



Neural Networks for SAR Oil Spill Detection

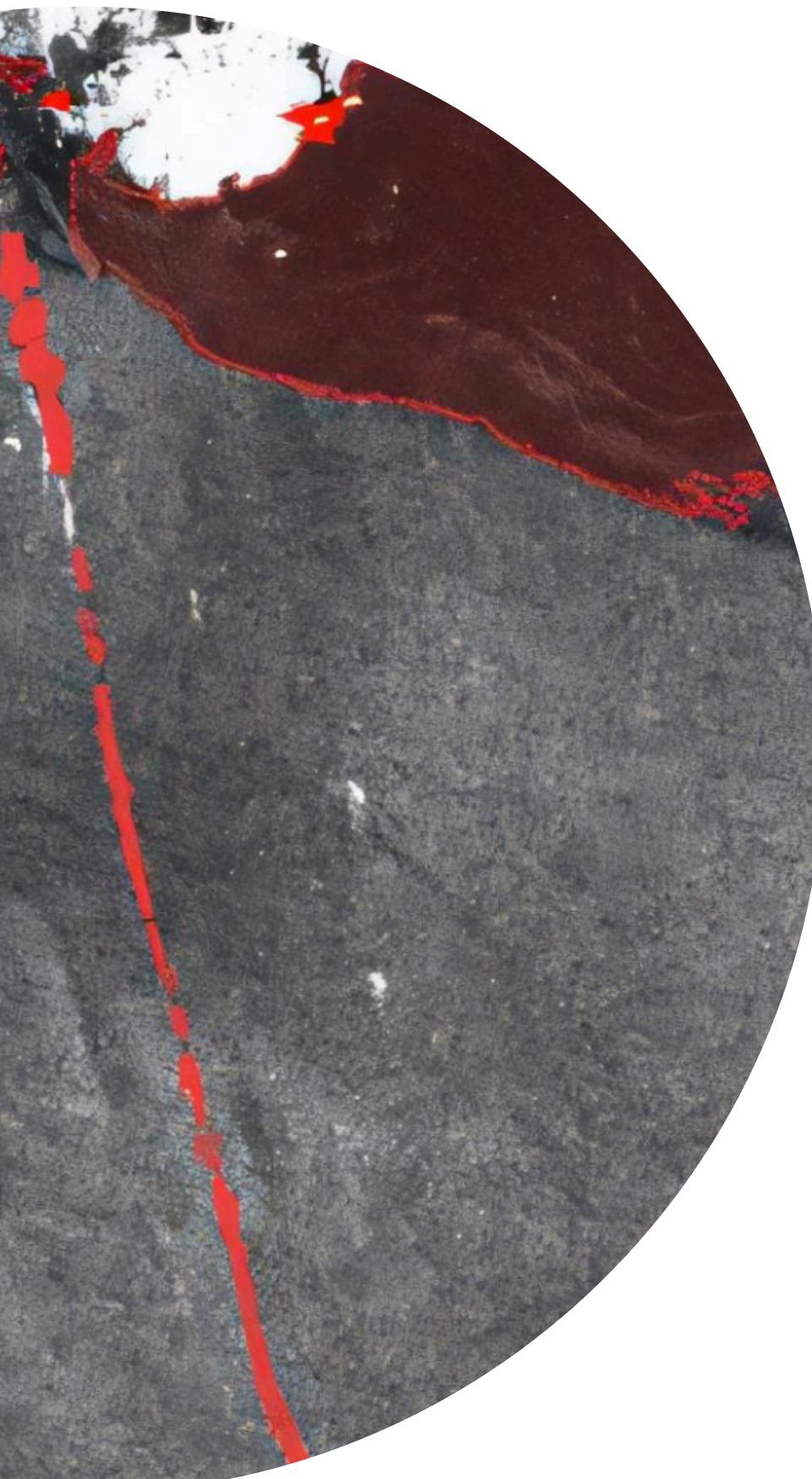
Applied Data Science Project

Bianca Bartoli - s322799

Carlo Marra - s334220

Alessandro Valenti - s328131

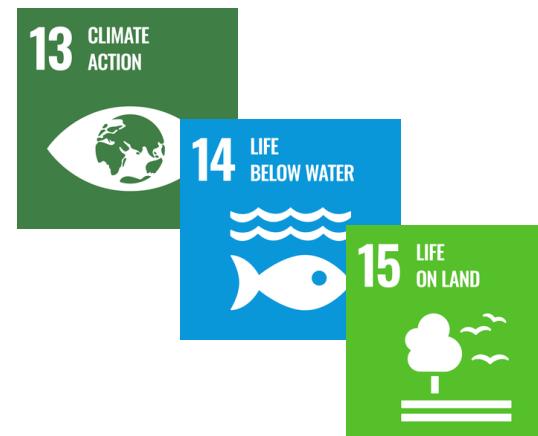
VALUE PROPOSITION



Oil spills pose a significant threat to marine ecosystems



Main goal of the project:
developing models to enable
continuous **monitoring** and
precise **detection** of oil spills



Environmental Impact

Faster response times can significantly reduce the environmental damage caused by oil spills

Response Efficiency

Early detection and response help contain the spill, reducing the resources needed to manage its spread

Scalability

The developed system can be applied to any marine environment with SAR imaging availability, offering potential for global deployment



TASKS

Introduction:

- ✓ Understand the oil spill issue
- ✓ Review reference papers and resources

Paper replication:

- ✓ Explore the dataset
- ⚠ Build a functional pipeline
- ⚠ Apply paper-specific augmentations
- ⚠ Reproduce all models
- Evaluate and compare results

Development of improvements:

- ⚠ Design tailored augmentations
- Implement CBDNet



GANTT

	October 2024	November 2024	December 2024	January 2025
Project Management				
Research and Familiarization				
Data Exploration & Preprocessing				
Model Implementation and Training				
Evaluation and Comparison				
Documentation & Communication				

Checkpoint 2



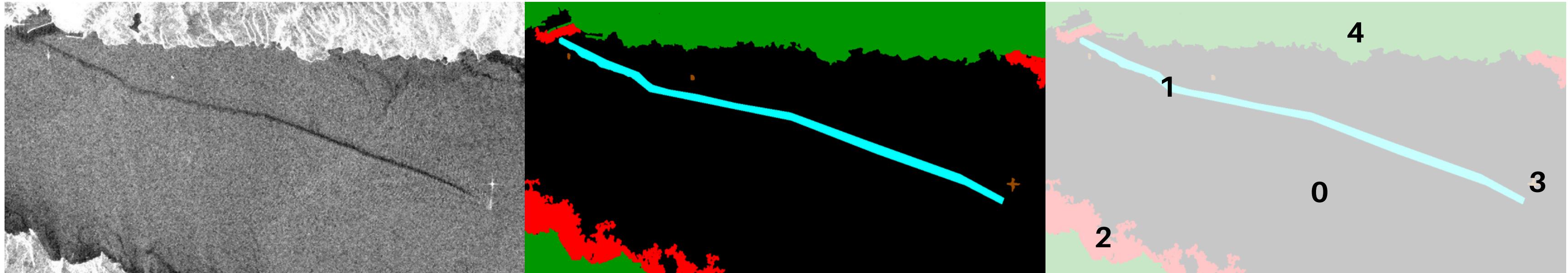


DATASET EXPLORATION

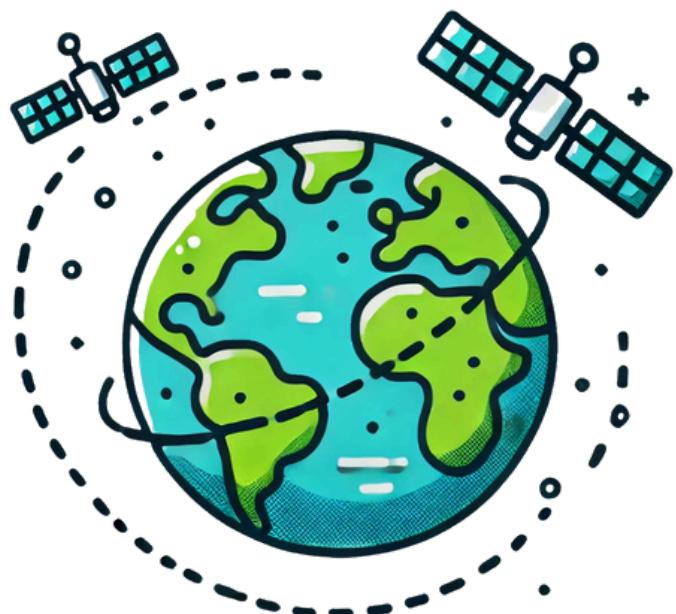
Images are taken by the ESA Satellite Sentinel-1

All images have dimensions 1250 x 650

Patches of dimension 320 x 320 are fed to models



Original SAR image
3 greyscale identical channels



Ground truth mask
3 rgb channels

	Sea Surface
	Oil Spill
	Look-alike
	Ship
	Land

Explicit labels mask
1 label channel

0	Sea Surface
1	Oil Spill
2	Look-alike
3	Ship
4	Land

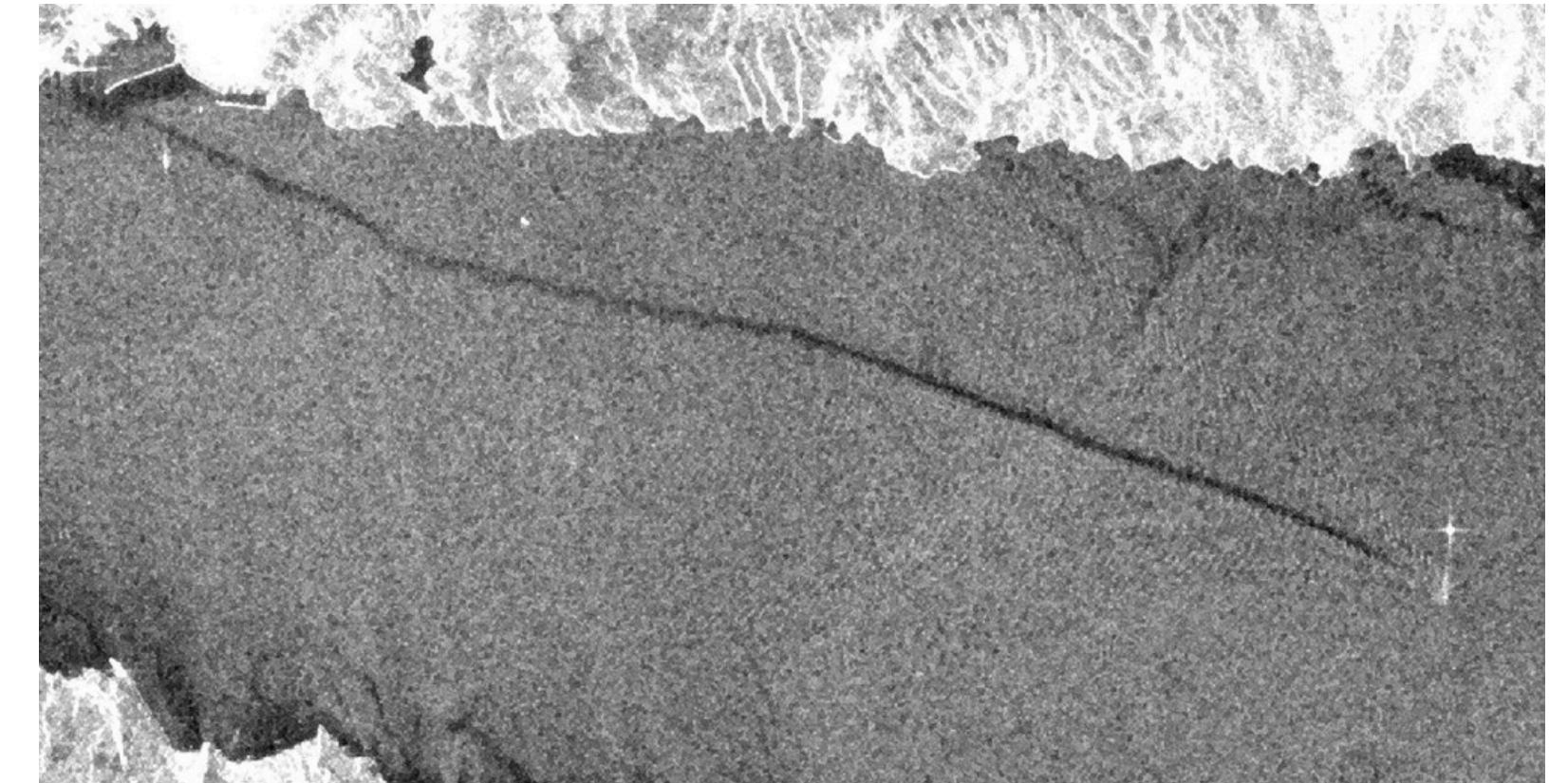




DATA PREPROCESSING

PAPER AUGMENTATIONS

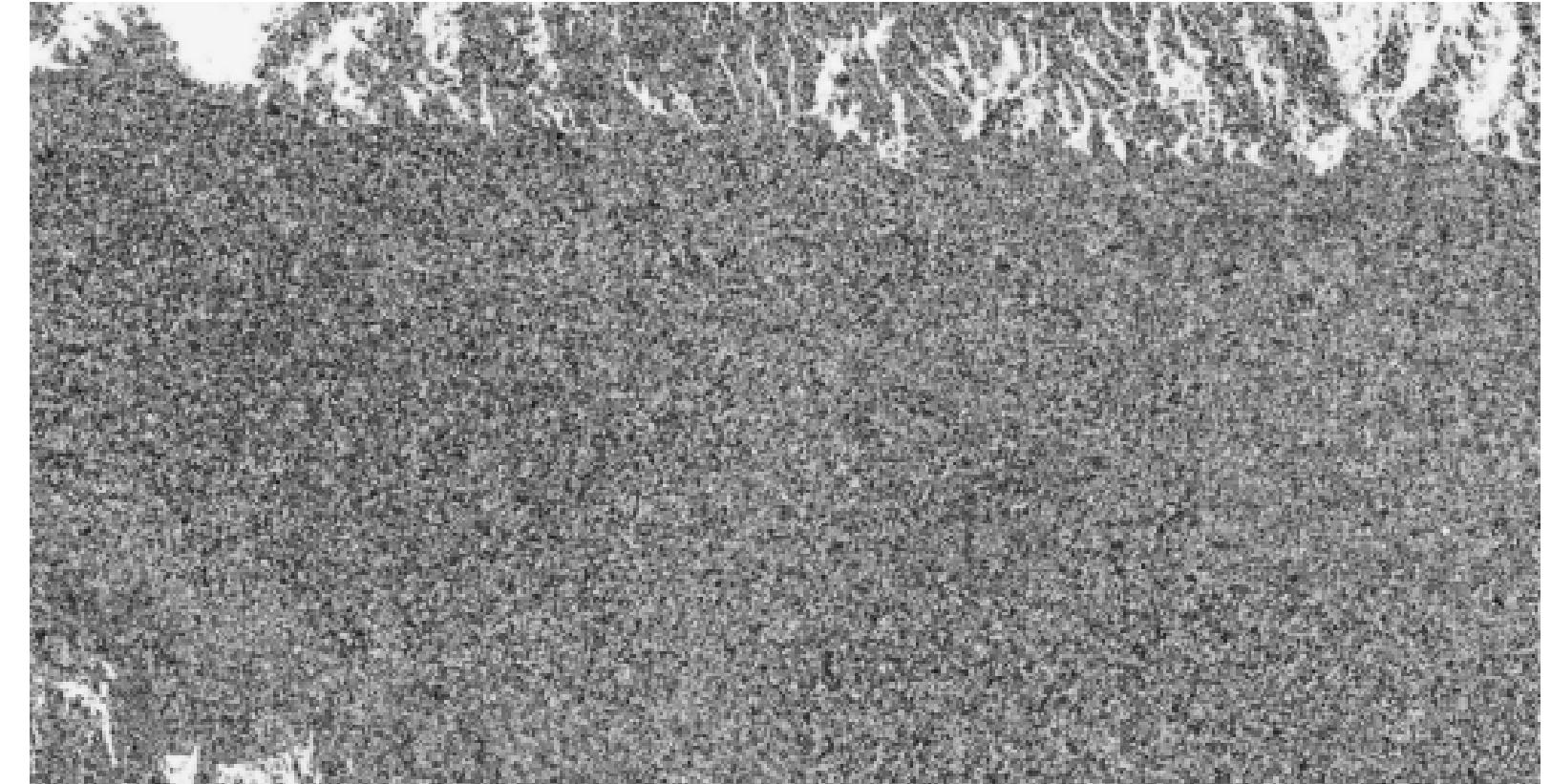
Starting from the original image, the original paper performs:



PAPER AUGMENTATIONS

Starting from the original image, the original paper performs:

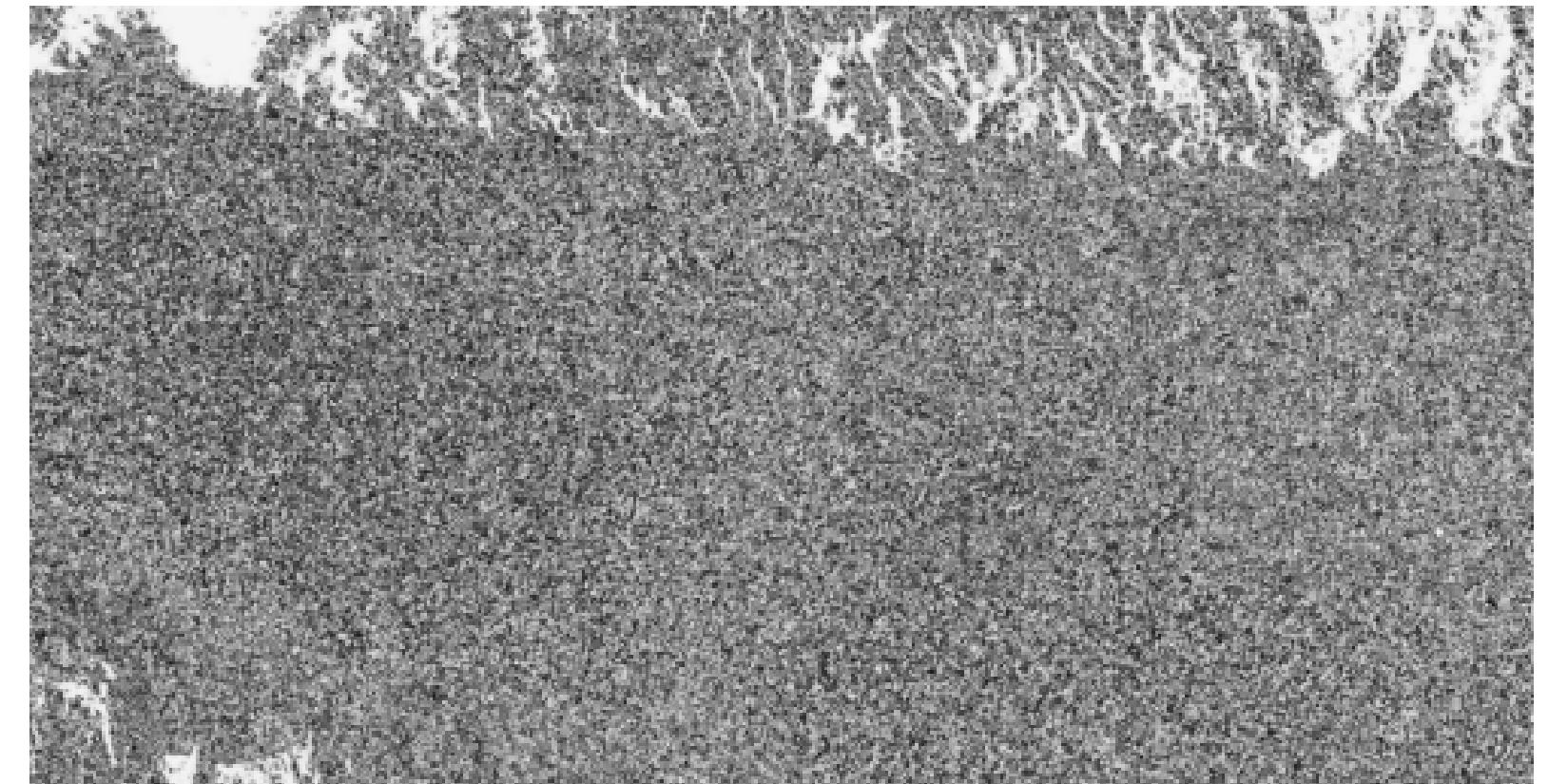
- Normalization



PAPER AUGMENTATIONS

Starting from the original image, the original paper performs:

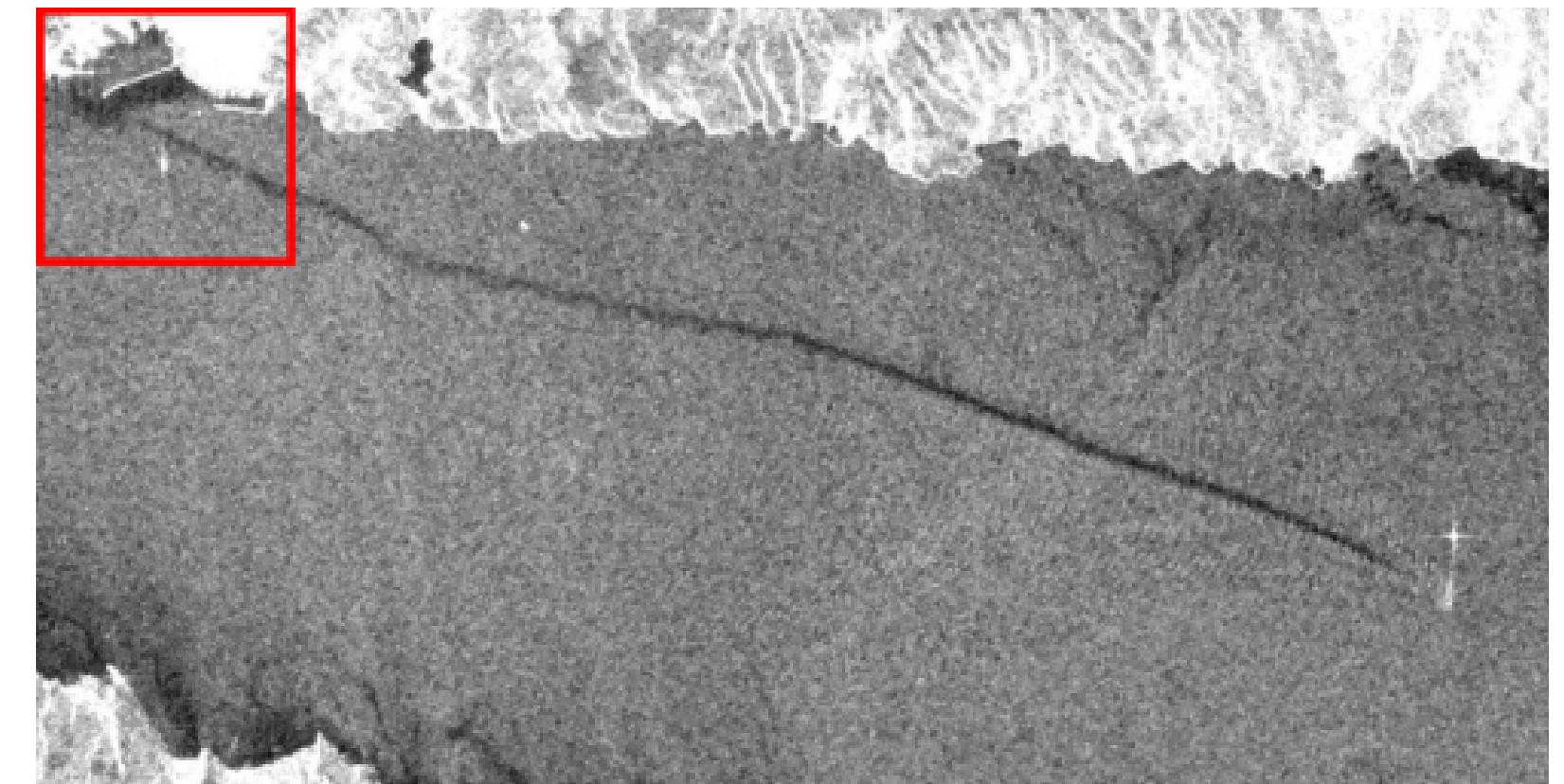
- **Normalization**
- **Random resize**
with a scale from $0.5x$ to $1.5x$



PAPER AUGMENTATIONS

Starting from the original image, the original paper performs:

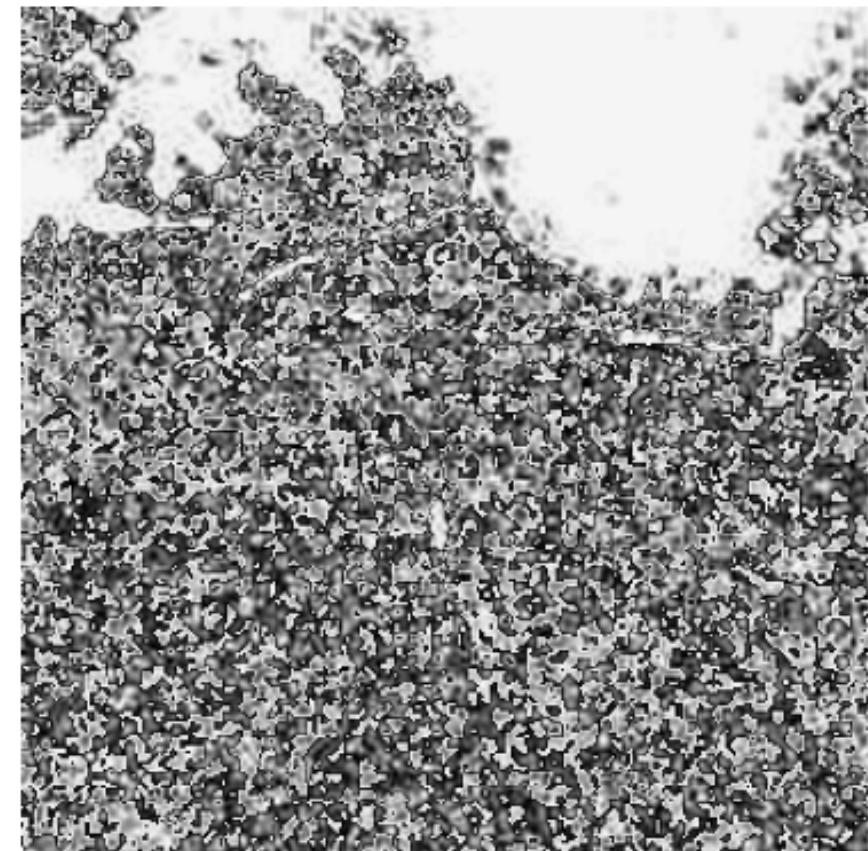
- Normalization
- Random resize
with a scale from $0.5x$ to $1.5x$
- **320x320 Crop**



PAPER AUGMENTATIONS

Starting from the original image, the original paper performs:

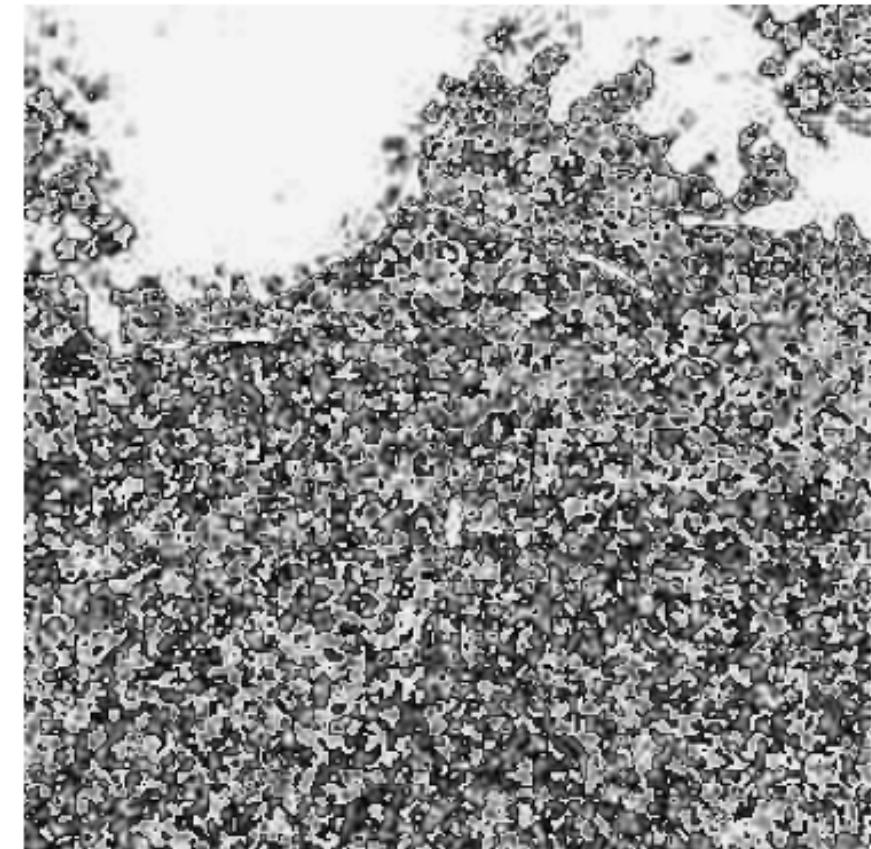
- **Normalization**
- **Random resize**
with a scale from $0.5x$ to $1.5x$
- **320x320 Crop**



PAPER AUGMENTATIONS

Starting from the original image, the original paper performs:

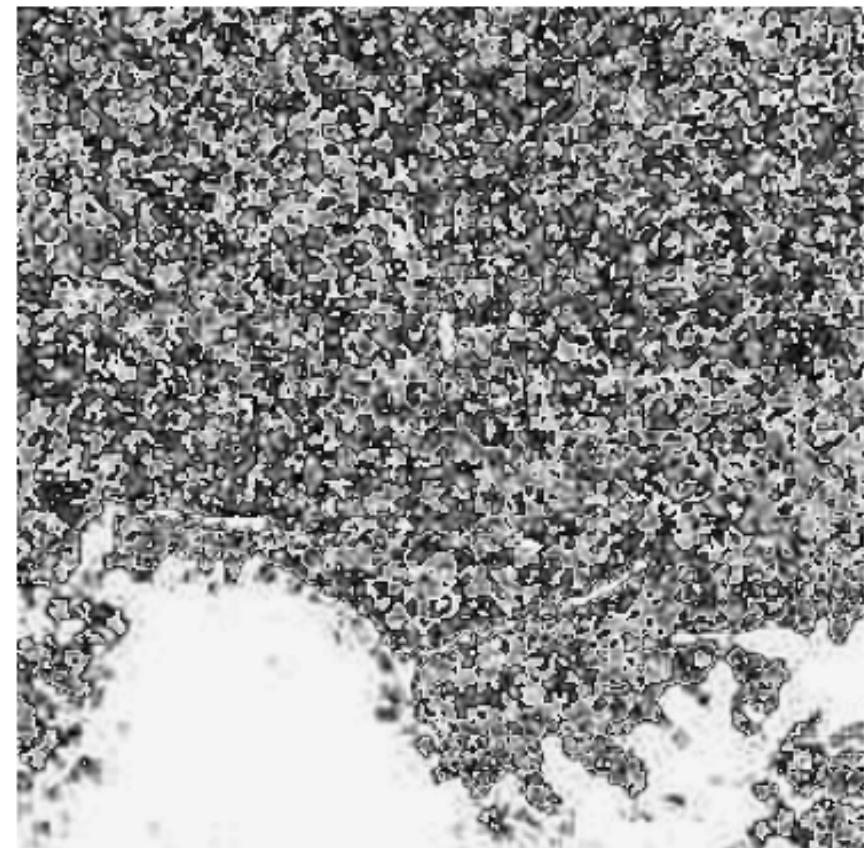
- **Normalization**
- **Random resize**
with a scale from $0.5x$ to $1.5x$
- **320x320 Crop**
- **Horizontal Flip**
with a probability of 50%



PAPER AUGMENTATIONS

Starting from the original image, the original paper performs:

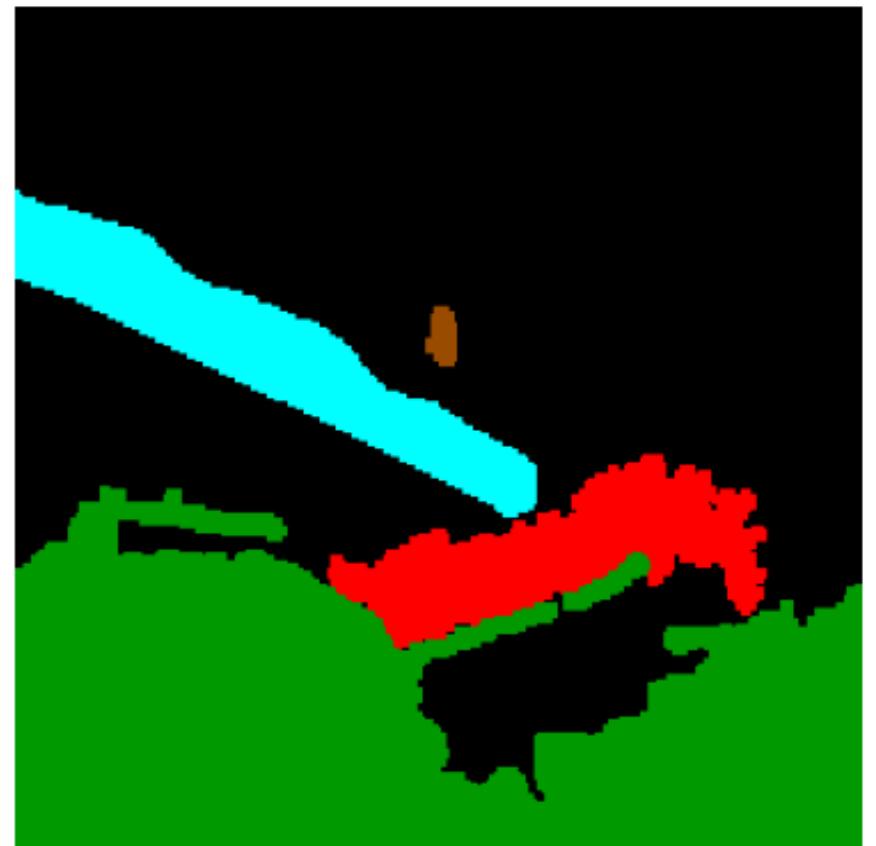
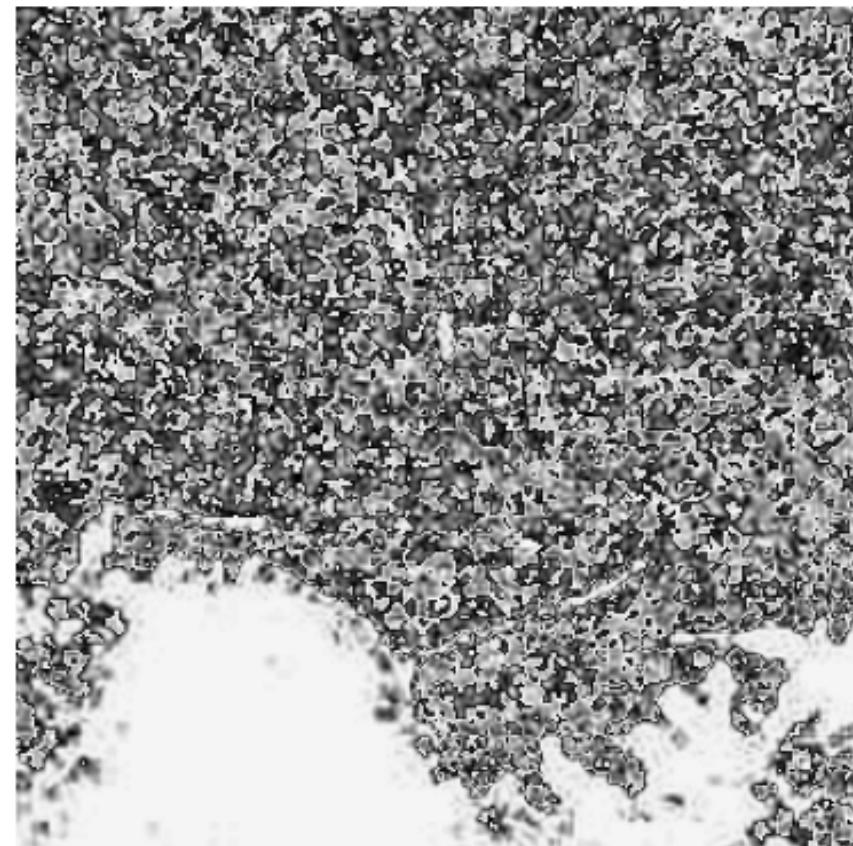
- **Normalization**
- **Random resize**
with a scale from $0.5x$ to $1.5x$
- **320x320 Crop**
- **Horizontal Flip**
with a probability of 50%
- **Vertical Flip**
with a probability of 50%



PAPER AUGMENTATIONS

Starting from the original image, the original paper performs:

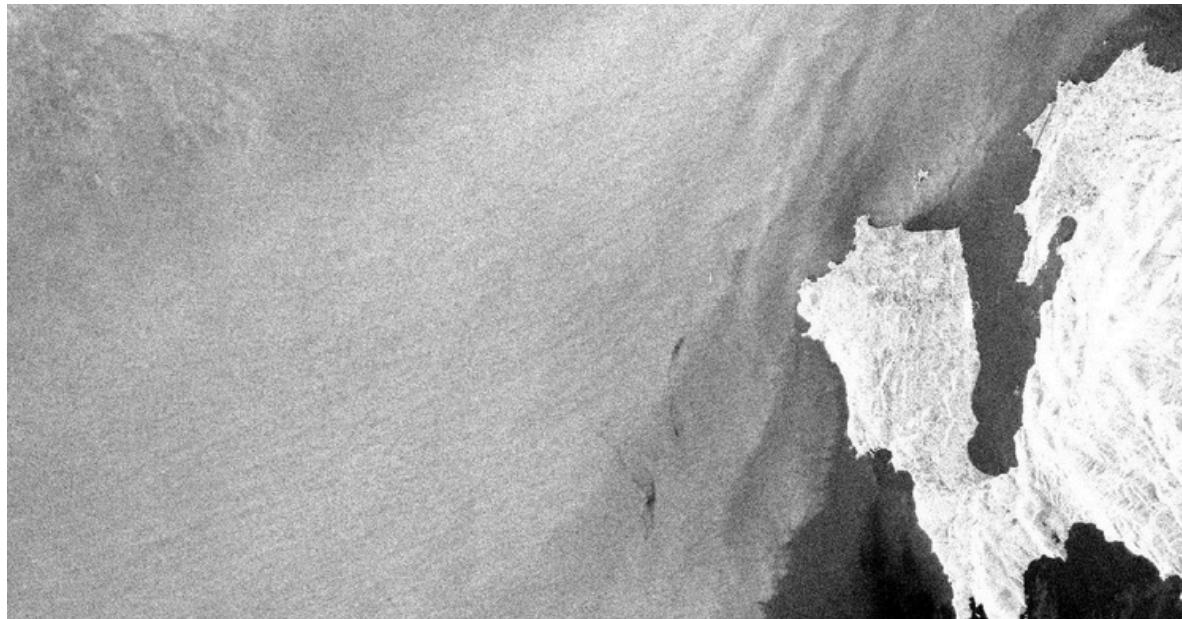
- **Normalization**
- **Random resize**
with a scale from $0.5x$ to $1.5x$
- **320x320 Crop**
- **Horizontal Flip**
with a probability of 50%
- **Vertical Flip**
with a probability of 50%



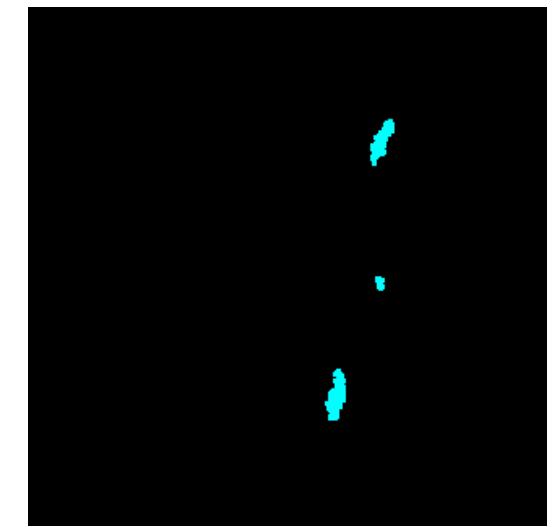
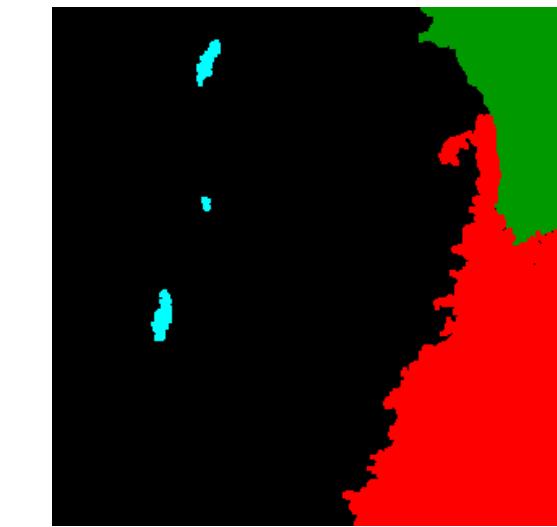
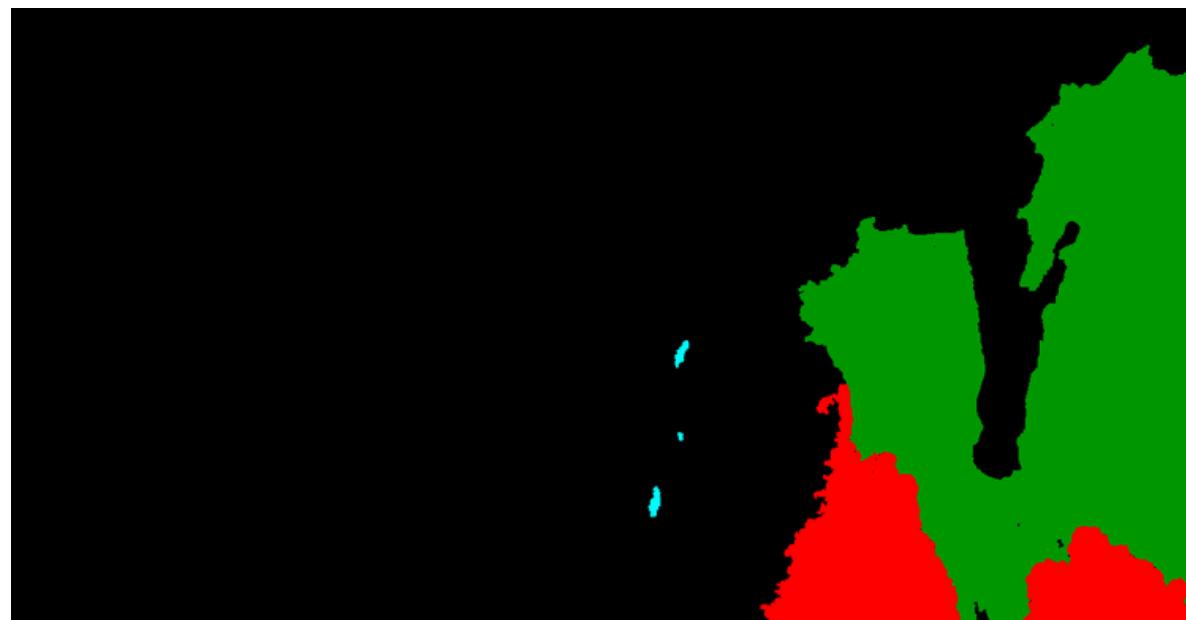
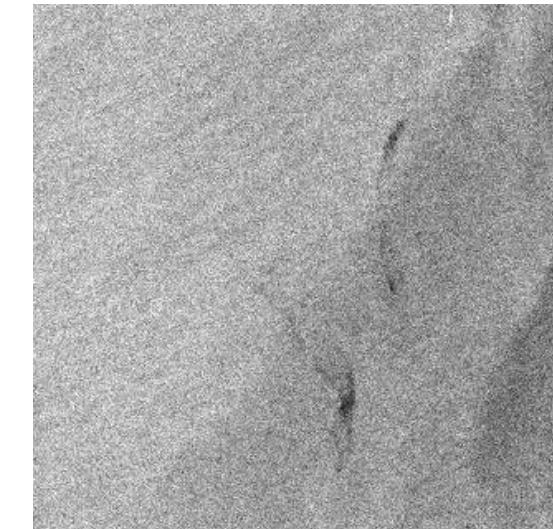
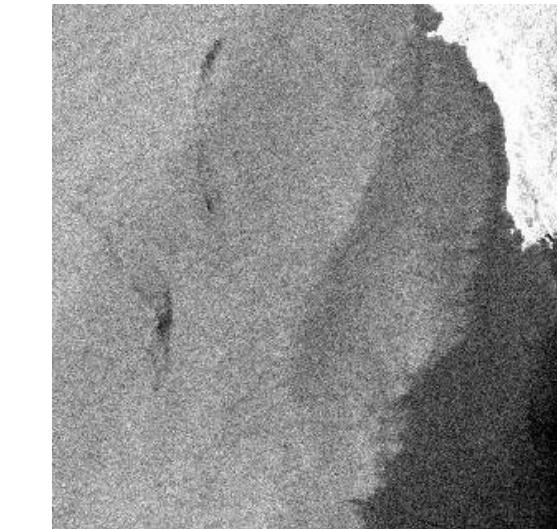
PROPOSED AUGMENTATIONS

- Focused Crop on Oil Spill

Original Image



Focused Crop Examples

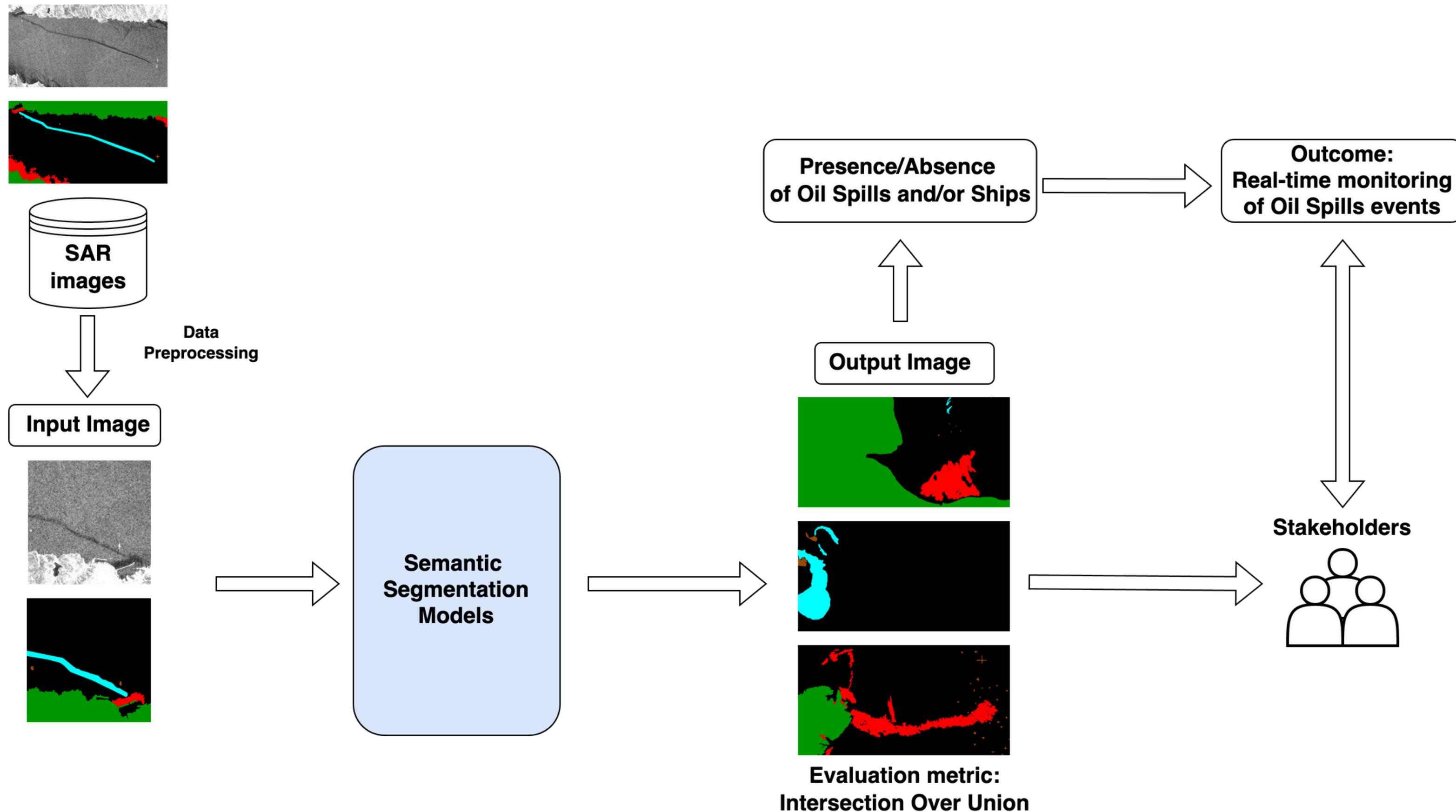


- Colors and shadows Augmentation
- ...



IMPLEMENTATION

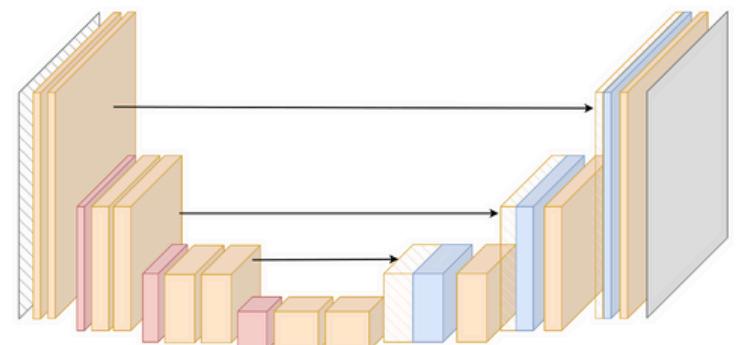
GENERAL OVERVIEW



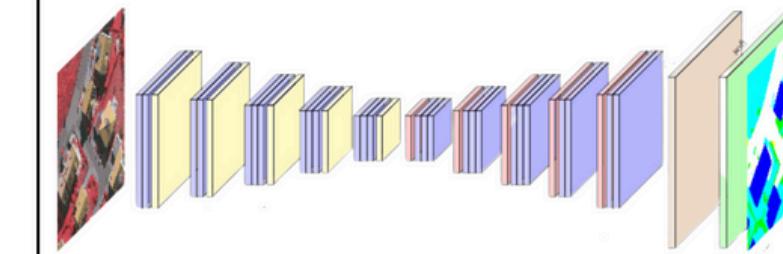
MODELS

Semantic
Segmentation
Models

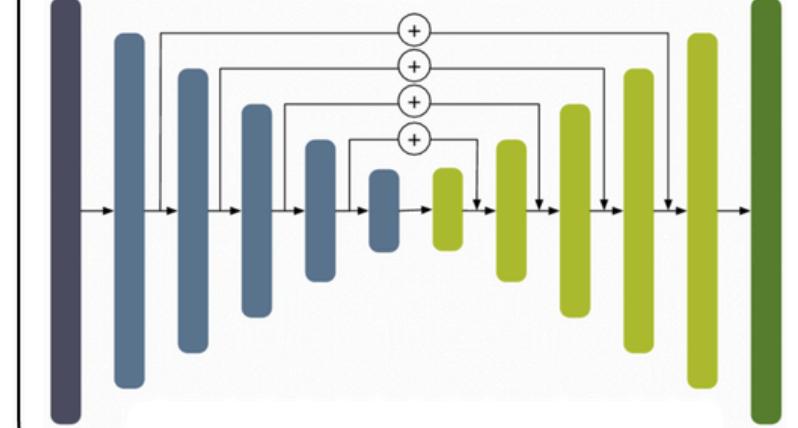
UNet



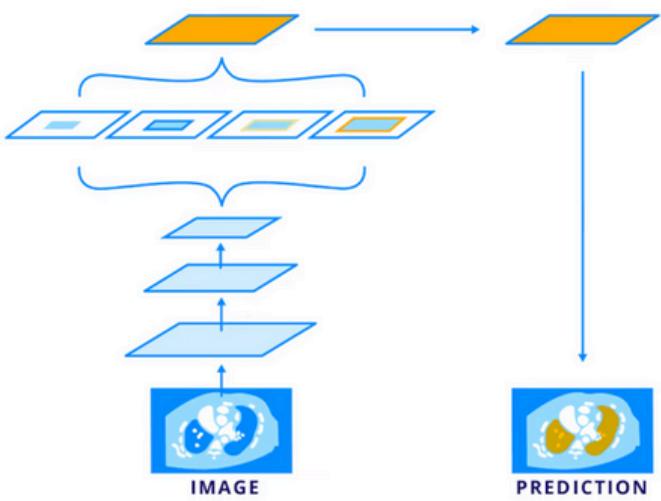
PSPNet



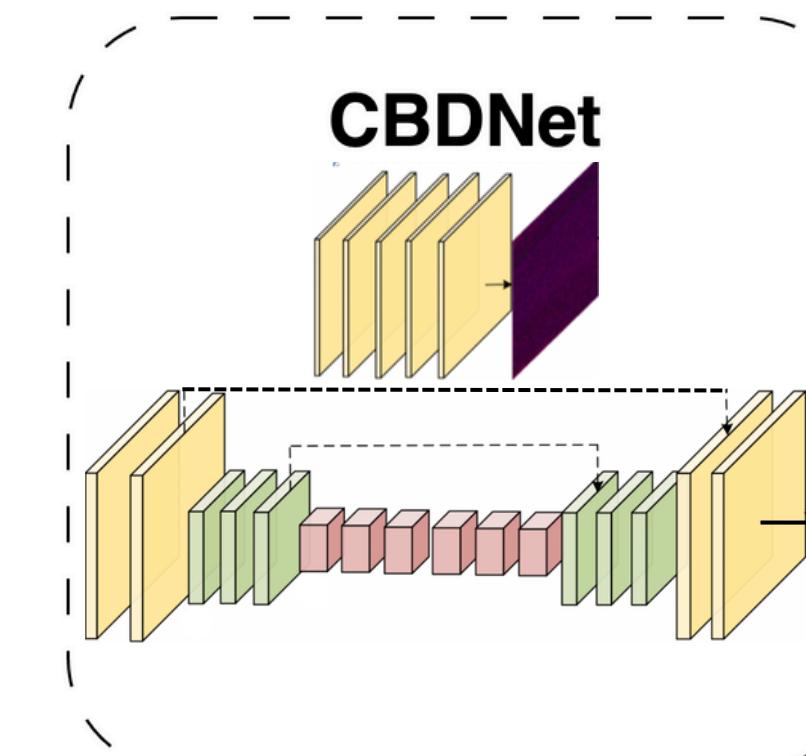
LinkNet



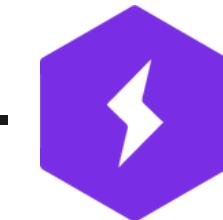
DeepLab



CBDNet



TRAINING -



PyTorch Lightning

The task:

Train and evaluate multiple models under varying preprocessing conditions to determine performance and robustness.

Chosen Framework:

We selected PyTorch Lightning as the primary framework for model development due to its support for modular workflows.

It streamlines data handling, training, and testing processes, making development more efficient and organized.

Dataset class:

```
class SARImageDataset(Dataset)
```

Loads and preprocesses raw data into usable data points for training

Data Module class:

```
class SARDataModule(LightningDataModule)
```

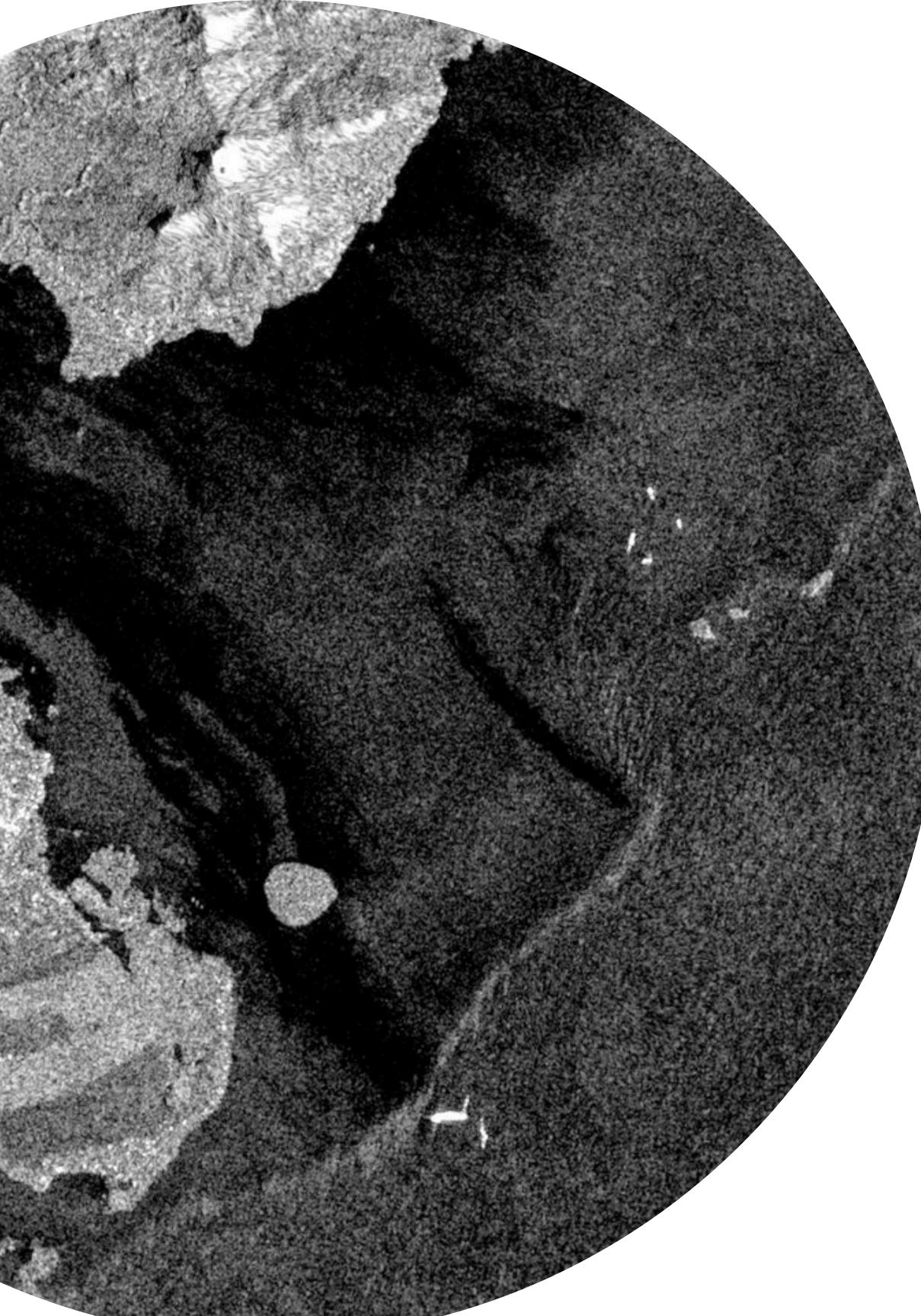
Manages data pipelines, including dataset splits and DataLoader creation

Lightning Module class:

```
class SARSegmentationModel(LightningModule)
```

Defines the model, training / validation / testing logic, and optimizers





Thank you for your attention

Any questions?

Bianca Bartoli - s322799

Carlo Marra - s334220

Alessandro Valenti - s328131

