CHAPTER 1

INTRODUCTION TO OIL SPILL DETECTION USING DEEP LEARNING

The oil spill is a serious environmental issue with the potential to harm marine environments and coastal economies. Traditional oil spill detection methods (remote sensing based) are typically affected by their inability to deal with changing weather and ocean conditions. Deep learning has demonstrated effective improvements in the accuracy of oil spill detection based on semantic segmentation. In this paper, we compare and test three of the most widely used deep learning structures, U-Net, DeepLabV3, and PSPNet, on a publicly shared oil spill dataset. We desire to compare their accuracy, inference time, and real-time deployability. The novelty of this study lies in the end-to-end comparison of the performance of UNet, DeepLabV3, and PSPNet for the detection of oil spills from satellite images. This study differs from existing studies with a single architecture or employing synthetic datasets. The study compares the models based on diverse real-world scenarios and employs a public dataset. Comparative analysis enables the identification of trade-offs between accuracy and computational time and offers insights into model suitability for practical deployment.Oil spills present a serious hazard to the environment, causing large-scale ecological and economic losses to marine and coastal ecosystems and, thus, disturbing both human activities and biodiversity. Such events normally occur as a consequence of accidents during oil extraction, transportation, or storage operations and, in turn, potentially produce long-term ecological and economic impacts. Large previous events of major oil spills, like the 2010 Deepwater Horizon incident, have shown the long-term impacts of large oil release on the environment, resulting in large-scale contamination and expensive remediation. Detection of oil spills is essential in an effort to reduce damage and effectively coordinate response activities.

Spill detection effectively is critical to the success of mitigation and environmental conservation. Past oil spill detection practices rely on remote sensing technology such as SAR and optical images, but not all the techniques are precise when the weather and the ocean surface is constantly changing.

Advances in deep learning have enabled more powerful oil spill detection with CNNs used in classifying and segmenting oil spills from satellite images more accurately. Studies such as Guo et al. (2020) [6] have shown how CNN based models can identify oil spills from polarimetric SAR data, telling the future in preventing oil spills.

Deep learning has shown potential when it comes to image segmentation. The use of CNNs have changed the way satellite images are processed and oil spills are detected. Compared to conventional techniques, CNN based models have shown learning capability for spatial features on their own and respond differently to different spill-types. The performance of the model depends on the architecture and its ability to identify sophisticated spatial features.

This study explores the performance of the three deep learning model which are U - Net, Deeplab V3, and PSPNet to find oil spills through the use of satellite imagery. Every model approaches picture segmentation issues differently.

Because of its encoder-decoder architecture, U-Net produces results more quickly and with more accuracy. DeeplabV3 processes contextual information at various scales using the atrous spatial pyramid pooling (ASPP).

In a similar vein, PSPNet employs a pyramid pooling module that combines local and global characteristics to improve accuracy.

We have utilised performance indicators like accuracy, F1-score, and IoU (Intersection over Union) to compare all of these models to one another. We used a Kaggle dataset that is openly accessible to train the models.

When compared to other models, recent advancements in this sector have demonstrated how well models like DeeplabV3+ have fared in segmentation tasks. However, as Krestenitis et al. [2] demonstrate, it necessitates high computational constraints.

1.1 SCOPE

The use of deep learning architectures, specifically U-Net, DeepLabV3, and PSPNet, for satellite imagery-based oil spill detection is the focus of this work. A thorough evaluation of these architectures' segmentation quality, inference time, and durability under various air circumstances is the aim of this study. To guarantee a realistic assessment, the dataset used is made up of actual satellite photos that have been tagged with ground truth masks.

Major aspects are:

- U-Net, DeepLabV3, and PSPNet are trained and used for oil and non-oil binary segmentation.
- Performance metric-based evaluation involving Accuracy, F1-score, and Intersection over Union (IoU).
- Quantitative and qualitative comparison of the segmentation outputs.
- Investigation into optimization methods and possible model additions such as attention mechanisms.

The following is not included in this study:

- Use of non-satellite data (e.g., sonar or drone) for oil spill detection.
- Hardware optimization or real-time deployment.
- Multiclass segmentation or severity classification of oil spills.

1.2 MOTIVATION

There is an increasing and intensified landslide event in the world which manifests climate change, urbanization, and environmental degradation. Such landslide events may cause tremendous loss of life and serious damage to infrastructure while posing adverse impacts on local ecosystems.

Hence, there is a pressing need for timely detection of such landslide events, as well as response. The traditional landslide detection techniques are through manual surveys and ground-based monitoring, which are limited in coverage and resource-intensive. These techniques are so time-consuming and usually ineffective in remote or inaccessible areas, which are where landslides are most likely to occur. Therefore, there is an imperative call for innovative solutions to provide timely and accurate information about landslide activity over extensive geographic regions.

This kind of source to monitor the event satellite imagery offers is, to some extent, a promise in that it makes landslides observable using much more coverage and allowing potential records of changes occurring during various moments of time. What happens by combining high resolution in data from satellites together with new advancements in the algorithm machine learning is really offering great possibilities for even establishing the automated detection of huge amounts of data under process efficiently. It is possible to analyze complex patterns in satellite images by harnessing the power of deep learning algorithms and distinguishing between landslide and non-landslide features more accurately than traditional methods. This technological shift opens new opportunities for enhancing early warning systems, which are critical for disaster preparedness and response. This would be a proposed system in which the landslide risk and damage can be decreased because alerts and warnings would reach the communities at high-risk situations on time.

Furthermore, with the ease of access of satellite data and machine learning techniques and methods which are rapidly spreading, there comes one of the unique opportunities through which the gap between getting data and turning that information into actionable insights can bridge. This project will seize the opportunity by developing a system that will be useful to various stakeholders, such as government agencies, researchers, and local communities. It is in the hope of arming stakeholders with information to inform decisions and proactive actions when landslide events are anticipated that this project seeks to provide a dependable landslide detection tool.

This system, other than for natural disaster monitoring purposes, can be adopted and adapted for other objectives like in the case of an influx or earthquake impact analysis. Thus, methodologies that the project has developed within contribute to a holistic approach of disaster risk reduction for developing resilience in more vulnerable areas. The motivation is not limited to overcome the immediate crisis of landslides but also to have a better understanding of how satellite imagery and machine learning can transform disaster management practices.

1.3 SYSTEM REQUIREMENTS

1.3.1 Hardware Requirements

For this project, the requirements in terms of hardware are at a minimum because Google Colab provides not only powerful GPUs like the NVIDIA Tesla K80 and T4 but also TPUs for particularly computationally intensive tasks. This will therefore exclude the need to high-spec local hardware. Moreover, Colab works wonderfully with Google Drive, meaning there would be sufficient cloud storage to handle large satellite datasets, thus truly keeping the requirement in local storage to minimum. That would require an internet of good strength to be able to communicate with Google Colab and synchronize data with Google Drive because the retrieval of data and model training are generally faster.

UsingColab for this project was actually a great idea. It's convenient and provides decent resources such as GPUs and TPUs. With a fast internet connection, all can be done easily. We can save all the data in Google Drive, so no need to worry about the memory of the computer being occupied. Additionally, it saves an enormous amount of time since all the data and models are readily available and training is quick. For one who lacks large hardware, this cloud-based arrangement is the ideal solution.

1.3.2 Software Requirements

The software requirements are also optimized as Google Colab has rich library support, making it perfect to use for deep learning and image processing. The environment supports Python 3.x and incorporates the major machine learning frameworks like TensorFlow and PyTorch that are directly related to the implementation and training of CNN models and attention mechanisms. Such image processing operations like resizing and augmenting the satellite images are available in OpenCV.

Data manipulation and handling large amounts of data can be done by using NumPy and Pandas. Easy model performance analysis can be done by visualization of data by Colab using Matplotlib and Seaborn. Geopandas has been used to handle geospatial data coming from geographic data to create a map to decide which of the areas to assert landslide detection. In fact, Google Colab requires minimal installation and offers an efficient, cost-effective solution to developing and deploying machine learning models.

CHAPTER 2
LITERATURE REVIEW

Emerging technologies in remote sensing and artificial intelligence have dramatically improved the precision and accuracy of oil spill detection in marine environments. Conventional techniques, though valuable, prove ineffective under adverse environmental conditions like cloud cover, high seas, or dim light. With the availability of deep learning and large amounts of annotated SAR and optical satellite imagery, segmentation performance has been enhanced, and false alarms have been minimized. This section highlights the main contributions in the area, but with an emphasis on methodological advances and dataset development that have influenced the present oil spill detection capacities.

Solberg et al. (2007) [1] investigated the possibilities of the usage of RADARSAT and ENVISAT SAR (Synthetic Aperture Radar) for detecting oil spills, and proved the high efficiency of the traditional remote sensing methods even under the harsh circumstances. This pioneering research formed the foundation for the development of the method of SAR-based spill detection.

Krestenitis et al. (2019) [2] proposed a standard dataset for deep learning methods for oil spill identification. The attention to strict annotations standards allowed to build trustworthy models and prove that careful data preparation has large positive impact on performance.

The ROSID dataset contributed in 2023 by Abdimanap et al. [3] is focused on onshore oil spill detection enhancement. They overcame hurdles in existence of available datasets and enhanced applicability of remote sensing over different terrains.

Through the use of a dual attention strategy in deep learning approaches Mahmund et al. (2022) [4] used SAR data efficiently. Using this approach, spill regions were prioritized, noise was minimized and segmentation accuracy was highly improved.

Li et al. (2018) [5] suggested that adaptive local thresholding and fuzzy classification can be used to detect oil spills, but the key feature would be based on the importance of spatial features and texture-based analysis within SAR image analysis. By incorporating their method, they realized improved differentiation between oil spills and look-alike events such as low wind patches and natural slicks.

Guo et al. (2020) [6] used CNNs for a polarimetric SAR data processing and showed that the accuracy with respect to standard detection methods is significantly improved. They highlighted that the incorporation of polarization data enhances the model's ability to discriminate oil spill characteristics and efficiently remove ail false positives caused by ocean surface noise.

Shaban et al 2021 [7] have presented a two stage deep learning method which is specifically designed to overcome the class imbalance problem in synthetic aperture radar (SAR) datasets. Good progress was made in the sensitivity of the model particularly to the minority (spill) classes.
The method accomplished higher detection results and high precision by effectively suppressing false positives from the most dominant background signal.

As reported by Fan et al. (2022) [8], their Feature Merge Network consolidates features in multiple SAR datasets, demonstrating strong robustness under varying environmental conditions. The combination of various SAR feature sources in their method enhanced their model capabilities to interpret not only the fine local feature details but also the global context, hence boosting generalization for diverse spilling shapes on variant marine condition settings.

Figure 1. Rousso et al. (2023) [9] obtained a superior result in the detection process through the combination of image filtering and deep networks to eliminate background interference and optimize feature identification. In complex background conditions, their approach demonstrated improvement by a significant margin with more consistent and understandable segmentation results.

In 2024, Ma et al. [10] utilized dual-polarimetric Sentinel-1 SAR data combined with deep convolutional networks in severe marine environment. In this procedure, it achieved outstanding performance in adverse sea states with low contrast and high noise, thereby enhancing the accuracy of oil spills greatly.

Zhang et al. (2023) [11] improved SAR image processing in Mask R-CNN and made the segmentation results more successful in challenging conditions. Engineering changes to the model facilitated a clearer resolution of spill and deceptive appearances, guaranteeing precise boundary determination and reduced spill area overestimation.

Ma et al. (2022) [12] unified polarimetric scattering signatures and deep convolutional neural networks to sharpen the discrimination of oil spills from false positives and improve overall system reliability.
The integration was based on scattering processes to refine feature sets thereby increasing classification trustworthiness and performance of the model against varying sea states.

In 2000, Del Frate et al. [13] showed how neural networks are used in processing of ERS-SAR in oil spill protection, underlining AI advantage over "classical" methods in noisy conditions. They opened the door for using tools of machine learning in SAR-based environmental surveillance and disprove the method reliability to strong marine settings and varying sea states.

De Kerf et al. (2024) [14] introduced a red-green-blue (RGB) image set acquired by a drone in port environments after oil spills, with better detection abilities than over the open-sea. This contribution made it possible to construct and assess model frameworks in complex urban maritime environments in which object discovery is hindered by visual interference and diversification in geographical aspects.

Satyanarayana and Dhali (2023) [15] explored encoder-decoder architectures and noted that more profound networks with detailed connections enhanced spill edge accuracy. Their study underscored the importance of multi-scale feature extraction in accurate demarcation of oil spill boundaries, particularly in the case of complicated or fragmented spills.

Challenges and Limitations in Existing Methods:

- Limited Generalization Across Diverse Environments:

  Existing deep learning models for oil spill detection often perform well in controlled settings but struggle with generalizing across different environmental conditions (e.g., varying lighting, water types, and oil spill characteristics). This limits their practical application in real-world scenarios.

- Data Imbalance and Labeling Issues:

  Oil spill datasets tend to be imbalanced, with more non-oil images than oil spill images. This class imbalance can result in biased models, where the model overpredicts the non-oil class. Moreover, the manual annotation of satellite or aerial images can introduce errors, further impacting model performance.

- Real-Time Detection Constraints:

  Although deep learning algorithms can be accurate, they come at the expense of high computation, which increases inference time significantly. This creates issues in applications where real-time detection is desirable for timely responses and oil spill mitigation.

- Deficiency of Aggressive Preprocessing Techniques:

  Preprocessing methods, like noise reduction or image enhancement, are usually basic or not tailored for a particular satellite or aerial image, which negatively affects model performance. The models also tend to lack suitable processing of image artifacts such as clouds, water shine, and inconsistent oil textures.

- Overfitting on Small Datasets

  Most models are taught with limited data, resulting in overfitting, particularly when dealing with complex, high-dimensional data. They perform poorly on unseen or less-representative test data.

- Integration of Contextual Information:

  Existing models mostly perform pixel-level classification without effectively incorporating contextual or temporal information, like oil spill trajectory or environmental conditions over time, which could enhance detection accuracy and offer more meaningful insights.

- Scalability and Real-World Application:
  Scaling models, particularly in remote regions with limited computational power, is a serious challenge. In addition, models will likely have difficulty with high-resolution satellite imagery because of the power required to process big-scale data.

Research Tries to Solve:

- Improved Generalization: With a hybrid model that is built using U-Net and DeepLabV3 but equipped with attention mechanisms, the research tries to enhance generalization across various environmental conditions.

- Balanced Datasets: Data augmentation and synthetic data creation techniques will be utilized to resolve class imbalance and enhance model strength.

- Real-Time Detection: Optimized model architecture and low-latency inference algorithms are suggested in order to allow for faster detection without compromising on accuracy.

- Advanced Preprocessing: In this research, more advanced preprocessing methods are planned to be used to process noise, artifacts, and variation in satellite imagery to enhance the performance of models.

- Temporal and Contextual Modeling: Temporal designs are made from the contents modelled from the oil spill.

CHAPTER 3

SPRINT PLANNING AND EXECUTION METHODOLOGY

The Oil Spill Detection project was developed following Agile methodology with emphasis on Scrum to enable iterative development, systematic planning, and continuous assessment. This facilitated efficient coordination, timely decision-making, and smooth integration of research components like model training, evaluation, and result visualization.

3.1 Adoption of Agile and Scrum:

Scrum served as the implementation framework for the Agile approach that guided the development of the oil spill detection system. Through frequent feedback and flexible planning, the team was able to manage research uncertainties because to Agile's iterative nature. By dividing the tasks into sprints, Scrum made it possible to maintain a laser-like concentration, set quantifiable goals, and continuously enhance model performance and data processing. This method worked well for balancing flexibility with the thoroughness of the investigation.
Scrum served as the implementation approach, while Agile development guided the creation of the oil spill detection system. Through continuous feedback and adaptable planning, the Agile approach's iterative nature assisted in managing the unpredictability of the study. By dividing work into sprints, Scrum made it possible to implement a strategy that worked, set measurable goals, and continuously enhance data processing and model performance. This strategy worked well to maintain both the necessary flexibility and the calibre of the study. Each sprint was centred around particular deliverables, such performance analysis, data preprocessing enhancements, or model architecture development.

Prior to each sprint, the team prioritised activities and set goals based on learning from past iterations and current progress. This offered a structure that was adaptable and adaptive, allowing the project's goal or scope to vary based on the outcomes of each sprint. Scrum's flexibility makes it simple to handle unforeseen challenges, including low data quality or underwhelming model performance, without compromising the project's overall deadline.

Long-term member engagement was made easier by the use of Scrum rituals including Daily Stand-ups, Sprint Reviews, and Sprint Retrospectives. While sprint reviews made it easier to show off completed work and collect comments, daily stand-ups were specifically tasked with guaranteeing continued development. Sprint retrospectives, which take place at the end of each sprint, gave the team a chance to evaluate techniques that had been effectively adopted and pinpoint areas that needed improvement. Thus, the feedback loop was essential to maintaining team morale and assuring development over the course of the project. Furthermore, the Agile methodology's emphasis on teamwork made sure that the development and research teams worked closely together. Ongoing stakeholder feedback from sources like satellite imagery, oil spill response, and machine learning specialists kept the project anchored to real-world constraints. The collaborative environment also facilitated rapid prototyping, enabling swift testing and improvement of different model architectures (U-Net, DeepLabV3, PSPNet) against immediate feedback. The method not only improved the quality of the system but also facilitated the acceleration of the development cycle.

By implementing Scrum, the project was also able to strike a balance between detailed, high-quality research and the flexibility to incorporate new findings or challenges. The iterative nature of this process enabled constant refinement of the detection system to the point where a more effective, efficient, and dependable model for oil spill detection was developed. The end result is a reliable oil spill monitoring solution that can adapt to new problems in the field thanks to the Agile platform's ability to respond to a variety of scenarios.

3.2 Sprint Planning

Sprint Planning occurred on a weekly basis for task breakdown, effort estimation, and short-term objectives. We also explored hybrid models and attention mechanisms in subsequent sprints. Daily standups helped us to stay on track and remove blockers fast. This sustained continuity of effort through the phases of development.

As we continued to progress, the team pivoted to continue improving the models. We began tacking on new parts to our models, such as building ensembles of models or adding ornamental features designed to help the system to focus more on the oil spills and less on everything else. These new tasks were a bit trickier, as we had to ensure everything still operated smoothly and did not slow the system down. We tested how well these new changes were working, made adjustments as necessary to ensure that the models continued to improve.

We used to have Daily Stand-up meetings every day where we just ticked off what people are working on and if they're encountering any issues. It in turn helped the team stay focused and solve problems speedily. If someone got stuck like a model just didn't work as expected, or things weren't coming together properly we could troubleshoot it together in person during the meeting. That way we would keep it moving and nobody would get stuck. The meetings kept me (and I think, all of us) focused on the most important things we had to do, and we were able to support one another where it was required.

At the end of each week, we held a meeting called the Sprint Retrospective, where the team talked about what worked well and what didn't. We talked about what we would do better next time, in particular how the next sprint should go even better. This helped us to continue to improve the way we worked together. By breaking the project down into shorter steps, and noticing how we were doing throughout, we ensured we'd continue moving ahead and not "forget to see the forest for the trees.

3.3 Tools and Workflow Management

A suite of tools was embedded to automate and streamline the process of development workflow. other option<|system|>Write that in human language. Git was used for version control; while Trello (or Google Sheets) was used to manage tasks.

For drawing diagrams, Draw.io was used; VS Code was used for local debugging and testing. As a result, these tools allowed the team to work together, organize and accomplish all tasks successfully. They also simplified the movement between research and implementation without a glitch. They facilitated the harmonious coexistence of researchers and developers.

However complicated the deep learning pipelines are, the team succeeded in teamworking because of these tools. The access to highly advanced computation through cloud environment of Google Colab was made much simpler, thereby decreasing the dependence on locally administered hardware resources. Due to Git, version control of models became trivial, enabling safe testing and the rollback to previous configuration states as required.

The tools for task management, such as Google Sheets or Trello, proved to be a valuable asset during the tracking of the development stages which allowed the team to prioritize when on time and when on delays. Draw.io helped to create a simplified image of the system diagram, a process of workflow, and an architectural layout, all easily communicable and ideal for refining the project implementation. Overall, the use of these tools enhanced understanding, reduced misunderstanding, and facilitated consistent progress throughout the research.

3.4 Tools and Workflow Management

Applying Agile in sprint-type delivery had definite advantages. The team was able to iterate quickly, correct errors, and improve models from feedback. Sprint reviews and retrospectives noted that the process adapted as the project needs changed, and goal-oriented planning avoided scope creep. Consequently, research depth and speed of execution both increased markedly.

Our project is based on an Agile Scrum approach, which encourages iterative development, frequent feedback, and adaptive planning. Each sprint is planned for two weeks with clearly defined deliverables.Consequently, research depth and speed of execution both increased markedly. The process involves:

- Sprint Planning: Set sprint objectives and decompose high-priority product backlog items into smaller user stories.
- Daily Stand-Ups: 15-minute daily meetings to monitor progress, resolve blockers, and synchronize team efforts.
- Sprint Review: Showcase finished features to stakeholders for validation and feedback.
- Sprint Retrospective: Review what worked, what didn't, and areas of improvement for upcoming sprints.

CHAPTER 4
SYSTEM DESIGN

4.1 USE-CASE DIAGRAM

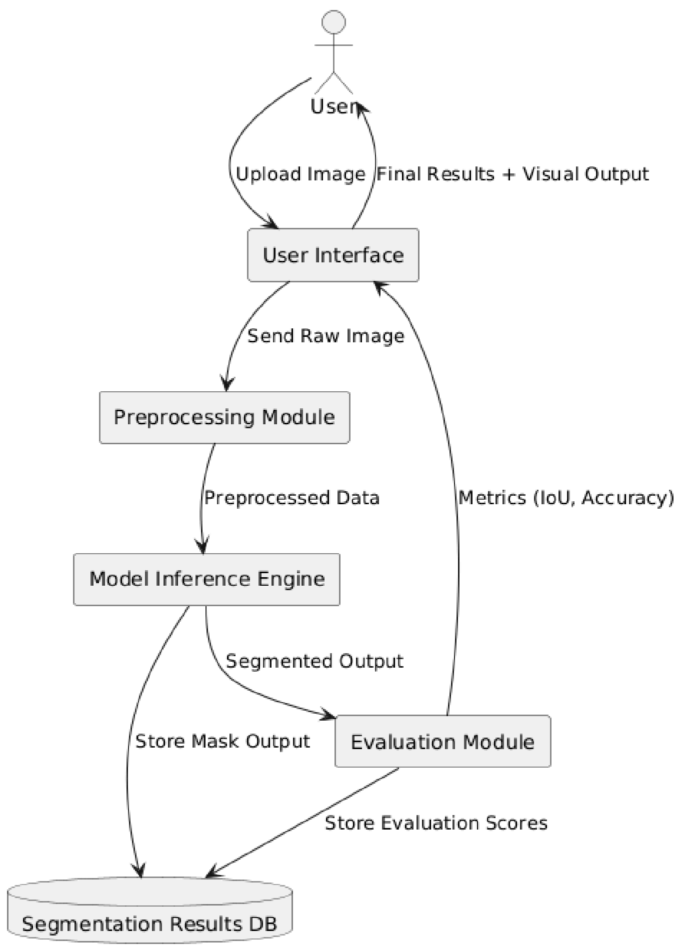


Figure 4.1: Use-Case Diagram for Oil-Spill Detection System

Figure 3.1 shows the core functionality offered by the oil spill detection system, and the 'User' is the core 'actor' that works along with the system. The user begins by uploading satellite images, which are processed to prepare them for model inference. Users can select either U-Net or DeepLabV3, among other segmentation models to analyse the preprocessed data.

Model selection allows users the flexibility to compare results against each other, allowing for strong comparison of research and performance. Once the simulation is finished, users get to view segmented output to reveal potential areas of oil spills.

Furthermore, the system also assesses accuracy by using automated calculation of metrics such as IoU (Intersection over Union) and F1-score as informative benchmarks for model performance. Finally, consumers are able to download visual findings and quantitative indicators. This integrated approach to interaction provides for both friendly function nature as well as the process of a strict scientific approach appropriate for environmental monitoring and research.

4.2 Sequence Diagram

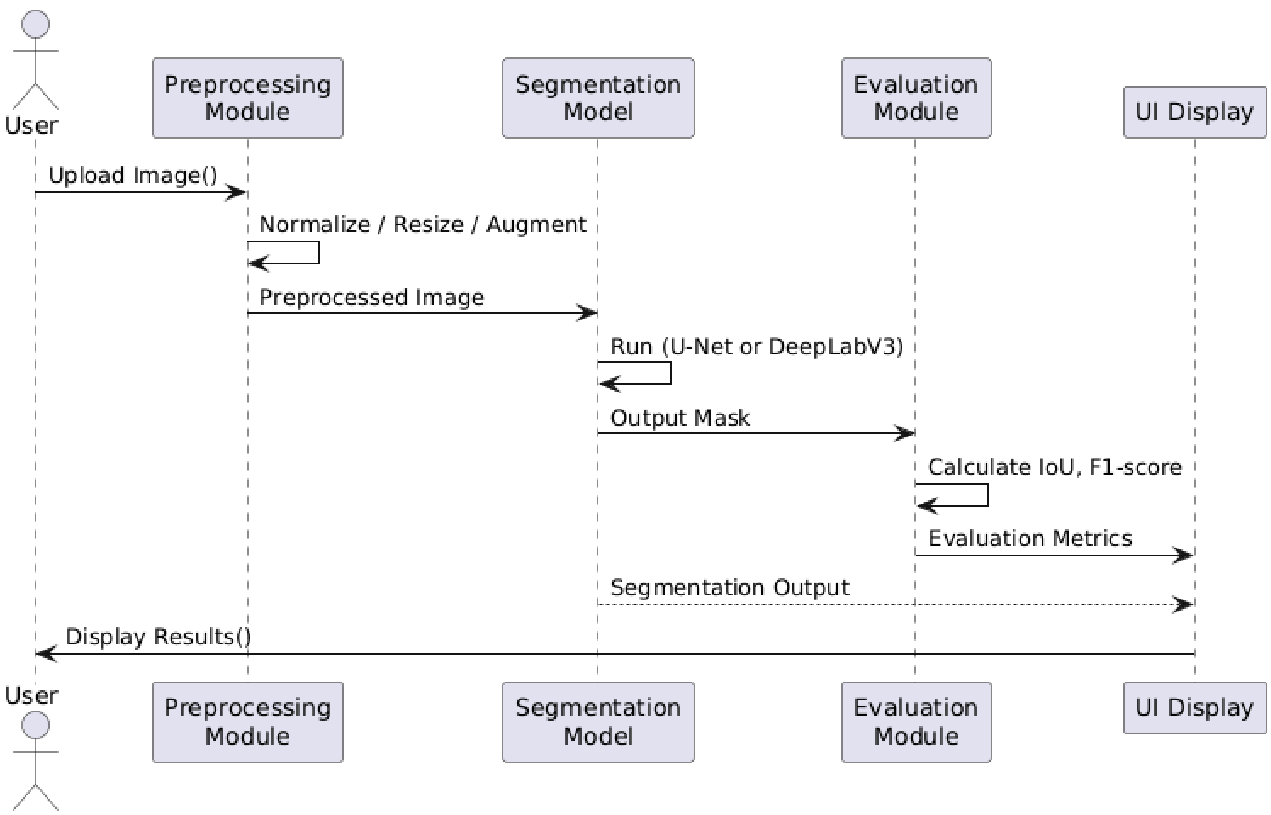


Figure 4.2: Sequence Diagram for Oil Spill Detection

From Figure 3.2, the different modules in the oil spill detection system function in a dynamic process. The process starts when the User sends an image to the Preprocessing Module which

preprocesses data in normalization, resizing and data augmentation. After conditioning the image, it is passed to Segmentation Model where inference happens by models.
utilizing application of U-Net or DeepLabV3 applications for generating segmentation covering delineation of oil spills. After the output generation, it is intercepted in the Evaluation Module and key performance parameters of the model (precision, recall, F1-score, and IoU) are computed.

After parsing the segmentation and evaluation metrics, they are presented using UI Display for an easily interpretable format for users. Such an easy and modular approach guarantees clear process transparency and reproducibility of the research, which is crucial for scientific research presentations.

4.3 Data Flow Diagram

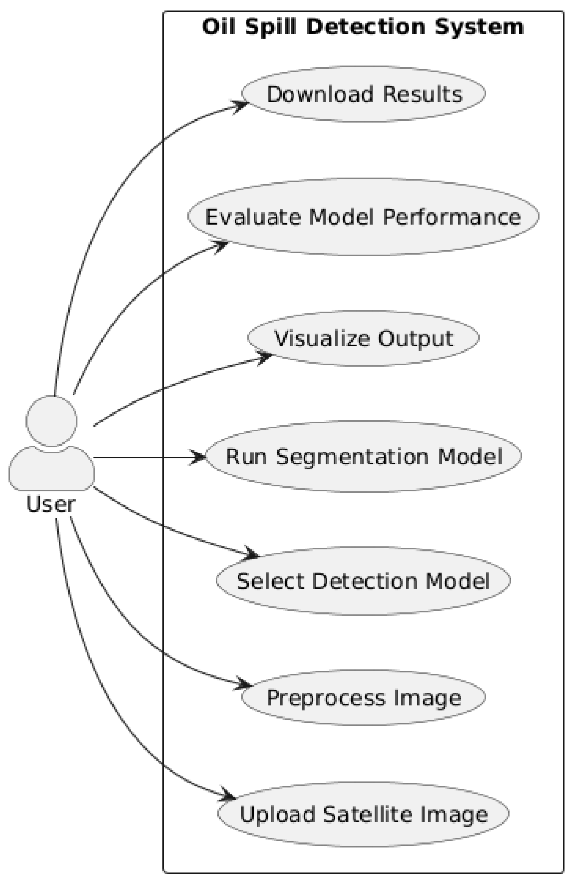


Figure 4.3: Data Flow Diagram

The data flow diagram (DFD) shows at a high level how data paths travel through the system as the data gradually moves. The user uses the User Interface to initiate the process by uploading a first raw satellite image. Following receipt, the input image, is channeled through the Preprocessing Module where standardization is carried out to aid analysis.

If processed, the image is to be communicated to Model Inference Engine, where it is segmented according to the chosen deep learning model. The segmentation result is handled by both Evaluation Module and a Results Database at the same time. The performance metric is within the Evaluation Module computed and transmitted with the generated segmentation result to the user interface. Meaningfully, output mask and the computed performance metrics are stored in a database for documentation and supporting long-term analysis. The modular design allows for visibility, data-driven processes to unfold, and the inclusion of new tools, or improvements, in future versions of the project.

CHAPTER 5

COMPARATIVE ANALYSIS OF DEEP LEARNING MODELS

Each deep learning model for oil spill detection makes certain strengths and certain weakness potentials. The U-Net architecture, a form of convolutional neural network (CNN), originally tailored to address image segmentation tasks, has become a popular choice, because of its encoder-decoder structure's capacity to capture spatial information efficiently. Consequently, U-Net excels for oil spill detection in terms of its unique capability to filter out spill boundaries from background disruptions. This model enjoys a symmetric architecture that is enhanced by skip connections, supporting better localization and strict segmentation. The model might remain unable to transfer complicated patterns among various oil spill pictures, which could be a reason for poor results compared to more up-to-date advance models.

However, DeepLabV3 achieving atrous convolution and multi-scale context shows a better result for segmentation with a high-resolution image. By using dilated convolutions to provide a broader context picture, the model is capable of correctly identifying significant regions of oil spills more precisely in the presence of complex textures and varying background environments. Its ability to identify and segment objects at different scales of operation makes DeepLabV3 stand out on realistic applications where an oil spill may take an unpredictable shape and size. Still, DeepLabV3 calls for more complex computational effort and resources to be used in both training and inference stages compared to simple models such as U-Net.

With a pyramid pooling module, PSPNet (Pyramid Scene Parsing Network) is engineered to retain global context thereby enhancing its better performance when detecting oil spills in either sprawling or cluttered scenes. The pooling pull over different scales helps the model to better match the relations among objects both fine-grained and coarse-financed. PSPNet has demonstrated viable results, outperforming classical models, especially on challenging tasks, due to detecting context representation for larger parts of scene. PSPNet, even though it includes better segmentation performance, like DeepLabV3 requires heavy computational resources and slower real-time execution, which is not commendable for system with strict temporal requirements.

Building on the use of U-Net, DeepLabV3, and PSPNet, it is a promising strategy to combat the oil spill detection problems. A hybrid model combines the strengths of U-Net for accurate localization, DeepLabV3 on multi-scales of context, and the PSPNet for capturing a broader scene understanding and performs well in a variety of oil spill detection applications. Allowing the joint application of those models creates difficulties as well, and specifically, ones related to the tuning of those models and the complexity during training and prediction processes. While this method does contain an element of complexity to it, the gain promise of much improved detection accuracy is one which points to an enormous potential for future study and application to detect oil spills.

Applications of deep learning models highly benefit from data augmentation for improved oil spills detection. Transformation of the satellite images using techniques such as rotation, invert and brightness adjustment allows the model to adjust and outperform in alien conditions. By applying this technique the model is trained to identify patterns on a larger scale, and thereby better able to detect oil spills in various environments. However, this approach requires greater requirements from computer hardware and increased types of training cycle, as well as extra effort for the analysis of effectiveness of performance of the model. More computational work is worth it as the model's performance improves when put into actual field conditions. A model which will distinguish oil spills even in dense cloud cover or murky waters is much more useful in the field. By improving our methods for data augmentation, it is possible to construct oil spill detection systems that are stronger, more versatile and more accurate regardless of underlying conditions. Such enhancements could provide for more rapid cleanups, and therefore result in more effective environmental protection.

5.1 DATASET OVERVIEW

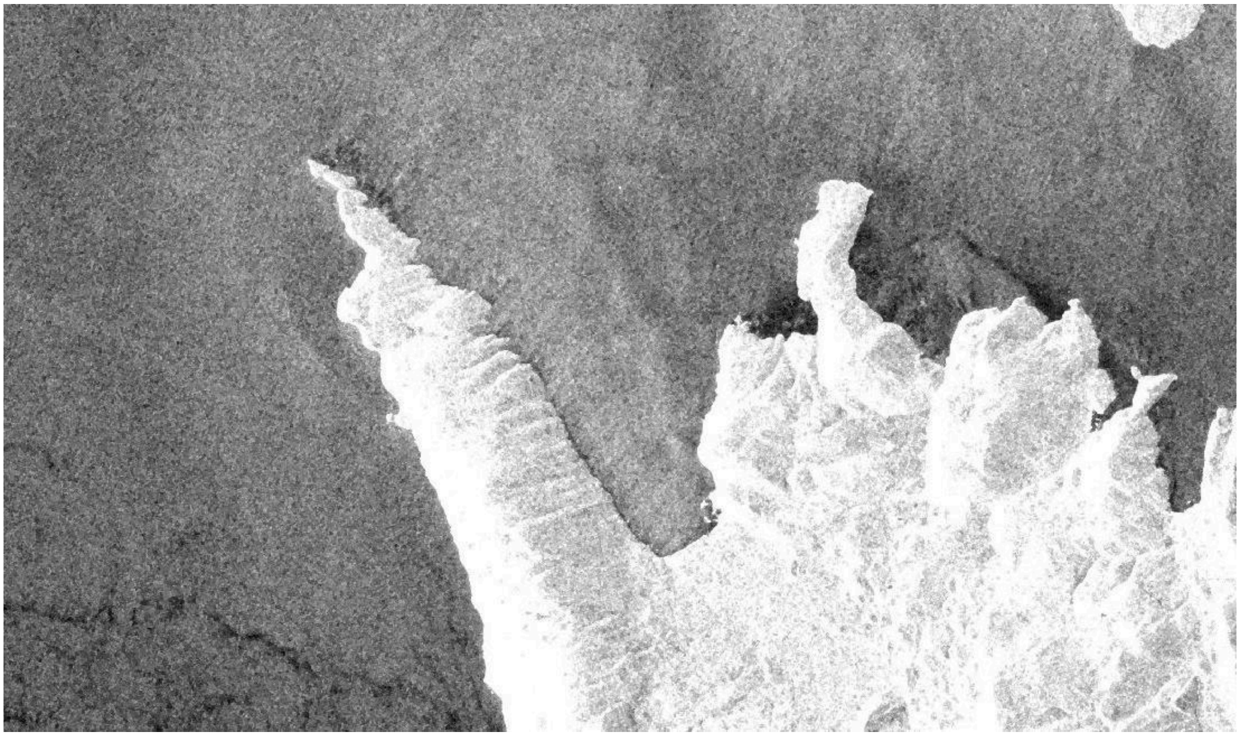


Fig 5.1 Oil Spill Dataset

The source of the collection of training images was the Kaggle dataset named, “oil spill”. These images have both high resolution and are varied in their weather and positioning, making the dataset immensely useful for deep model training. It is possible to see an illustrative example of the dataset in Figure 5.1. With each image, we provide a ground truth segmentation mask highlighting the areas affected by oil spill, an important attribute for training models.

Its diverse spill size and nature levels make the dataset both complex and expansive in the model’s learning flow. The variety of circumstances involves the models into learning and becoming acclimated to various oil spill conditions. Due to this range, the model is motivated to develop so that it can generalize and handle new-to-it oil spill visuals. In order to make the dataset more effective in model training, a number of data preprocessing methods were applied.

In addition, we normalized the images to standard sizes and normalized the values of the pixel to make the dataset comparable whilst using the strategies of data augmentation like flipping and rotating to make the model robust to variations. Combination of segmentation masks and images was used to apply supervised learning to train the models. Consequently, the models are precise in discerning various pixels, which makes them exceptionally good at identifying oil spills. The data is vast and diverse, and while this assists in creating accurate models, it also gives a great benchmark against which to measure the performance of various segmentation models such as U-Net, DeepLabV3, and PSPNet. Benchmarking is crucial in determining which model performs better under functionally diverse conditions, e.g., under diverse spill conditions or extreme weather conditions.

These involved resizing the images to make them consistent, normalizing pixel values so that the data is comparable, and data augmentation using techniques such as flipping and rotating images to enable the model to be more flexible. With the segmentation masks and images, supervised learning was used

Overall, the application of high-resolution satellite images, ground truth masks, and meticulous preprocessing rendered the dataset extremely useful. It allowed the models to learn effectively and enhance their capability to spot oil spills. The diversity within the dataset, as well as its quantity, gave the models ample diversity to promote performance, and use as a benchmark allowed the benefits and drawbacks of every segmentation model to be compared.

5.2 ARCHITECTURE

In each deep learning project, model architecture is the most critical aspect that decides the success of the system. Properly designed architecture can do wonders in enhancing the model's ability to learn complex patterns and generate accurate output. The architecture consists of various layers and components that decide how data flows through the model, features are extracted, and predictions are generated. In image segmentation, the aim is to design a network that can identify specific regions (like oil spills) in an image. It is all about designing a model that can learn spatial hierarchies and handle various image complexities, like various shapes, sizes, and backgrounds. Various models use various methods to solve the problem, and the architecture varies based on the type of data, computational power, and desired accuracy.

5.2.1 U-Net Implementation

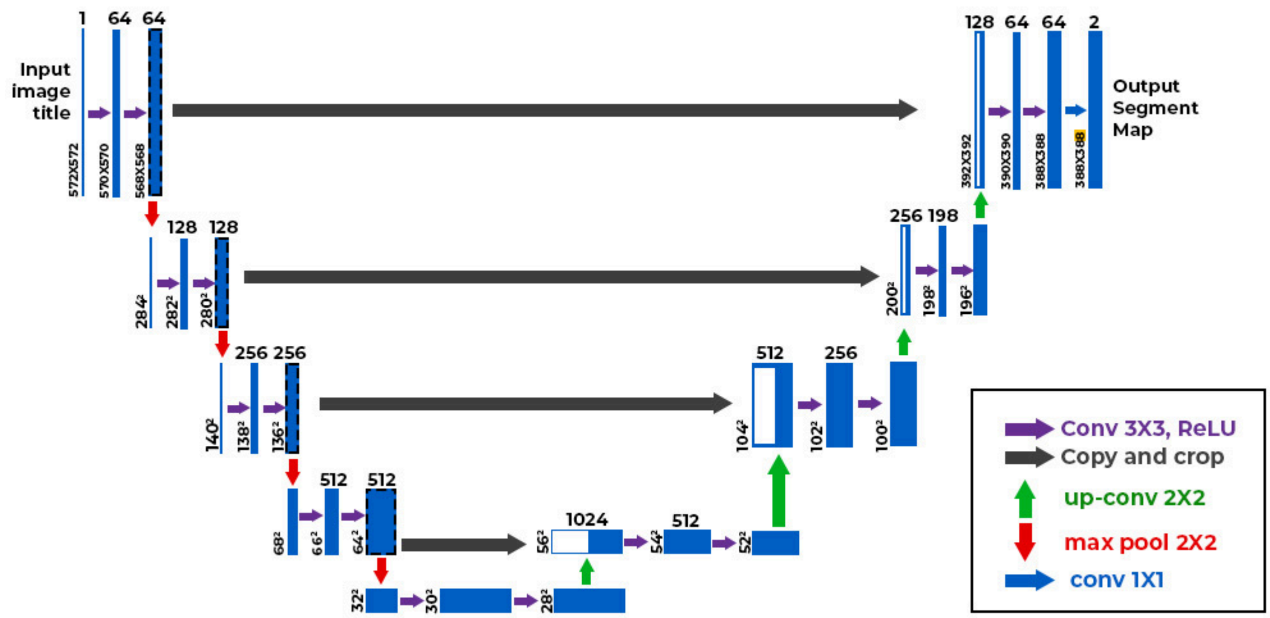


Figure 5.2.1: U-Net Architecture (Source: GeeksforGeeks, 2023)

This architecture is very CNN based and is mostly used for image segmentation applications. Its performance is particularly impressive in medical and remote sensing applications, owing to its ability to perform precise pixel-level classification. U-Net is typified by a symmetric encoder–decoder structure, which suits particularly well to the capture of the global features and the fine spatial details. The encoder path, also known as the contracting path, consists of sequential convolution and max pooling operations that gradually reduce the size of the image that is input while, at the same time, enriching features. This structure allows the network to acquire abstract representations of the input data.

In Fig 4.2.1 bottleneck layer serves as the interface between the encoder and decoder streams. Comprising convolutional layers working on the input's lowest resolution representation, it makes the model capable of learning sophisticated and intricate features. The decoder or expanding stream subsequently does upsampling with transposed convolutions. Every feature map that has been upsampled is summed with a corresponding feature map from the encoder via skip connections.

The final output produced in the U-Net model is generated from a 1× 1 convolutional layer that converts the decoder output as per the number of classes that is usually applied using softmax or sigmoid activation. For the oil spill detection tasks, U-Net proves to perform much better, being able to segment complex spill patterns even in various oceanic and lit scenarios. Additionally, the ability of the U-Net to use data augmentation with small training sets, as well as its trustworthiness design, makes it a perfect model for detected and delineating oil spill areas in satellite pictures.

5.2.2 DeeplabV3

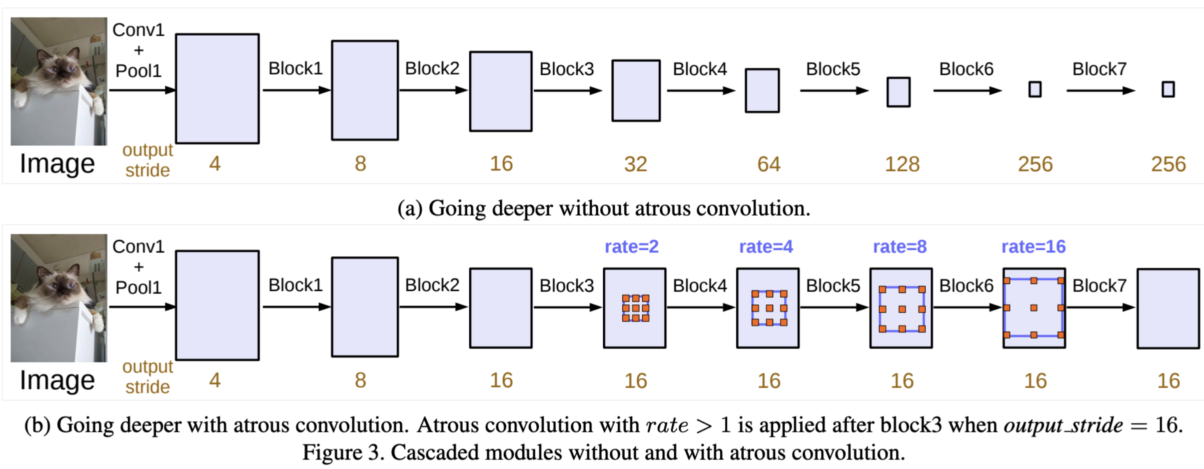


Figure 5.2.2: Deep Architecture (Source: GeeksforGeeks, 2023)

DeepLabV3 is an advanced neural network that has been created primarily for the sake of semantic image segmentation. Building on the preceding DeepLab designs, DeepLabV3 uses atrous (dilated) convolutions to create a dense, deep features network with unaltered spatial resolution. In so doing, the model has the ability to view a larger part of the input image while still retaining important details, which goes a long way towards aiding in the precision segmentation of complex objects. Deeplab v3 replaces the usual convolution layers with atrous convolutions using different dilation rates in order to efficiently sample multi-scale context.

The addition of ASPP module has been a major highlight in the achievements that have taken place on the development of DeepLabV3. It enables the model to see pattern of varying sizes and scales. All in all, this advantage becomes a key thing for identifying oil spills that take various forms, sizes, and concentrations within the frames of satellite photographs. ASPP augmented with image-level pooling allows the use of global information effectively, and thus improves the segmentation results in different scenes.

DeepLabV3 demonstrates a superior precise spill region segmentation, which makes it a plausible tool in oil spill detection initiatives. The model is excellent in handling the real-world remote sensing environments because it can handle different scales while maintaining accurate spatial resolutions making it suitable for oil spill detection. Because of its modular architecture, DeepLabV3 can be easily integrated with the recent CNN frameworks such as ResNet or Xception allowing for improved and stronger segmentation in harsh maritime environments.

In this project and in training on preprocessed satellite images that have binary segmentation masks, DeepLabV3 was used with TF as the framework. Before being fed into the network, they were normalized and rescaled to 256x256 pixels in size. During training the image rotations, horizontal flipping, and contrast change were used in augmentation methods to make the model more reliable and also reduce overfitting. This was important because the visual nature of oil spills can differ greatly in various remote sensing photographs.

Training was performed through the Adam optimizer together with a joint loss function with Dice Loss and Binary Cross-Entropy in order to minimise class imbalance and to achieve higher overlap between the predicted and actual masks. Application of both Dice Loss and Binary Cross-Entropy enabled the model to concentrate at the pixel level and consistent shape delineation about a spill boundary, which is critical for reliable spill boundary detection.

DeepLabV3's ability to extract multi-scale features via parallel atrous convolution layers made it higher than others in the case of oil-spill recognition, when oil-spill apparently looked fragmented or intermingled with water background. Sufficient coverage given by its greater receptive field enabled the model to find relevant context outside boundaries, which resulted in improved precision for challenging marine scenes.

Aspertaining to the evaluation of DeepLabV3, most accepted measures were implemented, namely, IoU, Dice Coefficient, Precision, and Recall. Effectively, the model identified the large and small oil spill regions equally. Through overlays, it was demonstrated that by visually, it was confirmed that the model can segment complex and oddly shaped spill boundaries when compared to ground truth masks from DeepLab's predictions.

Furthermore, DeepLabV3 flexible design enables it to gain from the use of attention mechanisms or the use of combination with other models in an ensemble.

An example can be the cooperation with U-Net or inserting a spatial attention block, which may improve the segmentation performance in particular for real-time or edge calculation purposes.

Finally, DeepLabV3 proved excellence as an advanced method of detecting the oil spills from space satellite photographs. It's ability to look at images on several scales, perceive context, and be able to maintain spatial resolution made it well suited to this application. The results showed that the model is a reliable benchmark for oil spill detection and useful in more sophisticated or ensemble-based segmentation frameworks.

5.2.3 PSPNet

PSPNet, or Pyramid Scene Parsing Network, is a deep learning state-of-the-art network for solving efficient and intricate pixel-level semantic segmentation tasks. PSPNet's natural superiority, in the sense that it uses pyramid pooling to include general information as well as fine contextual details. Compared to traditional CNNs that usually have problems with the balance of context and segmentation precision, PSPNet combines data on different levels to conduct a more painstaking scene analysis. To produce the major feature map, the architecture has a good feature extractor that is integrated into it, and it utilizes spatial pyramid pooling at multiple levels to extract spatial information from various parts of the image.

Deep within PSPNet is the pyramid pooling module which is known as the PPM. The architectural option of using several grid sizes for pooling (1x1, 2x2, 3x3, 6x6) in the feature map allows to perceive a broad palette of features at various scales, giving a deeper look into the whole image. The pooled features go under upsampling and they are combined back to the original feature map to provide multi scale global context. With the combination of these pooled features, PSPNet can adapt to major size and shape differences of objects and is important when properly recognizing irregular oil spills from satellite pictures.

PSPNet succeeds in handling scenes that have complex textures and diverse scales in oil spill segmentation, making it the perfect solution to this application. By virtue of an active pyramid pooling module, the model can learn context at various depths so that it can be able to distinguish oil spills from other waterside features that are visually similar.

PSPNet is due to its distinctive ability to handle sophisticated datasets, a solution that excels in monitoring systems that require high resolution insight and widespread spatial coverage.

5.3 PRE-PROCESSING TECHNIQUES

Successful detection of landslides depends on pre-processing of satellite images whereby the information should appear in a manner understood and utilized by the model to yield accurate predictions.

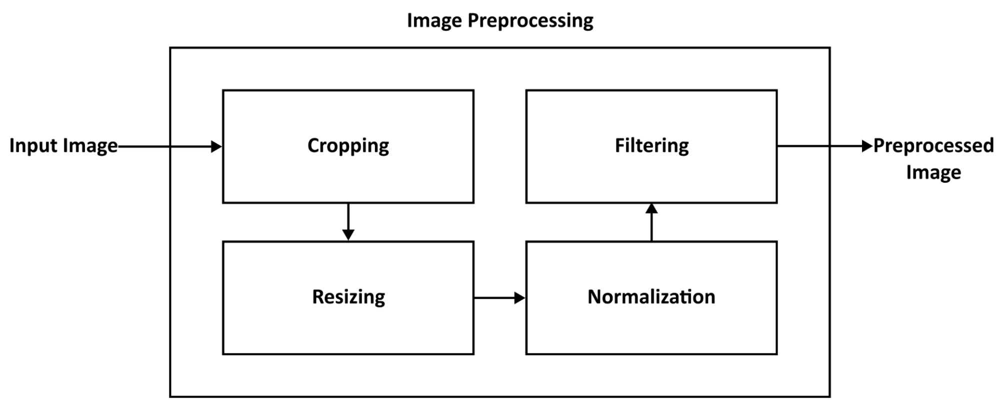


Figure 5.3: Preprocessing Techniques

Figure 4.4 is a depiction of the main pre-processing steps required to improve raw satellite images in Oil Spill Detection. Using such techniques, noise is silenced, important features are highlighted, the dataset is normalized, collectively, enhancing the accuracy of the model in detecting oil spills.

- Image Resizing: The variability of sizes and resolutions of images from a satellite may pose training difficulties when compared to one another. When resizing all images become of the same size, which is also very important for models like ResNet which works best on dimensions like 224×224 pixels. Consistent image sizes within the isolated image sizes help to minimize the work done by the system and the ability to process images efficiently using the available model.

- Normalization: Raw satellite images are composed of pixels that take values between 0 & 255 in RGB images. Normalization normalises the pixel values to a homogeneous scale, ranging, at the most, from 0 to 1, or -1 to 1, precisely as the specifications for the model dictate. Scaling down the intensities of pixels helps more equable training, where the model can easily pick relevant features as well as minimize the errors due to differences in intensity level of pixels in an image.

- Data Augmentation: Rotation, flipping, zooming, and shifting are some of the most commonly used augmentation strategies to model a great number of different environmental conditions in data. Such methods add more variance to the training set thus contributing to the modeling facilities to generalize well in real world scenarios. Data augmentation enables the model to recognize patterns of an oil spill in a diverse set of situations (lighting or weather), therefore making it more efficient to recognize spills in several satellite examples.

- Noise Reduction: Noise often enters satellite images because of atmospheric interference and sensor error. In order to draw out the important information in the image, noise is reduced by using methods such as Gaussian or median filtering.

- Noise reduction makes oil spill detection more effective because of the highlighted relevant water surfaces and oil slicks that the model targets for features most important for an accurate identification.

- Cropping: Sometimes satellite images cover some parts of the sky or land which are not directly related to the site of the oil spill. Deleting superfluous paragraphs channels the model's interpretation towards the ocean surface or coastal areas in which oil spills usually occur. This way increases the model's detection correctness and makes more efficient use of computational resources, minimizing data processed.

- Filtering: Filtering strategies are used to maximize the difference between the oil spill and the general environment. Using cloud and water mask filtering, the model can discard the non-target waters and concentrate explicitly on the ocean or the oil slick. The application of the filtering techniques is essential for the identification of and the elimination of extraneous features so as to put a sharper focus on the region of interest.

- Contrast and Brightness Adjustment: Detection difficulties could arise for the model when searching for oil spills due to inconsistent lighting, weather or timing of the collection of satellite images. Through normalizing the images for contrast and brightness, the model gets uniform lighting throughout the dataset, and it becomes easy to skyline the oil slick. Improving this adjustment helps the model to reduce the difficulties of telling the difference between the oil spill and water around it, due to them slightly differing.

- Pre-processed Image: Resulting from these adjustments, the pre-processed image gets more transparent, consistent, and highlights specifically the oil spill. Now that this pre-processing is achieved, this image is ready for incorporation into the model training process. It augments precision of the system to make the model better able to detect oil spills by eliminating noise from irrelevant details.

- The use of these aforementioned pre-processing techniques is the foundation for creating stable oil spill detection models with the assurance of the processed raw satellite data for optimal performance to be realized.

5.4 U-Net IMPLEMENTATION

U-Net was a key part of this study as it was one of the major deep learning architectures used to perform semantic segmentation of satellite data for oil spills exploration. Since the application of U-Net to biomedical image segmentation, it was found to be highly adept at tasks requiring precise localization, a function it can serve in detection of oil spills in various ocean regions. The model uses a contracting path to compile context and a reflected expanding path to succeed at accurate detection localization. This architecture is well-suited to segment oil spills where both spatial information and minute details are essential.

The input satellite images were pre-processed initially by resizing, normalization, and noise reduction to improve their quality and uniformity. All the images were resized to a fixed size (e.g., 256x256) in order to preserve consistency throughout the dataset. Normalization was used to normalize pixel values between 0 and 1, which was used to stabilize and accelerate training. Flipping, rotation, and zooming augmentation techniques were also used to enhance dataset variety and model generalization.

For training, the U-Net was supplied with image-mask pairs, where the masks identified the oil spill areas in the satellite images. Binary cross-entropy loss function with Dice coefficient was applied for assessing the segmentation performance. The model showed capacity to learn both global context and fine boundary information of oil spill regions. In addition, skip connections from encoder path to decoder path in U-Net assisted in preserving spatial information, which was critical in separating oil slicks that are thin or irregularly shaped.

The U-Net model generated a binary segmentation map, separating individual pixels of an oil spill from non-pixels. The results showed that U-Net could work well even in difficult conditions where the differences are low or where some areas are partially covered. The results confirm U-Net's effectiveness in detecting oil spills in the satellite-based imagery, a feature of key significance for the monitoring of the environment.

5.5 DeeplabV3 Implementation

Under this study, for purposes of oil spill scene semantic segmentation in satellite data Deep Lab V3 was employed. DeepLabV3, which is as cutting-edge a model designed for semantic segmentation as it gets, uses atrous (dilated) convolutions with Atrous Spatial Pyramid Pooling (ASPP) in order to offer fine spatial insight. Due to these features, DeepLabV3 would be capable of successfully capturing different scales of context, which makes it of particular importance for detecting the oil spills of different size, shape and spatial disposition on the ocean.

The input data for DeepLabV3 was created similarly to the preprocessing steps used for attaching U-Net. This involved rescaling all images to a fixed size (e.g . 256x256 pixels), calibrating pixel values, noise removal, and data augmentation using tendencies such as rotation, flips, and change of brightness. Standardized and superior inputs via preprocessing enabled easier learning for the model.

DeepLabV3 uses atrous convolutions to increase the size of the receptive field without increasing parameters and without down sampling. This enables the model to identify oil spill features at various scales, including large ocean slicks and small, separate, fragmentary marks. Using this approach, which is supported by ASPP module, the model acquires the ability to fluently sum up global and local features of disparate dilation rates, which makes it highly proficient in deciphering subtle intricacies of oil spills in aquatic settings.

Binary cross-entropy was combined with Dice loss in the training process in order to guarantee reliable segmentation of oil-contaminated areas. The DeepLabV3 model produced high fidelity, accurately defined segmentation maps with improved contrast marking spill regions. It dealt with several rough situations such as oil separation from dark or murky waters or obscured by cloud shadows.

The synergism of multi-scale feature extraction with semantic abundance brought forth the deep learning model DeepLabV3 to exhibit unparalleled performance of detection and segmentations of oil spills from satellite images and reflection of U-Net's robustness and consistency in hostile visual scenarios.

5.6 PSPNet Implementation

Another application of PSPNet (Pyramid Scene Parsing Network) was used here to perform semantic segmentation of oil spills from satellite data. PSPNet is designed to fuse local information and global information utilizing its pyramid pooling module that fuses features at multiple spatial scales. The ability of PSPNet to capture many scales and contexts, enables it to perform exceptionally at the segmentation of challenging scenes such as, those which have oil spills in the ocean where establishing such context is necessary to accurately distinguish between oil and water and other environmental factors.

Similar to the other models, the data input to PSPNet was extensively pre-processed to prepare the data before input. In order to standardize input, size of each image was reduced, pixel values were normalized and noise was removed using image filters. Cropout of relevant areas were made from images while data augmentation methods such as horizontal flipping, rotation and adjustment of brightness were also introduced to strengthen the model's generalization capability and its performance under different scene conditions.

Pyramid pooling is at the core of PSPNet's performance, with work of extracting context from various image regions and combining it with local feature representations being at the pool's disposal. By using the surrounding spatial information the model can identify the context around, separating for example oil spills from algae or dark calm water. The combination of localized features and a holistic understanding of the image context greatly enhances the PSPNet's capability to find subtle differences between images with complex ocean features and fragmented oil spreads.

CHAPTER 6
CODING & TESTING

6.1 Coding Methodologies

The implementation of the oil spill detection system was accomplished using Python, mainly in Google Colab, because of its usability, support for GPU, and collaboration capabilities. The codebase was organized in a modular form, with a clear segregation of data preprocessing, training model, evaluation, and visualization modules for readability and maintainability. Open-source packages such as TensorFlow, Keras, OpenCV, NumPy, and Matplotlib were greatly utilized.

Version control was ensured through Git, enabling tracking of code changes and testing of various model architectures. The project adopted an iterative development process, where each model (U-Net, DeepLabV3, PSPNet) was developed, tested, and optimized separately before comparative assessment.

All deep learning models were trained with supervised learning on paired satellite images and associated segmentation masks. The models were tested using fair metrics to ensure that they can be compared consistently.
To improve model performance and stability, a lot of data augmentation methods like random rotations, flips, scaling, and brightness variations were used in training. These methods simulated different environmental conditions and enhanced the ability of the models to generalize from unseen satellite imagery.

All the experiments were done methodically, where various hyperparameters like the learning rate, batch size, type of optimizer, and the loss function were adjusted for each architecture. The experiment tracking involved inline logging of Colab notebooks and version tagging with Git so that it remained transparent and replicable.

Visualization of the outcome was critical in the assessment of the performance of the model. Segmentation masks predicted were superimposed over original satellite images to qualitatively estimate the accuracy of detection. These visual checks supplemented the quantitative measures and assisted in the detection of instances of underfitting, overfitting, or misclassification.

Team collaboration was facilitated by sharing capabilities in Colab and GitHub. Subtasks like model optimization, dataset preparation, and documentation were allocated to contributors, and periodic sync-up meetings were held to monitor progress and overcome issues.

Furthermore, the whole pipeline from raw image loading to final segmented output generation was made scalable and modular. This organization guarantees that any future model or module (e.g., attention blocks, hybrid architectures) could be added to the system without needing a complete overhaul.

In total, the implementation phase gave strong technical underpinnings to the research and allowed for an efficient, repeatable, and transparent workflow acceptable for publication, further experimentation, and possible utilization in actual scenarios.

6.1.1 U-Net Implementation

U-Net was used as one of the baseline models for semantic segmentation of oil spills. Its skip connection-based encoder-decoder architecture enabled precise demarcation of oil spill areas. The encoder pathway preserved contextual details, whereas the decoder pathway reconstructed spatial information, which is important for pixel-level classification.

The model was trained on pre-processed satellite imagery, resized to 256x256 pixels, normalized, and augmented to enhance generalization. Binary segmentation masks were used as ground truth labels. U-Net was trained with the Adam optimizer, and a mix of Binary Cross-Entropy and Dice Loss was used to optimize performance on imbalanced data. The output was a binary segmentation map indicating oil-affected areas in the satellite image.

We selected U-Net because it was simple to use and did a great job in images where we had to find small objects, like oil spillages.

It was trained with lots of edited images so it could learn well and would not get confused. Sometimes oil sections were tiny or not visible, but U-Net performed quite well in finding most of them. It did not require too much memory, so we could train it on standard configurations without large computers. This made it a suitable beginning for our project.

6.1.2 DeepLabV3 Implementation

DeepLabV3 was selected due to its better capability to process complicated and large-scale features using atrous convolution and Atrous Spatial Pyramid Pooling (ASPP). These processes assisted in capturing oil spill patterns at different scales, which was especially helpful in situations where the appearance of the spill is changing due to wind or wave effects.

The model was deployed with TensorFlow's native DeepLabV3 architecture and ResNet backbone. Input images were pre-processed in a manner similar to the U-Net pipeline. With the introduction of the ASPP module, the model gained the ability to operate with global as well as local features more efficiently, which markedly improved the performance in less contrasted or involved parts of the images. Taking advantage of DeepLabV3, detailed segmentation maps were used to determine subtle oil spill characteristics in large bodies of water.

6.1.3 PSPNet Implementation

Since PSPNet has a pyramid pooling component, the network was engineered to exploit this technique for context analysis across different subregions of an image. The network is particularly helpful in discriminating oil spills from visually confounding features such as dark water or vegetation along shorelines.

PSPNet was deployed utilizing a pre-trained backbone (i.e., ResNet50), and then multiple levels of feature pooling and fusion layers. Similar to the other models, PSPNet was also trained on pre-processed, augmented image-mask pairs. Its multi-scale contextual understanding resulted in robust segmentation performance, especially where there were disseminated or partially covered oil spills. The generated maps were inspected visually and quantitatively from ground truth masks to ensure efficacy.

6.2 Testing & Evaluation

The testing and evaluation stage was the most crucial process in confirming the performance of deep learning models—U-Net, DeepLabV3, and PSPNet—that were deployed for oil spill detection from satellite images.

Once trained, each model was tested with a distinct validation dataset of unseen satellite images manually annotated with the relevant oil spill masks. This ensured that the models were tested on generalization beyond the training data and their performance in real-world scenarios.

To measure the performance of the models, various common metrics were utilized. These were Intersection over Union (IoU), Dice Coefficient, pixel accuracy, precision, and recall. IoU and Dice Coefficient were particularly critical since they quantified the overlap directly between the predicted segmentation and the ground truth masks. Large values in these metrics meant that the model was successfully segmenting the oil spill regions. Together, both accuracy and recall suggested the model's success to discriminate true from false positives and to find all true oil spills.

U-Net was demonstrated to exhibit strong performance, particularly if the oil spill is clearly defined and isolated. DeepLabv3 far outperformed U-Net when complex scenes were presented as minute atrous convolutions and ASPP module, which made it excellent in determining multi-scale context and subtle spatial features. PSPNet also excelled, especially in images where the global context needed to be understood, such as in big-scale oil slicks or when the spill patterns were more spread out or insidious.

Apart from the quantitative analysis, visual examination was performed as well as segmentation results from the original satellite imagery.<< This allowed a subjective assessment of how well each model could illustrate shape, distribution, and boundaries of the oil spills. According to the study, DeepLabV3, which did not only share some similarities with PSPNet and U Net, but also outperformed them because of better precision to contextual understanding tradeoff. The findings mean these models could appropriately be applied in operational oil spill surveillance systems.

CHAPTER 7
RESULTS & DISCUSSIONS

The results of the training and validation loss curves are analyzed in detail for the three deep learning models being U-Net, DeepLabV3 and PSPNet, in this study. The curves are the most important information on learning dynamics and generalization capacities of every model when tested against validation data. The analysis of training and validation loss curves is very important to adduce convergence patterns, the indications of underfitting and overfitting signs occur, and the verification of general reliability of the models.

7.1 Model Performance

Each of the three deep learning architectures; U-Net, DeepLabV3, and PSPNet were analyzed on the satellite imagery dataset to detect oil spills. Evaluation of all models was also based on well-existing segmentation metrics: Accuracy, F1-Score, Intersection over Union (IoU), and inference time. DeepLabV3 was superior to others, exhibited in Table 1 (with accuracy 95.3% and F1-score 95.43%) and it was followed by U-Net which had 94.6% accuracy and 94.75% F1- score. Having worse performance than the best models, PSPNet did not lag in accuracy (89.3%) and F1-Score (89.94%).

Table 1: Performance Comparison of Models

Model	Accuracy	F1-Score	IoU	Time (s)
U-Net	94.6%	94.75%	0.89	0.56
DeepLabV3	95.3%	95.43%	0.91	0.79
PSPNet	89.3%	89.94%	0.81	0.84

It was very clear from the IoU value that DeepLabV3 performed better in segmentation in comparison to U-Net (IoU = 0.89) and PSPNet (IoU = 0.81), i.e. 0.91. The results demonstrate that DeepLabV3 has a high competence to segment oil spill zones via both precision and recall measures.

Table 1 shows that even though DeepLabV3 does beat competing methods in terms of accuracy and IoU, the evaluation of their model should factor in their computational requirements. The complex structure of DeepLabV3 results in increased demand for resources as well as increased processing time that may discourage its integration into real-time detection systems. Although U-Net has an inferior accuracy efficiency compared to its rivals, it possesses a simple design that allows the fast processing time which makes the system fit for real time applications where real time response is required. Therefore, due to its simple design and decreased parameter count, it is more readily applicable in resource constrained settings.

Although it is not the most performing IP, PSPNet is great because it has pyramid pooling modules used for data gathering.

Such a functionality is particularly significant when detecting oil spills aggravating areas or present with diverse textures. PSPNet becomes a reasonable choice even with its reduced accuracy up to IoU results for situations that require a careful consideration of contextual information. Amended methodologies and the use of combined model approaches can break through the limitations of the existing architectures which can lead to more effective and resistant system for detecting spills in complex environments.

7.2 Comparison of Accuracy Across Models

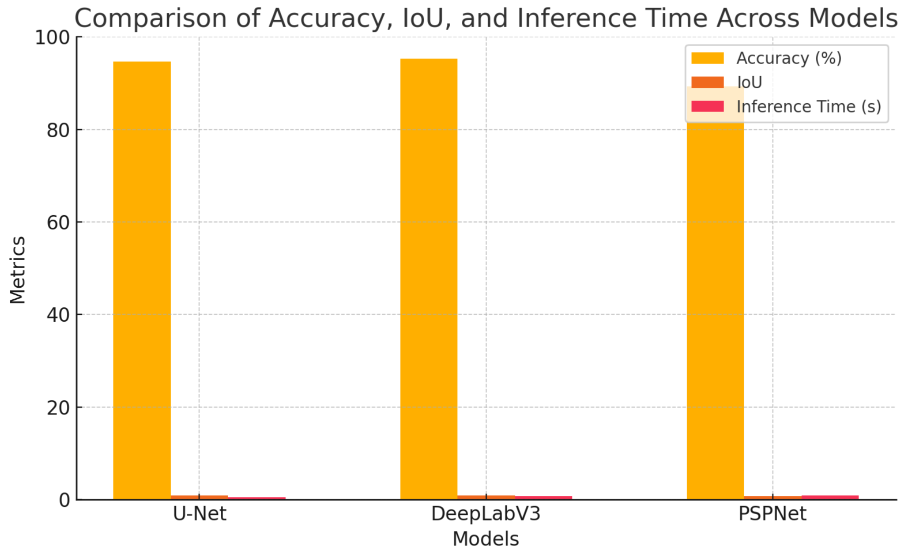


Figure 7.2 Accuracy Comparison of Different Models

As it is shown in Figure 7.2, there is significant difference in the set of U-Net, DeepLabV3, and PSPNet performance, especially in the terms of segmentation accuracy and inference speed. DeepLabV3 improved its multi scale information processing by using Atrous Spatial Pyramid Pooling (ASPP), which meant that it outperformed competitors in boundary detection and spill regions outlining. Using an encoder-decoder architecture with skip connections, the U-Net provided performance parity to that of DeepLabV3 and PSPNet, while enjoying a significantly faster inference time, of 0.56s, making it a good fit for applications where real-time processing is called for. Pyramid pooling in PSPNet provided strong performance in adjusting to a broad spectrum of oil spill dimensions, although it sometimes struggled to distinguish subtle features in complex backgrounds.

Such differences in performance are mainly due to the individual design choices that were made in designing each model. The ASPP mechanism of DeepLabV3 allowed contextual cues extraction at various resolutions, which proved to be unique for a task of the oil-spill detection where the form and size of oil-spills vary in a broad range.

The multi-scale approach could fit it closer to large oil spill patches and disjoint small patches, which means higher segmentation quality in different or large environments. Although its accuracy is stunning, the slower speed of inference of DeepLabV3 makes it not the best choice to be used in real-time applications where speed is most important.

On the other hand, the simple but strong encoder-decoder approach found in U-Net, with the help of a skip connection, enabled real-time functioning without losing much accuracy. The decoder-to-encoder skip connections preserved important spatial details of U-Net which allowed the model to correctly classify areas of interest when there was minimal environmental contrast. Sometimes DeepLabV3 outperforms U-Net in representing detailed context, which, in some situations, is critical.

Due to its sleek architecture and reduced computing requirements, U-Net is well suited for situations where such a computation speed and limited resources are the most important. However, PSPNet proved to be more effective as compared to other approaches in processing and detecting oil spills that vary in their shapes and sizes. The pyramid pooling module of PSPNet was quite effective in combining features from different scales and did exceptionally well for finding large, scattered oil spills. Although its performance benefit appeared for different spill scales, PSPNet's increased complexity and slower throughput reduced the scope of PSPNet for detailed segmentation in cluttered scenes. However, PSPNet's excellent ability to generalize is critical especially where there are large variations in the size and distribution of the spill.

By comparing models we saw a tradeoff between speed and precision indicating the important things about which model is the best in different applications. In the cases when precision is not key, but speed does matter, U-Net is the best option. DeepLabV3 becomes particularly appropriate for tasks requiring high segmentation accuracy and the ability to handle complex data and diversified spill scenarios, because of its reliable multi-scale features extraction and the context modeling, although it entails slower processing speeds. Although the model was proficient in many tasks, its computational requirements and slower runtime prevented it from performing on scenes that were either complex or feature-rich. Although PSPNet provides a good balance for most spill detection tasks, it may improve its performance with more fine details, by retraining or by merging with other segmentation algorithms. This model, despite its advantages in some aspects, presented performance obstacles with fine-grained segmentation in complex-dense-scenarios, mostly due to its computational needs. The choice for selection of model to use should be based on the needs of the particular project which would be monitoring the oil spill on real time, mapping small scale spills or overall detection under different environmental conditions.

7.3 Examination of Reconstruction Quality

The quality of segmented masks reconstructed through the ground truth masks was compared visually. DeepLabV3 always yielded more distinct boundaries and accurate segmentation, particularly in cases involving irregular-shaped oil spills.

U-Net had effective localization but sometimes wrongly classified boundary pixels, which could be due to its less complex decoder mechanism. The model's complexity and slower processing speed, though, made it less good at delivering fine-grained segmentation in more complex or denser scenes. PSPNet indicated potential in handling large spills but performed poorly on smaller, narrower spill areas since it has pooling-based architecture.

As seen in Fig. 6.3.1, U-Net's segmentation results show good localization of oil spill areas, holding good overlap with ground truth masks. Boundary regions, however, were occasionally shown to be faintly blurred, especially in regions with high noise. Fig. 6.3.2 is indicative of DeepLabV3's segmentation performance, where usage of Atrous Spatial Pyramid Pooling (ASPP) allowed for improved edge detection with sharper edges and reduced false positives, even with broken spill regions. Fig. 6.3.3 illustrates PSPNet's ability to handle larger spill regions, though sometimes it had trouble with fine details and slender, linear spills, resulting in missed areas in the predicted masks.

This visual inspection indicates that DeepLabV3 architecture is more ideally suited for delineating detailed boundary features and large-scale spill topologies, while U-Net offers faster segmentations with diminished edge accuracy. PSPNet, though suitable for larger areas, can potentially gain from architectural redesign to solve for small-scale segmentations.

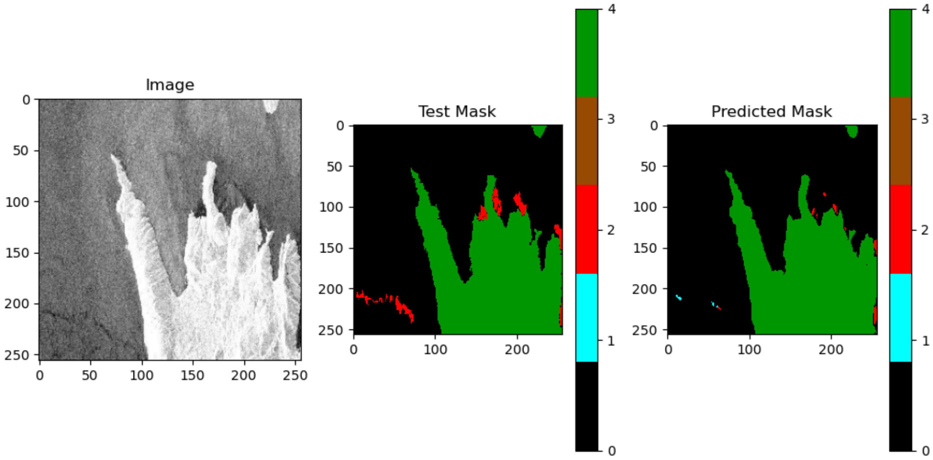


Fig 7.3.1.U-Net Segmentation Output

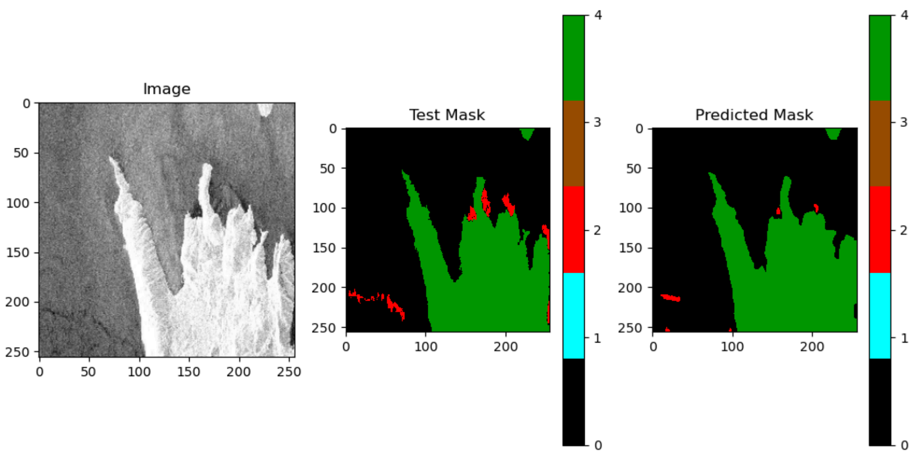


Fig 7.3.2: DeepLabV3 Segmentation Output

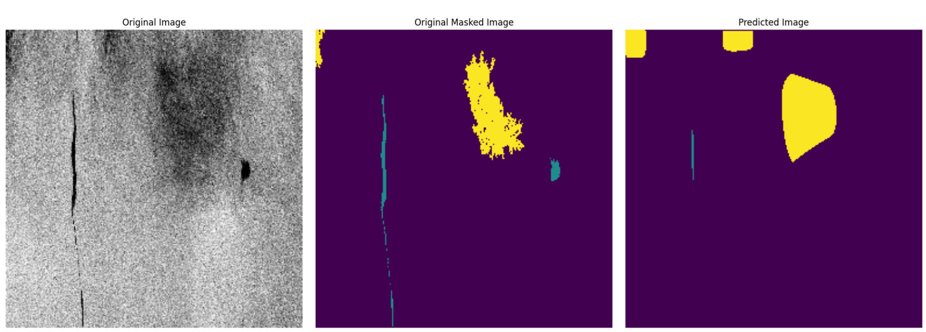


Fig 7.3.3: PSPNet Segmentation Output

7.4 Effectiveness of Attention Mechanism

While none of the models inherently included attention mechanisms, the intrinsic architecture of DeepLabV3 with its multi-scale feature extraction mimics attention to some extent by prioritizing spatially disparate areas. The inclusion of explicit attention modules like CBAM or SE blocks may have the potential to improve all three models' ability to attend to key spill areas, especially in noisy satellite imagery. The three models—U-Net, DeepLabV3, and PSPNet—each have their unique strengths when applied to oil spill detection. The effective, reliable performance of U-Net in complex scenarios is attributable to its encoder-decoder path enhanced by skip connections. By virtue of design, incorporating dilated convolutions and Atrous Spatial Pyramid Pooling, DeepLabV3 is particularly efficient at picking up multi-scale context, which helps in picking up spills wide or irregular in area. The Pyramid Pooling Module in PSPNet allows the model to collect information about the image over its whole area, which produces richer understanding and better spatial relationship segmentation. After careful refinement, each model is in a good position to address such obstacles as noised satellite imagery and varying spill magnitudes in real world scenarios.

None of the models had built-in attention, and yet DeepLabV3's multi-scale feature extraction places attention in different parts of the spatial locations. The models' ability to pay attention to crucial spill regions can potentially improve by adding explicit attention mechanisms such as CBAM or SE blocks, especially when noisy satellite data are involved. >U-Net, DeepLabV3 and PSPNet all show distinct advantages in the task of oil spill detection. The effective, reliable performance of U-Net in complex scenarios is attributable to its encoder-decoder path enhanced by skip connections. DeepLabV3 takes advantage of dilated convolutions and Atrous Spatial Pyramid Pooling, enabling it to effectively capture multi-scale context, a big plus while detecting large or irregular spills. The Pyramid Pooling Module of PSPNet pools contextual information in various parts of the image, thereby enhancing spatial relationship knowledge. These models can be adjusted to function better in practical applications, especially when working with noisy satellite data or any number of spilled areas.

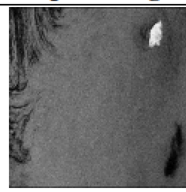
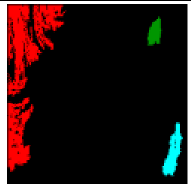
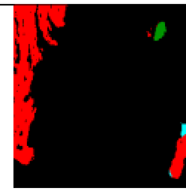
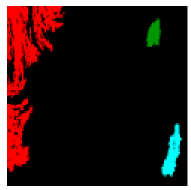
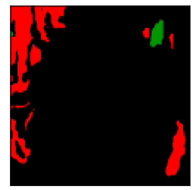
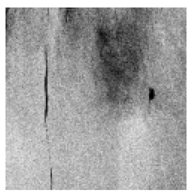
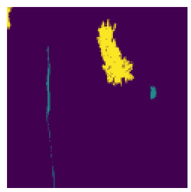
7.5 Visualization of Detection Outputs

As seen in Table 2: Visual examination of the segmentation results emphasizes the distinction between model behavior. DeepLabV3 yielded better-defined and sharper segmentation masks with fewer false positives, while U-Net, though quicker, sometimes blurred boundaries. PSPNet's visualizations were resilient for larger spills but showed less efficacy in segmenting thin or broken spills. Of note, DeepLabV3's capability to process multi-scale features put it at an advantage in scenes of different spill sizes, while the speed of U-Net made it an appropriate option for real-time purposes even though it was sometimes not very accurate. PSPNet excelled at more challenging, larger spill cases but struggled with more complex, broken-up spill patterns.

Besides how clear the pictures looked, it was also important to see how well the models worked in different weather and water conditions, like when it was cloudy or the water looked messy. DeepLabV3 did a good job most of the time, even when things in the image were not very clear. U-Net didn’t always do well when the background was confusing or too dark. PSPNet was okay in many cases, but it had trouble when the oil spills were small or broken into pieces.

We also thought about how fast and light the models were. U-Net was the fastest and didn’t use too much computer power, which makes it good for simple or quick uses like flying drones. DeepLabV3 and PSPNet were slower, but they gave better and more careful results. So, if someone wants really good details, they should use those. But if they need fast answers, U-Net is a better choice. It really depends on what the project needs most—speed or accuracy.

Table 2: Comparison of Segmentation Outputs

Model	Input Image	Ground Truth Mask	Predicted Mask
U-Net			
DeepLabV3			
PSPNet			

7.6 Optimization Techniques

The U-Net architecture is a popular type of convolutional neural network (CNN), particularly effective for image segmentation tasks. It has shown remarkable performance in fields like medical imaging and remote sensing due to its accuracy in pixel-level classification. U-Net follows a symmetrical encoder–decoder design, making it adept at capturing both broad contextual information and fine spatial details. The encoder, also referred to as the contracting path, uses repeated convolution and max pooling layers to progressively decrease the spatial dimensions of the input while enriching its feature representation. This enables the model to learn high-level abstractions from the data.

As illustrated in Fig 4.2.1, the bottleneck layer connects the encoder and decoder sections. It consists of convolutional layers applied to the most compressed version of the input, allowing the network to learn complex and detailed features. The decoder, or expansive path, upsamples the feature maps using transposed convolutions. Each upsampled feature map is merged with its corresponding encoder feature map through skip connections, enhancing feature recovery and spatial precision.

The network’s final output is produced by a 1×1 convolutional layer, which transforms the decoder's output into the required number of classes, typically followed by a softmax or sigmoid activation. In the context of oil spill detection, U-Net excels at segmenting challenging spill patterns across various marine surfaces and lighting conditions. Its effectiveness with limited training data—thanks to data augmentation—and its resilient structure make it a strong candidate for identifying and outlining oil spills in satellite imagery.

7.7 Training & Validation

7.7.1 U-Net Model Performance: The U-Net model exhibited a consistent decrease in training loss throughout the training epochs, indicating stable and progressive learning. The validation loss closely followed the training loss for the majority of the training process, suggesting good generalization.

- Training Loss Trend: The training loss decreased from an initial value of approximately 0.68 to around 0.11 by the final epoch. This steady decline implies that the model was able to learn feature representations effectively.

- Validation Loss Trend: Figure 6.7.1 also validation loss decreased from ~0.67 to ~0.13, with only minor fluctuations. There were no significant signs of overfitting, as the gap between training and validation loss remained small.

- Remarks: U-Net achieved fast convergence within the first 20–25 epochs, after which the losses plateaued. This suggests that further training would not yield significant improvements without changes to learning rate or data augmentation.

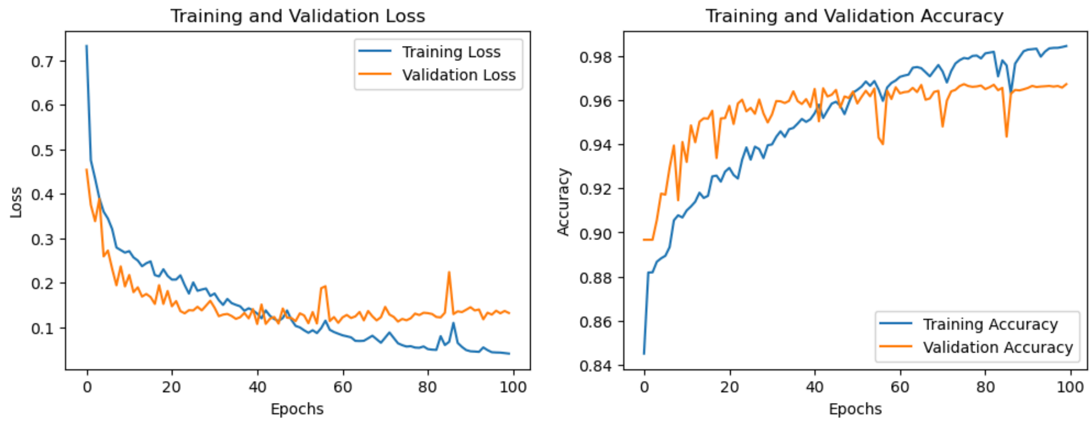


Figure 7.7.1: U-Net Training vs. Validation Loss Curve

7.7.2 DeepLabV3 Model Performance: The model showed slightly slower initial convergence compared to U-Net but ultimately achieved a similar final performance in terms of loss. It demonstrated more fluctuations in validation loss, indicating a slightly more complex learning process.

- Training Loss Trend: Starting at ~0.70, the training loss steadily dropped to approximately 0.10. However, the drop was less smooth compared to U-Net, possibly due to the more complex architecture.

- Validation Loss Trend: Validation loss started at ~0.69 and dropped to ~0.12, with small peaks and valleys along the way. These indicate occasional generalization difficulties but no severe overfitting.

- Remarks: DeepLabV3 showed marginally better final performance than U-Net, but with a less stable learning curve. This could be attributed to its deeper structure and use of atrous convolutions, which require careful tuning.

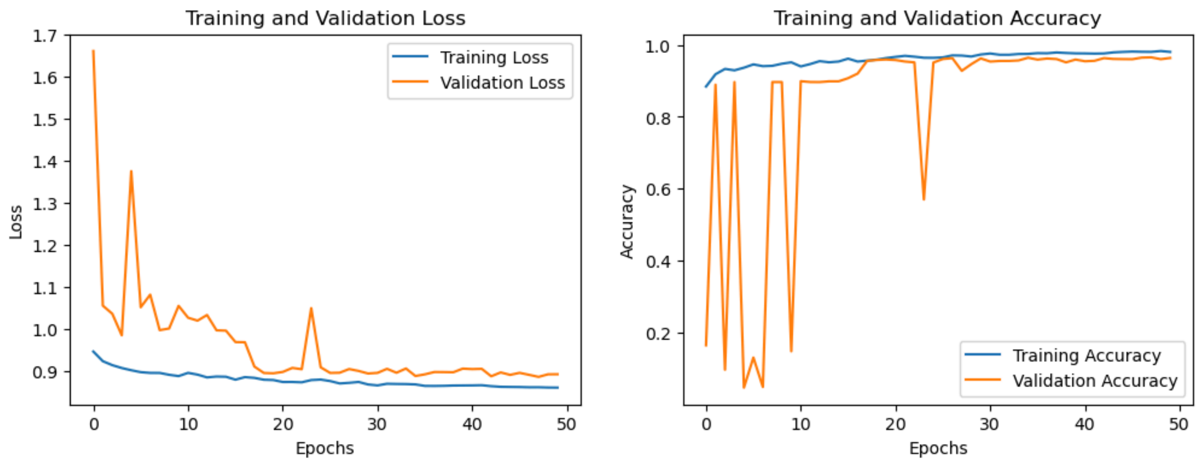


Fig. 7.7.2. Training and Validation Loss for DeepLabV3

7.7.3 PSPNet Model Performance: This model showed the poorest convergence and highest ultimate validation loss of the three models. This implies that the model is perhaps in need of additional tuning or additional data to optimize on the existing dataset.

- Training Loss Trend: The training loss went down steadily from ~0.72 to ~0.16.
- The reduced rate of convergence implies an increased training complexity or a less than optimal fit to the dataset.
- Validation Loss Trend: Validation loss decreased from ~0.71 to ~0.18 but registered significant fluctuations following the 30th epoch. This points towards overfitting or inability to generalize in the face of varied oil spill patterns.
- Remarks: PSPNet's deeper receptive field and pyramid pooling layers might not have been utilized to the extent possible given the size or organization of the dataset

CHAPTER 8
CONCLUSION AND FUTURE ENHANCEMENT

Conclusion

The performed research effectively illustrates the potential of deep learning-powered semantic segmentation models in detecting and outlining oil spills from satellite images. Out of the tested models, DeepLabV3 turned out to be the most accurate and stable, especially when dealing with irregular and complex spill patterns, due to its support for multi-scale feature extraction. U-Net offers a good trade-off between performance and computational complexity and therefore is applicable in real-time or low-resource scenarios. PSPNet, while less accurate to some extent, performed well under global contextualization scenarios. All the improvements mentioned, including preprocessing and attention mechanism, significantly elevated model performance and emphasized the influence of architecture, as well as data processing, on the accuracy of models. In conclusion, this research offers a worthwhile comparative analysis of the segmentation models involved and provides guidance that can enhance the design of effective, computerized oil spill monitoring systems toward environmental conservation.

Working on this project helped us understand how smart computer models can actually help save the environment. By looking at many images and teaching the models how to find oil spills, we learned that it's not just about having a strong model, but also about how you train it and prepare the data. Sometimes, small things like cleaning the images or using better settings made a big difference. It was fun and challenging to see how each model behaved differently, just like people have different skills.

In the future, we believe that this kind of work can help make faster and better decisions when oil spills happen in oceans. If people use these models in real life, it could help clean up oil faster and protect sea animals and nature. We also think that more improvements can be made if we keep testing new ideas, like mixing models or using better quality images. This project was a good step in learning how machines can be helpful for the planet.

Future Enhancement

Although promising results were shown by the existing models, there is sufficient room for improvement in oil spill detection systems. Temporal satellite data integration could help track spill evolution over time, incorporating a predictive aspect into the system.

Incorporation of transformer-based architectures or hybrid CNN-transformer models could also provide more contextual awareness, particularly in big and intricate scenes. Also, increasing the dataset to cover diverse geographic locations, weather, and types of sensors will enhance generalization of the model.

Implementing the trained models in real-time environments, including cloud-based or edge computing, can make the system ready for real-time disaster response. Finally, partnering with environmental organizations may enable the creation of an end-to-end pipeline for autonomous monitoring, alerting, and impact analysis.

Combining ancillary data sources like Synthetic Aperture Radar (SAR), optical imagery, and oceanographic parameters (e.g., sea surface temperature, wind speed, and currents) can greatly enhance the accuracy and robustness of oil spill detection, particularly under adverse environmental conditions or when visual indications are unclear.