PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
الجمهـــوريـــــة الجزائـريـــــة الديـمقـراطيـــــة الشـعـبيـــــة
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
وزارة التعليـم العالـــي والبحـــث العلمـي
HIGHER SCHOOL OF COMPUTER SCIENCE AND DIGITAL TECHNOLOGIES-BEJAIA
المـــدرسة العـــليا في علـــوم وتكنـولوجيـات الاعـــلام الالـــي بجـــاية

**Dissertation Submitted to the Department Of Computer Science in Partial Fulfillment of the Requirements for Master's Degree in Computer Science**

**Specialty: Artificial Intelligence and Data Sciences**

Submitted By: **Anis MOHAMMEDI**

# Seismic Image Segmentation based on Deep Learning for the Characterization of Hydrocarbon Reservoirs in the Oil & Gas Industry : Comparative Study

Supervised by: **Dr.** Abderrazek SEBAA (ESTIN)

**Dr.** Kamel SOUADIH (SONATRACH RTC-BEJAIA)

**Members of jury:**

| | | |
|---|---|---|
| ▪ **Dr.** Sylia ZENADJI | President | ESTIN |
| ▪ **Dr.** Chemseddine BERBAGUE | Examiner | ESTIN |
| ▪ **PhD Student.** Amine BECHAR | Examiner | ESTIN |
| ▪ **PhD Student.** Chawki ABBES | Examiner | ESTIN |

Academic year: 2023/2024

# Abstract

Salt Domes are subsurface geological formations critical for hydrocarbon exploration and storage. They often serve as traps for hydrocarbons, significantly contributing to the defining characteristics of reservoirs in the oil and gas industry. Accurately identifying and delineating these structures is essential for efficient resource management. Traditional methods for detecting salt domes rely heavily on manual interpretation, which is time-consuming and prone to errors. Recent advancements in Deep Learning, particularly in Semantic Segmentation, offer promising solutions to automate and enhance this process. This study explores the application of state-of-the-art Deep Learning models, including Convolutional Neural Networks (CNNs), U-Net architectures, and Transformer-based approaches, for the Semantic Segmentation of Salt Domes in Seismic Images. Generative AI techniques are also examined for data augmentation and enhancing model robustness. The integration of these advanced models aims to improve the precision and reliability of Salt Dome identification, potentially transforming the field of geophysical exploration.

***Keywords—*** Salt Domes, Semantic Segmentation, Deep Learning, Generative AI, CNN, U-Net, Transformers, Seismic Imaging, Hydrocarbon Exploration

As when a fog disperseth gradually
Our vision traces what the mist involves
Condens'd in air; so piercing through the gross
And gloomy atmosphere, as more and more
We near'd toward the brink, mine error fled ...

The Divine Comedy, Vol. 1 (Inferno) - Canto XXXI , Dante Alighieri

# Acknowledgment

I would like to extend my sincere gratitude to everyone who has supported and guided me throughout the completion of my master's degree thesis.

First and foremost, I am deeply grateful to my advisor, Dr. SOUADIH Kamel, for his exceptional guidance, insightful advice, and unwavering support. His expertise and encouragement have been crucial in the development and completion of this thesis.

I would also like to thank the faculty and staff of the School for providing a conducive learning environment and the resources necessary for my research. I also extend my sincere thanks to Dr. SEBAA Abderrazek for his valuable notices and advices. His critical insights and constructive feedback have significantly improved the quality of this work. His attention to detail and willingness to share his knowledge have been greatly appreciated.

I am particularly thankful to my classmates and friends who have been a constant source of support and motivation. Their constructive feedback and camaraderie have significantly enriched my academic experience.

I would like to acknowledge the financial and logistical support provided by the library stuff especially, Mr. ADOUANE Nacer. Your contributions have been instrumental in enabling this research.

A heartfelt thank you goes to my family for their endless support, patience, and love. To my parents and my dear sister, your belief in my abilities and your constant encouragement have been my greatest strength.

Finally, I would like to thank everyone who has contributed, directly or indirectly, to the successful completion of this thesis. Your support and encouragement have been greatly appreciated.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

**2D** Two-Dimensional. 12

**3D** Three-Dimensional. 12

**ADADELTA** Stochastic Optimization Technique. 52

**AI** Artificial Intelligence. 9

**AlexNet** AlexNet is the name of a convolutional neural network (CNN) architecture, designed by Alex Krizhevsky with others. 17

**BART** Bidirectional and Auto-Regressive Transformers. 30

**BCE** Binary Cross Entropy. 43

**BRIEF** Binary Robust Independent Elementary Features. 7

**CAE** Convolutional AutoEncoder. 32

**CNF** Conditional Normalizing Flow. 34

**CNN** Convolutional Neural Networks. 11

**ConvLay** Convolutional Layer. 12

**CV** Computer Vision. 27

**DL** Deep Learning. 3

**DSC** Dice Similarity Coefficient. 37

**FC** Fully Connected Layer. 12

**FCN** Fully Convolutional Networks. 19

**FFN** Feed-Forward Network. 28

**FLOP** Floating Point Operations. 38

**FN** False Negative. 37

**FP** False Positive. 37

**FPNet** Feature Pyramid Network. 24

**FPS** Frames Per Second. 38

**GAN** Generative Adversarial Networks. 31

**GOFAI** Good Old Fashion AI. 10

**GPT** Generative Pre-trained Transformer. 30

**GPU** Graphics Processing Unit. 51

**HD** Hausdorff distance. 38

**IoU** Intersection over Union. 37

**MAD** Mean Absolute Distance. 38

**MIoU** Mean Intersection over Union. 39

**ML** Machine Learning. 9

**MLP** Multi Layer Perceptron. 12

**NLP** Natural Language Processing. 27

**PA** Pixel Accuracy. 37

**PSPNet** Pyramid Scene Parsing Network. 26

**QKV** Query-Key-Value. 28

**ReLU** Rectified Linear Units. 17

**ResNet** Residual Network. 17

**RGB** RED - GREEN - BLUE. 12

**RNN** Recurrent Neural Network. 27

**SE** Squeeze-and-Excitation Networks. 43

**SIFT** Scale-Invariant Feature Transform. 7

**SSL** Semi-supervised learning. 35

**SURF** Speeded-Up Robust Features. 7

**TN** True Negative. 37

**TP** True Positive. 37

**TPU** Tensor Processing Unit. 52

**TTA** Test-Time Augmentation. 48

**U-net** A type of convolutional neural network developed for biomedical image segmentation, also used in other domains. 22

**VaE** Variational AutoEncoder. 32

**VGG** Visual Geometry Group. 19

**ViT** Vision Transformer. 30

# General Introduction

The exploration and extraction of hydrocarbon resources, such as oil and natural gas, have long been crucial endeavors for meeting the world's ever-increasing energy demands. One of the key challenges in this domain lies in accurately identifying and mapping geological structures that can serve as potential reservoirs for these valuable resources. Among these structures, salt domes have garnered significant attention due to their propensity to trap hydrocarbons within their intricate formations.

Salt domes are massive underground deposits of salt that have been pushed upward through overlying sedimentary rock layers over millions of years, creating dome-like structures. These formations can act as traps for hydrocarbons, making their precise delineation a critical step in the exploration and extraction processes. Traditionally, the interpretation of seismic data has relied on manual analysis by skilled experts, a time-consuming and labor-intensive task that can take weeks or even months to complete. With the advent of deep learning techniques, particularly in the field of computer vision, new opportunities have emerged to automate and expedite the interpretation of seismic data.

Semantic segmentation, a foundational task in computer vision, aims to partition an image into semantically meaningful regions, offering a promising solution for the accurate delineation of salt domes in seismic images. By leveraging state-of-the-art deep learning architectures, this research endeavors to develop robust and accurate models for automating the detection and segmentation of salt domes in seismic data.

The overarching goal of this research is to explore and evaluate various deep learning techniques for semantic segmentation, with a specific focus on their application in the detection and delineation of salt domes in seismic images. Through a comprehensive literature review, dataset acquisition and preprocessing, model implementation and fine-tuning, and rigorous evaluation, this study aims to contribute to the advancement of automated seismic data interpretation.

The work plan for this research will involve several key steps, including:

1. Conducting a thorough literature review to understand the current state-of-the-art techniques and methodologies employed in semantic segmentation, with a particular emphasis on their applications in seismic data analysis.

2. Evaluate the performance of the developed models using appropriate metrics, such as intersection over union (IoU) and compare their results against established baselines and state-of-the-art methods.

3. Investigate techniques for semi-supervised learning and domain adaptation to address the challenges of limited labeled data and domain gaps across different seismic datasets, analyze the strengths and limitations of the employed deep learning techniques, identifying areas for further improvement and potential future research directions.

By addressing these objectives, this research aims to contribute to the advancement of automated seismic data interpretation, ultimately facilitating more efficient and cost-effective hydrocarbon exploration while promoting sustainable energy practices.

# Chapter 1

# Fundamental Concepts

## 1.1 Introduction

This chapter covers the essential concepts underpinning our study, beginning with Seismic Imaging and its pivotal role in geosciences. We then explore the geological formation and significance of Salt Domes, followed by Semantic Segmentation, highlighting its importance in precise image analysis and interpretation. Finally we delve Deep Learning (DL) principles, particularly Neural Networks.

## 1.2 Seismic Survey and Salt Domes

### 1.2.1 Salt Domes

Salt Domes, also known as salt diapirs, are geological structures that form when underground salt layers or beds rise toward the Earth's surface [1]. These salt layers, composed primarily of halite (rock salt), are less dense than the surrounding sedimentary rocks. As a result, they can migrate vertically over geological time, pushing their way through the layers of sedimentary rock above them (see Figure 1.1).

Over millions of years, the salt can intrude and deform the overlying rock layers, causing the formation of a dome-like structure. These Salt Domes can vary in size from a few meters to several kilometers in height and width [1].

Salt Domes hold significant relevance in hydrocarbon exploration, serving as potential reservoirs for oil and gas, making them crucial in the energy industry. Understanding and detecting Salt Domes are essential for efficient hydrocarbon resource exploration for theses reasons [2] :

- **Hydrocarbon Traps** : Salt Domes can serve as traps for hydrocarbons, such as oil and natural gas. Over millions of years, salt can create structural deformations in the overlying sedimentary rock layers, forming pockets where hydrocarbons can accumulate. This makes Salt Domes important targets for oil and gas exploration.

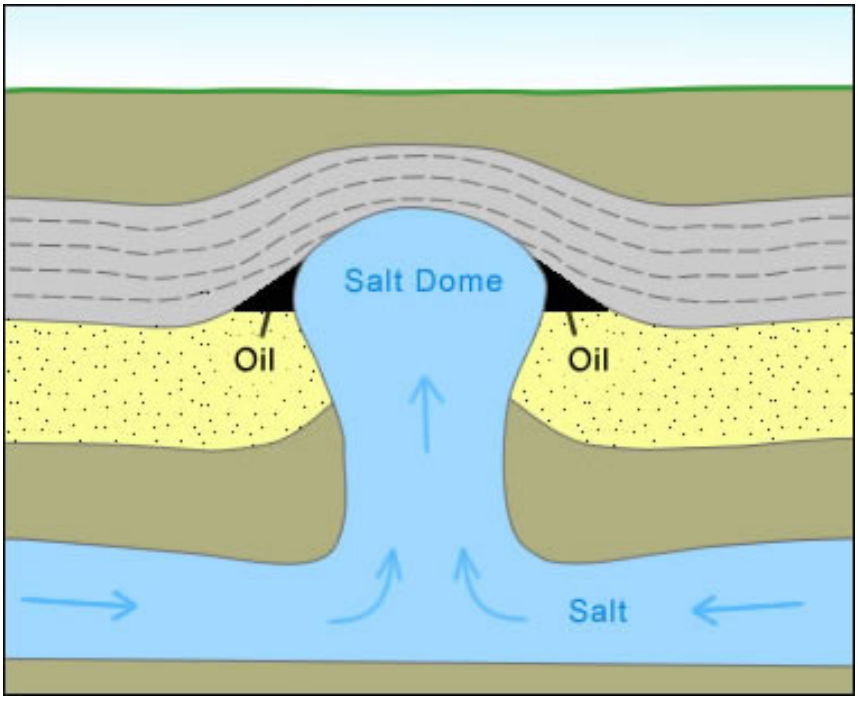



Figure 1.1: Formation of a Salt Dome through Sedimentary Rock layers [2].

- **A Source of Sulfur** : Sulfur recovery from Salt Domes with sulfur-rich cap rock involves drilling a well and injecting superheated water and air to melt and bring the sulfur to the surface. However, this method is generally less cost-effective than obtaining sulfur as a byproduct of crude oil refining and natural gas processing, which is the primary source of sulfur production today.
- **Salt Production** : Certain Salt Domes have been utilized for underground mining operations, extracting salt used as a raw material in the chemical industry and for de-icing snowy highways. In some instances, Salt Domes are mined through a solution mining technique, where hot water is injected down a well to dissolve the salt. The saline solution is then brought to the surface through production wells, where the salt is recovered through evaporation or employed in chemical processes.
- **Underground Storage Reservoirs** : Certain salt Dome mines have been deliberately sealed and repurposed as storage facilities for oil, natural gas, and hydrogen. In both the United States and Russia, Salt Domes are utilized as national repositories for government reserves of helium gas. The unique quality of salt as a rock with exceptionally low permeability[1] enables it to securely retain even the smallest helium atoms.
- **Waste Disposal** : Salt's impermeable nature and ability to self-seal fractures make Salt Domes suitable for hazardous waste disposal, including oil field drilling waste, while also being considered for high-level nuclear waste storage in the United States.

[1]Permeability is how easily a liquid or gas can move through a material with tiny holes, like water soaking through a sponge.

There are a variety of techniques used for detecting Salt Domes but the cheapest, the least dangerous, the effective method is by using Seismic Surveys.

### 1.2.2 Seismic Survey

As stated in [3], Seismic Surveying is an important geophysical exploration method because it can detect subsurface features of all sizes. Seismic methods involve sending sound waves into the Earth and measuring their reflections to determine the shapes and properties of subsurface layers. Early oil explorers found oil by drilling at natural oil seeps and large folds in exposed rocks. Once these easy targets were depleted, geologists turned to seismic surveying to find more elusive oil and gas traps. Seismic technology has been used to measure water depths and detect icebergs since the early 1900s. In 1924, seismic data was first used to discover an oil field in Texas. Seismic Surveys rely on the propagation of seismic waves through the Earth. These waves are generated artificially at the surface using specialized equipment and then travel through the subsurface layers. By recording and analyzing the arrival times and characteristics of these waves at different receivers (seismic sensors), geophysicists can infer information about the subsurface structure. Once seismic data is collected, it undergoes processing to remove noise, correct for variations in the Earth's surface, and enhance the signal. After processing, geophysicists interpret the data to create subsurface images, maps, and models. Interpretation involves identifying geological features, such as rock layers, faults, and fluid reservoirs, based on seismic wave reflections, refractions, and diffractions.

## 1.3 Influence of Salt Domes in Characterization of Hydrocarbon Reservoirs

Salt Domes significantly influence the characterization of oil and gas reservoirs. They create trapping structures, sealing layers, and compartmentalized reservoirs with varying properties. Salt domes pose challenges for seismic imaging and drilling operations due to their unique properties and mobility. Understanding salt behavior is crucial for field development, production strategies, and long-term management as salt movement impacts well performance and reservoir drainage. Integrating geological, geophysical, and engineering data is essential for accurately characterizing Salt Domes and optimizing exploration, development, and production from hydrocarbon reservoirs influenced by these geological features. Additionally, studying Salt Dome structures provides insights into depositional environments, hydrocarbon generation timing, and basin evolution, aiding in understanding reservoir distribution and characteristics. Seismic data and advanced imaging techniques are vital for identifying and mapping Salt Domes, while well data refines the understanding of Salt Dome structures and their impact on reservoirs (from the introduction of [4]).

## 1.4 Seismic Surveying Challenges

Interpreting salt structures in the context of geology and geophysics can be particularly challenging due to several factors. In fact, The density of Salt Domes and the surrounding

rocks can vary significantly. Salt is typically less dense than sandstone and limestone, but more dense than shale (2.16 g/cm³ for Halite, whereas 2.71g/cm³for sandstone). This can lead to significant acoustic impedance contrasts between Salt Domes and the surrounding rocks. Acoustic impedance is a measure of how much a material resists the passage of sound waves. Therefore, seismic waves travel through salt at much higher velocities than through other rocks. This is because salt is more elastic than other rocks (4.5km/s for salt structures whereas 3.5km/s for the surrounding structures) [5].

Here are some problems which make salt interpretation difficult [6]:

- Interpret seismic data using a prior model based on the geological knowledge of the region.
- Steeply inclined flanks are difficult to map because seismic waves graze on them, making it hard to define their boundaries. This can be improved by increasing the distance between the source and receiver.
- High impedance contrast between salt and sediments traps seismic waves inside salt structures, generating multiples that interfere with primary events, making sub-saline sediments difficult to image.
- Seismic waves follow complex paths in the vicinity of salt domes, resulting in weak correlation between real and predicted data, lower image quality, and prism waves. Prism waves are generated by two reflections along the travel path from source to receiver, creating spurious artifacts. They are common in regions with steeply sloping flanks, such as salt domes.

This is why it can take weeks and demand higher quality experts to detect Salt Domes in seismic images [7], efforts have been made to expedite the process. With the mention that tracing 1 $km^2$ of the subsurface can generate large amount of data up to 600 Gigabytes of space which first demand high computational hardware especially for 3D visualisation (it's just an accumulation of 2D images) it takes years to interpret all the scans manually [8](see Figure 1.2). That's led to an increasing demand to automate the interpretation process. Since it's is a relatively recent research focus, numerous methods were proposed but for the scope of the study we will remain only on Artificial Intelligence (especially Deep Learning) contributions.

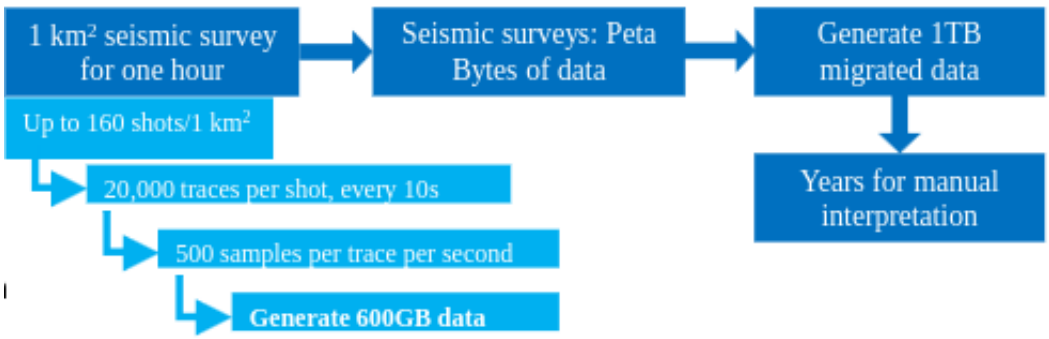



Figure 1.2: Data aquisation challenge [8].

## 1.5 Semantic Segmentation

Before we explain the concept of Semantic Segmentation, we should first give to the reader some basics of Computer Vision since the segmentation task involves Computer Vision techniques.

Based on [9]: Computer vision is a multidisciplinary domain focused on enabling computers to interpret digital images and videos akin to human visual perception. It aims to automate tasks that humans can perform with their vision. At its core, Computer Vision involves extracting, analyzing, and comprehending useful information from individual images or sequences of images. This field spans theoretical development and algorithmic design to achieve automatic visual comprehension. It addresses various data formats, including video streams, multi-camera perspectives, and complex medical imaging data. OpenCV (https://opencv.org/), a widely used library, embodies real-time Computer Vision and integrates with Artificial Intelligence frameworks (Deep Learning for instance) for image and video processing. Fundamentally, Computer Vision revolves around pixel extraction from images to discern and understand the contents within them. Here are several fundamental aspects that Computer Vision endeavors to identify within photographs [10]:

- Object Detection : Identifying the precise location of objects within the image.
- Object Recognition : Recognizing the objects present in the image and determining their respective positions.
- Object Classification : Categorizing objects into broader classes or categories.
- Object Segmentation : Segmenting and isolating the pixels that belong to each object within the image.

Over the years, advancements in Computer Vision have greatly enhanced the accuracy and speed of processing images captured from cameras. Artificial Intelligence (Machine Learning & Deep Learning) notably, has emerged as a dominant tool in this field. Previously, Computer Vision relied heavily on image processing algorithms, primarily focusing on feature extraction. Detecting elements like color, edges, corners, and objects constituted initial steps in Computer Vision tasks. These features, engineered by humans, directly influenced the accuracy and reliability of models. Traditional approaches, such as SIFT, SURF, and BRIEF, were instrumental in extracting features from raw images as shown in Figure 1.3. However, a drawback of this feature extraction approach in image classification is the need to predefine which features to seek in each image. As the number of classification categories increases or image clarity diminishes, traditional Computer Vision algorithms struggle to adapt effectively.

Due to Artificial Intelligence techniques (especially Deep Learning), Semantic segmentation have witnessed a progression from coarse to fine inference within the realm of Computer Vision research [11]. It is not an isolated field but rather an evolutionary step building upon automated classification methods. Initially rooted in classification techniques, which predict the class of objects in an image, Semantic Segmentation delves deeper by assigning labels to individual pixels based on the objects or regions they belong to. This process aims for precise inference by delineating boundaries between objects. Object Detection, a sub-

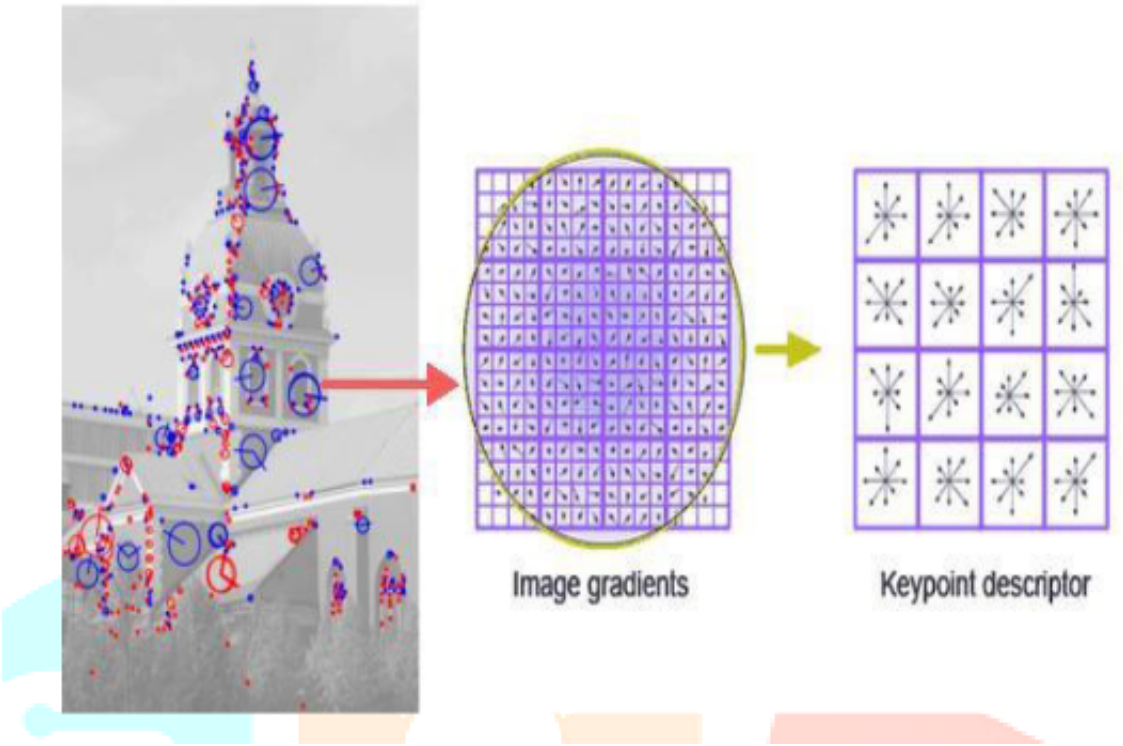



Figure 1.3: Traditional way of Feature Extraction [10].

sequent step, refines this inference by not only classifying objects but also providing their spatial locations through bounding boxes or centroids (see Figure 1.6). Previously, among the research realm, there is claiming that among the best methods used before Deep Learning was Boosting and Random Forest. There is also Instance Segmentation, but it differs from Semantic Segmentation, Instance Segmentation further refines the process by assigning distinct labels to separate instances of objects within the same class, essentially combining Object Detection and Semantic Segmentation tasks. This approach aims to achieve simultaneous solutions to these tasks. Part-based segmentation takes this evolution further by breaking down segmented objects into their constituent sub-components. Overall, Semantic Segmentation and its related tasks represent a continuum of research efforts aiming for increasingly granular and accurate understanding of visual data (see Figure 1.4).

In recent years, semantic segmentation has been applied more and more widely. It plays an important role in medical image analysis [12], automatic driving [13], virtual/augmented reality [14], video surveillance [15] and more, thanks to the advancement of Deep Learning.

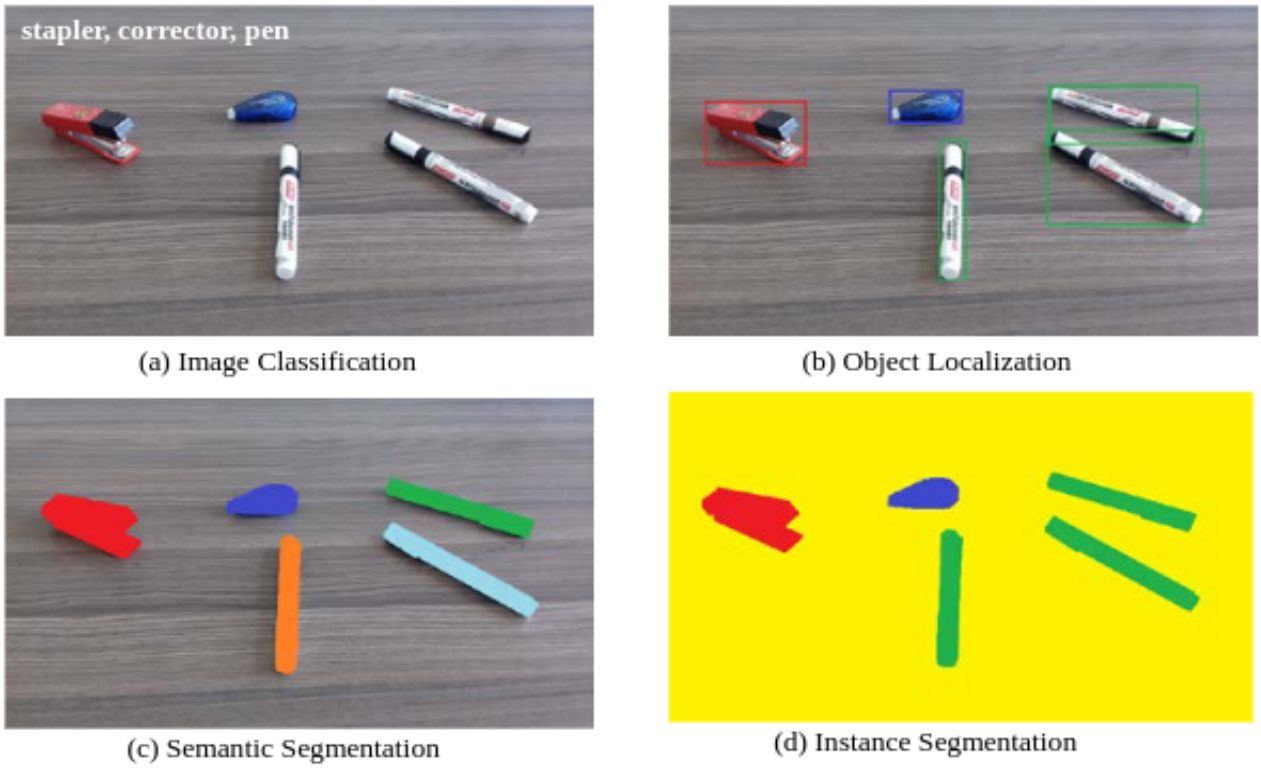



Figure 1.4: An example of different vision tasks [11].

## 1.6 Machine Learning and Deep Learning

### 1.6.1 Machine Learning

Many academics and researchers perceive "Machine Learning" (ML) as an integral component of Artificial Intelligence (AI), although it is often used interchangeably with the broader term [16]. In the literature, researchers have proposed numerous definitions for ML. One notable definition (Koza et al., 1996) offers insights into the nature of this field and its underlying principles:

“ML describes a set of methods commonly used to solve a variety of real-world problems with the help of computer systems, which can learn to solve a problem instead of being explicitly programmed to do so.”

Usually AI and ML are often used interchangeably. However, it is important to note that while ML is a part of AI, the two terms are not synonymous. ML is a technology that enables computers to learn directly from data and examples, distinguishing it from traditional programming methods that rely on predefined rules. In ML, systems are provided with a specific task and a substantial amount of data to learn from, either as examples or patterns. Through this process, the system learns how to achieve the desired output by analyzing and extracting insights from the given data. ML can be considered as a form of narrow AI (AI), as it focuses on training intelligent systems to learn specific functions based on provided data.

On the other hand, AI encompasses a broader spectrum of technologies, ranging from traditional AI, such as Good Old Fashion AI (GOFAI), to more advanced connectionist architectures like DL. ML is a subfield of AI that specifically deals with learning algorithms and their application to data and it encompasses various techniques [17].

### 1.6.2 Deep Learning

The term "Deep Learning" (DL) made its initial appearance in the field of ML during a conference by Dechter in 1986. It was further associated with Artificial Neural Networks in a publication by Aizenberg et al. in 2000 [18]. The introduction of this term marked a significant milestone in the advancement of both fields, paving the way for the development and exploration of DL algorithms and architectures [19]. DL, a subset of Artificial Neural Networks (ANNs) (see Figure 1.5), has emerged as a powerful approach in the field of ML. Inspired by the information processing mechanisms in biological systems, Artificial neural networks are flexible structures inspired by biological systems that consist of interconnected processing units called neurons. ANNs can be modified for various ML contexts and are characterized by the transmission of signals between neurons, which are weighted and adjusted during the learning process. Neurons are organized into networks with input, hidden, and output layers, and the connections between them allow for non-linear mapping.

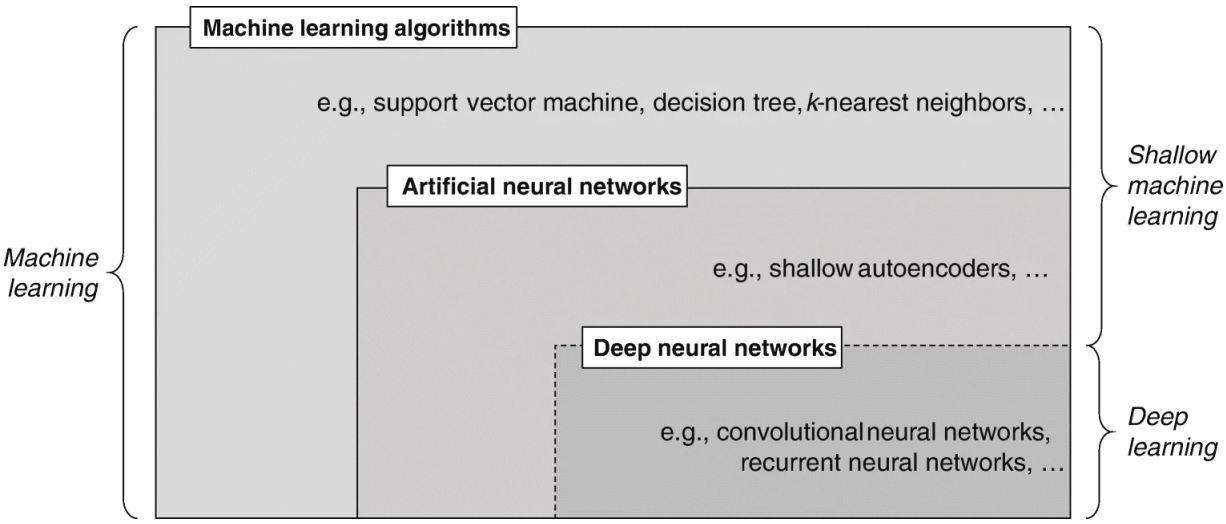
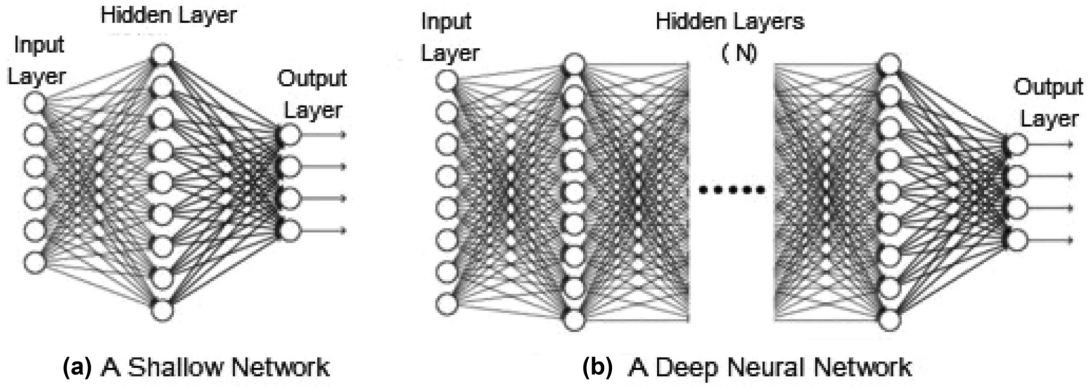



Figure 1.5: Venn diagram of machine ML algorithms learning concepts and classes (inspired by Goodfellow et al. 2016, p. 9) [20].

Deep Neural Networks, a type of ANN, have multiple hidden layers and advanced neurons that enable them to automatically learn representations from raw input data, known as DL (see Figure 1.6). DL excels in processing large and high-dimensional data like text, images, and audio. However, for low-dimensional data and limited training data, shallow ML algorithms can still produce superior and more interpretable results. While DL can achieve superhuman performance in certain tasks, it falls short in solving problems requiring strong AI capabilities [20].

For instance, DL models excel at pattern recognition and prediction tasks but fall short




**(a)** A Shallow Network **(b)** A Deep Neural Network

Figure 1.6: A general architecture of a shallow network with one hidden layer and a deep neural network with multiple hidden layers [21].

when it comes to tasks requiring strong AI capabilities, such as literal understanding and intentionality.

There are numerous DL models, but for the scope of this study, we stick in one type of the models : Convulutional Neural Networks (CNN).

CNNs are widely recognized and extensively employed in DL, offering several advantages over previous algorithms. One notable benefit is their ability to autonomously identify relevant features without human supervision. CNNs have found applications in various domains, including Computer Vision, speech processing, and face recognition. The structure of CNNs is inspired by the neural networks present in human and animal brains, particularly the visual cortex (See 2.2) .

## 1.7 Conclusion

In this chapter, we have presented the fundamental concepts to understanding Semantic Segmentation in Seismic Imaging of Salt Domes. We covered the basics of Seismic Imaging and its significance, the geological formation and importance of Salt Domes, the essential role of Semantic Segmentation in image analysis and the definition of Deep Learning with a focus on Neural Networks. These concepts provide the necessary foundation for delving into advanced segmentation methodologies and their applications in the subsequent chapters.

# Chapter 2

# Deep Learning Techniques for Semantic Segmentation

## 2.1 Introduction

In this chapter, we will conduct a comprehensive review of the state-of-the-art deep learning architectures utilized in Semantic Segmentation tasks. Where we will delve into key Deep Learning architectures, elucidating their fundamental principles and applications in Semantic Segmentation. Additionally, advanced techniques and methods aimed at enhancing segmentation performance are discussed, alongside evaluation metrics and real-world applications across various domains. This chapter serves as a foundation for understanding the advancements and complexities in Semantic Segmentation using Deep Learning.

## 2.2 Concepts of CNNs

A Convolutional Neural Network (CNN) is a specialized type of Artificial Neural Network designed to effectively process structured grid data, such as images or videos.

Based on what was mentioned on [22], CNNs leverage shared weights and local connections to exploit the 2D and 3D structures of input data, such as image signals. This approach reduces the number of parameters, simplifies training, and improves network speed. The idea is similar to how cells in the visual cortex perceive small portions of a scene rather than the entire scene, capturing local patterns akin to local filters applied across the input.

A commonly used CNN architecture, which is a Multi-Layer Perceptron (MLP), comprises Convolution Layers (ConvLay) followed by sub-sampling (Pooling) layers, with Fully Connected (FC) layers at the end. Figure 2.1 provides an example of a CNN architecture designed for image classification.

In a CNN model, the input data is structured across three dimensions: height, width, and depth (which represents the number of channels) or $n{*}m{*}r$ where $n{=}m$. For example, in an RGB image, the depth $r$ is equal to three. Each Convolutional Layer consists of multiple

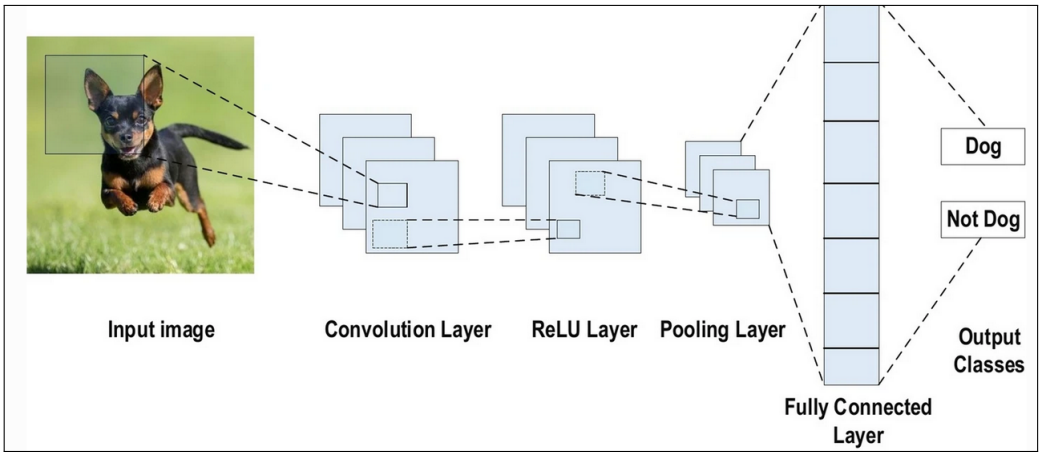



Figure 2.1: CNN model [22].

kernels $k$ (filters) with the same dimensions as the input image where they are of size $n*n*q$, but with a smaller depth where $n<m$ and $q \leq r$. These kernels are responsible for generating feature maps by convolving with the input using shared parameters (bias $b^k$ and weight $x^k$) as shown in Figure 2.2.

The Convolutional Layer conducts a mathematical operation by taking the dot product between the input and its weights, and subsequently applies a nonlinearity or activation function (Formula (2.1)).

$$h^k = f(w^k \cdot x + b^k) \tag{2.1}$$

where $h^k$ are the features maps got by for each convolution operation of size $m - n - 1$. Subsequently, the sub-sampling layers downsample each feature map By using a Pooling function (like Max or Average Pooling), you aggregate information from neighboring regions of a defined length $p$, that is, of size $p * p$. This reduces the network parameters, speeds up training, and helps address overfitting as shown in Figure 2.3.

The fully connected (FC) layers take the extracted mid- and low-level features and transform them into higher-level abstractions, resembling the operation of a traditional neural network. The last-stage layers produce classification scores using techniques like softmax. These scores represent the probabilities of different classes for a given input instance, the whole process is shown in figure 2.4.

### 2.2.1 Advantages and Limitations of CNN

**Advantages**

[23] mentioned that CNNs offer several advantages that render them potent tools for analyzing both visual and sequential data. One primary strength lies in their capacity to acquire hierarchical representations of input data, wherein higher-level features are constructed atop lower-level features. This capability enables the model to discern intricate patterns within the data and yield precise predictions.

Furthermore, CNNs possess the capability to accommodate inputs with varying sizes and aspect ratios. This adaptability stems from the Convolutional Layers ability to discern fea-

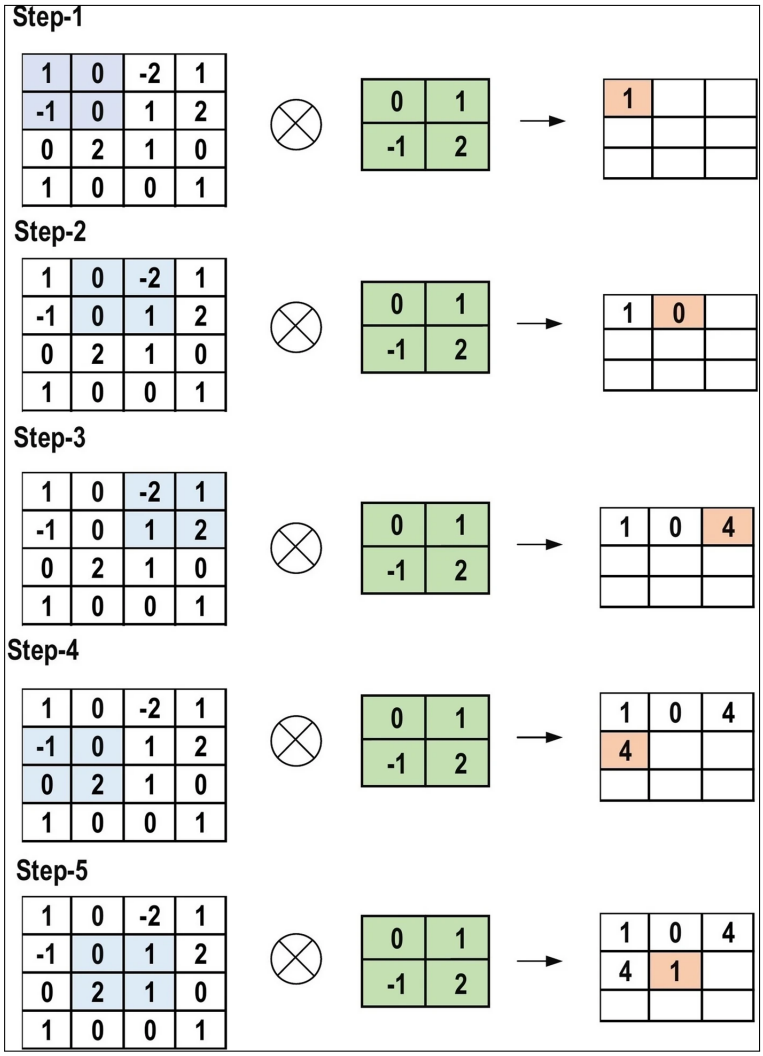



Figure 2.2: The process of Convolution [22].

tures that exhibit translation invariance, allowing recognition of patterns irrespective of their location within the input. Additionally, the integration of Max-Pooling layers aids in Downsampling the input, facilitating the handling of diverse input sizes and aspect ratios.

Another notable advantage of CNNs lies in their adeptness at generalization to novel data, signifying their ability to make accurate predictions on previously unseen data. This propensity arises from the models acquisition of task-relevant features rather than memorization of specific instances from the training data. Additionally, leveraging transfer learning can further enhance model accuracy by transferring knowledge from pre-trained models to new tasks, thereby mitigating the need for extensive training data.

**Limitations**

While CNNs offer numerous benefits, they also present certain limitations that warrant consideration during their usage as state in this work [23]. One drawback is their dependency on sizable labeled training datasets for optimal performance. Due to the vast number of parameters in CNNs, extensive data variation is necessary to prevent overfitting, a pro-

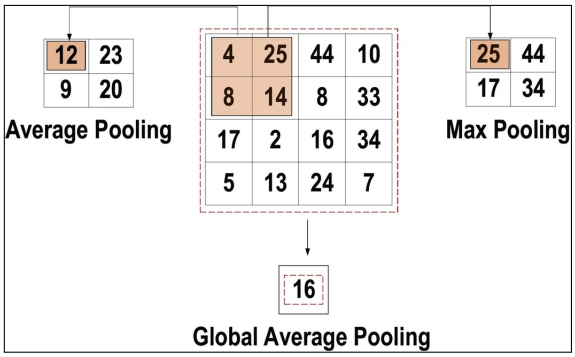



Figure 2.3: The process of Pooling [22].

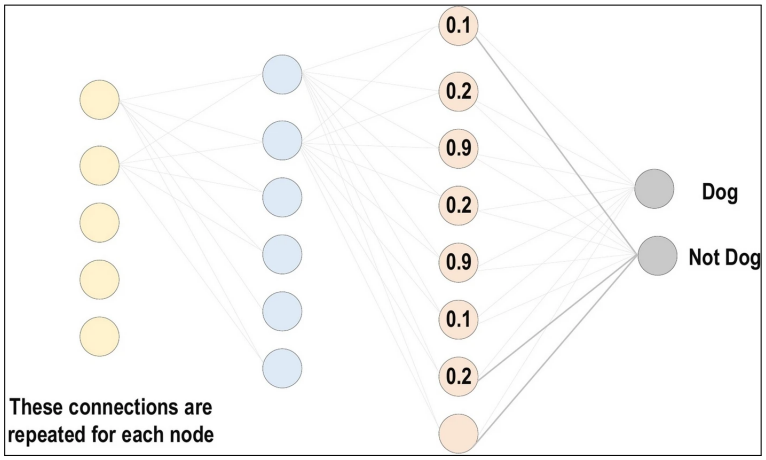



Figure 2.4: Fully connected layer [22].

cess that can be arduous and costly, particularly for tasks demanding detailed classification or segmentation.

Moreover, CNNs can exhibit sensitivity to the quality of training data, especially in the presence of noise or biases. If the data contains artifacts, the model might prioritize learning these nuances over the underlying patterns, underscoring the importance of meticulous curation and preprocessing to ensure data fidelity.

Additionally, the computational demands of training and evaluating CNNs, especially for complex architectures, can be substantial. With their reliance on ample memory and processing power, training large CNNs may pose challenges. However, advancements in hardware and software have facilitated training on single or multiple GPUs, alongside the development of techniques like model pruning and compression to mitigate computational burdens.

Lastly, while CNNs excel in various tasks, they may not always be the most suitable choice. For instance, if the input data follows a different structure, such as graphs or trees, alternative Neural Network architectures like Graph Neural Networks or Recursive Neural Networks might be preferable. Similarly, tasks necessitating reasoning over symbolic or logical representations may benefit from models like Rule-Based or Logic-Based systems. Thus, selecting the appropriate model architecture should be guided by the specific require-

ments of the task at hand.

## 2.3 The Difference between Context Information and Feature Enhancement

The following table shows a comparaison between methods often repeated in various works which Deep Learning models for Semantic Segmentation based on them, we should mention here that the methods listed below depend on labeled data, so they are supervised learners :

| Aspect | Context Information | Feature Enhancement |
|---|---|---|
| Definition | Broader understanding of scene, including spatial relationships and global context | Improving feature quality by leveraging semantic and spatial information from different layers |
| Focus | Providing complementary information beyond pixel-level appearance | Enhancing model's ability to capture both semantic and spatial details |
| Mechanisms | Context aggregation modules, multi-scale feature fusion | Skip connections, feature recalibration (e.g, attention mechanisms), adaptive feature selection |
| Usage | Incorporating global context features, spatial relationships between objects | Improving feature richness for accurate segmentation |

Table 2.1: Comparison between context information and feature enhancement techniques.

## 2.4 Common Deep Network Models for Semantic Segmentation

### 2.4.1 General Overview of Techniques Used for Semantic Segmentation

Deep learning methods, particularly CNNs, have achieved remarkable success in various high-level computer vision tasks. These tasks include image classification and object detection, where supervised approaches utilizing CNNs have demonstrated superior performance. This success has spurred interest in leveraging deep learning techniques for pixel-level labeling tasks like Semantic Segmentation.

One of the primary advantages of deep learning models is their capacity to automatically learn meaningful feature representations directly from data. Unlike traditional methods that rely on hand-crafted features requiring domain expertise and extensive fine-tuning, deep learning models can learn these representations in an end-to-end fashion. This means

that the model learns to extract relevant features for the specific task, such as pixel labeling, without the need for manual feature engineering. This capability greatly simplifies the development process and often leads to improved performance on diverse datasets. There are various methods which have managed to perform the segmentation task and then considered as state-of-the art solutions, we will not discuss all of them (Figure 2.5 visualise some of the methods will be not discussed listed in 'A Review on Deep Learning Techniques Applied to Semantic Segmentation' (- A. Garcia-Garcia & al. - 2017) ), we will select few of them which considered as relevant and used frequently in the industry and academic realms.

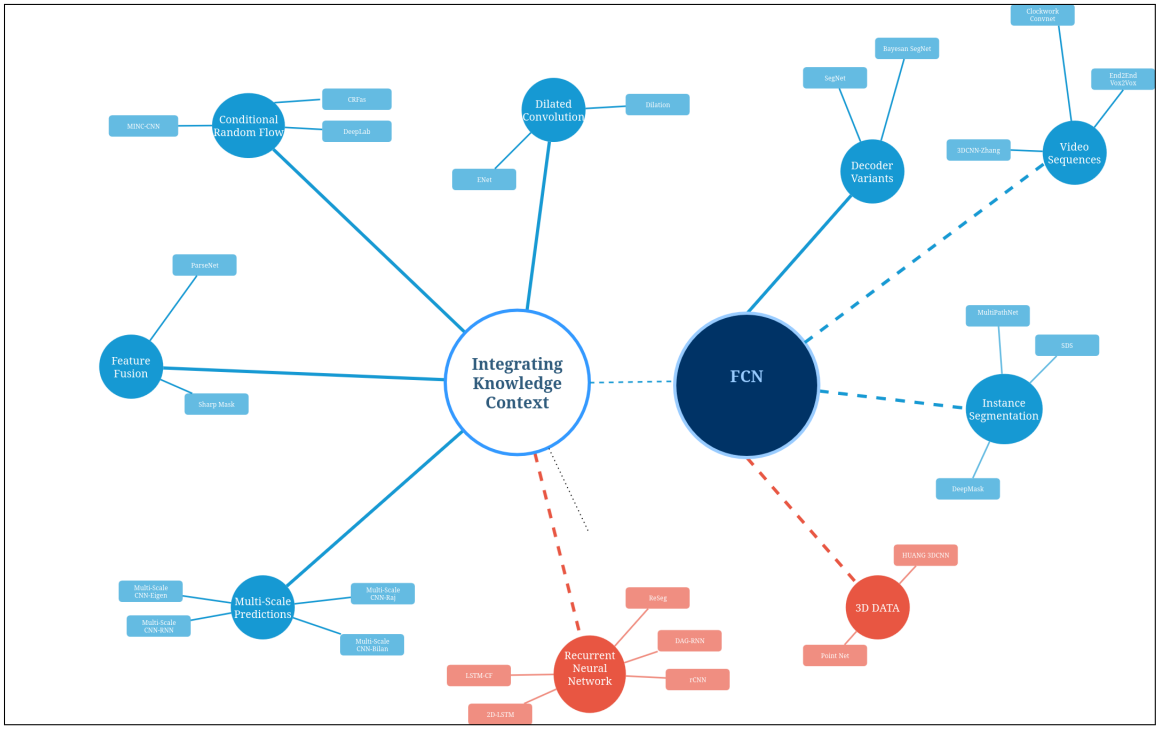



Figure 2.5: Visualization of methods didn't mention on this chapter. Image got from [24]

### 2.4.2 AlexNet

AlexNet was the pioneering Deep CNN that won the ILSVRC-2012 with a TOP-5 test accuracy of 84.6% while the closest competitor, which made use of traditional techniques instead of deep architectures, achieved a 73.8% accuracy in the same challenge. The architecture presented by Krizhevsky et al. [25] was relatively simple. It consists of five convolutional layers, max-Pooling ones, Rectified Linear Units (ReLUs) as non-linearities, three fully-connected layers, and dropout as shown on Figure 2.6.

### 2.4.3 ResNet (Residual Networks)

ResNet is a CNN architecture developed by Kaiming He et al. in 2015, initialy known as RESNET50 which introduced the concept of residual connections (see Figure 2.7) to address the problem of vanishing gradients in deep networks. The ResNet architecture consists of multiple residual blocks, each of which includes multiple ConvLays and shortcut connections that bypass the ConvLays and add the original input to the output of the block. The

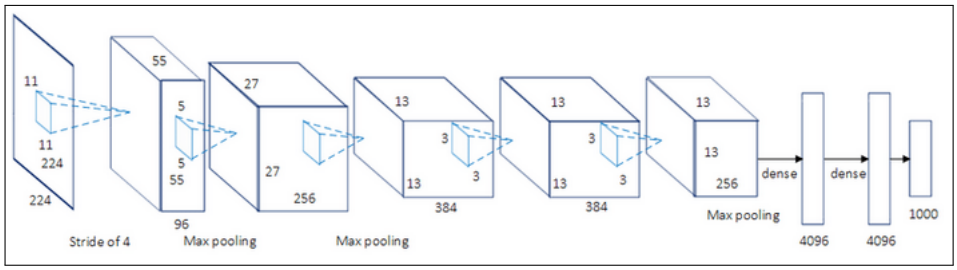



Figure 2.6: The architecture of AlexNet taken from [25].

Fully Connected Layers use a softmax activation to output a probability distribution over the possible classes. A large number of Semantic Segmentation methods make use of it as their backbone, e.g, the DeepLab family and PSPNet, just to name but a few [23].

Resnets (see Figure 2.8) relie on Context information methodology. In the context of Semantic Segmentation, it refers to the broader understanding of a scene beyond individual pixel-level appearance. It encompasses spatial relationships, object interactions, and global scene context, providing valuable cues for segmenting objects accurately. By analyzing context, Semantic Segmentation models gain semantic awareness, allowing them to make more informed decisions about the class labels of pixels based on their surrounding context. Integrating context information enhances the model's ability to understand the semantic meaning of pixels within the context of the entire scene, leading to more robust and accurate segmentation results.

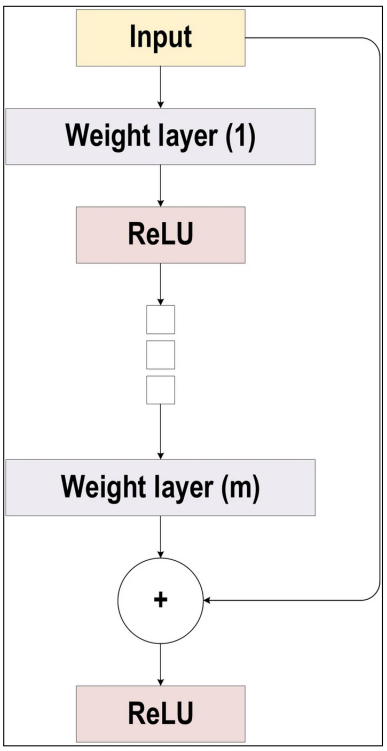



Figure 2.7: Residual connection [22].

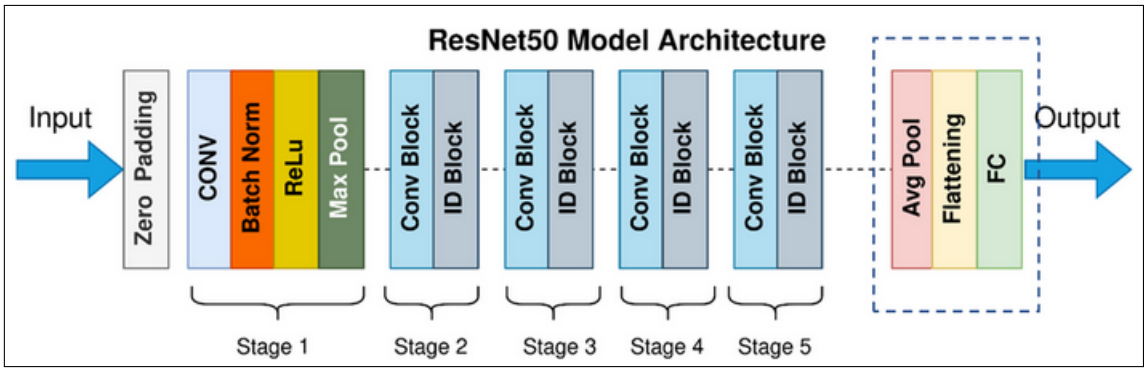



Figure 2.8: Resnet50 backbone architecture [26].

### 2.4.4 VGG Network

VGG (see Figure 2.9) is a CNN architecture developed by Karen Simonyan and Andrew Zisserman in 2014, which achieved state-of-the-art performance on the ImageNet dataset. The VGG architecture consists of multiple layers of small 3 × 3 Convolutional Filters, followed by Max PoolLays. The Fully Connected Layers use ReLU activation and dropout regularization to prevent overfitting. There are different versions for VGG networks according to the layer number, such as VGG-13, VGG-16, and VGG-19. They are also like ResNets, relie on Context information methodology [23].

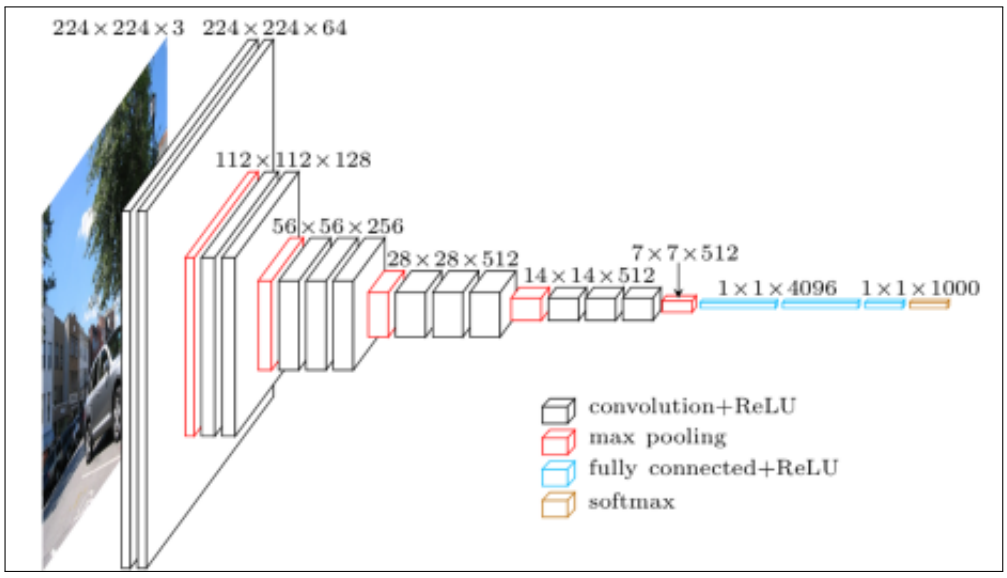



Figure 2.9: VGG backbone architecture.The image is taken from [27]

### 2.4.5 Fully Convolutional Network (FCN)

Fully Convolutional Networks (FCNs) [28] played a crucial role in the transition from classification to segmentation, enabling end-to-end pixel-wise classification while preserving spatial information through techniques like $1 * 1$ convolutions [1]. This evolution from classi-

[1] A 1 x 1 Convolution is a convolution with some special properties in that it can be used for dimensionality reduction, efficient low dimensional embeddings, and applying non-linearity after convolutions. It maps an

fiers to Semantic Segmentation reflects a deeper understanding and more nuanced analysis of visual content, paving the way for applications ranging from autonomous driving to medical imaging.

FCN (see Figure 2.10) stands out as one of the preferred models for Semantic Segmentation tasks. A distinguishing feature of FCNs compared to traditional CNNs is the absence of a Fully Connected Layer as the final layer. Instead, FCNs modify the number of output channels through convolutional operations, facilitating the integration of information. This design choice offers the advantage of enhanced flexibility in input size, as FCNs are not constrained by the limitations imposed by Fully Connected Layers.

A significant challenge inherent in this architecture is the uniform use of padding across all Convolution Layers. This choice aims to maintain the output image's dimensions identical to the input image. However, this insistence on preserving full-resolution incurs a considerable computational cost.

Reducing the number of layers could be a tempting solution, but it would severely compromise performance.

Unlike classification tasks, where only the presence of a single object matters, Semantic Segmentation demands preserving spatial information. Downsampling images through Pooling, a common practice in classification, is unsuitable for segmentation tasks.

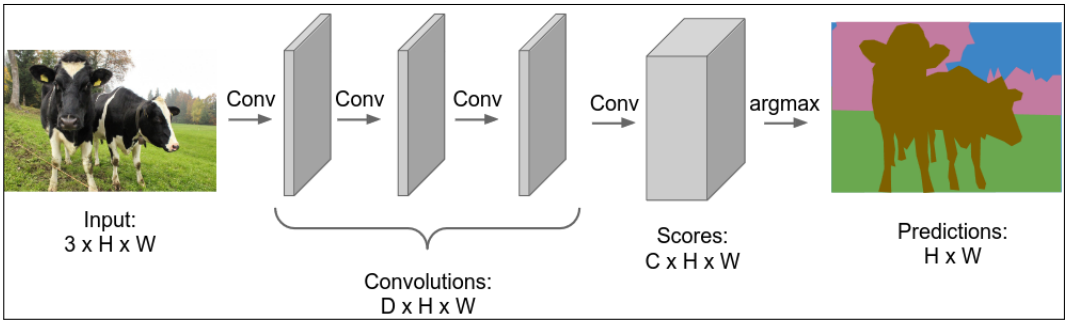



Figure 2.10: The architecture of FCN taken from [29].

There is also a variant of FCN that tackled the problem : Fully Convolutional Network – with Downsampling and Upsampling inside the network (see Figure 2.11). In the initial stages of the model, the authors employ a technique where they gradually decrease the clarity of the image while creating detailed feature representations. With each convolutional step, they enhance their comprehension of the image's intricacies. While this method is excellent for distinguishing between different classes effectively, it sacrifices precise spatial information. To address this issue, they incorporate an Upsampling step following Downsampling (which includes Pooling and Strided Convolutions). This Upsampling phase amalgamates multiple lower-resolution images to generate a high-resolution segmentation map.

input pixel with all its channels to an output pixel which can be squeezed to a desired output depth. It can be viewed as an MLP looking at a particular pixel location.

#### 2.4.5.1 Skip-Connections

In Fully Convolutional Networks (FCNs), Downsampling within the network significantly reduces the input resolution, posing a challenge during the subsequent Upsampling phase to accurately reconstruct finer details, even with advanced techniques like Transpose Convolution. Consequently, the output often appears coarse.

To address this limitation, the authors of [28] incorporate 'skip connections' during the Upsampling stage from earlier layers proves beneficial. These connections allow the flow of additional information from earlier layers to later ones by summing the corresponding feature maps. By blending fine-grained details with coarser representations, this approach enhances the network's ability to delineate precise segmentation boundaries. Consequently, the integration of both fine and coarse layers facilitates local predictions while preserving nearly accurate global spatial structure (see Figure 2.13) .

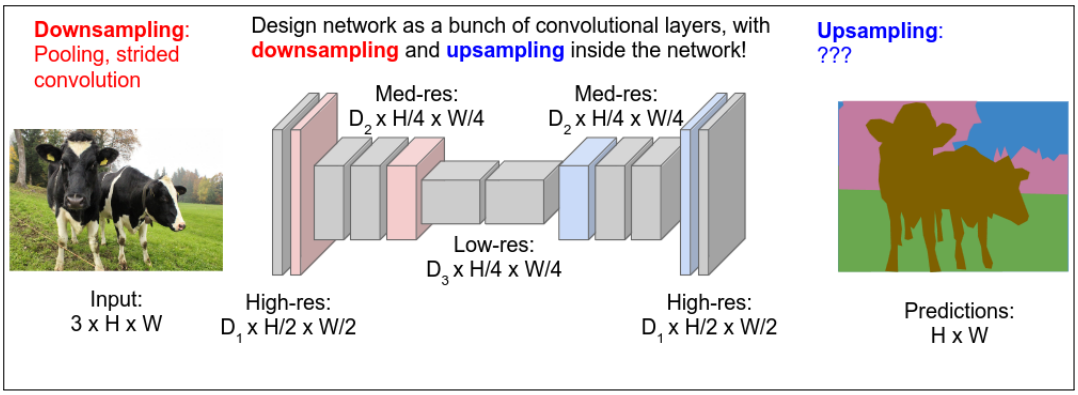



Figure 2.11: The architecture of FCN with Upsampling and Downsampling.The image taken from [29].

#### 2.4.5.2 Upsampling techniques

Upsampling techniques are crucial in tasks such as image segmentation, where the resolution of feature maps needs to be increased to match the original input image's dimensions. Here are some commonly used Upsampling methods (see Figures 2.12) :

- **Max-UnPooling :** From [30], Max UnPooling improves upon traditional Pooling methods (Nearest Neighbor for instance) by capitalizing on the symmetry inherent in Downsampling-Upsampling Networks. In these networks, each Downsampling layer corresponds to an Upsampling layer. Instead of using fixed cells like traditional methods, max unPooling selects the maximum value from each region in the Downsampling layer and fills the corresponding cell in the Upsampling layer with that value. This approach efficiently preserves crucial features, contributing to the network's overall performance.
- **Deconvolution :** From [30], UnPooling layers make the feature maps bigger but not very detailed, while Deconvolutional Layers make them both bigger and more detailed. Unlike regular layers, which combine lots of information into one output, Deconvolutional Layers spread out one piece of information into many outputs. This

makes the maps denser, with edges that match the size of the previous unPooling layer. The filters in Deconvolutional Layers help rebuild the shapes of objects in the image. They start with the big shapes and add in smaller, more specific details as they go along. This helps the network understand different shapes better, which is really important for tasks like finding objects in pictures.

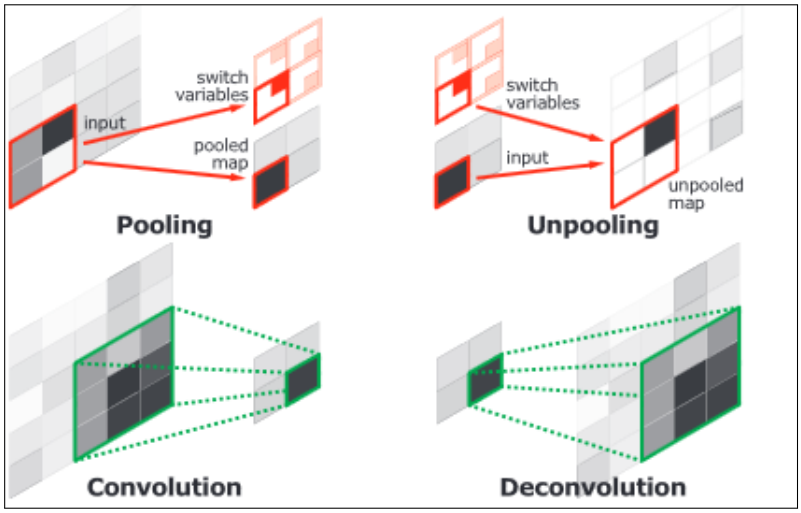



Figure 2.12: Illustration of Deconvolution and UnPooling operations from [30].

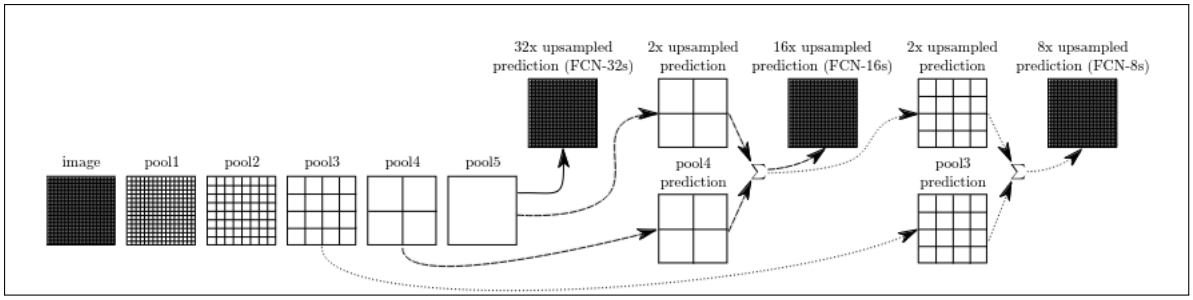



Figure 2.13: A netowrk with skip connection in place from [28].

## 2.4.6 U-Net Model

U-net is a Neural Network architecture primarily designed for image segmentation. Most convolutional networks are designed for whole-image classification, lacking pixel-level context information necessary for medical image analysis. U-net addresses this limitation and significantly improves medical image segmentation performance. It builds upon the concept of fully convolutional networks and was introduced by Ronneberger et al. in 2015. The U-net implementation surpassed previous state-of-the-art methods, winning the ISBI cell tracking challenge in 2015 and achieving superior performance in the ISBI 2012 challenge, U-net relies on what is known feature-enhancement-based methods [31].

### 2.4.6.1 Architecture Overview

The basic U-net network (see Figure 2.14) consists of two main components: the contracting path and the expansive path. In the contracting path, a typical CNN architecture is used

like a Resnet, where each block includes two consecutive 3x3 convolutions followed by a ReLU activation and a max-Pooling layer. This process is repeated several times to extract hierarchical features from the input image, but fully-connected layers that make up the classification head are not used, rather it followed up directly by the expansive path. The deeper is the contracting path, the resolution of the feature maps are more smaller but the channel dimension get more wider, that's the case in the existant variants of U-net.

The expansive path is where the U-Net architecture stands out. Here, the feature map undergoes upsampling using 2x2 up-convolutions. Concurrently, the corresponding feature map from the contracting path is cropped and merged with the upsampled feature map through concatenation. This merging enables the network to integrate high-resolution features from the expansive path with contextual information captured by the contracting path. Following concatenation, two successive 3x3 convolutions with ReLU activation refine the features.

In the final stage of U-net, a 1x1 convolution is added to reduce the feature map to the required number of channels, which produces the segmented image. The cropping operation during the expansive path is crucial because it discards pixel features from the edges, which contain minimal contextual information. This U-net architecture takes the form of a "U," which facilitates the propagation of contextual information throughout the network, enabling the segmentation of objects in a given area using context from a larger overlapping region.

#### 2.4.6.2 Real World Applications

U-net offers several notable advantages, especially in the field of segmentation where labeled data is often limited. One key advantage is its ability to generate detailed segmentation maps even with a small number of training samples. This is accomplished through techniques like random elastic deformation, which enables the network to learn variations and generalize well without relying on additional labeled data.

Additionally, U-net addresses the challenge of segmenting touching objects of the same class. It achieves this by utilizing a weighted loss function that penalizes the model for ineffective separation, ensuring accurate segmentation results.

Another advantage of U-net is its faster training compared to many other segmentation models. This is attributed to its context-based learning approach, which allows the network to leverage contextual information effectively and learn efficiently during the training process. While initially introduced for biomedical image segmentation, the versatility of U-Net extends beyond its original application. Over time, U-Net has been employed in a diverse range of tasks across various domains. For exemple, satellite and aerial imagery analysis, autonomous vehicles and others applications [32, 33, 34].

### 2.4.7 Linknet Model

LinkNet [35] is a Neural Network architecture designed for Semantic Segmentation tasks. It features an encoder-decoder structure, with ResNet18 used as the encoder for feature extraction. Linking each encoder layer to its corresponding decoder output facilitates efficient

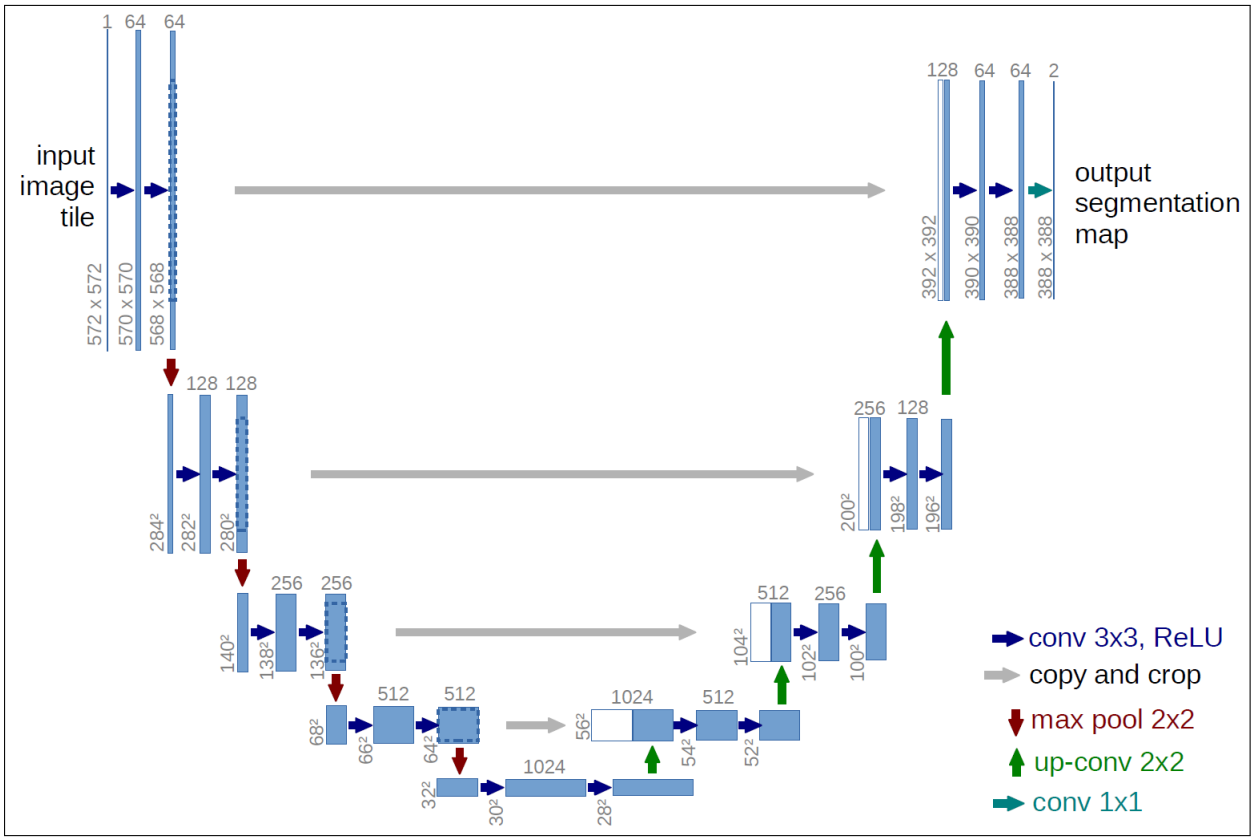



Figure 2.14: Unet model [31].

spatial information recovery, resulting in real-time operation and competitive performance.

#### 2.4.7.1 Architecture Overview

The architecture of LinkNet , depicted in Figure 2.15, consists of an encoder-decoder structure. The encoder, situated on the left side of the network, commences with an initial block that convolves the input image with a 7x7 kernel and stride of 2, followed by spatial max-Pooling with a 3x3 area and stride of 2. Subsequently, residual blocks, denoted as encoder-block (i), are employed for feature extraction. Similarly, decoder-blocks are utilized in the decoder section of the network. Each encoder layer's input is also directly linked to its corresponding decoder output, facilitating the recovery of spatial information lost during Downsampling operations. This innovative linking mechanism, in conjunction with the sharing of knowledge between encoder and decoder layers, enables the decoder to operate efficiently with fewer parameters. Unlike traditional segmentation architectures, LinkNet employs trainable parameters to link encoder and decoder layers, enhancing spatial information recovery and overall network efficiency for real-time operation.

### 2.4.8 FPNet Model

FPNet, or Feature Pyramid Network [36] is a Neural Network architecture designed to extract features at multiple scales from images, facilitating tasks like object detection and

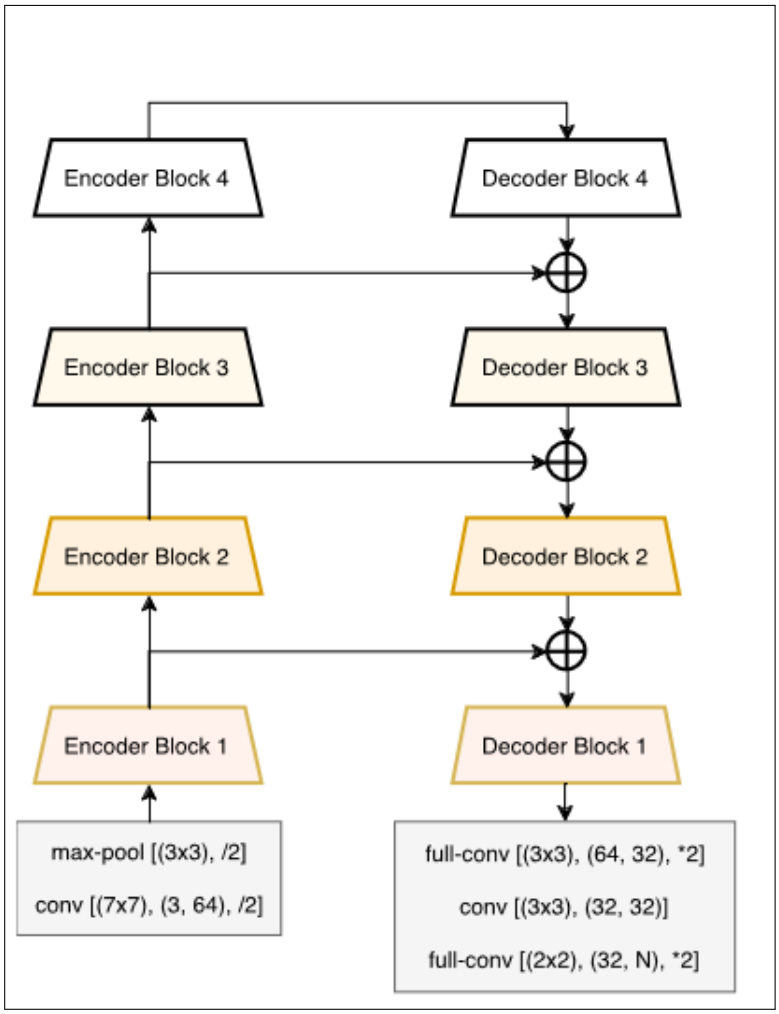



Figure 2.15: Linknet architecture [35].

segmentation by integrating high-level semantic information across different resolutions.

#### 2.4.8.1 Architecture Overview

The aim of the authors is to utilize the hierarchical feature structure of ConvNets(another naming refered to CNNs), constructing a Feature Pyramid Network (FPN) that integrates high-level semantics across multiple scales. FPN is adaptable for various tasks, with a focus here on sliding window-based proposal generators (Region Proposal Network, RPN) and detectors that operate on regions (Fast R-CNN), while also extending to instance segmentation proposals. The method operates on single-scale images of arbitrary sizes, producing proportionally sized feature maps at multiple levels in a fully convolutional manner. This architecture includes a bottom-up pathway, a top-down pathway, and lateral connections. The bottom-up pathway involves the feedforward process of the backbone ConvNet, creating hierarchical features at various scales. The top-down pathway upscales coarse features from higher levels of the pyramid and integrates them with bottom-up features through lateral connections. Consistency in feature dimensions across all pyramid levels is main-

tained by shared classifiers or regressors. The authors maintain simplicity in design, with empirical findings demonstrating robustness to various architectural choices. While more sophisticated blocks yield marginal improvements, thier focus remains on simplicity and efficiency (see Figure 2.16).

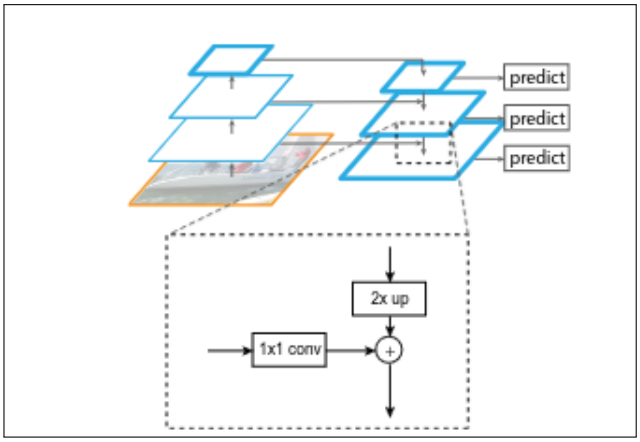



Figure 2.16: FPNet architecture [36].

## 2.4.9 PSPNet Model

PSPNet, or Pyramid Scene Parsing Network [37] is a Semantic Segmentation model that utilises a pyramid parsing module (or Pyramid Pooling Module) that utilizes global context information through region-based context aggregation, combining local and global cues to enhance the reliability of the final prediction.

### 2.4.9.1 Pyramid Pooling Module (PPM) :

At the heart of PSPNet lies the Pyramid Pooling Module (PPM), a pivotal component designed to consolidate contextual information across various scales. The PPM partitions the input feature map into distinct regions of interest (ROIs) and employs Pooling operations to gather features from each ROI at diverse scales. This multi-scale aggregation enables the network to encompass both local and global context, significantly enhancing its capability to discern intricate spatial relationships within the input scene. As a result, PSPNet achieves heightened segmentation accuracy by effectively integrating information from different scales, thereby facilitating more comprehensive scene understanding and precise pixel-wise predictions.

### 2.4.9.2 Architecture Overview

PSPNet, illustrated in Figure 2.17, is founded on the Pyramid Pooling Module (PPM), a core element of its architecture. The workflow begins with an input image (Figure 2.17a), processed by a CNN model which is a pretrained ResNet model leveraging the dilated network strategy[2] to yield a feature map (Figure 2.17b), downscaled to one-eighth the size of the

[2]Dilated Convolutions are a type of convolution that "inflate" the kernel by inserting holes between the kernel elements. An additional parameter $l$ (dilation rate) indicates how much the kernel is widened. There

input image. Atop this map, the pyramid Pooling module (Figure 2.17c) strategically integrates contextual information. Utilizing a four-level pyramid, Pooling kernels span various portions of the image, amalgamating them into a global prior. This prior is then concatenated with the original feature map in the latter part of Figure 2.16c, followed by a convolution layer generating the final prediction map (Figure 2.17d). PSPNet furnishes a potent global contextual prior for pixel-level scene parsing, with the pyramid Pooling module adept at capturing hierarchical levels of information, surpassing the representational capabilities of global Pooling. Notably, PSPNet maintains computational efficiency against FCN (as it is mentioned in the original article), and in the realm of end-to-end learning, optimization of the global pyramid Pooling module and the local FCN feature occurs concurrently.

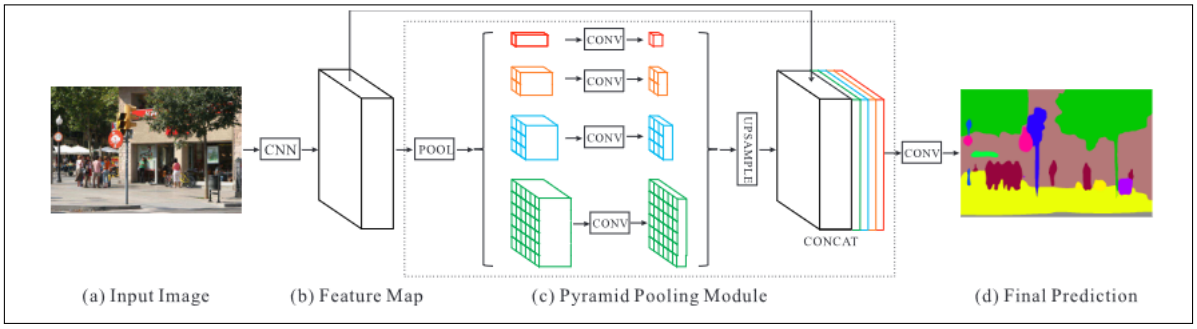



Figure 2.17: PSPNet architecture [37].

### 2.4.10 Transformers & Attention Models

The Transformer model, introduced by Vaswani et al. in 2017 [38], has garnered significant attention and utilization across a range of domains, including natural language processing (NLP) [39, 40], computer vision (CV) [41]. Initially conceptualized as a sequence-to-sequence model by Sutskever et al. in 2014 for machine translation, the Transformer has since found extensive application and adoption in various fields. Transformers have become the preferred architecture in NLP and computer vision, surpassing Recurrent Neural Network (RNNs) [42] as the typical model to process sequential data due to their ability to address the vanishing gradient problem. Unlike RNNs, Transformers have parallel processing capabilities and no recurrent connections, allowing gradients to flow directly through the network during backpropagation. This results in faster training times and the ability to train larger models. Additionally, the self-attention mechanism in Transformers enables them to focus on different parts of the input sequence, regardless of their distance apart. This ability to handle complex patterns and long sequences is crucial for tasks that invloves processing sequential data as texts, images and videos.

#### 2.4.10.1 Architecture of Transformers

The vanilla Transformer is a sequence-to-sequence model that includes both an encoder and a decoder. Each component, the encoder and decoder, is constructed using multiple

are usually $l - 1$ spaces inserted between kernel elements.

Note that concept has existed in past literature under different names, for instance the algorithme a trous, an algorithm for wavelet decomposition (Holschneider et al., 1987; Shensa, 1992).

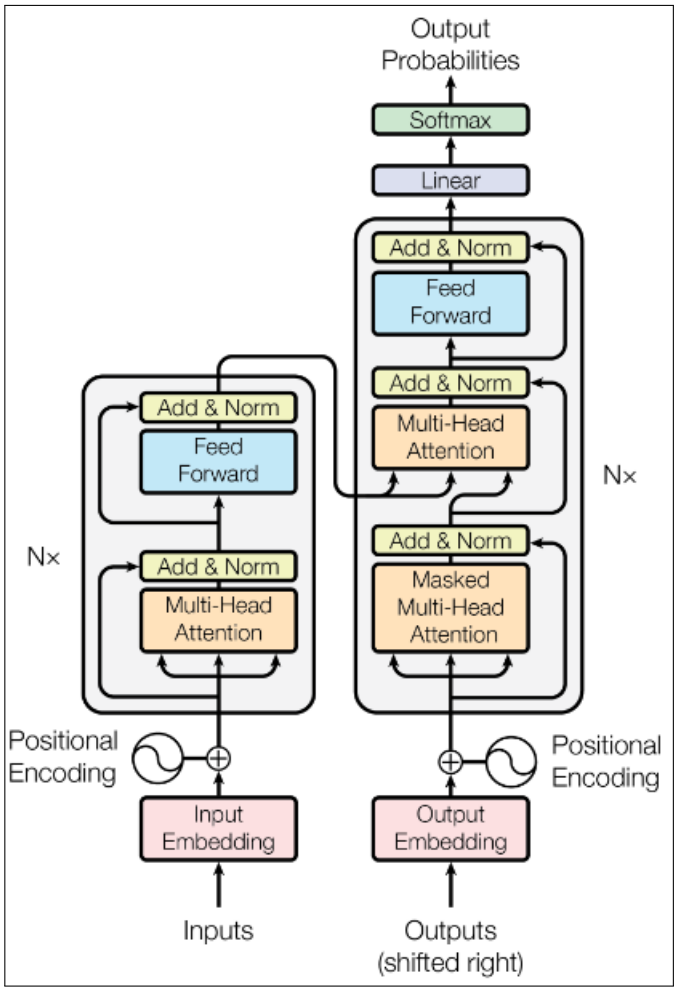



Figure 2.18: The Transformer model [38]

identical stacked blocks (see Figure 2.18).

In the encoder, each block primarily consists of a multi-head self-attention module and a position-wise feed-forward network (FFN). To create a deeper model, residual connections are utilized around each module, followed by Layer Normalization. These blocks enable the encoder to process effectively the input sequence. In the decoder, cross-attention modules are introduced between the multi-head self-attention modules and the position-wise FFN. This integration enables the decoder to utilize information from the encoder's output as it processes inputs. Additionally, the self-attention mechanisms in the decoder are adjusted to restrict each position from attending to subsequent positions, thereby ensuring correct sequential generation.

**Attention modules**

The Transformer architecture [43] incorporates attention mechanism using the Query-Key-Value (QKV) model. It utilizes scaled dot-product attention, where the dot-products of queries and keys are normalized by dividing them by the square root of the dimension.

This approach helps mitigate the issue of vanishing gradients (Formula (2.2)).

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{D_k}}\right) \cdot V \tag{2.2}$$

where the Queries $Q$ are matrices $Q \in \mathbb{R}^{N \times D_k}$, the Keys $K$ are matrices $K \in \mathbb{R}^{M \times D_k}$ and the Value $V$ are matrices $V \in \mathbb{R}^{M \times D_v}$. $N$ and $M$ denote the lengths of queries and keys (or values); $D_k$ and $D_v$ denote the dimensions of keys (or queries) and value, $\sqrt{D_k}$ helps to alleviate gradient vanishing problem of the softmax function.

In addition to single attention function, Transformer utilizes multi-head attention $D_m$. It transforms the original queries, keys, and values into distinct dimensions (respecitevly $D_k$, $D_q$ and $D_v$) and calculates attention for each transformed projection, and concatenates the outputs for further projection , back to $D_m$. The whole process is given by Formula (2.3).

$$\text{MultiHeadAttention}(Q, K, V) = \text{Concat}\left(head_1, ..., head_H\right) \cdot W^O \tag{2.3}$$

$$\text{where } head_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V)\right) \tag{2.4}$$

There are three types of attention in Transformer:

1. **Self-attention :** In the Encoder.
2. **Masked Self-attention :** In the Transformer decoder , it's known under autoregressive or causal attention, a type of self-attention, imposes a restriction by applying a mask function to the unnormalized attention matrix $\hat{A}(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{D_k}}\right)$. In this approach, certain positions($i < j$) are masked out by setting their corresponding values to $-\infty$, so the model cannot "cheat" by looking ahead at future information.
3. **Cross-attention :** In the attention mechanism, queries are projected based on the outputs of the preceding layer in the decoder. Meanwhile, keys and values are projected using the outputs from the encoder.

**Position-wise FFN**

The position-wise FFN is simply a fully connected feed-forward module that operates separately and identically on each position (Formula (2.5).

$$\text{FFN}(H') = \text{Relu}\left(H'W + b\right) + W^2 \tag{2.5}$$

where : $W \in \mathbb{R}^{D_m \times D_f}$ and $W \in \mathbb{R}^{D_f \times D_m}$ are the weights, whereas $b \in \mathbb{R}^{D_f}$ and $b \in \mathbb{R}^{D_m}$ are the bias. Together form the parameters of FFN module.

**Residual connection and normalization**

Each Transformer encoder module uses residual connection [44], followed by Layer Normalization [45], the process is expressed by (2.6) and (2.7).

$$\text{H'} = \text{LayerNorm}\left(\text{SelfAttention}(x) + x\right) \tag{2.6}$$

$$\text{H} = \text{LayerNorm}\left(\text{FFN}(H') + H'\right) \tag{2.7}$$

**Position encodings**
It helps to annotated the position and the ordering of each token in each sequence at the input of the decoder, it's like a contextuel embedding.

#### 2.4.10.2 Use cases of Transformer

The transformer can be used as :

1. **Encoder_Decoder** : the whole architecture is used in Sequence-To-Sequence modeling like BART [46] and T5 [40].
2. **Encoder only** : where the outcome results of the encoder is only needed , and it's often used for Sequence Understanding and for input sequence representation for instance like Bert [47] and RoBerta [48].
3. **Decoder only**: This is generally used for sequence generation like GPT3 [49].

#### 2.4.10.3 Variants of Transformer

Since Transformers model has gained a wide sucess in natural language processing (NLP) tasks, researches were motivited to invistigate a way further research in other domains, including computer vision. However, there was a challenge to represent an image sequentually and perserve the spatial domaines between pixels, which makes a quadratic cost for the operation [41].

Several years later, researchers found a potential solution by dividing images into patches, converting each patch into a vector, applying a linear transformation, and treating these vectors as embedded words. This approach allowed them to leverage the power of Transformers for image recognition tasks. The original paper titled "An image is worth 16x16 words: Transformers for image recognition at scale" introduced the architecture known as Vision Transformer (ViT) (see Figure 2.19) [50].

The ViT architecture consists of multiple Transformer encoder-layers stacked together. Each encoder-layer processes the input image patches, allowing the model to capture spatial dependencies and learn meaningful representations for image recognition. By adapting the Transformer framework to images, researchers were able to achieve impressive results in various computer vision tasks, such as Image Classification, Object Detection, and Image Generation and Semantic Segmentation recently since the transformers architecture, maintain the spatial information throughout the network as CNN's do [51].

Another model was inspired from the ViT : It's the MT-U-net (Mixed transformer U-net) [52].

Mixed Transformer U-Net (MTU-Net) is a deep learning model originated for medical image segmentation that combines the strengths of CNNs and transformer networks. It is based on the popular U-Net architecture, but incorporates transformer layers to enhance the model's ability to capture long-range dependencies and contextual information.

The proposed Mixed Transformer U-Net (MT-UNet) (see Figure 2.20) consists of an encoder-decoder structure with skip connections. The encoder is composed of a series of convolu-

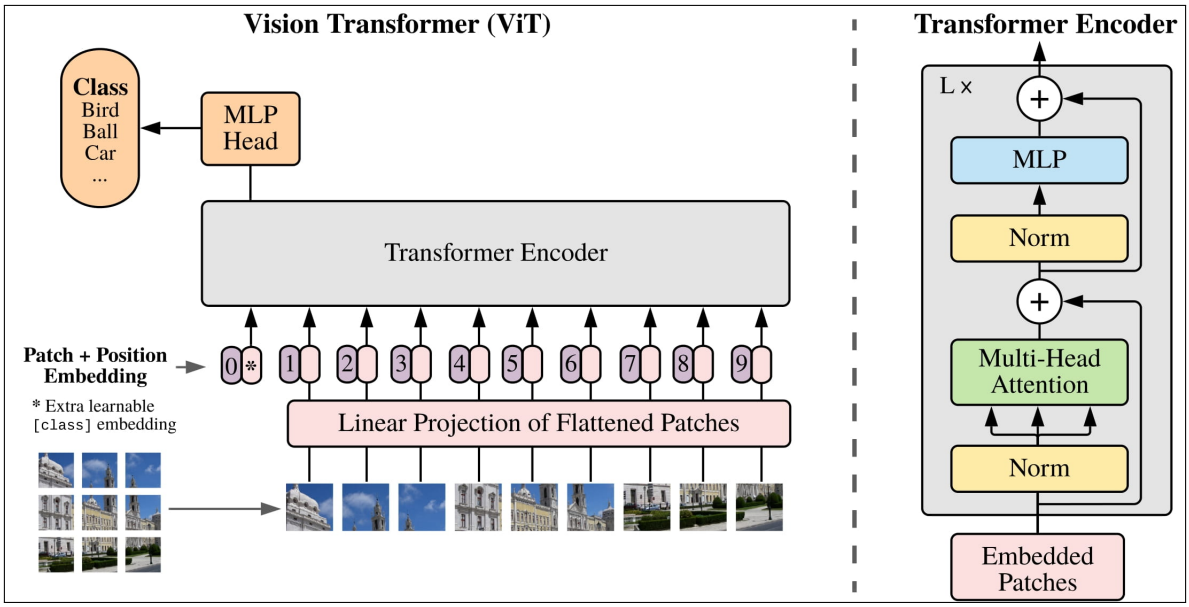



Figure 2.19: The Vit model [50]

tional layers followed by Transformer modules. The decoder is composed of a series of transposed convolutional layers followed by Transformer modules. The skip connections connect the corresponding layers in the encoder and decoder, allowing the model to learn features at different scales.

The Transformer modules in MT-UNet are based on the Mixed Transformer Module (MTM). MTM consists of two main components: Local-Global Gaussian-Weighted Self-Attention (LGG-SA) and External Attention (EA). LGG-SA is designed to capture local and global dependencies in the input data. EA is designed to capture dependencies between different samples in the dataset.

**LGG-SA**: Captures local and global dependencies in the data by dividing the data into local windows, applying self-attention to each window, and then aggregating the results.

**EA**: Captures dependencies between different samples in the dataset by creating a set of memory vectors, computing attention weights for each sample, and then aggregating the features from the different samples.

### 2.4.11 Generative Models

Generative models, such as VaEs, GANs, Flows, and Diffusions, learn to approximate the data distribution by iteratively adjusting their internal parameters. They generate new data samples by transforming random noise, essentially learning to map this noise to meaningful data representations. In essence, these models capture the underlying structure of the data distribution and produce realistic samples by sampling from this learned distribution.

we will delve into VaE's and Flows for the scope of the study, but we will give here a brief definitions of what are GANs and Diffusions.

**GANs** or Generative Adversarial Networks, introduced by Goodfellow et al. in 2014, oper-

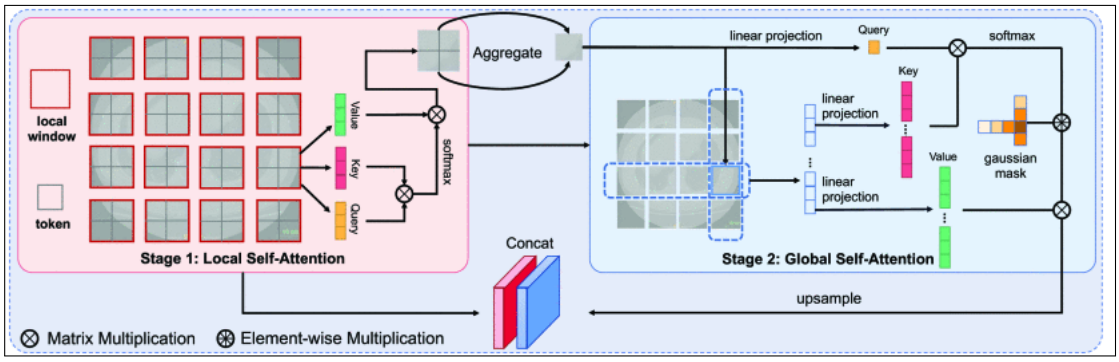



Figure 2.20: Detailed structure of the proposed Local-Global Gaussian-Weighted Self-Attention [52].

ate as a two-player game between a generator and a discriminator. The generator creates synthetic data samples, while the discriminator's task is to differentiate between real and generated samples. Through adversarial training, the generator improves by fooling the discriminator, while the discriminator improves at distinguishing real from fake data. As training progresses, the generator becomes adept at generating realistic samples, reaching a point where the discriminator can no longer reliably distinguish between real and synthetic data [53].

**Diffusions models** inspired by non-equilibrium thermodynamics and introduced by Ho, Jain, and Abbeel (2020), learn the latent structure of data by simulating the gradual diffusion of data points in space. These models utilize diffusion probability models to synthesize high-quality images by capturing the intricate relationships between data points over time. By effectively modeling the diffusion process, these models excel at generating realistic and diverse samples from complex datasets [53].

#### 2.4.11.1 Convolutional AutoEncoder (CAE)

[54] mentioned that Convolutional Autoencoders (CAEs) (see Figure 2.21) leverage convolution and Pooling operations from CNNs to preserve the two-dimensional spatial structure of images. Unlike traditional autoencoders, which process images as one-dimensional vectors, CAEs encode and decode input data using convolution, extracting local features and reducing dimensionality through deconvolution. This architecture combines the benefits of CNNs and autoencoders, utilizing shared weights and a loss function akin to regularization autoencoders, resulting in improved feature extraction and reconstruction of image data. The loss function is given in Formula (2.8) :

$$J_{\text{CAE}}(\theta) = J(X, X_d) + \lambda \|W\|_2^2 \tag{2.8}$$

#### 2.4.11.2 Variational AutoEncoder

A variational autoencoder (VaE) [56] is a Neural Network designed to learn a condensed representation of input data by encoding it into a lower-dimensional latent space and subsequently reconstructing the original data from thiscompressed representation. Unlike tra-

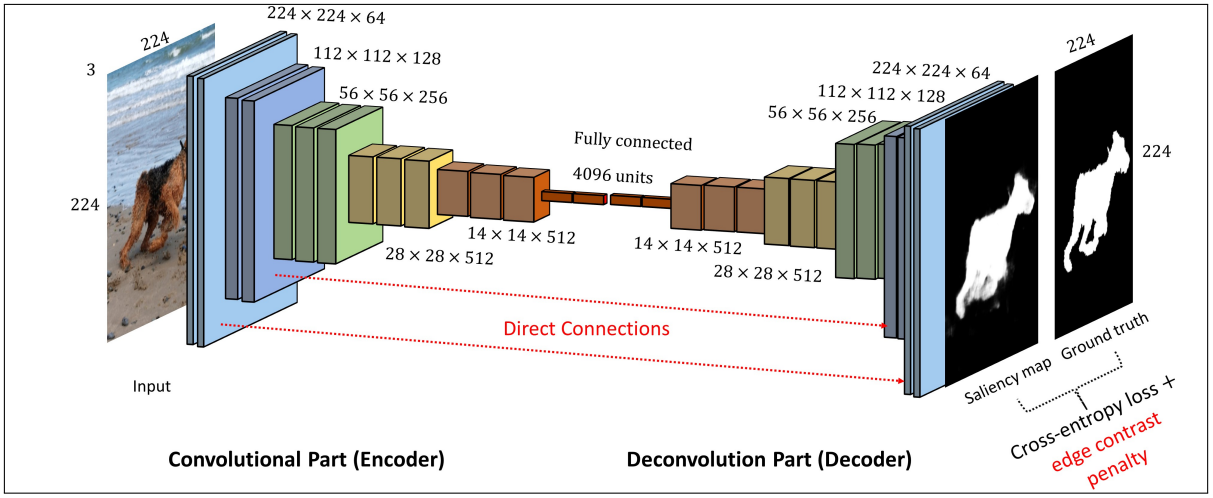



Figure 2.21: Convolutional AutoEncoder architecture [55].

ditional autoencoders, VaEs employ a probabilistic approach during the encoding and decoding process.

This probabilistic framework allows VaEs to capture the inherent structure and variability in the data, enabling them to generate new data samples from the learned latent space. VaEs find utility in diverse tasks, including anomaly detection, data compression, as well as image and text generation. The workby Kingma and Welling in 2013 introduced the concept of VaEs and their probabilistic nature

**VaE architecture**

Autoencoder Neural Networks consist of an encoder and a decoder. The encoder learns to transform the input sequence into a latent space, and the decoder reconstructs the original input sequence. Vanilla autoencoders typically lack regularity in the latent space, which complicates interpolation for missing data points (see Figure 2.22).

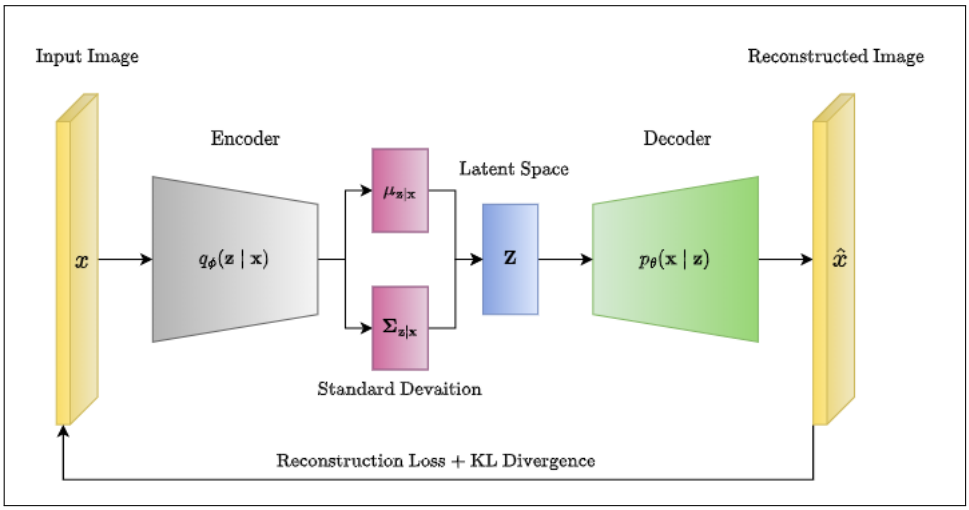



Figure 2.22: VaE model [56].

Variational autoencoders (VaEs) address this by introducing KL diver- gence regularization. The VaE encoder (E) maps data to vectors representing the mean and standard deviation of a Gaussian distribution. Minimizing the Lprior loss encourages the encoder to compress the input into a Gaussian distribution. This regularization improves reconstruction in the decoder, which samples from a continuous distribution. The decoder loss is computed using the distance between the reconstructed sequence $(\hat{x})$ and the original input $(x)$. Both Lprior (Formula (2.9)) and Lreconstruction (Formula (2.10)) losses are backpropagated to train VaE parameters. This regularization and reconstruction loss allow VaEs to learn a structured latent space, facilitating interpolation and generating new data samples.

$$L_{\text{prior}} = \text{DKL}(E(x)||N(0,1)) \tag{2.9}$$

$$L_{\text{reconstruction}} = L_{\text{prior}} + |\hat{x} - x|^2 \tag{2.10}$$

#### 2.4.11.3 Conditional Normalizing Flow (CNF)

CNFs work by conditioning the entire generative process on input data, known as conditioning data. This allows them to learn the conditional distribution of the data given the conditioning data.

Mathematically, a CNF model can be represented as follows: $p(x|y) = p(f(x|y)) * |det(df/dx)|$ where :

- $x$ and $y$ are two random variables
- $p(x)$ and $p(y)$ are their respective probability density functions. $df/dx$ is the Jacobian matrix of the transformation from $x$ to $y$

This formula shows that the probability density function of $y$ can be obtained by transforming the probability density function of $x$ and multiplying by the absolute value of the determinant of the Jacobian matrix.

CNFs use this formula to transform a simple base distribution $p(z)$ into a more complex distribution $p(x|y)$ by applying a sequence of invertible transformations f. The Jacobian matrix of each transformation is computed efficiently, and the product of all the Jacobians gives the determinant of the overall transformation f.

**Mathematical Assumptions of Conditional Normalizing Flow Models**
Conditional Normalizing Flow (CNF) models make the following mathematical assumptions [57]:

- **Invertibility :** The sequence of transformations f used to define the CNF must be invertible. This means that for every transformation $f_i$ in the sequence, there exists an inverse transformation $f_i^{-1}$ such that $f_i^{-1}(f_i(x)) = x$ and $f_i(f_i^{-1}(y)) = y$.
- **Efficient Jacobian computation :** The Jacobian matrix of each transformation $f_i$ must be efficiently tractable [58]. This is necessary for calculating the determinant of the overall transformation $f$, which is used to compute the probability density function of the data $x$ given the conditioning data $y$.

- **Base distribution :** The base distribution $p(z)$ must be a simple distribution for which the probability density function can be easily computed. Common choices for the base distribution include the standard normal distribution.

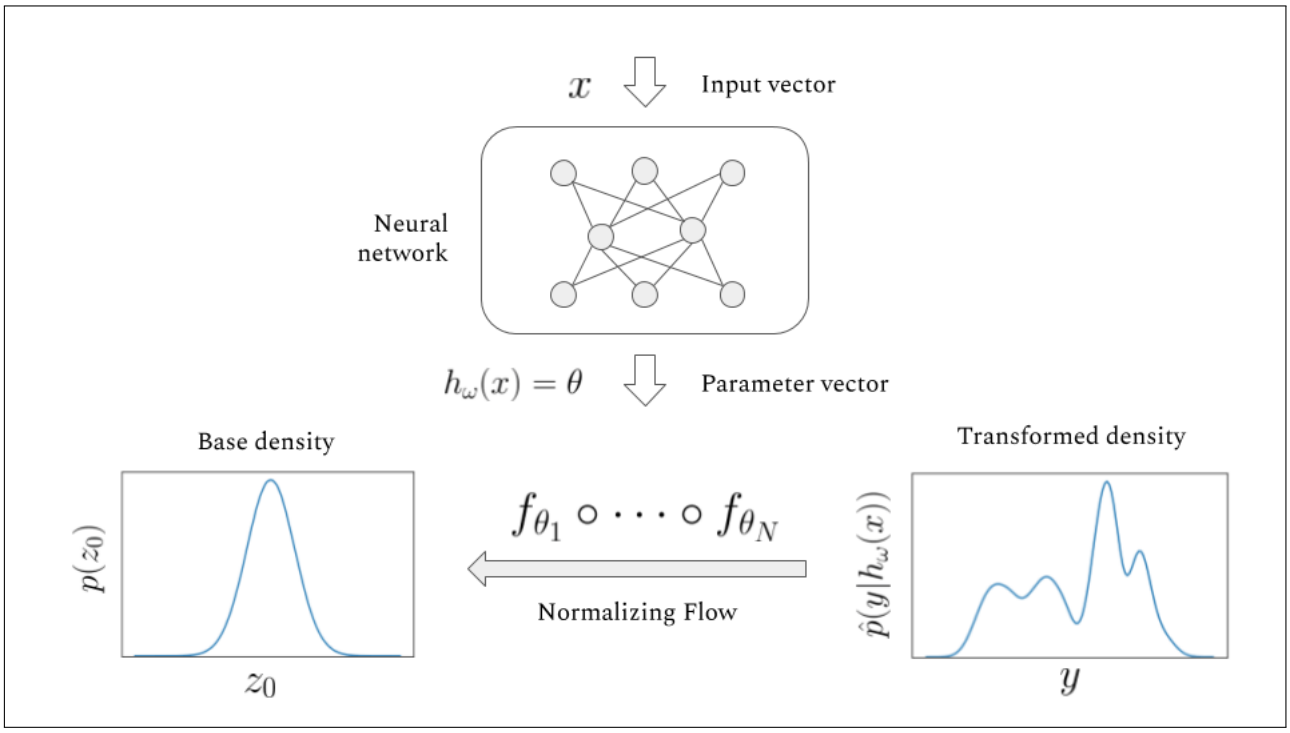



Figure 2.23: Normalized Flow model [59].

### 2.4.12 Semi-Supervised Learning

Semi-supervised learning (SSL) is a type of learning that falls between supervised and unsupervised learning. In SSL, the algorithm is provided with some labeled data, but not necessarily for all examples. Often, the labeled data will consist of the targets associated with some of the examples [60].

In the standard setting of SSL, the data set $X = (x_i)_{i=1}^n$ can be divided into two parts:

- The points $X_l = (x_1, ..., x_l)$, for which labels $Y_l = (y_1, ..., y_l)$ are provided.
- The points $X_u = (x_{l+1}, ..., x_{l+u})$, the labels of which are not known.

SSL algorithms can use the labeled data to learn a model that can predict the labels of the unlabeled data. This can be done by using the labeled data to learn a mapping from the input data to the output labels. The mapping can then be used to predict the labels of the unlabeled data.

With the advent of deep learning techniques has significantly enhanced the quality of segmentation outcomes, much like it has revolutionized numerous other computer vision challenges, where there are some studies have been carried out on semantic segmentation based on semi-supervised Deep Learning methods including Generative methods and some other

techniques like data augmentation which have shown satisfactory results [**pelÃąezvegas2023survey**, 61].

The existant seismic salt domes's dataset are known for its class imbalance between salt and non-salt images, and for the non-abondance of them. Few works proposed solutions to evercome this issues, we mention :

- The authors introduce a semi-supervised approach for segmenting salt bodies in seismic images [62]. In this method, the labeled dataset contains seismic images annotated with the locations of salt bodies, while the unlabeled dataset comprises seismic images that lack such annotations.

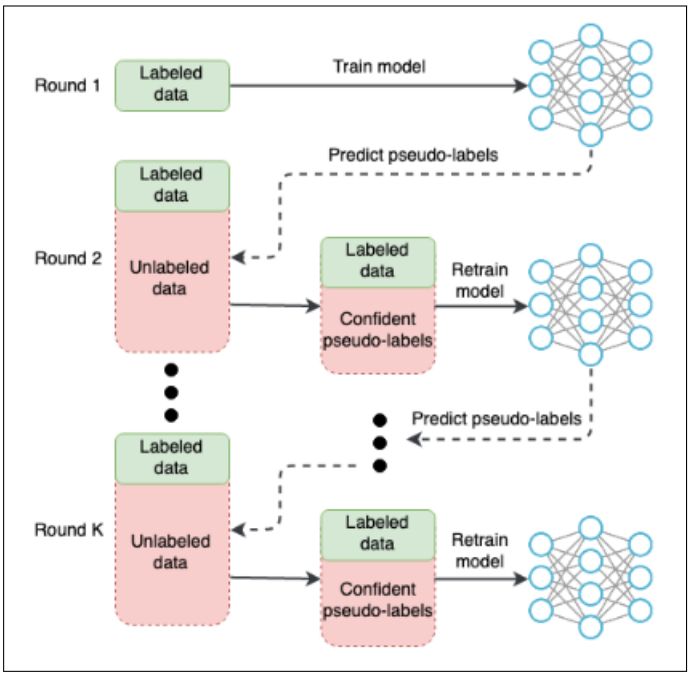



Figure 2.24: Multi-phase Self training [62].

The proposed method uses an ensemble of CNNs to reduce error amplification in the self-training process. Self-training involves utilizing the model itself to assign labels to unlabeled data. The model is then retrained on the combined dataset of labeled and unlabeled data. Error amplification is a problem that can occur in self-training when the model makes mistakes on the unlabeled data. These errors can subsequently affect the labeled data, potentially reducing the accuracy of the model.

The authors obtained top-performing results on the TGS Salt Identification Challenge dataset.

- The proposed methodology [63] involves training two generative models: a Variational Autoencoder (VaE) to generate salt body masks and a Conditional Normalizing Flow (CNF) to generate seismic image patches conditioned on the masks. The VaE is trained on dataset masks to learn their distribution, while the CNF is trained on pairs of masks and image patches to learn the conditional distribution of patches given masks. To generate data augmentation samples, the VaE is used to sample salt

masks, which are then used as input to the CNF to generate corresponding seismic image patches. This approach allows for the generation of realistic and diverse samples that can be used to augment training data for Semantic Segmentation of salt bodies.

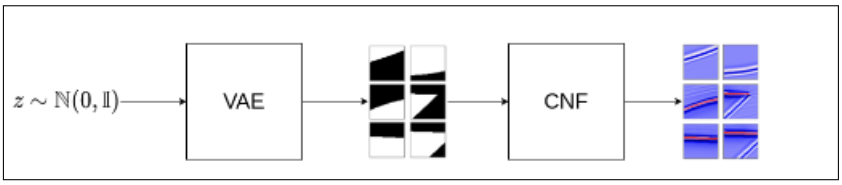



Figure 2.25: VaE-CNF model [63].

## 2.5 Metrics for Semantic Segmentation

For the task of Semantic Segmentation, we use various metrics for evaluating the efficiency and the accuracy for some real world application. For clarity in our explanation, we use the following notation: we consider a total of $k+1$ classes (ranging from $L_0$ to $L_k$, including a void or background class). Here, $Y_{ij}$ denotes the number of pixels from class $i$ that are predicted to belong to class $j$. Specifically, $Y_{ii}$ indicates the number of true positives (TP), whereas $Y_{ij}$ and $Y_{ji}$ are typically regarded as false positives (FP) and false negatives (FN), respectively (although they can also represent the sum of both false positives and false negatives). We demonstrate the popular ones:

### 2.5.1 For Accuracy

- **Intersection over Union (IoU) :** IoU means the rate between the intersection and union which means it is equal to $TP/(TP+FP+FN)$ where TN stands for true negative.The equation below is for Mean IoU across all classes (2.11):

$$\text{MIoU} = \frac{1}{k+1} \sum_{i=0}^{k} \frac{Y_{ii}}{\sum_{j=0}^{k} Y_{ij} + \sum_{j=0}^{k} Y_{ji} - Y_{ii}} \tag{2.11}$$

- **Pixel Accuracy (PA) :** it is the simplest metric, simply computing a ratio between the amount of properly classified pixels and the total number of them. The equation below is across all classes known as MPA (2.12):

$$\text{Mean Pixel Accuracy} = \frac{1}{k+1} \sum_{i=1}^{k} \frac{Y_{ii}}{\sum_{j=1}^{k} Y_{ij}} \tag{2.12}$$

- **The Dice Similarity Coefficient (DSC) :** The dice Coefficient (DSC),one of the common methods for evaluating segmentation results, indicates a level of similarity between the reference (manual segmentation) and segmented result ( (Raina et al., 2023). The formulation of DSC is given by (2.13):

$$DSC(A,B) = \frac{2|A \cap B|}{|A| + |B|}, \quad DSC \in [0,1] \tag{2.13}$$

- **The Hausdorff distance (HD) :** HD is metric represents the spatial distance between two point sets. HD is defined as follows (2.14):

$$HD(A,B) = \max\left\{\sup_{a\in A}\inf_{b\in B} d(a,b), \sup_{b\in B}\inf_{a\in A} d(b,a)\right\} \tag{2.14}$$

  where $d(a,b)$ is the distance metric between points $a$ and $b$, and $A$ and $B$ are sets of points.

- **The Mean Absolute Distance (MAD) :** MAD is a measure of the average distance between each data point and the mean of the data set. It is a robust measure of variability, meaning that it is not as sensitive to outliers as some other measures, such as the standard deviation. It is given as follows (2.15):

$$MAD(A,B) = \frac{1}{|A|}\sum_{a\in A}\min_{b\in B} d(a,b) \tag{2.15}$$

  where $A$ is the predicted set and $B$ is the Ground Truth set.

### 2.5.2 For Efficiency

- **Model complexity :** Model parameters and Floating Point Operations (FLOPs) are commonly used metrics to assess model complexity. Typically, models with a high number of parameters and FLOPs exhibit lower implementation efficiency. Importantly, these metrics are not influenced by the implementation environment.

- **Implementation speed :** is another crucial factor for evaluating a model's efficiency. Metrics such as runtime and Frames Per Second (FPS) are commonly used for this purpose. However, it is important to note that these metrics are influenced by the hardware and software environment.

## 2.6 Common Challenging Issues

### 2.6.1 Balance between Accuracy and Efficiency

Efficiency and accuracy are two fundamental metrics used to evaluate the performance of various algorithms and models in machine learning and computer vision tasks, such as Semantic Segmentation.

**Effeciency** refers to how quickly or resourcefully a model can perform its task. In Semantic Segmentation, efficiency can be measured in terms of computational resources (such as memory and processing power) and time required to process an image. A more efficient model can achieve satisfactory performance with fewer computational resources and faster processing times.

**Accuracy** refers to how close a model's predictions are to the true values or labels in a dataset. In the context of Semantic Segmentation, accuracy measures how accurately the model assigns each pixel in an image to its corresponding class or category. Higher accuracy indicates that the model is making fewer mistakes in segmenting the objects in the image.

[64] states that balancing accuracy and efficiency is crucial in practical applications. While high accuracy ensures precise segmentation results, it often comes at the cost of increased computational complexity and longer processing times, making it less suitable for real-time or resource-constrained environments. Conversely, highly efficient models may sacrifice some accuracy to achieve faster processing speeds or to be deployable on low-power devices.

Finding the right trade-off between accuracy and efficiency depends on the specific requirements of the application. In some cases, sacrificing a bit of accuracy for significantly improved efficiency may be acceptable, while in others, the highest possible accuracy is non-negotiable despite longer processing times.

### 2.6.2 Dependency on high-quality training data

High-quality training data are crucial for achieving accurate Semantic Segmentation. Yet, obtaining such data, which require pixel-level annotation, is a laborious and time-consuming task. This dependency poses a significant challenge in Semantic Segmentation. To address this challenge, researchers have developed weakly- and semi-supervised methods, assuming only low-quality training data are accessible. However, despite these efforts, the performance of these methods still lags behind fully supervised approaches, highlighting the need for further investigation.

### 2.6.3 Domain Gap across different datasets

The domain gap presents a significant challenge in Semantic Segmentation and related vision tasks. For instance, while PSPNet achieves an impressive 85.4% Mean Intersection over Union (MIoU) on the PASCAL VOC 2012 dataset, its performance drops considerably to 44.94% MIoU on the more complex ADE20K dataset, which comprises 150 semantic classes and intricate scenes. This disparity arises due to variations across datasets, including differences in category numbers, scene appearances, dataset sizes, object sizes, and other factors. Consequently, these differences exacerbate the divide between heterogeneous domains. It's essential for researchers and developers to address the domain gap issue when applying Semantic Segmentation techniques in practical vision applications .

## 2.7 Transfer Learning

Training a deep Neural Network from scratch is often impractical due to the need for large datasets and lengthy convergence times. Even with sufficient data, using pre-trained weights instead of starting from scratch is beneficial for faster convergence and better performance. Fine-tuning pre-trained networks, a common transfer learning method, allows for effective adaptation to new tasks. Yosinski et al. demonstrated that transferring features from different tasks generally outperforms random initialization, though the effectiveness decreases as task dissimilarity increases. However, transfer learning involves specific architectural constraints and adjustments in the training process. Typically, higher-level layers are fine-tuned while lower-level layers, containing generic features, are less modified. A smaller learning rate is often used to preserve the pre-trained weights. Due to the difficulty

of acquiring extensive per-pixel labeled segmentation datasets, which are much smaller than classification datasets like ImageNet , fine-tuning pre-trained classification networks is a prevalent and successful strategy for segmentation tasks [24].

## 2.8 Conclustion

This chapter has provided a detailed exploration of Deep Learning architectures employed in Semantic Segmentation. It has highlighted the evolution from classical image processing techniques to sophisticated deep learning models specifically designed for pixel-level classification tasks. Throughout the discussion, various challenges and advancements in the field have been examined, illustrating the continuous refinement and adaptation of these models to achieve higher accuracy and efficiency in real-world applications. This review underscores the pivotal role of Deep Learning in advancing Semantic Segmentation capabilities and sets the stage for further exploration into future research directions.

# Chapter 3

# Semantic Segmentation of Salt Images in The Literature

## 3.1 Introduction

Recent developments in Deep Learning have generated considerable interest in advancing the analysis of salt dome seismic images. This chapter explores state-of-the-art Semantic Segmentation techniques tailored for accurately delineating salt structures and other geological features. By leveraging the power of artificial intelligence, these methods aim to automate and optimize the interpretation process, offering profound implications for hydrocarbon exploration and reservoir management. Through a concise review of cutting-edge methodologies, this chapter provides insights into the evolving landscape of Deep Learning applications in salt dome imaging, setting the stage for further advancements in this critical field.

## 3.2 CNNs Based Models

CNN-based Semantic Segmentation methods have shown remarkable performance in delineating salt dome structures in seismic images, leveraging their ability to capture complex spatial features. These models excel at adapting to diverse imaging conditions and learning from large datasets, leading to enhanced segmentation accuracy. However, challenges like data scarcity and model generalization persist, warranting further research.

### 3.2.1 Robust Concurrent Detection of Salt Domes and Faults in Seismic Surveys Using an Improved UNet Architecture

One of the most relevent work was proposed by M.Derriche & al. in 2020 [65], supported by the Center for Energy and Geo Processing (CeGP) at King Fahd University of Petroleum and Minerals. This paper introduces an innovative Semantic Segmentation method for simultaneous detection of salt domes and faults in seismic surveys, employing an enhanced

U-Net architecture. Here are the fundamental aspects of their methodology and the results they have achieved:

**Methodology :**

- The authors propose two variants of the U-Net architecture: UNet-VGG19 and UNet-ResNet34, where the encoders are VGG19 and ResNet34 networks pre-trained on natural images from ImageNet dataset.
- Transfer learning is utilized by employing pre-trained encoders and fine-tuning them on seismic data to mitigate the shortage of labeled seismic data.
- The models are trained on a small dataset of 84 labeled seismic images (61 for salt domes and 43 for faults) from the Netherlands offshore F3 block.
- To handle class imbalance, faults are manually thickened, and a balanced cross-entropy loss function is used during training.

**Experimental Results:**

- **Qualitative Evaluation:**
  - Visualizations of the predictions on test images show accurate detection of salt domes by all four networks (UNet-VGG19, UNet-ResNet34, and their counterparts trained from scratch).
  - The pre-trained networks (UNet-VGG19 and UNet-ResNet34) accurately detect most fault structures, while the networks trained from scratch struggle with fault detection.
- **Quantitative Evaluation:**
  - The metrics employed include Precision, Recall, F1-score, and IoU (Intersection over Union).
  - The pre-trained networks achieve high accuracy for background and salt detection but lower performance for faults.
  - UNet-VGG19: IoU of 0.6588 for faults and 0.9776 for salt domes.
  - UNet-ResNet34: IoU of 0.5419 for faults and 0.9727 for salt domes.
  - The basic U-Net model (baseline) shows good performance but misses most fault structures (low recall).

**Comparison with Other Works mentioned in the work**

- Most existing works focus on single-event detection (either salt domes or faults) and lack quantitative evaluation on real data.

- The proposed method surpasses the multiresolution approach in (M. Alfarraj et al., 2018) for simultaneous detection, achieving IoU scores of 0.2656 for faults and 0.5261 for salt domes.
- For salt dome delineation, the proposed method achieves better segmentation results compared to SegNet (Y. Shi and al. 2018) and U-Net (Y. Zeng and al. 2017) & Y. X. Zhang and al. 2019) based methods, which either lack quantitative evaluation or show poorer performance.
- For fault detection, the proposed method outperforms the U-Net based method in (S. Li, C. Yang and al. 2019) (IoU of 0.500) and the CNN-based method in the work given by A. Pochet and al. 2019 trained on synthetic data (F1-score of 0.80).

### 3.2.2 Using Deep Learning based methods to classify salt bodies in seismic images

Another work, similar to the first one above, proposed by Muhammad Saif ul Islam (2020) [66], the paper suggests employing a Deep Learning approach that combines U-Net with SE-ResNet to automatically classify and delineate salt bodies in seismic images. Here are the main aspects of their methodology and the results they have achieved:

**Methodology :**

- The U-Net architecture with modified ResNet blocks and addition of Squeeze-and-Excitation Networks (SE-ResNet) is used.
- The model is trained in two stages: first, for 300 epochs using a blend of Binary Cross Entropy (BCE) and Dice loss, and subsequently, for another 300 epochs with emphasis on Lovasz loss to enhance the Intersection over Union (IoU) metric.
- Data preparation involves reading seismic images, converting to grayscale and numpy arrays, normalizing pixel values.
- Stratified training/validation split is done based on salt coverage calculated from image masks.

**Experimentation Results:**

- **Quantitative Results :**
  - 10-fold cross-validation was performed with 600 epochs per fold.
  - Average IoU score achieved was 0.842 on validation data using Lovasz loss
  - With BCE + Dice loss, average validation IoU was lower at 0.819.
- **Qualitative Results:**

  - Predicted salt masks on unseen validation data show the method is able to accurately classify and segment the salt body regions (visualized in Figure 9 on the paper).

**Comparison with Other Works mentioned on the paper :**

- Traditional methods relied on manually engineered seismic attributes, requiring expert knowledge.
- Earlier machine learning methods also utilized seismic attributes as input features.
- The proposed Deep Learning approach operates directly on the raw seismic image data, autonomously learning pertinent features throughout the training process.
- The author didn't perform a comparative table but it outperforms prior attribute-based and traditional methods, achieving an IoU of 0.842 compared to Zeng et al. (2019) 90% for example and to lower metrics reported in other studies (e.g. accuracy of 96% in Shi et al. (2019)).

The paper mentioned that the classical methods can be abondent where we can eliminate the need for manual seismic attribute extraction and expert guidance since DL methods now show a better performance, and his work was an End-to-end learning of relevant features from raw data.

### 3.2.3 Identification of Salt Deposits on Seismic Images Using Deep Learning Method for Semantic Segmentation

Finally, another work was published by Aleksandar Milosavljevi´c (2020)[67], the paper introduces a new deep CNN architecture tailored for semantic segmentation of salt deposits in seismic images. Key aspects of their approach and the results obtained are as follows:

**Methodology :**

- The architecture is inspired by the U-Net model and integrates ideas from ResNet and DenseNet architectures.
- It consists of three main block types: C-blocks with convolutional layers and skip connections inspired by ResNet/DenseNet, D-blocks for downsampling, and U-blocks for upsampling and concatenating feature maps.
- The network was implemented in Python using the Keras library with TensorFlow backend.

**Experimental Results:**

- **Quantitative Results :**

  - The proposed architecture achieved a score of 0.85241 on the private TGS Salt Identification Challenge leaderboard, ranking in the top 14% out of 3221 teams.
  - In post-competition experiments, an ensemble of 5 models with the proposed architecture (without dropout) achieved the best private score of 0.85219.
  - It outperformed standard segmentation models like U-Net (0.84404), LinkNet (0.84565), and PSPNet (0.81138) on the same data.
  - The FPN model achieved the best public score of 0.83623, slightly better than the proposed method's 0.83181.
- **Qualitative Results:**
  - Visualizations show the proposed method producing accurate segmentation masks on easier examples, comparable to other models.
  - On more challenging cases with no salt present, the method (like others) tends to incorrectly detect salt deposits.

**Comparison with Other Works mentioned on the paper :**

- Traditional methods used handcrafted features and texture analysis for seismic segmentation.
- Recent works have applied Deep Learning, including CNN architectures like FCN, DeconvNet, U-Net, SegNet for Semantic Segmentation.
- Some specifically tackled salt deposit identification using approaches like CNNs on image patches, 3D CNNs on data cubes, ensemble methods, etc.
- The winning solution in the TGS challenge used an ensemble of ResNet/ResNeXt encoders with skip connections, self-training, and achieved a higher score of 0.89646.

We expose below a comparaison table of Deep Learning Methods CNN based methods for Salt Body Detection 3.1 :

Table 3.1: Comparison of Deep Learning Methods CNN based methods for Salt Body Detection

| **Aspect** | **Paper 1 (Milosavljević)** | **Paper 2 (Islam)** | **Paper 3 (Alfarhan et al.)** |
|---|---|---|---|
| Methodology | U-Net-based architecture with modifications inspired by ResNet and DenseNet (C-blocks, D-blocks, U-blocks) | U-Net architecture combined with SE-ResNet blocks | Improved U-Net architecture with VGG19 or ResNet34 as encoder, along with transfer learning |

Table 3.1: Comparison of Deep Learning Methods CNN based methods for Salt Body Detection (continued)

| Aspect | Paper 1 (Milosavljević) | Paper 2 (Islam) | Paper 3 (Alfarhan et al.) |
|---|---|---|---|
| Data Augmentation | Flipping, translation, intensity scaling/shifting | Grayscale conversion, normalization | Not explicitly mentioned |
| Qualitative Results | Visualizations of sample images, ground truth, and predictions; struggles with some hard cases | Visualizations of images, ground truth, and predictions from proposed and other methods; accurate salt region identification | Visualizations of salt dome and fault predictions; accurate detection in most cases |
| Quantitative Results | IoU = 0.85241 on TGS Salt Identification Challenge leaderboard | Average IoU = 0.84201 after 10-fold cross-validation on TGS dataset; highest IoU = 0.85 with BCE + Dice Loss and Lovász Loss | Precision, Recall, F1-score, and IoU metrics for salt and fault detection; superior performance compared to baseline U-Net |
| Pros | Novel architecture, comparison with standardized models, extensive data augmentation | SE-ResNet blocks, stratified training/validation split, exploration of loss functions | Transfer learning, concurrent salt dome and fault detection, robustness to event similarities |
| Cons | Struggles with hard cases, no transfer learning mentioned | No detailed comparison with other models, limited data augmentation | Limited information on data augmentation and comparison with other methods |

## 3.3 Transformers and Attention Gates inspired models

Transformers or attention-based Semantic Segmentation methods have recently gained attention for their ability to capture long-range dependencies and contextual information in images. By incorporating self-attention mechanisms, these models can effectively weigh the importance of different image regions, enhancing segmentation accuracy, particularly in tasks involving complex structures like salt domes. Their capability to process global context enables more informed decision-making, leading to superior performance in delineating subtle features within seismic imagery.

### 3.3.1 Transformer Model for Fault Detection from Brazilian Pre-salt Seismic Data

Let´ıcia Bomfim & al. from niversity of Campinas, Brazil proposed a transformer based architecture to delineate salt domes seismical images (2023) [68], this work investigates

using a transformer model called TransUNet for the task of fault detection from seismic data of the Brazilian pre-salt region. Faults and fractures are important geological features that can impact fluid flow and hydrocarbon reservoirs. The fundamental components of their approach and the results they have attained are as follows:

**Methodology :**

- Used seismic amplitude and fault interpretation volumes from the Santos Basin pre-salt area as input and target data.
- Preprocessed data by extracting 2D sub-images, data augmentation via flipping, normalization
- Employed the TransUNet model which combines a convolutional U-Net with a transformer encoder to leverage both CNN spatial features and transformer's global context modeling
- Compared to conventional CNN models: U-Net, U-Net++, SegNet
- Initial dataset of 9369 2D sub-images extracted with size 1401×1481 pixels,renormalized between -1 and 1
- After augmentation: 9242 images for training, 1914 for validation
- Used binary cross-entropy loss
- Evaluated using Dice coefficient and Intersection over Union (IoU) metrics
- Trained for 100 epochs
- Used batch size 16 for TransUNet (due to higher memory usage)
- Learning rate of 1e-4
- Same training settings for baselines (U-Net, U-Net++, SegNet) for fair comparison

**Experimental Results:**

- **Quantitative Results :**
  - TransUNet achieved the best performance with a Dice coefficient of 88.34% and IoU of 84.34%
  - Outperformed U-Net (85.99% Dice), U-Net++ (83.41% Dice), SegNet (83.31% Dice)
  - Also had higher accuracy, precision, recall and F1-score
- **Qualitative Results:**
  - Visualizations show TransUNet could better detect faults, especially smaller structures.

  - Reconstructed full 3D seismic volume shows TransUNet captured more fault details with better vertical continuity compared to U-Net.

**Comparison with Other Works mentioned on the paper :**

- Same training settings for baselines (U-Net, U-Net++, SegNet) for fair comparison
- Highlighted recent success of transformer models in computer vision tasks
- Builds on prior work using CNNs for seismic fault detection

### 3.3.2 Automatic salt deposits segmentation: A deep learning approach

Another work, tackled the same problem and get inspired form attention gates, made by Mikhail Karchevskiy & al. from Russia (2018) [69].Here are the crucial aspects of their methodology and the results they have obtained:

**Methodology :**

- Utilized a U-Net framework incorporating a ResNeXt-50 encoder pretrained on ImageNet as the foundational model.
- Incorporated multiple methods: Spatial-Channel Squeeze & Excitation, Lovasz loss, CoordConv, Hypercolumn, and Attention Gates.
- Used data augmentation: horizontal flip, brightness changes, horizontal shifts, rotations.
- Trained 5-fold model with horizontal flip test-time augmentation (TTA).
- Extended input to 128x128 pixels with relative depth channel and CoordConv channels.
- Used cyclic learning rate scheduling and snapshot ensembling of last 10 best models.

**Experimental Setup:**

- Dataset: 8000 train, 18000 test 101x101 pixel seismic images
- Trained on single Nvidia GTX 1080 Ti GPU
- 24 hours for full training and prediction cycle
- Batch size 20, Adam optimizer.
- The model was trained for 80 epochs using Binary Cross-Entropy (BCE) loss, followed by an additional 50 epochs using a combination of 0.1 BCE loss and 0.9 Lovasz loss.

**Quantitative Results:**

- Achieved 27th place (top 1%) on Kaggle competition
- Intersection over Union (IoU) metric used to evaluate performance at different thresholds(0.84%)
- Score increase with different components:
  - ResNet152 encoder: 1.26%
  - Lovasz hinge loss: 0.62%
  - Hypercolumn: 0.85%
  - SE-ResNeXt50 encoder: 0.72%
  - Spatial/channel attention gates: 2.25%
  - TTA horizontal flip: 0.12%

The paper does not explicitly provide or discuss **qualitative results** in terms of visualizations or examples comparing the segmentation outputs. The only mention related to qualitative analysis is this line:

"Higher resolution input images: provided no noticeable improvements and caused much slower learning." ,we think that because of the results of Dice and Jaccard coefficients where they acheived 0.5% as a maximum level.

We expose below a comparaison table between Transformer-Attention based model 3.2 :

Table 3.2: Comparison between Transformer-attention based models

| **Aspect** | **TransUNet for Fault Detection** | **Salt Deposit Segmentation** |
|---|---|---|
| Methodology | - TransUNet (U-Net + Transformer)<br>- ResNeXt-50 CNN encoder<br>- Spatial-Channel Squeeze & Excitation<br>- CoordConv layers<br>- Hypercolumn technique | - U-Net with ResNeXt-50 encoder<br>- Spatial-Channel Squeeze & Excitation<br>- Lovasz loss<br>- CoordConv channels<br>- Spatial/channel attention gates<br>- Hypercolumn technique<br>- 5-fold model, horizontal flip TTA |

Table 3.2: Comparison between Transformer-attention based models (continued)

| Aspect | TransUNet for Fault Detection | Salt Deposit Segmentation |
|---|---|---|
| Qualitative Results | - Visualizations show better fault detection, especially smaller structures<br>- Better continuity in 3D volumes | - No qualitative results or visualizations discussed |
| Quantitative Results | - Dice: 88.34%<br>- IoU: 84.34%<br>- Outperformed U-Net, U-Net++, SegNet | - 27th place (top 1%) on Kaggle<br>- IoU score improvements:<br>+ ResNet152 encoder: +1.26%<br>+ Lovasz loss: +0.62%<br>+ Hypercolumn: +0.85%<br>+ SE-ResNeXt50: +0.72%<br>+ Attention gates: +2.25%<br>+ TTA horizontal flip: +0.12% |
| Pros | - Transformer captures global context<br>- Good for discontinuous structures<br>- Qualitative results show improvement | - Used multiple advanced techniques<br>- Ensembling and TTA |
| Cons | - Higher memory usage<br>- Slightly more complex architecture | - No qualitative analysis<br>- Less innovative architecture |

## 3.4 Generative Models and Semi-Supervised Learning Works

Generative-based Semantic Segmentation and with semi-supervised learning, offers a compelling approach to address data scarcity issues in seismic imaging. These methods can effectively learn from both labeled and unlabeled samples. This hybrid approach enhances segmentation accuracy by exploiting the rich information embedded within unlabeled data while still benefiting from the supervision provided by labeled examples.

### 3.4.1 Generating data augmentation samples for Semantic Segmentation of salt bodies in a synthetic seismic image dataset

As an exemple, Luis Felipe Henriques & al. from Pontifical Catholic University of Rio de Janeiro, Brazil proposed a technique for augmenting data in semantic segmentation of salt bodies in seismic images (2021) [63]. Here are the essential elements of their approach and the results they have achieved:

**Methodology**

- Two Deep Learning models are trained:
  - A Variational Autoencoder (VAE) to generate salt body masks
  - A Conditional Normalizing Flow (CNF) model inputs generated salt masks to produce corresponding patches of seismic images.
- During data augmentation, the VAE first generates salt masks, then the CNF model uses those masks to generate realistic seismic image patches containing salt bodies
- This allows generating pairs of image patches and salt masks focused on the boundaries of salt bodies, which are challenging areas.

**Experimental Results:**

- The experiments were conducted using a dataset derived from two publicly available synthetic seismic models designed to replicate salt bodies found in the Gulf of Mexico.
- Assessed the performance of 10 different state-of-the-art Semantic Segmentation models when trained with and without the suggested data augmentation techniques.
- With the generated augmentations, IoU (Intersection over Union) metric improved by an average of 8.57% across all models
- Best improvement was 25.1% for DeepLabV3+ with Xception_71 backbone
- Best overall result was 95.17% IoU achieved by DeepLabV3+ with ResNet_v1_101 backbone when using augmentations (2.14% improvement)
- all models are trained using the Adam Optimizer for 40K training iterations and a mini-batch of 20 examples per iteration on an NVidia Tesla P100 graphic processing unit (GPU).

**Comparisons**

- The proposed method exceeded the performance of seven other standalone data augmentation methods from the Albumentations library, all trained using the settings detailed in section 6.2. Training took an average of 1 hour and 10 minutes on an Nvidia Tesla P100 GPU.
- Composing the proposed augmentations with the ElasticTransform method achieved the highest 90.39% IoU score
- Using a larger context size of 128x128, incorporating augmentations improved the Intersection over Union (IoU) from 84.85% to 90.55% for DeepLabV3+ with the MobileNetV3_large backbone. Each DeepLabV3+ model underwent three rounds of training, totaling 40,000 iterations, for both augmented and non-augmented versions, av-

eraging approximately 6 hours and 20 minutes per model. Table 3 presents a comparison of the model's performance, highlighting the best validation scores achieved in each variant.

- Scaling the method to larger context sizes retained substantial performance improvements.
- Regarding the generated data, the CNF model achieved a loss value of 1.81 after six days and 259,200 training iterations on a Google Cloud TPU v2-8. In contrast, the VAE loss settled at approximately 61.17 following 14 hours and 28,800 training iterations on an NVidia Tesla P100 GPU.
- Further exploration of context sizes has been constrained in this study due to the extensive experiments conducted and the significant computational resources needed.

They did not address qualitative results but suggested future directions, such as testing the proposed method on alternative datasets like the TGS Salt Identification Challenge, applying it to other seismic tasks such as seismic facies segmentation with adaptations for natural images, and exploring the use of different generative models like Conditional GANs.

### 3.4.2 Salt Detection Using Segmentation of Seismic Image

Another work, a paper proposes using a deep convolutional neural network (DCNN) with an autoencoder architecture for Semantic Segmentation of salt bodies in seismic images by Mrinmoy Sarkar(2022 -USA) [70].

**Methodology**

- The autoencoder consists of an encoder section comprising convolutional and pooling layers, and a decoder section comprising convolutional and upsampling layers.
- The encoder condenses the input seismic image into a latent representation.
- The decoder reconstructs the output segmentation mask from the compressed representation.
- The loss function employed is the binary cross-entropy loss with sigmoid activation.
- The network undergoes training with the ADADELTA optimization algorithm.

**Experimental Results:**

- Evaluated on a dataset of 4,000 seismic images and corresponding binary masks indicating salt regions.
- Dataset split into 3,200 images for training and 800 for testing.
- After 50,000 training epochs, achieved a training loss of 0.1307 and test loss of 0.1452.

- Performed 10-fold cross-validation after 20,000 epochs, mean cross-validation error was 0.19.
- Showed examples of ground truth masks and masks that were forecasted by the trained model.
- Prediction time is fast after training the model

**Qualitative Assessment:**

- The proposed DCNN with autoencoder architecture could successfully segment salt bodies in seismic images.
- Reduced need for expert human interpretation by automating the segmentation process.
- No explicit qualitative assessment.

The paper does not provide explicit numerical comparisons to other methods, but claims the autoencoder DCNN approach performs well on this application of segmenting salt bodies from seismic imagery, but at least the architecture outperform manual interpretation methods.

### 3.4.3 Semi-Supervised Segmentation of Salt Bodies in Seismic Images using an Ensemble of Convolutional Neural Networks

The paper (Y.Babakhin & al. 2019) [62] proposes a semi-supervised approach for segmenting salt bodies in seismic images using convolutional neural networks (CNNs).

**Methodology**

- It uses an iterative self-training procedure with K rounds, where each round:
  - Trains the model using labeled data alongside confident pseudo-labels derived from the paraphrased versions from the previous round.
  - Generating new pseudo-labels entails predicting labels for unlabeled data based on confident predictions from the model in the previous round, essentially using these predictions as surrogate ground truth labels.
- To reduce error accumulation during self-training, it uses an ensemble of two CNN models (U-ResNet34 and U-ResNeXt50) and averages their predictions for pseudo-labeling.
- The CNN architectures use attention mechanisms like Squeeze-and-Excitation, Feature Pyramid Attention, and Hypercolumns to improve performance.
- Images resized to 202x202 pixels and padded to 256x256 pixels.

- Training Procedure:
  - **Round 1**: Trained only on 4000 labeled images.
  - **Rounds 2 & 3**: Trained for T=200 epochs on pseudo-labels, then fine-tuned T=200 more epochs on labeled data.
  - Used all 18000 pseudo-labeled images, did not filter low-confidence ones.
  - Initialized model weights from ImageNet pretraining.
  - Performed data augmentation (horizontal flips for test-time augmentation).
- Ensembling:
  - Used 4 model snapshots per fold by saving every 50 epochs.
  - For inference, ensembled 20 snapshots for single model, 40 for U-ResNet34 + U-ResNeXt50.
  - Ensembled by averaging predictions.
- **Optimization:** Used warm-up by training first 50 epochs with binary cross-entropy also Then minimized Lovasz loss for 150 epochs to optimize IoU directly, for the algorithm itself is not explicitly stated, but likely stochastic gradient descent since they mentioned the cosine annealting rate schedule from 0.001 to 0.0001 every 50 epochs (as a reference ).
- **Loss function :** Binary cross-entropy loss for initial 50 epochs and Lovasz loss for remaining 150 epochs to directly optimize intersection-over-union (IoU) (as mentioned above).
- The dataset was taken from TGS Salt Identification Challenge dataset with 4000 labeled and 18000 unlabeled 101x101 seismic image patches.

**Experimental Results :**

- Quantitative: Achieved top score of 0.8964 mean average precision (mAP) on private test set, surpassing the previous best by 0.9%.
- Qualitative: Visualizations show self-training improves validation mAP over rounds (Figure 1 in the paper).

**Comparison to Other Works:**

- Outperformed approach by Karchevskiy et al. by 0.9% mAP on same dataset(the paper discussed already in section 3.3.2).
- Claims to achieve state-of-the-art performance, ranking 1st among 3234 competitors.

We expose below a comparaison table between Transformer-Attention based model 3.3 :

Table 3.3: Comparison between works using self-supervised methods or/and generative models

| **Aspect** | **Henrique & al. 2021** | **Sarkar 2022** | **Y.Babakhin & al. 2019** |
|---|---|---|---|
| Methodology | Uses VAE to generate salt masks and CNF to generate seismic patches from masks. Data augmentation by sampling from VAE and CNF | DCNN autoencoder architecture with encoder to compress input and decoder to reconstruct segmentation mask. Direct training on seismic images and masks | Semi-supervised self-training approach using ensemble of CNNs. Iterative training on labeled data and pseudo-labels from unlabeled data |
| Quantitative Results | 8.57% average improvement in IoU across 10 models. Best: 95.17% IoU (DeepLabV3+ ResNet_v1_101). Improved IoU from 84.85% to 90.55% on 128x128 patches | Training loss: 0.1307. Test loss: 0.1452. 10-fold CV error: 0.19 | mAP 0.8964 on private test set (1st place on TGS Salt Identification Challenge leaderboard). 0.9% improvement over previous best approach |
| Qualitative Results | No explicit qualitative assessment | No explicit qualitative assessment | No direct qualitative assessment, but ranks 1st on real-world benchmark |
| Pros | Consistently improves performance across different models. Outperforms other augmentation methods. Adaptable to larger image sizes | Directly tackles Semantic Segmentation task. Reasonable quantitative performance. Fast prediction after training | Effective semi-supervised approach using unlabeled data. Ensemble mitigates error accumulation. State-of-the-art results |
| Cons | Complex method using two models. No direct qualitative evaluation | No comparisons to other methods. Long training time | No direct qualitative evaluation |

## 3.5 Comparaison between The Methodologies Proposed

Semantic segmentation, a fundamental task in computer vision, involves classifying each pixel in an image into a specific category. Various methodologies, including CNN-based methods, transformers, and self-supervised/generative models, have been employed to tackle this task.For segmenting salt domes, we've discussed the value added by each methodology to solve the problem, each approach offers distinct advantages and drawbacks, catering to

different use cases and requirements.

### 3.5.1 CNN-Based Methods

Convolutional Neural Networks (CNNs) have been the cornerstone and the conventional state-of-the art of many computer vision tasks, including Semantic Segmentation. CNNs excel at learning spatial features and capturing local patterns in images. They are well-suited for tasks where understanding local relationships and patterns is critical. Some key considerations regarding CNN-based methods include:

- **Use Cases :** CNNs are effective for tasks with spatial dependencies, such as image segmentation. They are widely used in various computer vision applications due to their ability to capture local features effectively.
- **Pros :**
  - Well-established architecture with numerous pre-trained models available, allowing for efficient transfer learning.
  - Efficient at learning spatial features, making them suitable for tasks like Semantic Segmentation.
- **Cons :**
  - Limited ability to capture global context in images, which can be crucial for understanding relationships across the entire input.
  - CNNs often overfit, particularly when trained on small datasets.

### 3.5.2 Transformers

Transformers, initially developed for natural language processing (NLP), have shown promising results in computer vision tasks, including Semantic Segmentation recently. Unlike CNNs, transformers excel at capturing long-range dependencies and global context in data. They offer superior performance in tasks where understanding relationships across the entire input is essential. Key points to consider regarding transformers include:

- **Use Cases :** Transformers are suitable for tasks requiring capturing long-range dependencies and understanding global context. They are increasingly utilized in computer vision tasks, especially in scenarios where understanding relationships across the entire image is crucial.
- **Pros :**
  - Excellent at capturing global relationships in data, making them effective for tasks like Semantic Segmentation in large images.
  - Can be combined with CNNs to leverage both local and global information, offering a hybrid approach to tackle complex tasks
- **Cons :**

  - Computationally expensive, especially for large input sizes, which can limit their scalability.
  - Requires large amounts of training data to generalize well, which may pose challenges in datasets with limited annotations.

### 3.5.3 Attention Gates

Attention mechanisms have been widely adopted in various Deep Learning tasks to selectively focus on relevant information while filtering out noise. In Semantic Segmentation, attention gates play a crucial role in enhancing feature representations and improving segmentation accuracy. They enable the model to attend to the most informative regions of the input, effectively incorporating contextual information into the segmentation process. Here's a closer look at attention gates:

- **Use Cases :** Attention gates are beneficial in tasks where capturing contextual dependencies is critical, such as Semantic Segmentation. They facilitate better feature selection and refinement, leading to more accurate segmentation results.
- **Pros :**
  - Enhance feature representations by selectively attending to relevant regions of the input.
  - Improve segmentation accuracy by incorporating contextual information into the segmentation process.
  - Help mitigate the impact of noise and irrelevant features, leading to more robust segmentation performance.
- **Cons :**
  - May introduce additional computational overhead, particularly in deep neural network architectures.
  - Require careful tuning of hyperparameters to balance model complexity and performance.
  - Implementation and integration into existing architectures may require additional effort and expertise.

In the context of Semantic Segmentation, attention gates and transformers serve similar purposes in capturing contextual information and improving segmentation accuracy. However, there are fundamental differences in the way they incorporate attention mechanisms .For attention gates operate at a local level within the network, while transformers capture global relationships across the entire input.That means if the images say are independent variables attention gates are a good option to learn the spatial correlation between the pixels in the image ,whereas if the images are correlated in the way they are temporally dependend (that's depend how the images are obtained) or large-scale images.Also it depends on the hardware requirements ,since they are both gourmand of data.

### 3.5.4 Self-Supervised and Generative Models

Self-supervised learning and generative models offer innovative approaches to Semantic Segmentation tasks, particularly in scenarios where labeled data is limited or expensive to obtain. These models leverage unlabeled data to pretrain networks or augment existing datasets, improving model generalization and performance. Key considerations for self-supervised and generative models include:

- **Use Cases :** Self-supervised and generative models are beneficial when labeled data is scarce or expensive to obtain. They can generate synthetic data to augment training datasets and improve model robustness.
- **Pros :**
  - Can generate synthetic data, augmenting the training dataset and improving model generalization.
  - Effective in scenarios where labeled data is limited, as they can utilize unlabeled data for training, reducing the reliance on annotated samples.
- **Cons :**
  - Often require complex architectures and training procedures, which can increase the computational overhead.
  - May suffer from mode collapse or generate unrealistic samples, requiring careful evaluation of the generated data.

### 3.5.5 Comparative Study between The Techniques

Figure 3.1 shows Taxonomy of works discussed in this chapter, Table 3.4 displays Comparison of Salt Body Segmentation Methodologies discussed in this chapter, Table 3.5 shows summary of Semantic Segmentation Method.

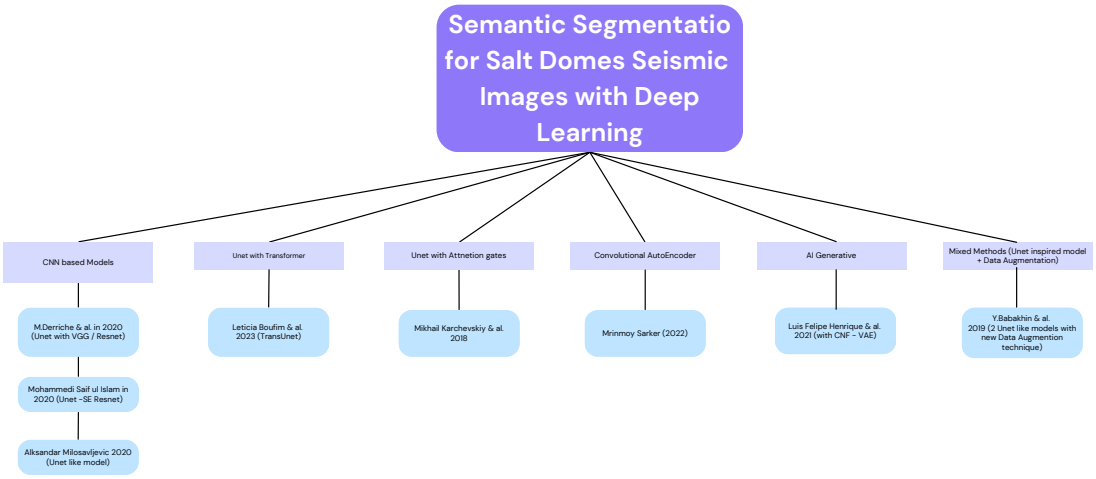



Figure 3.1: Taxonomy of works discussed in this chapter.

Table 3.4: Comparison of Salt Body Segmentation methodologies

| Authors | Methodology | Data Aug. | Qual. Results | Quantitative Results |
|---|---|---|---|---|
| Milosavljević (2020) | U-Net with ResNet and DenseNet-inspired modifications | ✓ | ✓ | IoU = 0.85241 |
| Islam (2020) | U-Net with SE-ResNet blocks | ✓ | ✓ | IoU = 0.85 |
| Alfarhan et al. (2020) | Improved U-Net with VGG19 or ResNet34 encoder, transfer learning | ✗ | ✓ | ResNet34 (pre-trained) IoU = 0.9782 |
| Bomfim et al. (2023) | - TransUNet (U-Net + Transformer) - ResNeXt-50 CNN encoder - Spatial-Channel Squeeze & Excitation - CoordConv layers - Hypercolumn technique | ✗ | ✓(better in 3D) | - Dice: 88.34% - IoU: 84.34% - Outperformed U-Net, U-Net++, SegNet |
| Karchevskiy et al. (2018) | - U-Net with ResNeXt-50 encoder - Spatial-Channel Squeeze & Excitation - Lovász loss - CoordConv channels - Spatial/channel attention gates - Hypercolumn technique - 5-fold model, horizontal flip TTA | ✗ | ✗ | - 27th place (top 1%) on Kaggle - IoU improvements: + ResNet152 encoder: +1.26% + Lovász loss: +0.62% + Hypercolumn: +0.85% + SE-ResNeXt50: +0.72% + Attention gates: +2.25% + TTA horizontal flip: +0.12% |
| Henrique et al. (2021) | VAE for salt masks, CNF for seismic patches. Data augmentation by sampling from VAE and CNF | ✗ | ✗ | - 8.57% avg. improvement in IoU across 10 models - Best: 95.17% IoU (DeepLabV3+ ResNet_v1_101) - Improved IoU from 84.85% to 90.55% on $128 \times 128$ patches |
| Sarkar (2022) | DCNN autoencoder: encoder compresses input, decoder reconstructs mask. Direct training on seismic images and masks | ✗ | - | - Training loss: 0.1307 - Test loss: 0.1452 - 10-fold CV error: 0.19 |
| Babakhin et al. (2019) | Semi-supervised self-training with CNN ensemble. Iterative training on labeled data and pseudo-labels | ✗ | No direct assessment, but ranks 1st on real-world benchmark | - mAP 0.8964 on private test set - 1st place on TGS Salt Identification Challenge - 0.9% improvement over previous best |

Table 3.5: Summary of Semantic Segmentation Methods

| Method | Use Cases | Pros | Cons |
|---|---|---|---|
| CNN-Based Methods | Tasks with spatial dependencies, image segmentation | Efficient at learning spatial features, well-established architecture, suitable for local patterns | Limited ability to capture global context, prone to overfitting with limited data |
| Transformers | Tasks requiring capturing long-range dependencies, global context | Excellent at capturing global relationships, effective for understanding relationships across the entire input | Computationally expensive, requires large amounts of training data for generalization |
| Self-Supervised and Generative Models | Tasks with limited labeled data, data augmentation | Can generate synthetic data, effective with limited labeled data, improves model robustness | Complex architectures and training procedures, may suffer from mode collapse or generate unrealistic samples |
| Attention Gates | Tasks requiring selective focus on relevant features, refining segmentation results | Enhances feature representations, improves segmentation accuracy at a local level | Integrated within CNN-based architectures, limited to spatial domain, additional computational overhead |

## 3.6 Conclusion

In this chapter, we have discussed the methodologies used in Deep Learning for Salt Domes Semantic Segmentation and how they acheived state-of-the art to tackle the problem.When selecting a methodology for Semantic Segmentation tasks, practitioners should consider the specific requirements of the task, the availability of labeled data, computational resources, and the complexity of the problem. CNN-based methods offer efficient spatial feature learning and are suitable for tasks where local patterns are crucial. Transformers excel at capturing global context and long-range dependencies, making them effective for understanding relationships across the entire input. Self-supervised and generative models leverage unlabeled data to improve model performance and robustness, making them valuable in scenarios with limited annotated samples. By understanding the characteristics, benefits, and limitations of each methodology, practitioners can make informed decisions about which approach to use based on the specific requirements and constraints of their Semantic Segmentation task.

# Conclusion

In this thesis, we explored and compared various deep learning approaches for the semantic segmentation of salt domes in seismic images. This task is vital for the oil and gas industry, as salt domes often trap hydrocarbons, making their precise identification crucial for efficient resource management. Historically, the detection of these geological formations has relied heavily on manual interpretation, a method fraught with potential for error and inefficiency. The advent of deep learning has opened new avenues for automating and enhancing this process, promising significant advancements in the accuracy and reliability of seismic image analysis.

Throughout our study, we investigated the performance of several cutting-edge deep learning models. Convolutional Neural Networks (CNNs) have been a cornerstone in image processing tasks due to their ability to capture local features effectively. Models such as AlexNet and ResNet demonstrated robust performance in extracting features from seismic images, providing a solid baseline for comparison. These networks, with their deep layers and convolutional structures, excel at recognizing patterns and textures inherent in complex geological data.

We also examined the U-Net architecture and its variations, which have gained prominence in tasks requiring detailed image segmentation. Originally developed for biomedical image analysis, U-Net's strength lies in its ability to accurately delineate boundaries, an essential feature for identifying salt domes. Its symmetrical encoder-decoder structure, enhanced with skip connections, allows it to capture both fine and coarse features, making it particularly adept at handling the diverse scales present in seismic images. Variants like Linknet and PSPNet extend these capabilities by incorporating advanced techniques such as multi-scale context aggregation and pyramid pooling, further enhancing segmentation performance.

Transformer-based models, renowned for their success in natural language processing, have also shown promise in the field of image segmentation. These models leverage self-attention mechanisms to capture long-range dependencies in the data, which is particularly beneficial for identifying large-scale structures like salt domes in seismic images. Transformers' ability to model complex relationships and their flexibility in handling various input sizes and shapes provide a significant advantage over traditional CNN-based approaches.

Generative models and semi-supervised learning techniques were also explored as part of our comprehensive approach. These methods offer solutions to some of the inherent chal-

lenges in seismic image analysis, such as the scarcity of labeled training data. By generating synthetic data or utilizing unlabeled data more effectively, these techniques enhance model robustness and improve segmentation accuracy.

Our comparative study revealed that while each model has its strengths, the choice of the best model often depends on the specific requirements of the task at hand. CNNs provide a solid foundation for feature extraction, U-Nets excel in tasks requiring precise boundary delineation, and Transformer-based models are particularly effective in capturing global context and long-range dependencies. Generative models and semi-supervised approaches offer valuable enhancements in scenarios with limited labeled data.

In conclusion, the integration of these advanced deep learning models into the process of seismic image analysis marks a significant step forward in the field of geophysical exploration. By automating and improving the accuracy of salt dome identification, these technologies not only enhance the efficiency of hydrocarbon exploration but also pave the way for more innovative applications in earth sciences. Future research could focus on hybrid models that combine the strengths of different architectures or explore new deep learning paradigms to further advance the capabilities of seismic image segmentation.

# Bibliography


[3] N. Mondol and K. Bjørlykke. “Seismic Exploration”. en. In: (Sept. 2010), pp. 375–402.

[4] E. Salmazo, José Mendes, and Kazuo Miura. “THE INFLUENCE OF SALT DOMES IN DRILLING WELL ACTIVITIES”. In: *Brazilian Journal of Petroleum and Gas* 7 (June 2013), pp. 43–55. DOI: 10.5419/bjpg2013-0004.

[5] Jingjing Zong et al. “Salt densities and velocities with application to Gulf of Mexico salt domes”. In: 2015. URL: https://api.semanticscholar.org/CorpusID:149448165.

[6] Javier Abreu-Torres. “Salt Tectonic Imaging at Crustal and Experimental Scales by Seismic Migration and Adjoint Method: Offshore Application Context”. English. ffNNT: 2022TOU30130ff, fftel-03813706ff. PhD thesis. Université Paul Sabatier - Toulouse III, 2022.

[8] Haibin Di and Ghassan Alregib. “Seismic Multi-attribute Classification for Salt Boundary Detection - A Comparison”. In: June 2017. DOI: 10.3997/2214-4609.201700919.

[9] Haritha Thilakarathne. *Deep Learning vs. Traditional Computer Vision*. https://naadispeaks.wordpress.com/2018/08/12/deep-learning-vs-traditional-computer-vision/. 2018.

[10] Ravula Samatha, Rani, and Pole Laxmi Devi. “A Literature Survey on Computer Vision Towards Data Science”. In: 2020. URL: https://api.semanticscholar.org/CorpusID:247382624.

[11] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. “A survey on instance segmentation: state of the art”. In: *International Journal of Multimedia Information Retrieval* 9.3 (July 2020), pp. 171–189. ISSN: 2192-662X. DOI: 10.1007/s13735-020-00195-x. URL: http://dx.doi.org/10.1007/s13735-020-00195-x.

[12] R. Naqvi, D. Hussain, and W. Loh. “Artificial intelligence-based semantic segmentation of ocular regions for biometrics and healthcare applications”. In: *Computers, Materials & Continua* 66.1 (2021), pp. 715–732.

[13] X. Tang et al. “DFFNet: An IoT-perceptive dual feature fusion network for general real-time semantic segmentation”. In: *Information Sciences* 565 (2021), pp. 326–343.

[14] T. Leonardo et al. “Real-time deep learning semantic segmentation during intra-operative surgery for 3D augmented reality assistance”. In: *International Journal of Computer Assisted Radiology and Surgery* 16.9 (2021), pp. 1435–1445.

[15] S. Nedevschi. “Weakly supervised semantic segmentation learning on UAV video sequences”. In: *Proc. of 29th European Signal Processing Conf. (EUSIPCO)*. Dublin, Ireland, 2021, pp. 731–735.

[16] N. Kühl et al. “Artificial intelligence and machine learning”. In: *Electronic Markets* 32 (2022), pp. 2235–2244. DOI: 10.1007/s12525-022-00598-0.

[17] Corien Prins, Haroon Sheikh, and Erik Schrijvers. *Mission AI: The new system technology*. English. Research for Policy. Germany: Springer, Jan. 2023. ISBN: 978-3-031-21447-9. DOI: 10.1007/978-3-031-21448-6.

[18] Jürgen Schmidhuber. “Deep Learning”. In: Jan. 2017, pp. 338–348. DOI: 10.1007/978-1-4899-7687-1_909.

[19] Ilya Aizenberg, Natalia N. Aizenberg, and Joos P. Vandewalle. *Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications*. Springer Science & Business Media, 2013.

[20] Christian Janiesch, Patrick Zschech, and Kai Heinrich. “Machine learning and deep learning”. In: *Electronic Markets* 31.3 (Sept. 2021), pp. 685–695.

[21] Iqbal H. Sarker. “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions”. In: *SN Computer Science* 2.420 (2021). DOI: 10.1007/s42979-021-00815-1. URL: https://doi.org/10.1007/s42979-021-00815-1.

[22] Laith Alzubaidi et al. “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions”. In: *Journal of Big Data* 8.1 (2021). Epub 2021 Mar 31, p. 53. DOI: 10.1186/s40537-021-00444-8.

[23] Moez Krichen. “Convolutional Neural Networks: A Survey”. In: *Computers* 12.8 (2023). ISSN: 2073-431X. DOI: 10.3390/computers12080151. URL: https://www.mdpi.com/2073-431X/12/8/151.

[24] Alberto Garcia-Garcia et al. “A Review on Deep Learning Techniques Applied to Semantic Segmentation”. In: *CoRR* abs/1704.06857 (2017). arXiv: 1704.06857. URL: http://arxiv.org/abs/1704.06857.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Neural Information Processing Systems* 25 (Jan. 2012). DOI: 10.1145/3065386.

[26] Iqra Kiyani et al. “Deep learning based Glaucoma Network Classification (GNC) using retinal images”. In: *International Journal of Imaging Systems and Technology* 34 (Dec. 2023), n/a–n/a. DOI: 10.1002/ima.23003.

[28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *CoRR* abs/1411.4038 (2014). arXiv: 1411.4038. URL: http://arxiv.org/abs/1411.4038.

[30] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. "Learning Deconvolution Network for Semantic Segmentation". In: *ArXiv* (May 2015). DOI: 10.1109/ICCV.2015.178.

[31] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2015, pp. 234–241.

[32] Manisha Agarwal, Shashi Kumar Gupta, and Krishnendu Kumar Biswas. "Plant Leaf Disease Segmentation Using Compressed UNet Architecture". In: *International Conference on Advanced Machine Learning Technologies and Applications*. Springer. 2021, pp. 14–23.

[33] Zhiyong Fan et al. "JAUNet: A U-Shape Network with Jump Attention for Semantic Segmentation of Road Scenes". In: *Applied Sciences* 13.3 (2023). ISSN: 2076-3417. DOI: 10.3390/app13031493. URL: https://www.mdpi.com/2076-3417/13/3/1493.

[34] N. Subraja and D. Venkatasekhar. "Satellite Image Segmentation using Modified U-Net Convolutional Networks". In: *2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*. 2022, pp. 1706–1713. DOI: 10.1109/ICSCDS53736.2022.9760787.

[35] Abhishek Chaurasia and Eugenio Culurciello. "LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation". In: *CoRR* abs/1707.03718 (2017). arXiv: 1707.03718. URL: http://arxiv.org/abs/1707.03718.

[36] Tsung-Yi Lin et al. "Feature Pyramid Networks for Object Detection". In: *CoRR* abs/1612.03144 (2016). arXiv: 1612.03144. URL: http://arxiv.org/abs/1612.03144.

[37] Hengshuang Zhao et al. "Pyramid Scene Parsing Network". In: *CoRR* abs/1612.01105 (2016). arXiv: 1612.01105. URL: http://arxiv.org/abs/1612.01105.

[38] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in neural information processing systems* 30 (2017).

[39] Hang Yan et al. "TENER: adapting transformer encoder for named entity recognition". In: *arXiv preprint arXiv:1911.04474* (2019).

[40] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 5485–5551.

[41] Mark Chen et al. "Generative Pretraining from Pixels". In: *International Conference on Machine Learning*. See page 9. PMLR. 2020, pp. 1691–1703.

[42] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning Internal Representations by Error Propagation*. Tech. rep. ICS 8504. San Diego, California: Institute for Cognitive Science, University of California, Sept. 1985.

[43] Tianyang Lin et al. "A survey of transformers". In: *AI Open* 3 (2022), pp. 111–132. ISSN: 2666-6510. DOI: https://doi.org/10.1016/j.aiopen.2022.10.001. URL: https://www.sciencedirect.com/science/article/pii/S2666651022000146.

[44] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[45] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. "Layer normalization". In: *arXiv preprint arXiv:1607.06450* (2016).

[46] Mike Lewis et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension". In: *arXiv preprint arXiv:1910.13461* (2019). URL: https://arxiv.org/abs/1910.13461.

[47] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[48] Yinhan Liu et al. "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[49] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[50] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021.

[51] Maithra Raghu et al. "Do Vision Transformers See Like Convolutional Neural Networks?" In: (2022). arXiv: 2108.08810 [cs.CV]. URL: https://arxiv.org/abs/2108.08810.

[52] Hongyi Wang et al. "Mixed Transformer U-Net For Medical Image Segmentation". In: (2021). arXiv: 2111.04734 [eess.IV]. URL: https://arxiv.org/abs/2111.04734.

[53] Guipeng Lan et al. "Active learning inspired method in generative models". In: *Expert Systems with Applications* 249 (2024), p. 123582. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2024.123582. URL: https://www.sciencedirect.com/science/article/pii/S0957417424004470.

[54] Pengzhi Li, Yan Pei, and Jianqiang Li. "A comprehensive survey on design and application of autoencoder in deep learning". In: *Applied Soft Computing* 138 (2023), p. 110176. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2023.110176. URL: https://www.sciencedirect.com/science/article/pii/S1568494623001941.

[56] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *CoRR* abs/1312.6114 (2013).

[57] C. Winkler et al. “Learning Likelihoods with Conditional Normalizing Flows”. In: *ArXiv* abs/1912.00042 (2019).

[58] Diederik P. Kingma and Prafulla Dhariwal. “Glow: Generative Flow with Invertible 1x1 Convolutions”. In: *NeurIPS*. 2018.

[59] Simon Boehm. “The Normalizing Flow Network”. In: (Aug. 2019). URL: https://siboehm.com/articles/19/normalizing-flow-network.

[60] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, eds. *Semi-supervised Learning*. Adaptive Computation and Machine Learning. Library of Congress Cataloging-in-Publication Data. City: The MIT Press, 2006. P. ISBN: 978-0-262-03358-9.

[61] L. Li et al. “Semi-Supervised Remote Sensing Image Semantic Segmentation Method Based on Deep Learning”. In: *Electronics* 12.2 (2023), p. 348. DOI: 10.3390/electronics1202 URL: https://doi.org/10.3390/electronics12020348.

[62] Yauhen Babakhin, Artsiom Sanakoyeu, and Hirotoshi Kitamura. “Semi-Supervised Segmentation of Salt Bodies in Seismic Images using an Ensemble of Convolutional Neural Networks”. In: (2019). arXiv: 1904.04445 [cs.CV]. URL: https://arxiv.org/abs/1904.04445.

[63] Luis Felipe Henriques et al. “Generating Data Augmentation samples for Semantic Segmentation of Salt Bodies in a Synthetic Seismic Image Dataset”. In: (2021). arXiv: 2106.08269 [cs.CV]. URL: https://arxiv.org/abs/2106.08269.

[64] Shijie Hao, Yuan Zhou, and Yanrong Guo. “A Brief Survey on Semantic Segmentation with Deep Learning”. In: *Neurocomputing* 406 (2020), pp. 302–321. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.11.118. URL: https://www.sciencedirect.com/science/article/pii/S0925231220305476.

[65] Mustafa Alfarhan, Mohamed Deriche, and Ahmed Maalej. “Robust Concurrent Detection of Salt Domes and Faults in Seismic Surveys Using an Improved UNet Architecture”. In: *IEEE Access* 10 (2022), pp. 39424–39435. DOI: 10.1109/ACCESS.2020.3043973.

[66] Saif ul Islam. “Using deep learning based methods to classify salt bodies in seismic images”. In: *Journal of Applied Geophysics* 178 (May 2020), p. 104054. DOI: 10.1016/j.jappgeo.2020.104054.

[67] Aleksandar Milosavljević. “Identification of Salt Deposits on Seismic Images Using Deep Learning Method for Semantic Segmentation”. In: *ISPRS International Journal of Geo-Information* 9.1 (2020), p. 24. DOI: 10.3390/ijgi9010024.

[68] Letícia Bomfim et al. “Transformer Model for Fault Detection from Brazilian Pre-salt Seismic Data”. In: Oct. 2023, pp. 3–17. ISBN: 978-3-031-45388-5. DOI: 10.1007/978-3-031-45389-2_1.

[69] Mikhail Karchevskiy, Insaf Ashrapov, and Leonid Kozinkin. “Automatic salt deposits segmentation: A deep learning approach”. In: (2018). arXiv: 1812.01429 [cs.LG]. URL: https://arxiv.org/abs/1812.01429.

[70] Mrinmoy Sarkar. “Salt Detection Using Segmentation of Seismic Image”. In: *Journal Name* Volume Number.Issue Number (2022), Page Range. DOI: DOI. URL: https://arxiv.org/abs/2203.13721.

# Webography


[1] Encyclopaedia Britannica. *Salt dome*. URL: https://www.britannica.com/science/salt-dome (visited on 04/02/2024).

[2] *What is a Salt Dome?* URL: https://shorturl.at/uvX78 (visited on 04/02/2024).

[7] Lameez Omarjee. *Seismic Surveys: Scientists Call for Tighter Laws, Greater Oversight*. Jan. 2022. URL: https://www.news24.com/fin24/economy/seismic-surveys-scientists-call-for-tighter-laws-greater-oversight-20220114 (visited on 03/29/2024).

[27] Heuritech. *A Brief Report of the Heuritech Deep Learning Meetup #5*. Feb. 2016. URL: https://blog.heuritech.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/ (visited on 03/29/2024).

[29] Fei-Fei Li, Andrej Karpathy, and Justin Johnson. *CS231N: Convolutional Neural Networks for Visual Recognition*. Stanford University. 2017. URL: https://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf (visited on 03/29/2024).

[55] Arthur Meyer. *Saliency Detection Convolutional Autoencoder*. URL: https://github.com/arthurmeyer/Saliency_Detection_Convolutional_Autoencoder (visited on 04/05/2024).