model to learn new tasks; (2) domain-incremental, which introduces new data to the model; (3) class-incremental, for adding new classes into the network knowledge, which can introduce issues such as confusion between similar classes, especially when they are not presented simultaneously during training, as illustrated in Figure 21; and (4) modality-incremental, which adds new input data, such as text into an image, sensor information, and other information [110].

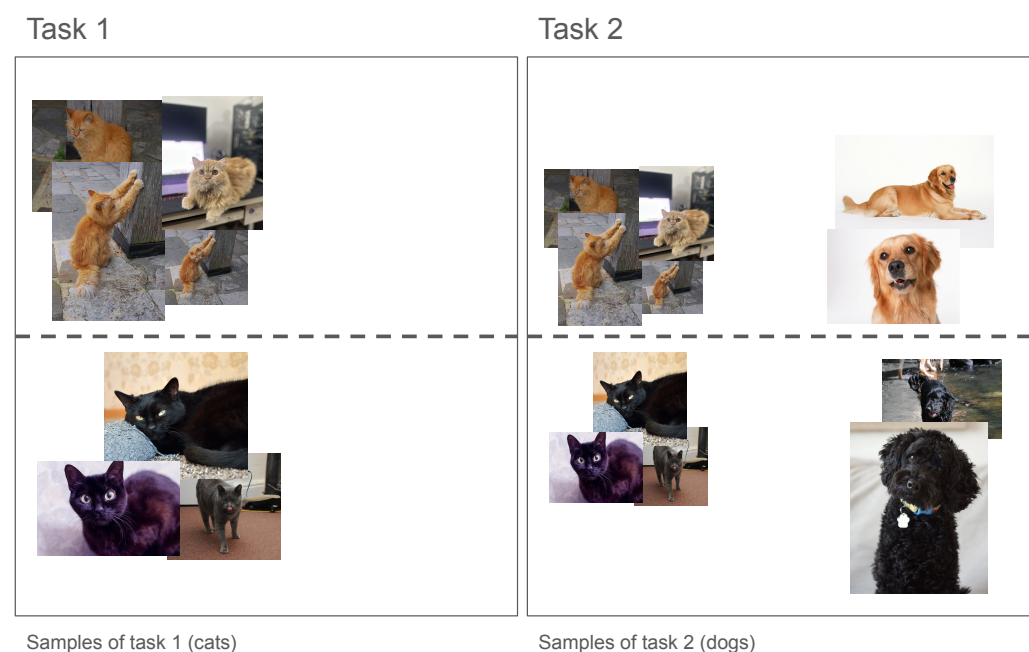


Figure 21. Visualisation of class-incremental learning with errors. The network never learns to differentiate between cats and dogs since it was never trained at the same time with both.

Retraining a model with new information may lead to an event known as catastrophic forgetting. This occurs when the model loses previously acquired knowledge during the process of learning new data. Semantic drift is another challenge, created from the addition of new background information in the data, potentially causing previously learned classes to be misclassified as background, as shown in Figure 22. This contributes to the loss of previously learned knowledge, harming the model's overall performance [111].

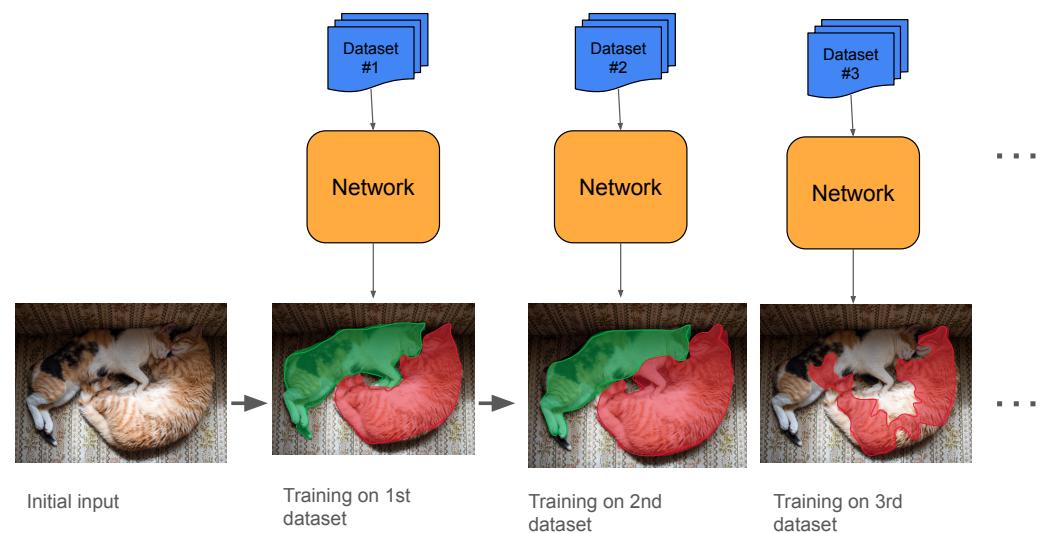


Figure 22. Example of what catastrophic forgetting can look like after several rounds of retraining with new data. The previously learned objects can begin to be forgotten and classified as background.

Moreover, this type of continual learning is resource-intensive, incurring significant costs. Obtaining and managing the data also poses a challenge, as it may be inaccessible due to privacy concerns or constrained by storage limitations [112]. Therefore, the issues associated with continuous learning in neural networks require balancing improved performance with the practical constraints of data availability and storage. While multiple approaches exist, two predominant methods stand out: data-free and data-replay [110,111].

Data-free methods include all techniques that do not store data and aim to keep the existing knowledge in the model while teaching it new information. There are three well-known techniques to achieve this: self-supervision, regularization, and dynamic architecture. Self-supervision utilizes labeled data and uses image manipulation techniques, as in rotation, and context reconstruction. Regularization can consist, for example, of freezing weight parameters to maintain old knowledge, as done in elastic weight consolidation [113]. Another related method, learning without forgetting [114], preserves prior outputs as soft targets to prevent forgetting during new task training. Finally, dynamic architecture restructures the network's design for each continual learning task, for example, by adding new modules. A significant advantage in data-free methods lies in not needing to store any historical data, which may accumulate and significantly increase storage and processing costs. Additionally, this approach decreases concerns related to the storage of potentially sensitive or private information. Data-free methods are appealing in scenarios where resource efficiency and privacy are important, for example, medical image segmentation [115,116].

Conversely, data-replay methods involve storing a certain amount of older data or utilizing a GAN to present the model, with generated images mimicking the old data. However, the effectiveness of GAN-based data-replay hinges on the performance of the GAN. This method provides a mechanism to reinforce the model's memory of past experiences but introduces the challenge of maintaining the quality and fidelity of the replayed data [117,118].

In summary, choosing between data-free and data-replay methods involves a trade-off between resource efficiency, privacy concerns, and the fidelity of retained information. Striking a balance between these considerations is crucial for designing a continuous learning solution that is both effective and ethically sound [110,111]. To possibly mimic the stability and plasticity of real human brains with networks and algorithms, research in continuous learning can contribute to creating better solutions that require models to be

flexible, especially in problems where learning new knowledge is critical, such as the case of self-driving cars.

12. Datasets

This section describes several 2D and 2.5D datasets used in image segmentation tasks (2.5D datasets include image depth information). The selected datasets and their main characteristics are listed in Table 1. These datasets were selected based on their diversity, data volume, and frequent use in benchmarking within related literature.

Among the most commonly used datasets are PASCAL VOC 2012 [119] and Cityscapes [17]. PASCAL VOC 2012 offers a diverse collection of real-world images labeled for 20 object classes. In contrast, PASCAL Context [120], derived from the same dataset family, provides denser semantic annotations with a significantly larger set of classes, including both objects and background elements.

These datasets are ideal for testing neural networks' capabilities in recognizing varied object types and for supporting transfer learning due to their diversity. However, larger datasets often require greater memory and computational resources during training. When training across all available classes is unnecessary, using a subset of the dataset (in terms of images or class labels) can be a more efficient alternative.

Some datasets are created with specific application domains in mind. For example, Cityscapes, CamVid [121], and KITTI [122–125] are designed for urban scene understanding and are widely used in research on autonomous driving. For tasks involving weakly supervised object detection and object tracking in videos, the YouTube-Objects dataset [126] offers short video clips centered around 10 object categories from PASCAL VOC, though it does not include full-frame annotations or support real-time processing out of the box. The PASCAL Part dataset [127] builds on PASCAL VOC by adding part-level annotations for several object classes, providing finer-grained information such as heads, wings, wheels, or legs, depending on the object category.

Table 1. Common datasets for semantic segmentation, most of which are used in the works discussed in this paper.

Type	Dataset	Use Cases	Size	Classes	Notes
2D	PASCAL VOC 2012 [119]	People, animals, vehicles, objects	11,530	20	27,450 ROI and 6929 segmentations
	PASCAL Context [120]	Objects, stuff, and hybrids	10,103	400	9637 testing images
	MSCOCO [128]	Objects and stuff	330K	171	80 object classes and 91 stuff categories
	Cityscapes [17]	Street scenes and objects	25K	30	5K fine annotated and 20K coarsely annotated images
	ADE20K [129]	Scene images	25,574	150	Validation set of 2K images
	BSDS500 [130]	Contour and edge detection	300		Training set of 200 and test set of 100 images 12K hand labelled ROIs
	YouTube-Objects [126]	YouTube videos		10	Each class has 9–24 videos of 3 s to 3 min length
	CamVid [121]	Road driving scenes	701	32	Samples taken at 1 fps and 15 fps and manually annotated
	SBD [131]	Contour and edge detection	11,355	20	Images taken from PASCAL VOC 2011

Table 1. Cont.

Type	Dataset	Use Cases	Size	Classes	Notes
2D	PASCAL Part [127]	Body parts of each object	10K+		Testing set of 9637 images
	OpenEarthMap [132]	Aerial images	5000	8	Over 64 regions, across 6 continents
	SIFTFlow [133]	Scene images		33	Two types of labelling: semantic and geometrical, has unannotated images
	Stanford Background [134]	Scene images	715	11	8 object classes and 3 geometric classes
	KITTI [122–125]	Road driving scenes			pixel-level and instance-level segmentation images and videos
	BraTS [135]	Brain tumor segmentation	775K+	4	Multimodal MRI Dataset
	CHAOS [136]	Abdominal organ segmentation	4.3K+	3	CT and Multimodal MRI (T1-Dual, T2-SPIR)
	ISIC [137]	Skin lesion segmentation	25K+	1	Dermoscopic RGB Images
	DeepGlobe [138]	Satellite image segmentation	24K+	7	High-resolution Satellite RGB Imagery (0.5 m/pixel)
	SpaceNet [139]	Building footprint segmentation	40K+	1	Satellite RGB and Multispectral Imagery
2.5D	NYU-Depth V2 [140]	Indoor scenes			Video sequences of indoor scenes
	SUN RGB-D [141]	RGB-D indoor scenes	10,335		Includes depth and segmentation masks
	ScanNet [142]	Indoor scenes	1513		Instance level with 2D and 3D data
	Stanford 2D-3D [143]	Indoor scenes	70K+		Includes raw sensor data, depth, surface normals, and semantic annotations

Beyond the datasets discussed thus far, domain-specific segmentation datasets—such as those used in medical imaging, remote sensing, and off-road environments—introduce additional challenges, including class imbalance, sensor-induced noise, and complex object geometries. For instance, medical datasets such as BraTS [135], CHAOS [136], and ISIC [137] focus on segmenting tumors, organs, and skin lesions from modalities such as CT, MRI, and dermoscopy. These datasets are commonly used to evaluate model performance under conditions of high class imbalance, limited labeled data, and modality-specific artifacts. In the field of remote sensing, high-resolution aerial or satellite datasets such as DeepGlobe [138] and SpaceNet [139] are designed for the segmentation of land use, buildings, and road networks. These datasets challenge models with scale variation, visual similarity between classes, and geometric complexity. Including such domain-specific datasets in benchmarking enables a more comprehensive evaluation of model generalization and real-world applicability.

13. Implementation Considerations

When implementing semantic segmentation models, it is important to consider the software frameworks and hardware limitations that could affect the development and training process.

Most of the architectures discussed in this paper are implemented using popular deep learning frameworks such as PyTorch [144] and TensorFlow [145], which provide modular APIs, pretrained models, and active open-source communities. These frameworks support GPU acceleration and allow fine-grained customization of model architecture, training procedures, and loss functions. PyTorch, in particular, is widely adopted in research and

industry due to its dynamic computation graph and broad community support. It is often used for rapid prototyping and experimentation with the most recent state-of-the-art models such as transformers and NAS-based models. TensorFlow is commonly used in production environments due to its robust deployment tools (e.g., TensorFlow Lite, TensorFlow Serving).

Training deep neural networks for semantic segmentation presents several practical challenges. One significant issue is high memory consumption, particularly when processing high-resolution images. This problem becomes more pronounced with larger batch sizes or architectures that generate numerous intermediate feature maps, often exceeding the memory limits of standard GPUs. To address this, various optimization strategies are employed, including gradient checkpointing, mixed precision training (as described by Micikevicius et al. [146]), and patch-based training approaches. Another major challenge is the extended training time required by deep models—especially transformer-based architectures—when applied to large-scale datasets. In such cases, distributed training methods, such as data or model parallelism across multiple GPUs, are commonly used to reduce convergence time. Additionally, semantic segmentation tasks frequently suffer from class imbalance, wherein certain classes (e.g., background) dominate the dataset, adversely affecting model convergence and generalization. To mitigate this, specialized loss functions such as Dice loss, focal loss [147], or class-weighted cross-entropy are often utilized. Finally, effective preprocessing and data augmentation techniques—including random cropping, flipping, rotation, and color jittering—are crucial for improving model generalization and robustness during training.

Considering these training challenges, high-performance GPUs with large memory are recommended for training state-of-the-art models. In low-resource scenarios, transfer learning or training on lower-resolution images are alternative options to run deep models. To reduce the model size while preserving performance, deployment on edge devices may require model pruning, quantization, or knowledge distillation techniques [148,149].

14. Metrics

The performance of neural networks for 2D image segmentation tasks can be determined using various metrics. These metrics allow for comparing different networks, architectures, and techniques, offering some insights into how well a given configuration will perform and its ability to provide consistent results. In the following subsections, we discuss five performance metrics widely used in 2D image segmentation.

14.1. Pixel Accuracy

Pixel Accuracy (PA), given by Equation (4), calculates the percentage of correctly classified pixels in the output image [150]. In Equation (4), K is the number of classes, p_{ii} is the number of correctly identified pixels from class i , and p_{ij} is the number of pixels belonging to class i that were predicted as class j . Where an object does not occupy a large enough region, the classifier might still correctly classify most pixels even if it fails every single object pixel. This method is very vulnerable to class imbalances.

$$PA = \frac{\sum_{i=1}^K p_{ii}}{\sum_{i=1}^K \sum_{j=0}^K p_{ij}} \quad (4)$$

14.2. Mean Pixel Accuracy

Mean Pixel Accuracy (mPA), shown in Equation (5), is a variation on PA that involves averaging the PA value between all classes in an image. The symbols in Equation (5) have

the same meaning as the symbols in Equation (4). It is less vulnerable to class imbalances in comparison to PA [151].

$$mPA = \frac{1}{K} \sum_{i=1}^K \frac{p_{ii}}{\sum_{j=1}^K p_{ij}} \quad (5)$$

14.3. Intersection over Union

Intersection Over Union (*IoU*) is the most commonly used method for measuring performance. As highlighted in Equation (6), IoU determines the overlap coefficient between the ground-truth labels and predicted labels, which makes it more consistent in detecting smaller objects in an image. Unlike PA or mPA, IoU does not compare all categorized pixels in an image, just the ones in the predicted and ground truth areas [152].

$$IoU = J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (6)$$

14.4. Mean IoU

The Mean IoU (*mIoU*) is given by Equation (7), where IoU_i is the IoU for a certain class i . This metric provides a score to the network and how well it performs while taking into account precision (how many predictions are true positives) and recall (how many true positives were identified as such) [153].

$$mIoU = \frac{1}{K} \sum_{i=1}^K IoU_i \quad (7)$$

14.5. Dice Coefficient

The Dice coefficient, as expressed in Equations (8) and (9), quantifies the similarity between two images by calculating twice the area of their overlap divided by the sum of the total number of pixels in both images. The first equation expresses the Dice coefficient in terms of set overlap, while the second uses confusion matrix components (true positives, false positives, and false negatives) commonly found in classification tasks. This metric is closely related to the IoU method. Both metrics typically provide consistent evaluations, such that if the IoU indicates that one model's output is superior to another, the Dice coefficient will likely concur [154]. As highlighted in Equation (9), the Dice coefficient is also mathematically equivalent to the F1-score in the binary classification setting, though differences may arise in multi-class or multi-label segmentation tasks.

$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|} \quad (8)$$

$$\text{Dice} = \frac{2TP}{2TP + FP + FN} = F_1 \quad (9)$$

15. Discussion

This section provides a discussion of various models and their performance on the datasets they were trained on. Table 2 highlights the performance, using the mIoU metric, of each network on common and less common datasets, respectively. Due to differences in evaluation settings and incomplete reporting across studies, some entries are missing. This table is provided on a best-effort basis to illustrate general performance trends rather than to serve as a unified benchmark.

The results presented in Table 2, highlighting a selection of network architectures spanning from 2014 to 2023, offer some insights into the progression of state-of-the-art semantic segmentation [96]. NAS research is more common among image classification

and object detection, probably due to the higher difficulty and more complex architectures of image segmentation. The increased use of transformer architectures may lead to a corresponding rise in NAS-derived architectures since the modular nature of the former simplifies the latter's search space [105].

Although Table 2 suggests a general trend of improved performance over time, inconsistencies in evaluation protocols and dataset usage throughout studies make precise comparisons difficult. The evolution in research for this field started with a focus on CNNs but found better success with later architectures and techniques, such as RNNs and transformers [66,94]. Nonetheless, RNNs have been dropped in favor of transformers due to having shorter memory.

Although these results outline improvements in mIoU, such gains can also be attributed to key architectural shifts. Early CNN-based models relied heavily on local receptive fields and deep hierarchical representations, both of which struggled with context-aware segmentation. Techniques such as pyramid pooling and dilated convolutions were developed to address this issue, but long-range spatial dependencies remained a limitation.

Transformer-based models introduced a paradigm shift by using self-attention mechanisms, which directly model the relationships between any two points in an image—regardless of their distance. This technique for capturing global context represents an improvement in the field of semantic segmentation, where semantically related regions may be spatially distant.

Table 2. The mean Intersection over Union (mIoU) of different networks, from 2014 to 2023, in more common datasets.

Model	Year	Cityscapes	VOC 2011	VOC 2012
FCN [155]	2014	-	42.5	-
CRF-RNN [156]	2015	-	72.4	-
BoxSup [157]	2015	-	-	75.2
Efficient Piecewise [158]	2015	-	-	75.3
DPN [159]	2015	-	-	77.5
DeepLab-CRF [160]	2016	-	-	79.7
Wide ResNet [161]	2016	-	-	82.5
PSPNet [162]	2016	-	-	85.4
GCN [163]	2017	-	-	82.2
BiSeNet [164]	2018	76.8	-	-
Auto-DeepLab [105]	2019	82.04	-	-
RefineNet [165]	2020	73.6	-	-
Dynamic Routing [165]	2020	-	-	79
MOSAIC [166]	2021	75.67	-	-
WaveMix-Lite [167]	2022	75.32	-	-
M-FasterSeg [168]	2022	69.8	-	-
ZiCo [169]	2023	78.62	-	-
ZiCo-BC [169]	2023	79.71	-	-
GeNAS [170]	2023	72.58	-	-
FMAS [171]	2023	-	-	67.39

Hybrid approaches, such as TransUNet [60], combine the spatial precision of convolutional encoders with the contextual modeling of transformers to achieve a better balance between accuracy and hardware costs. The scalability of transformer-based models is also a significant advantage as datasets grow larger and more complex.

In addition to this performance improvement using transformer-based models, transformers require substantially more computational resources and are susceptible to overfitting when working with small datasets. Thus, techniques such as pretraining, transfer learning, and data augmentation are often necessary to ensure generalization.

The evolution in segmentation research has started with a focus on CNNs but found better results with later architectures and techniques such as RNNs and transformers [66,94]. Although this paper organizes models by their dominant architectural paradigms—such as FCNs, GCNs, encoders–decoders, and attention-based networks—it is important to note that many modern semantic segmentation models span multiple categories. For example, transformer-based models often adopt encoder–decoder structures, and hybrid approaches like TransUNet combine convolutional backbones with attention mechanisms. These overlapping characteristics reflect the modular and compositional nature of recent models, which increasingly draw from multiple architectural ideas to improve performance and adaptability.

Orthogonal to these architectural advances, continuous learning in image segmentation remains uncommon, with most networks being trained using closed datasets. Training on closed datasets means that networks learn their tasks once and cannot be updated with new or improved data. Different methods, or the optimization of existing methods, in continual semantic segmentation could result in networks that can learn and improve continuously. Further research could lead to networks that can learn new tasks or correct previous knowledge [110].

16. Current and Future Directions in Semantic Segmentation

Recent developments in computer vision have introduced a new class of models that significantly enhance the capabilities of semantic segmentation by moving beyond fixed-class labels and closed-world assumptions. One major advancement is the integration of Large Language Models (LLMs) into vision systems, leading to the emergence of multimodal approaches. Architectures such as CLIP [172] and LLaVA [173] combine visual encoders with language representations, allowing segmentation models to generalize to previously unseen object categories through text prompts. This enables open-vocabulary or zero-shot segmentation [174], where target classes are defined dynamically rather than being constrained to a predefined label set.

Another notable contribution is the Segment Anything Model (SAM) [175], which has been trained on over one billion masks spanning eleven million images. SAM supports the segmentation of arbitrary objects using point, box, or mask prompts and can generalize to new tasks and domains without the need for retraining or fine-tuning. In the realm of open-vocabulary and zero-shot segmentation, models such as MaskCLIP [176] and ViL-Seg [177] extend CLIP-based visual-language embeddings to pixel-level tasks, enabling segmentation of categories that were not present during training.

Furthermore, the diffusion paradigm—originally developed for image generation—has recently been adapted for semantic segmentation. Methods like Efficient Semantic Segmentation with Diffusion Models [178] and SEEM [179] leverage denoising diffusion processes to produce high-resolution segmentation masks, offering a new generative approach to this task.

17. Conclusions

The field of semantic segmentation includes a wide range of techniques and tools, many of which are still under active research and development. Starting with early solutions based on CNNs, the field has since evolved to include more advanced architectures that offer improved accuracy and efficiency. While this paper primarily focuses on image semantic segmentation, many of these methods are also being adapted for video semantic segmentation, which requires real-time detection and segmentation. The techniques and datasets reviewed in this work provide a foundational understanding of the various subdomains within this research area.

In addition to the techniques developed in recent years, newer methods such as NAS have also been analyzed. NAS, in particular, still faces challenges in reducing computational costs. However, being a relatively recent approach, NAS may benefit significantly from future research in the optimization techniques themselves, as well as from the introduction of newer architectures, which could offer more robust baseline networks to optimize, combine, and enhance further.

The availability of diverse datasets is also of great importance for testing and deploying these networks, as they must meet the varying requirements of researchers. Despite the existing datasets, networks may need to continuously learn from new data as they become available, introducing a unique set of challenges. These challenges are actively being addressed within the field of continual semantic segmentation.

In summary, and despite the significant challenges lying ahead for the field, semantic segmentation has made significant strides in the last few years, particularly with the development of advanced neural network architectures and techniques, supported by the growing availability of diverse datasets. Thus, it seems clear that the future of semantic segmentation lies in creating more adaptive, efficient models capable of real-time processing and continual learning from diverse data sources.

Author Contributions: Conceptualization, S.S. and J.P.M.-C.; methodology, S.S.; software, S.S.; validation, S.S., N.F. and J.P.M.-C.; formal analysis, N.F. and J.P.M.-C.; investigation, S.S.; resources, S.S.; data curation, S.S., N.F. and J.P.M.-C.; writing—original draft preparation, S.S.; writing—review and editing, S.S., N.F. and J.P.M.-C.; visualization, N.F. and J.P.M.-C.; supervision, N.F. and J.P.M.-C.; project administration, N.F. and J.P.M.-C.; funding acquisition, N.F. and J.P.M.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by Fundação para a Ciência e a Tecnologia (FCT) under Grants Copelabs ref. UIDB/04111/2020, Centro de Tecnologias e Sistemas (CTS) ref. UIDB/00066/2020, LASIGE Research Unit, ref. UID/00408/2025, and COFAC ref. CEECINST/00002/2021/CP2788/CT0001; and, Instituto Lusófono de Investigação e Desenvolvimento (ILIND) under Project COFAC/ILIND/COPELABS/1/2024.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
CCTV	Closed Circuit Television
CRF	Conditional Random Fields
VSS	Video Semantic Segmentation
VPS	Video Panoptic Segmentation
FCN	Fully Convolutional Network
FPN	Feature Pyramid Network
NAS	Neural Architecture Search
GCN	Graph Convolutional Networks
CT	Computed Tomography
MRI	Magnetic Resonance Imaging
RNN	Recurrent Neural Network
GAN	Generative Adversarial Network
NLP	Natural Language Processing
STN	Spatial Transformer Network
RL	Reinforcement Learning
GPT	Generative Pre-training Transformer
PA	Pixel Accuracy
mPA	Mean Pixel Accuracy
IoU	Intersection Over Union
mIoU	Mean Intersection Over Union

References

- Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440. [\[CrossRef\]](#)
- Xian, M.; Zhang, Y.; Cheng, H.; Xu, F.; Zhang, B.; Ding, J. Automatic Breast Ultrasound Image Segmentation: A Survey. *arXiv* **2017**, arXiv:1704.01472. [\[CrossRef\]](#)
- Li, B.; Liu, S.; Xu, W.; Qiu, W. Real-time object detection and semantic segmentation for autonomous driving. In Proceedings of the MIPPR 2017: Automatic Target Recognition and Navigation, Xiangyang, China, 28–29 October 2017; Liu, J., Udupa, J.K., Hong, H., Eds.; International Society for Optics and Photonics, SPIE: California, CA, USA, 2018; Volume 10608, pp. 167–174. [\[CrossRef\]](#)
- Yasuno, M.; Yasuda, N.; Aoki, M. Pedestrian detection and tracking in far infrared images. In Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop, Washington, DC, USA, 27 June–2 July 2004; p. 125. [\[CrossRef\]](#)
- Victoria Priscilla, C.; Agnes Sheila, S.P. Pedestrian Detection—A Survey. In Proceedings of the First International Conference on Innovative Computing and Cutting-Edge Technologies (ICICCT 2019), Istanbul, Turkey, 30–31 October 2019; Jain, L.C., Peng, S.L., Alhadidi, B., Pal, S., Eds.; Springer Nature: Cham, Switzerland, 2020; pp. 349–358. [\[CrossRef\]](#)
- Deepak, G.D.; Bhat, S.K. A comparative study of breast tumour detection using a semantic segmentation network coupled with different pretrained CNNs. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **2024**, *12*, 2373996. [\[CrossRef\]](#)
- Kohler, R. A segmentation system based on thresholding. *Comput. Graph. Image Process.* **1981**, *15*, 319–338. [\[CrossRef\]](#)
- Gómez, O.; González, J.A.; Morales, E.F. Image Segmentation Using Automatic Seeded Region Growing and Instance-Based Learning. In *Progress in Pattern Recognition, Image Analysis and Applications*; Rueda, L., Mery, D., Kittler, J., Eds.; Springer Nature: Berlin/Heidelberg, Germany, 2007; pp. 192–201.
- Roslin, A.; Marsh, M.; Provencher, B.; Mitchell, T.; Onederra, I.; Leonardi, C. Processing of micro-CT images of granodiorite rock samples using convolutional neural networks (CNN), Part II: Semantic segmentation using a 2.5D CNN. *Miner. Eng.* **2023**, *195*, 108027. [\[CrossRef\]](#)
- Lapa, P.A.F. Conditional Random Fields Improve the CNN-Based Prostate Cancer Classification Performance. Master's Thesis, NOVA Information Management School, Lisbon, Portugal, 2019.
- Zhang, M.; Dong, B.; Li, Q. Deep active contour network for medical image segmentation. In Proceedings of the Medical Image Computing and Computer Assisted Intervention—MICCAI 2020: 23rd International Conference, Lima, Peru, 4–8 October 2020; Proceedings, Part IV 23; Springer: Berlin/Heidelberg, Germany, 2020; pp. 321–331.

12. Li, P.; Xia, H.; Zhou, B.; Yan, F.; Guo, R. A Method to Improve the Accuracy of Pavement Crack Identification by Combining a Semantic Segmentation and Edge Detection Model. *Appl. Sci.* **2022**, *12*, 4714. [\[CrossRef\]](#)
13. Yuheng, S.; Hao, Y. Image Segmentation Algorithms Overview. *arXiv* **2017**, arXiv:1707.02051. [\[CrossRef\]](#)
14. Tian, D.; Han, Y.; Wang, B.; Guan, T.; Gu, H.; Wei, W. Review of object instance segmentation based on deep learning. *J. Electron. Imaging* **2021**, *31*, 041205. [\[CrossRef\]](#)
15. Kim, D.; Woo, S.; Lee, J.; Kweon, I.S. Video Panoptic Segmentation. *arXiv* **2020**, arXiv:2006.11339. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Kirillov, A.; He, K.; Girshick, R.; Rother, C.; Dollar, P. Panoptic Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9396–9405. [\[CrossRef\]](#)
17. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223. [\[CrossRef\]](#)
18. Prakash, V.J.; Nithya, L.M. A Survey on Semi-Supervised Learning Techniques. *arXiv* **2014**, arXiv:1402.4645. [\[CrossRef\]](#)
19. Chen, T.; Kornblith, S.; Swersky, K.; Norouzi, M.; Hinton, G.E. Big Self-Supervised Models are Strong Semi-Supervised Learners. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 22243–22255.
20. Ran, L.; Li, Y.; Liang, G.; Zhang, Y. Pseudo Labeling Methods for Semi-Supervised Semantic Segmentation: A Review and Future Perspectives. *IEEE Trans. Circuits Syst. Video Technol.* **2025**, *35*, 3054–3080. [\[CrossRef\]](#)
21. Zhang, B.; Zhang, Y.; Li, Y.; Wan, Y.; Guo, H.; Zheng, Z.; Yang, K. Semi-supervised deep learning via transformation consistency regularization for remote sensing image semantic segmentation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 5782–5796. [\[CrossRef\]](#)
22. Xie, J.; Shuai, B.; Hu, J.F.; Lin, J.; Zheng, W.S. Improving fast segmentation with teacher-student learning. *arXiv* **2018**, arXiv:1810.08476. [\[CrossRef\]](#)
23. Wang, W.; Zhou, T.; Porikli, F.; Crandall, D.J.; Gool, L.V. A Survey on Deep Learning Technique for Video Segmentation. *arXiv* **2021**, arXiv:2107.01153. [\[CrossRef\]](#)
24. Jung, S.; Heo, H.; Park, S.; Jung, S.U.; Lee, K. Benchmarking Deep Learning Models for Instance Segmentation. *Appl. Sci.* **2022**, *12*, 8856. [\[CrossRef\]](#)
25. Portillo-Portillo, J.; Sanchez-Perez, G.; Toscano-Medina, L.K.; Hernandez-Suarez, A.; Olivares-Mercado, J.; Perez-Meana, H.; Velarde-Alvarado, P.; Orozco, A.L.S.; García Villalba, L.J. FASSVid: Fast and Accurate Semantic Segmentation for Video Sequences. *Entropy* **2022**, *24*, 942. [\[CrossRef\]](#) [\[PubMed\]](#)
26. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587. [\[CrossRef\]](#)
27. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Dumoulin, V.; Visin, F. A guide to convolution arithmetic for deep learning. *arXiv* **2018**, arXiv:1603.07285. [\[CrossRef\]](#)
29. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3523–3542. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **2004**, *36*, 193–202. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6. [\[CrossRef\]](#)
32. Scherer, D.; Müller, A.; Behnke, S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In Proceedings of the Artificial Neural Networks—ICANN 2010, Thessaloniki, Greece, 15–18 September 2010; Diamantaras, K., Duch, W., Iliadis, L.S., Eds.; Springer Nature: Cham, Switzerland, 2010; pp. 92–101. [\[CrossRef\]](#)
33. O’Shea, K.; Nash, R. An Introduction to Convolutional Neural Networks. *arXiv* **2015**, arXiv:1511.08458. [\[CrossRef\]](#)
34. Estrach, J.B.; Szlam, A.; LeCun, Y. Signal recovery from Pooling Representations. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 22–24 June 2014; Xing, E.P., Jebara, T., Eds.; Proceedings of Machine Learning Research, Volume 32, pp. 307–315.
35. Wang, S.H.; Lv, Y.D.; Sui, Y.; Liu, S.; Wang, S.J.; Zhang, Y.D. Alcoholism detection by data augmentation and Convolutional Neural Network with stochastic pooling. *J. Med. Syst.* **2017**, *42*, 2. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Kassani, S.H.; Kassani, P.H.; Wesolowski, M.J.; Schneider, K.A.; Deters, R. Breast Cancer Diagnosis with Transfer Learning and Global Pooling. In Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 16–18 October 2019; pp. 519–524. [\[CrossRef\]](#)
37. Lv, Y.; Ma, H.; Li, J.; Liu, S. Attention guided u-net with atrous convolution for accurate retinal vessels segmentation. *IEEE Access* **2020**, *8*, 32826–32839. [\[CrossRef\]](#)

38. Qiao, S.; Chen, L.C.; Yuille, A. DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10208–10219. [\[CrossRef\]](#)
39. Kyrkou, C.; Theodoridis, T. EmergencyNet: Efficient aerial image classification for drone-based emergency monitoring using atrous convolutional feature fusion. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 1687–1699. [\[CrossRef\]](#)
40. Zhou, Y.; Chang, H.; Lu, Y.; Lu, X. CDTNet: Improved image classification method using standard, Dilated and Transposed Convolutions. *Appl. Sci.* **2022**, *12*, 5984. [\[CrossRef\]](#)
41. Odena, A.; Dumoulin, V.; Olah, C. Deconvolution and Checkerboard Artifacts. *Distill* **2016**. [\[CrossRef\]](#)
42. Gao, H.; Yuan, H.; Wang, Z.; Ji, S. Pixel transposed convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 1218–1227. [\[CrossRef\]](#) [\[PubMed\]](#)
43. Zhang, S.; Tong, H.; Xu, J.; Maciejewski, R. Graph convolutional networks: A comprehensive review. *Comput. Soc. Netw.* **2019**, *6*, 11. [\[CrossRef\]](#) [\[PubMed\]](#)
44. Tenenbaum, J.B.; de Silva, V.; Langford, J.C. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* **2000**, *290*, 2319–2323. [\[CrossRef\]](#) [\[PubMed\]](#)
45. Roweis, S.T.; Saul, L.K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* **2000**, *290*, 2323–2326. [\[CrossRef\]](#) [\[PubMed\]](#)
46. Belkin, M.; Niyogi, P. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In *Advances in Neural Information Processing Systems*; Dietterich, T., Becker, S., Ghahramani, Z., Eds.; MIT Press: Cambridge, MA, USA, 2001; Volume 14.
47. Perozzi, B.; Al-Rfou, R.; Skiena, S. DeepWalk: Online Learning of Social Representations. *arXiv* **2014**, arXiv:1403.6652. [\[CrossRef\]](#)
48. Grover, A.; Leskovec, J. node2vec: Scalable Feature Learning for Networks. *arXiv* **2016**, arXiv:1607.00653. [\[CrossRef\]](#)
49. Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; Weinberger, K. Simplifying Graph Convolutional Networks. In Proceedings of the 36th International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; Chaudhuri, K., Salakhutdinov, R., Eds.; Proceedings of Machine Learning Research, Volume 97, pp. 6861–6871.
50. Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; Li, Y. Simple and Deep Graph Convolutional Networks. In Proceedings of the 37th International Conference on Machine Learning, PMLR, Online, 13–18 July 2020; Daumé, H., Singh, A., Eds.; Proceedings of Machine Learning Research, Volume 119, pp. 1725–1735.
51. Zhang, L.; Li, X.; Arnab, A.; Yang, K.; Tong, Y.; Torr, P.H.S. Dual Graph Convolutional Network for Semantic Segmentation. *arXiv* **2019**, arXiv:1909.06121. [\[CrossRef\]](#)
52. Bian, T.; Xiao, X.; Xu, T.; Zhao, P.; Huang, W.; Rong, Y.; Huang, J. Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 549–556. [\[CrossRef\]](#)
53. Schulte-Sasse, R.; Budach, S.; Hnisch, D.; Marsico, A. Graph Convolutional Networks Improve the Prediction of Cancer Driver Genes. In Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2019: Workshop and Special Sessions, Munich, Germany, 17–19 September 2019; Tetko, I.V., Kůrková, V., Karpov, P., Theis, F., Eds.; Springer Nature: Cham, Switzerland, 2019; pp. 658–668. [\[CrossRef\]](#)
54. Wang, H.; Zhao, M.; Xie, X.; Li, W.; Guo, M. Knowledge Graph Convolutional Networks for Recommender Systems. In Proceedings of the The World Wide Web Conference (WWW '19), San Francisco, CA, USA, 13–17 May 2019; pp. 3307–3313. [\[CrossRef\]](#)
55. Sun, M.; Zhao, S.; Gilvary, C.; Elemento, O.; Zhou, J.; Wang, F. Graph convolutional networks for computational drug development and discovery. *Briefings Bioinform.* **2019**, *21*, 919–935. [\[CrossRef\]](#) [\[PubMed\]](#)
56. Yao, L.; Mao, C.; Luo, Y. Graph Convolutional Networks for Text Classification. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 7370–7377. [\[CrossRef\]](#)
57. Ghosh, S.; Das, N.; Das, I.; Maulik, U. Understanding Deep Learning Techniques for Image Segmentation. *arXiv* **2019**, arXiv:1907.06119. [\[CrossRef\]](#)
58. Oktay, O.; Schlemper, J.; Folgoc, L.L.; Lee, M.C.H.; Heinrich, M.P.; Misawa, K.; Mori, K.; McDonagh, S.G.; Hammerla, N.Y.; Kainz, B.; et al. Attention U-Net: Learning Where to Look for the Pancreas. *arXiv* **2018**, arXiv:1804.03999. [\[CrossRef\]](#)
59. Alom, M.Z.; Yakopcic, C.; Hasan, M.; Taha, T.M.; Asari, V.K. Recurrent residual U-Net for medical image segmentation. *J. Med. Imaging* **2019**, *6*, 014006. [\[CrossRef\]](#) [\[PubMed\]](#)
60. Chen, J.; Lu, Y.; Yu, Q.; Luo, X.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. *arXiv* **2021**, arXiv:2102.04306. [\[CrossRef\]](#)
61. Alom, M.Z.; Hasan, M.; Yakopcic, C.; Taha, T.M.; Asari, V.K. Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation. *arXiv* **2018**, arXiv:1802.06955. [\[CrossRef\]](#)
62. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. *arXiv* **2017**, arXiv:1612.03144. [\[CrossRef\]](#)

63. Hu, M.; Li, Y.; Fang, L.; Wang, S. A2-FPN: Attention Aggregation Based Feature Pyramid Network for Instance Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 15338–15347. [\[CrossRef\]](#)
64. Kirillov, A.; Girshick, R.; He, K.; Dollar, P. Panoptic Feature Pyramid Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 6392–6401. [\[CrossRef\]](#)
65. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [\[CrossRef\]](#)
66. Ribeiro, A.H.; Tiels, K.; Aguirre, L.A.; Schön, T. Beyond exploding and vanishing gradients: Analysing RNN training using attractors and smoothness. In Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, PMLR, Online, 26–28 August 2020; Chiappa, S., Calandra, R., Eds.; Proceedings of Machine Learning Research, Volume 108, pp. 2370–2380.
67. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#) [\[PubMed\]](#)
68. Cho, K.; van Merriënboer, B.; Gülçehre, Ç.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078. [\[CrossRef\]](#)
69. Miao, Y.; Gowayyed, M.; Metze, F. EESN: End-to-end speech recognition using deep RNN models and WFST-based decoding. In Proceedings of the 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Scottsdale, AZ, USA, 13–17 December 2015; pp. 167–174. [\[CrossRef\]](#)
70. Mikolov, T.; Kombrink, S.; Burget, L.; Černocký, J.; Khudanpur, S. Extensions of recurrent neural network language model. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 5528–5531. [\[CrossRef\]](#)
71. Li, D.; Qian, J. Text sentiment analysis based on long short-term memory. In Proceedings of the 2016 First IEEE International Conference on Computer Communication and the Internet (ICCCI), Wuhan, China, 13–15 October 2016; pp. 471–475. [\[CrossRef\]](#)
72. Visin, F.; Romero, A.; Cho, K.; Matteucci, M.; Ciccone, M.; Kastner, K.; Bengio, Y.; Courville, A. ReSeg: A Recurrent Neural Network-Based Model for Semantic Segmentation. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 426–433. [\[CrossRef\]](#)
73. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015; Bengio, Y., LeCun, Y., Eds.; ICLR: Appleton, WI, USA, 2015.
74. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2014; Volume 27.
75. Zhang, X.; Jian, W.; Chen, Y.; Yang, S. Deform-GAN: An Unsupervised Learning Model for Deformable Registration. *arXiv* **2020**, arXiv:2002.11430. [\[CrossRef\]](#)
76. Dai, Z.; Yang, Z.; Yang, F.; Cohen, W.W.; Salakhutdinov, R.R. Good Semi-supervised Learning That Requires a Bad GAN. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
77. Aggarwal, A.; Mittal, M.; Battineni, G. Generative adversarial network: An overview of theory and applications. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100004. [\[CrossRef\]](#)
78. Zhan, F.; Zhu, H.; Lu, S. Spatial Fusion GAN for Image Synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [\[CrossRef\]](#)
79. Tran, N.T.; Tran, V.H.; Nguyen, N.B.; Nguyen, T.K.; Cheung, N.M. On Data Augmentation for GAN Training. *IEEE Trans. Image Process.* **2021**, *30*, 1882–1897. [\[CrossRef\]](#) [\[PubMed\]](#)
80. Liang, X.; Lee, L.; Dai, W.; Xing, E.P. Dual Motion GAN for Future-Flow Embedded Video Prediction. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017. [\[CrossRef\]](#)
81. Neyshabur, B.; Bhojanapalli, S.; Chakrabarti, A. Stabilizing GAN Training with Multiple Random Projections. *arXiv* **2017**, arXiv:1705.07831. [\[CrossRef\]](#)
82. Zhang, Z.; Li, M.; Yu, J. On the Convergence and Mode Collapse of GAN. In Proceedings of the SIGGRAPH Asia 2018 Technical Briefs, Tokyo, Japan, 4–7 December 2018. [\[CrossRef\]](#)
83. Oeldorf, C.; Spanakis, G. LoGANv2: Conditional Style-Based Logo Generation with Generative Adversarial Networks. In Proceedings of the 2019 18th IEEE International Conference on Machine Learning And Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 462–468. [\[CrossRef\]](#)
84. Andreini, P.; Bonechi, S.; Bianchini, M.; Mecocci, A.; Scarselli, F. Image generation by GAN and style transfer for agar plate image segmentation. *Comput. Methods Programs Biomed.* **2020**, *184*, 105268. [\[CrossRef\]](#) [\[PubMed\]](#)

85. Majurski, M.; Manescu, P.; Padi, S.; Schaub, N.; Hotaling, N.; Simon, C., Jr.; Bajcsy, P. Cell Image Segmentation Using Generative Adversarial Networks, Transfer Learning, and Augmentations. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–17 June 2019; pp. 1114–1122. [\[CrossRef\]](#)
86. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems*; Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2015; Volume 28.
87. Chowdhary, K.R. Natural Language Processing. In *Fundamentals of Artificial Intelligence*; Springer: New Delhi, India, 2020; pp. 603–649. [\[CrossRef\]](#)
88. Parvaiz, A.; Khalid, M.A.; Zafar, R.; Ameer, H.; Ali, M.; Fraz, M.M. Vision Transformers in medical computer vision—A contemplative retrospection. *Eng. Appl. Artif. Intell.* **2023**, *122*, 106126. [\[CrossRef\]](#)
89. Strudel, R.; Garcia, R.; Laptev, I.; Schmid, C. Segmenter: Transformer for Semantic Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 7242–7252. [\[CrossRef\]](#)
90. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in Vision: A Survey. *ACM Comput. Surv.* **2022**, *54*, 1–41. [\[CrossRef\]](#)
91. Xu, M.; Dai, W.; Liu, C.; Gao, X.; Lin, W.; Qi, G.J.; Xiong, H. Spatial-Temporal Transformer Networks for Traffic Flow Forecasting. *arXiv* **2021**, arXiv:2001.02908. [\[CrossRef\]](#)
92. Giuliari, F.; Hasan, I.; Cristani, M.; Galasso, F. Transformer Networks for Trajectory Forecasting. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 10335–10342. [\[CrossRef\]](#)
93. Dwivedi, V.P.; Bresson, X. A Generalization of Transformer Networks to Graphs. *arXiv* **2020**, arXiv:2012.09699. [\[CrossRef\]](#)
94. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A Survey on Visual Transformer. *arXiv* **2020**, arXiv:2012.12556. [\[CrossRef\]](#)
95. Liu, C.; Zoph, B.; Neumann, M.; Shlens, J.; Hua, W.; Li, L.J.; Fei-Fei, L.; Yuille, A.; Huang, J.; Murphy, K. Progressive Neural Architecture Search. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer Nature: Cham, Switzerland, 2018; pp. 19–35. [\[CrossRef\]](#)
96. Ren, P.; Xiao, Y.; Chang, X.; Huang, P.Y.; Li, Z.; Chen, X.; Wang, X. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–34. [\[CrossRef\]](#)
97. White, C.; Safari, M.; Sukthankar, R.; Ru, B.; Elsken, T.; Zela, A.; Dey, D.; Hutter, F. Neural Architecture Search: Insights from 1000 Papers. *arXiv* **2023**, arXiv:2301.08727. [\[CrossRef\]](#)
98. Zoph, B.; Le, Q.V. Neural Architecture Search with Reinforcement Learning. *arXiv* **2016**, arXiv:1611.01578. [\[CrossRef\]](#)
99. Mellor, J.; Turner, J.; Storkey, A.; Crowley, E.J. Neural Architecture Search without Training. In Proceedings of the 38th International Conference on Machine Learning, PMLR, Online, 18–24 July 2021; Meila, M., Zhang, T., Eds.; Proceedings of Machine Learning Research, Volume 139, pp. 7588–7598.
100. Xie, L.; Chen, X.; Bi, K.; Wei, L.; Xu, Y.; Wang, L.; Chen, Z.; Xiao, A.; Chang, J.; Zhang, X.; et al. Weight-sharing neural architecture search: A battle to shrink the optimization gap. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–37. [\[CrossRef\]](#)
101. Real, E.; Aggarwal, A.; Huang, Y.; Le, Q.V. Regularized Evolution for Image Classifier Architecture Search. *arXiv* **2018**, arXiv:1802.01548. [\[CrossRef\]](#)
102. Real, E.; Moore, S.; Selle, A.; Saxena, S.; Leon-Suematsu, Y.I.; Le, Q.V.; Kurakin, A. Large-Scale Evolution of Image Classifiers. *arXiv* **2017**, arXiv:1703.01041. [\[CrossRef\]](#)
103. White, C.; Neiswanger, W.; Savani, Y. Bananas: Bayesian optimization with neural architectures for neural architecture search. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 10293–10301. [\[CrossRef\]](#)
104. Weng, Y.; Zhou, T.; Li, Y.; Qiu, X. NAS-Unet: Neural Architecture Search for Medical Image Segmentation. *IEEE Access* **2019**, *7*, 44247–44257. [\[CrossRef\]](#)
105. Liu, C.; Chen, L.C.; Schroff, F.; Adam, H.; Hua, W.; Yuille, A.L.; Fei-Fei, L. Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 82–92. [\[CrossRef\]](#)
106. Fan, Z.; Hu, G.; Sun, X.; Wang, G.; Dong, J.; Su, C. Self-attention neural architecture search for semantic image segmentation. *Knowl.-Based Syst.* **2022**, *239*, 107968. [\[CrossRef\]](#)
107. Zhang, X.; Xu, H.; Mo, H.; Tan, J.; Yang, C.; Wang, L.; Ren, W. DCNAS: Densely Connected Neural Architecture Search for Semantic Image Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13951–13962. [\[CrossRef\]](#)
108. Liu, H.; Simonyan, K.; Yang, Y. DARTS: Differentiable Architecture Search. *arXiv* **2018**, arXiv:1806.09055. [\[CrossRef\]](#)
109. Shaw, A.; Hunter, D.; Landola, F.; Sidhu, S. SqueezeNAS: Fast Neural Architecture Search for Faster Semantic Segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Seoul, Republic of Korea, 27–28 October 2019; pp. 2014–2024. [\[CrossRef\]](#)

110. Yuan, B.; Zhao, D. A Survey on Continual Semantic Segmentation: Theory, Challenge, Method and Application. *arXiv* **2023**, arXiv:2310.14277. [\[CrossRef\]](#) [\[PubMed\]](#)
111. Wu, W.; Zhao, Y.; Li, Z.; Shan, L.; Zhou, H.; Shou, M.Z. Continual Learning for Image Segmentation with Dynamic Query. *arXiv* **2023**, arXiv:2311.17450. [\[CrossRef\]](#)
112. González, C.; Sakas, G.; Mukhopadhyay, A. What is Wrong with Continual Learning in Medical Image Segmentation? *arXiv* **2020**, arXiv:2010.11008. [\[CrossRef\]](#)
113. Álvarez, L.; Valverde, S.; Rovira, À.; Lladó, X. Mitigating catastrophic forgetting in Multiple sclerosis lesion segmentation using elastic weight consolidation. *NeuroImage Clin.* **2025**, *46*, 103795. [\[CrossRef\]](#) [\[PubMed\]](#)
114. Douillard, A.; Chen, Y.; Dapogny, A.; Cord, M. PLOP: Learning without Forgetting for Continual Semantic Segmentation. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 4039–4049. [\[CrossRef\]](#)
115. Liu, H.; Zhou, Y.; Liu, B.; Zhao, J.; Yao, R.; Shao, Z. Incremental learning with neural networks for computer vision: A survey. *Artif. Intell. Rev.* **2022**, *56*, 4557–4589. [\[CrossRef\]](#)
116. Tian, M.; Yang, Q.; Gao, Y. Multi-scale Multi-task Distillation for Incremental 3D Medical Image Segmentation. In Proceedings of the Computer Vision—ECCV 2022 Workshops, Tel Aviv, Israel, 23–27 October 2022; Proceedings, Part III; Springer Nature: Berlin/Heidelberg, Germany, 2023; pp. 369–384. [\[CrossRef\]](#)
117. Michieli, U.; Zanuttigh, P. Continual Semantic Segmentation via Repulsion-Attraction of Sparse and Disentangled Latent Representations. *arXiv* **2021**, arXiv:2103.06342. [\[CrossRef\]](#)
118. Maracani, A.; Michieli, U.; Toldo, M.; Zanuttigh, P. RECALL: Replay-based Continual Learning in Semantic Segmentation. *arXiv* **2021**, arXiv:2108.03673. [\[CrossRef\]](#)
119. Everingham, M.; van Gool, L.; Williams, C.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [\[CrossRef\]](#)
120. Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.G.; Lee, S.W.; Fidler, S.; Urtasun, R.; Yuille, A. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 891–898. [\[CrossRef\]](#)
121. Lepelaars, C. CamVid (Cambridge-Driving Labeled Video Database). 2020. Available online: <https://www.kaggle.com/datasets/carlolepelelaars/camvid> (accessed on 14 May 2025).
122. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 3354–3361. [\[CrossRef\]](#)
123. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets Robotics: The KITTI Dataset. *Int. J. Robot. Res. (IJRR)* **2013**, *32*, 1231–1237. [\[CrossRef\]](#)
124. Fritsch, J.; Kuehnl, T.; Geiger, A. A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms. In Proceedings of the International Conference on Intelligent Transportation Systems (ITSC), The Hague, The Netherlands, 6–9 October 2013; pp. 1693–1700. [\[CrossRef\]](#)
125. Menze, M.; Geiger, A. Object Scene Flow for Autonomous Vehicles. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3061–3070. [\[CrossRef\]](#)
126. Brox, T.; Malik, J. Object Segmentation by Long Term Analysis of Point Trajectories. In Proceedings of the Computer Vision—ECCV 2010, Heraklion, Crete, Greece, 5–11 September 2010; Daniilidis, K., Maragos, P., Paragios, N., Eds.; Springer Nature: Berlin/Heidelberg, Germany, 2010; pp. 282–295. [\[CrossRef\]](#)
127. Chen, X.; Mottaghi, R.; Liu, X.; Fidler, S.; Urtasun, R.; Yuille, A. Detect What You Can: Detecting and Representing Objects using Holistic Models and Body Parts. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1979–1986. [\[CrossRef\]](#)
128. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, L. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer Nature: Cham, Switzerland, 2014; pp. 740–755. [\[CrossRef\]](#)
129. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene Parsing through ADE20K Dataset. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5122–5130. [\[CrossRef\]](#)
130. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In Proceedings of the Proceedings Eighth IEEE International Conference on Computer Vision, ICCV 2001, Vancouver, BC, Canada, 7–14 July 2001; Volume 2, pp. 416–423.
131. Hariharan, B.; Arbelaez, P.; Bourdev, L.; Maji, S.; Malik, J. Semantic Contours from Inverse Detectors. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011; pp. 991–998. [\[CrossRef\]](#)

132. Xia, J.; Yokoya, N.; Adriano, B.; Broni-Bediako, C. OpenEarthMap: A Benchmark Dataset for Global High-Resolution Land Cover Mapping. *arXiv* **2022**, arXiv:2210.10732. [[CrossRef](#)]
133. Liu, C.; Yuen, J.; Torralba, A. SIFT Flow: Dense Correspondence Across Scenes and Its Applications. In *Dense Image Correspondences for Computer Vision*; Hassner, T., Liu, C., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 15–49. [[CrossRef](#)]
134. Gould, S.; Fulton, R.; Koller, D. Decomposing a Scene into Geometric and Semantically Consistent Regions. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 1–8. [[CrossRef](#)]
135. Menze, B.H.; Jakab, A.; Bauer, S.; Kalpathy-Cramer, J.; Farahani, K.; Kirby, J.; Burren, Y.; Porz, N.; Slotboom, J.; Wiest, R.; et al. The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). *IEEE Trans. Med. Imaging* **2015**, *34*, 1993–2024. [[CrossRef](#)] [[PubMed](#)]
136. Kavur, A.E.; Gezer, N.S.; Barış, M.; Aslan, S.; Conze, P.H.; Groza, V.; Pham, D.D.; Chatterjee, S.; Ernst, P.; Özkan, S.; et al. CHAOS Challenge-combined (CT-MR) healthy abdominal organ segmentation. *Med. Image Anal.* **2021**, *69*, 101950. [[CrossRef](#)] [[PubMed](#)]
137. Codella, N.; Rotemberg, V.; Tschandl, P.; Celebi, M.E.; Dusza, S.; Gutman, D.; Helba, B.; Kalloo, A.; Liopyris, K.; Marchetti, M.; et al. Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC). *arXiv* **2019**, arXiv:1902.03368. [[CrossRef](#)]
138. Demir, I.; Koperski, K.; Lindenbaum, D.; Pang, G.; Huang, J.; Basu, S.; Hughes, F.; Tuia, D.; Raskar, R. DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018. [[CrossRef](#)]
139. Etten, A.V.; Lindenbaum, D.; Bacastow, T.M. SpaceNet: A Remote Sensing Dataset and Challenge Series. *arXiv* **2019**, arXiv:1807.01232. [[CrossRef](#)]
140. Silberman, N.; Hoiem, D.; Kohli, P.; Fergus, R. Indoor Segmentation and Support Inference from RGBD Images. In Proceedings of the 12th European Conference on Computer Vision (ECCV 2012), Florence, Italy, 7–13 October 2012; Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C., Eds.; Springer Nature: Berlin/Heidelberg, Germany, 2010; pp. 746–760. [[CrossRef](#)]
141. Song, S.; Lichtenberg, S.P.; Xiao, J. SUN RGB-D: A RGB-D scene understanding benchmark suite. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 567–576. [[CrossRef](#)]
142. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.A.; Nießner, M. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. *arXiv* **2017**, arXiv:1702.04405. [[CrossRef](#)]
143. Armeni, I.; Sax, A.; Zamir, A.R.; Savarese, S. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *arXiv* **2017**, arXiv:1702.01105. [[CrossRef](#)]
144. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. In Proceedings of the NIPS'19: 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
145. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: <https://www.tensorflow.org> (accessed on 14 May 2025).
146. Micikevicius, P.; Narang, S.; Alben, J.; Diamos, G.; Elsen, E.; Garcia, D.; Ginsburg, B.; Houston, M.; Kuchaiev, O.; Venkatesh, G.; et al. Mixed precision training. *arXiv* **2018**, arXiv:1710.03740. [[PubMed](#)]
147. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2999–3007. [[CrossRef](#)]
148. Matos-Carvalho, J.P.; Correia, S.D.; Tomic, S. Sensitivity Analysis of LSTM Networks for Fall Detection Wearable Sensors. In Proceedings of the 2023 6th Conference on Cloud and Internet of Things (CIoT), Lisbon, Portugal, 20–22 March 2023; pp. 112–118. [[CrossRef](#)]
149. Correia, S.D.; Matos-Carvalho, J.P.; Tomic, S. Quantization with Gate Disclosure for Embedded Artificial Intelligence Applied to Fall Detection. In Proceedings of the GoodIT '24 2024 International Conference on Information Technology for Social Good, Bremen, Germany, 4–6 September 2024; pp. 84–87. [[CrossRef](#)]
150. Buongiorno, R.; Germanese, D.; Colligiani, L.; Fanni, S.C.; Romei, C.; Colantonio, S. Chapter 9—Artificial intelligence for chest imaging against COVID-19: An insight into image segmentation methods. In *Artificial Intelligence in Healthcare and COVID-19*; Chatterjee, P., Esposito, M., Eds.; Intelligent Data-Centric Systems; Academic Press: Cambridge, MA, USA, 2023; pp. 167–200. [[CrossRef](#)]
151. Ulku, I.; Akagündüz, E. A Survey on Deep Learning-based Architectures for Semantic Segmentation on 2D Images. *Appl. Artif. Intell.* **2022**, *36*, 2032924. [[CrossRef](#)]
152. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 658–666. [[CrossRef](#)]
153. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Learning a discriminative feature network for semantic segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 1857–1866. [[CrossRef](#)]

154. Setiawan, A.W. Image Segmentation Metrics in Skin Lesion: Accuracy, Sensitivity, Specificity, Dice Coefficient, Jaccard Index, and Matthews Correlation Coefficient. In Proceedings of the 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 17–18 November 2020; pp. 97–102. [\[CrossRef\]](#)
155. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *arXiv* **2014**, arXiv:1411.4038. [\[CrossRef\]](#)
156. Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; Torr, P.H.S. Conditional Random Fields as Recurrent Neural Networks. *arXiv* **2015**, arXiv:1502.03240. [\[CrossRef\]](#)
157. Dai, J.; He, K.; Sun, J. BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation. *arXiv* **2015**, arXiv:1503.01640. [\[CrossRef\]](#)
158. Lin, G.; Shen, C.; Reid, I.D.; van den Hengel, A. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv* **2015**, arXiv:1504.01013. [\[CrossRef\]](#)
159. Liu, Z.; Li, X.; Luo, P.; Loy, C.C.; Tang, X. Semantic Image Segmentation via Deep Parsing Network. *arXiv* **2015**, arXiv:1509.02634. [\[CrossRef\]](#)
160. Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *arXiv* **2016**, arXiv:1606.00915. [\[CrossRef\]](#) [\[PubMed\]](#)
161. Wu, Z.; Shen, C.; van den Hengel, A. Wider or Deeper: Revisiting the ResNet Model for Visual Recognition. *arXiv* **2016**, arXiv:1611.10080. [\[CrossRef\]](#)
162. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. *arXiv* **2016**, arXiv:1612.01105. [\[CrossRef\]](#)
163. Peng, C.; Zhang, X.; Yu, G.; Luo, G.; Sun, J. Large Kernel Matters—Improve Semantic Segmentation by Global Convolutional Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1743–1751. [\[CrossRef\]](#)
164. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation. *arXiv* **2018**, arXiv:1808.00897. [\[CrossRef\]](#)
165. Li, Y.; Song, L.; Chen, Y.; Li, Z.; Zhang, X.; Wang, X.; Sun, J. Learning Dynamic Routing for Semantic Segmentation. *arXiv* **2020**, arXiv:2003.10401. [\[CrossRef\]](#)
166. Wang, W.; Howard, A. MOSAIC: Mobile Segmentation via decoding Aggregated Information and encoded Context. *arXiv* **2021**, arXiv:2112.11623. [\[CrossRef\]](#)
167. Jeevan, P.; Viswanathan, K.; Sethi, A. WaveMix-Lite: A Resource-efficient Neural Network for Image Analysis. *arXiv* **2022**, arXiv:2205.14375. [\[CrossRef\]](#)
168. Wu, J.; Kuang, H.; Lu, Q.; Lin, Z.; Shi, Q.; Liu, X.; Zhu, X. M-FasterSeg: An Efficient Semantic Segmentation Network Based on Neural Architecture Search. *arXiv* **2022**, arXiv:2112.07918. [\[CrossRef\]](#)
169. Bhardwaj, K.; Cheng, H.P.; Priyadarshi, S.; Li, Z. ZiCo-BC: A Bias Corrected Zero-Shot NAS for Vision Tasks. *arXiv* **2023**, arXiv:2309.14666. [\[CrossRef\]](#)
170. Jeong, J.; Yu, J.; Park, G.; Han, D.; Yoo, Y. GeNAS: Neural Architecture Search with Better Generalization. *arXiv* **2023**, arXiv:2305.08611. [\[CrossRef\]](#)
171. Xiong, Z.; Amein, M.; Therrien, O.; Gross, W.J.; Meyer, B.H. FMAS: Fast Multi-Objective SuperNet Architecture Search for Semantic Segmentation. *arXiv* **2023**, arXiv:2303.16322. [\[CrossRef\]](#)
172. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. *arXiv* **2021**, arXiv:2103.00020. [\[CrossRef\]](#)
173. Liu, H.; Li, C.; Wu, Q.; Lee, Y.J. Visual Instruction Tuning. *arXiv* **2023**, arXiv:2304.08485. [\[CrossRef\]](#) [\[PubMed\]](#)
174. Xu, M.; Zhang, Z.; Wei, F.; Lin, Y.; Cao, Y.; Hu, H.; Bai, X. A Simple Baseline for Open-Vocabulary Semantic Segmentation with Pre-trained Vision-language Model. *arXiv* **2022**, arXiv:2112.14757. [\[CrossRef\]](#)
175. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment Anything. *arXiv* **2023**, arXiv:2304.02643. [\[CrossRef\]](#) [\[PubMed\]](#)
176. Dong, X.; Bao, J.; Zheng, Y.; Zhang, T.; Chen, D.; Yang, H.; Zeng, M.; Zhang, W.; Yuan, L.; Chen, D.; et al. MaskCLIP: Masked Self-Distillation Advances Contrastive Language-Image Pretraining. *arXiv* **2023**, arXiv:2208.12262. [\[CrossRef\]](#)
177. Liu, Q.; Wen, Y.; Han, J.; Xu, C.; Xu, H.; Liang, X. Open-world Semantic Segmentation via Contrasting and Clustering Vision-Language Embedding. *arXiv* **2022**, arXiv:2207.08455. [\[CrossRef\]](#)
178. Baranchuk, D.; Rubachev, I.; Voynov, A.; Khrulkov, V.; Babenko, A. Label-Efficient Semantic Segmentation with Diffusion Models. *arXiv* **2022**, arXiv:2112.03126. [\[CrossRef\]](#)
179. Zou, X.; Yang, J.; Zhang, H.; Li, F.; Li, L.; Wang, J.; Wang, L.; Gao, J.; Lee, Y.J. Segment Everything Everywhere All at Once. *arXiv* **2023**, arXiv:2304.06718. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.