

**Università Politecnica delle Marche**

**Dipartimento di Ingegneria dell'Informazione**

**Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione**

**Rilevazione di Oil Spill nelle immagini SAR Palsar e  
Sentinel-1 di bassa qualità**

**Progetto per l'esame di Computer Vision e Deep Learning**



**Relatore**

Prof. Lucia Migliorelli

**Studenti**

Angelo Kollcaku

Joshua Sgariglia

**Supervisore**

Prof. Alessandro Galdelli

Prof. Adriano Mancini

Prof. Stefano Mereu

**A.A. 2024/2025**

# 1 - Introduzione

Gli sversamenti di petrolio, o *Oil Spill*, costituiscono un grande problema per l'ecologia marina: sono pericolosi per la flora e la fauna marina, oltre a rappresentare una fonte di inquinamento ambientale. Il petrolio si diffonde rapidamente sulla superficie marina a causa del vento e delle correnti, aumentando l'area interessata e rendendo sempre più difficile l'arginamento dello sversamento.

Per questo motivo, negli ultimi anni l'uso di immagini SAR (*Synthetic Aperture Radar*) è diventato sempre più rilevante per il monitoraggio ambientale, in particolare per la rapida identificazione di sversamenti di petrolio in mare aperto. La capacità delle immagini SAR di operare indipendentemente da condizioni meteorologiche e luminose rende questa tecnologia estremamente efficace per applicazioni di sorveglianza marittima.

Tuttavia, l'analisi automatica di queste immagini rappresenta una sfida complessa a causa della natura del rumore *speckle*<sup>1</sup> e delle caratteristiche spesso ambigue degli oil spill. Per questo motivo, l'impiego di modelli di *deep learning*, e in particolare di architetture *U-Net*, si è diffuso ampiamente per il compito di segmentazione semantica.

Lo scopo di questo progetto è quello di implementare e confrontare diverse varianti di architetture U-Net, insieme a modelli di benchmark proposti nella letteratura, al fine di valutarne le prestazioni nell'identificazione di eventi di oil spill su immagini SAR fornite da satelliti.

---

<sup>1</sup> Tipo di rumore che influisce sulla varianza dei valori dei pixel piuttosto che sulla loro intensità media. Il rumore speckle è più evidente nelle aree più luminose dell'immagine, dove l'intensità del segnale è maggiore. Si presenta con un caratteristico pattern punteggiato.

## 2 - Analisi dello stato dell'arte

Il rilevamento di oil spill rappresenta un task di segmentazione. Allo stato attuale esistono tre principali insiemi di metodologie utilizzate a tal scopo: algoritmi basati su un *threshold*, tecniche di *machine learning* e tecniche di deep learning. I primi due metodi si basano esclusivamente o principalmente sulla scelta degli iperparametri del modello, il che può portare a scarsa capacità di generalizzare. L'utilizzo di reti neurali convoluzionali è la strada nata più recentemente ed è ad oggi la principale direzione di ricerca in questo campo.

In ambito deep learning, un primo contributo<sup>[1]</sup>, pubblicato su MDPI nel 2023, offre una versione migliorata del dataset SOS (*Deep SAR Oil Spill*)<sup>[2]</sup>, correggendo errori nelle etichette. Lo studio si concentra dunque sulle immagini SAR di bassa qualità del dataset, introducendo anche modelli avanzati come *TransUNet* e pipeline ibride che integrano tecniche di denoising (*FFDNet*), mostrando un netto miglioramento rispetto a modelli classici come *UNet*, *SegNet* e *DeepLabV3+*. Un altro lavoro recente<sup>[3]</sup>, pubblicato su ScienceDirect nel 2024, esplora sistematicamente 90 configurazioni di modelli deep learning per ottimizzare la classificazione e segmentazione di oil spill in immagini SAR di *Sentinel-1*, proponendo un nuovo dataset annotato<sup>[4]</sup>. Complementare a questi approcci, uno studio pubblicato nel 2022 su MDPI<sup>[5]</sup> propone una metodologia che utilizza il clustering in superpixel tramite l'algoritmo SLIC (*Simple Linear Iterative Clustering*), segmentando l'immagine in aree omogenee prima dell'analisi. A queste regioni viene poi applicata una tecnica di stretching per evidenziare la differenza tra petrolio e superficie marina, migliorando l'efficacia della classificazione con reti CNN. L'approccio è applicato a immagini radar dual-pol<sup>2</sup>, permettendo una rappresentazione più ricca delle caratteristiche fisiche delle superfici. I risultati mostrano un aumento del MIOU di oltre il 7% rispetto ai metodi che non usano stretching. Infine, un ulteriore studio del 2022 su IEEE<sup>[6]</sup> descrive una pipeline completa per la rilevazione di oil spill tramite dati Sentinel-1 dual-pol, impiegando decomposizioni polarimetriche.

Questi contributi rappresentano lo stato dell'arte più aggiornato. Nel progetto proposto si è scelto di basarsi sul dataset SOS corretto nel 2023 e messo a disposizione dagli autori del primo articolo, cercando di migliorare i risultati da loro ottenuti, tramite test su UNet e modelli da esso derivati. In particolar modo, si è osservato come nella letteratura sulla segmentazione di oil spill non sia presente l'uso della rete *UNet++* (2019)<sup>[7]</sup> e della famiglia dei modelli Mamba. Per questo motivo, la *UNet++* e la *LightM-UNet* (2024)<sup>[8]</sup> sono due delle reti che si è scelto di utilizzare.

---

<sup>2</sup> Immagini radar che utilizzano due polarizzazioni, una verticale (V) e l'altra orizzontale (H) per la trasmissione e ricezione del segnale. Questo permette di ottenere informazioni aggiuntive su ciò che si sta osservando, migliorando le capacità di identificazione e rilevamento.

## 3 - Materiali e metodi

### 3.1 - Dataset

Il dataset SOS corretto nel 2023 raccoglie le immagini scattate da due distinti gruppi di satelliti, *Palsar* e *Sentinel-1A*. Si tratta di circa 4000 immagini per ciascun gruppo, già divise in train set e test set in percentuali 80/20. Le immagini, di dimensioni 256x256 pixel, sono caratterizzate da una forte componente di rumore. Tutte, comprese le maschere, sono memorizzate in scala di grigi su tre canali uguali, in formato PNG. L'intero dataset occupa uno spazio di circa 1,3 GB. Le immagini sono considerate di scarsa qualità per via del rumore e della bassa risoluzione.

Secondo l'utilizzo che se fa nell'articolo del 2023, le maschere presentano due classi di segmentazione, con il bianco a indicare le macchie di petrolio e il nero per l'oceano. In alternativa, il petrolio può essere visto come unica classe di segmentazione, con il nero a indicare tutto il resto. In fase di training si è adottato quest'ultimo punto di vista, mentre in fase di valutazione del modello si è rimasti fedeli alla scelta fatta nell'articolo, per poter confrontare i risultati.

Gli autori dell'articolo hanno fatto uso della sezione "test" dei dataset sia come validation set che come test set. Nel progetto si è invece deciso di separare i due insiemi di dati, estraendo il validation set dal test set originale, in modo che le proporzioni finali siano 80/10/10.

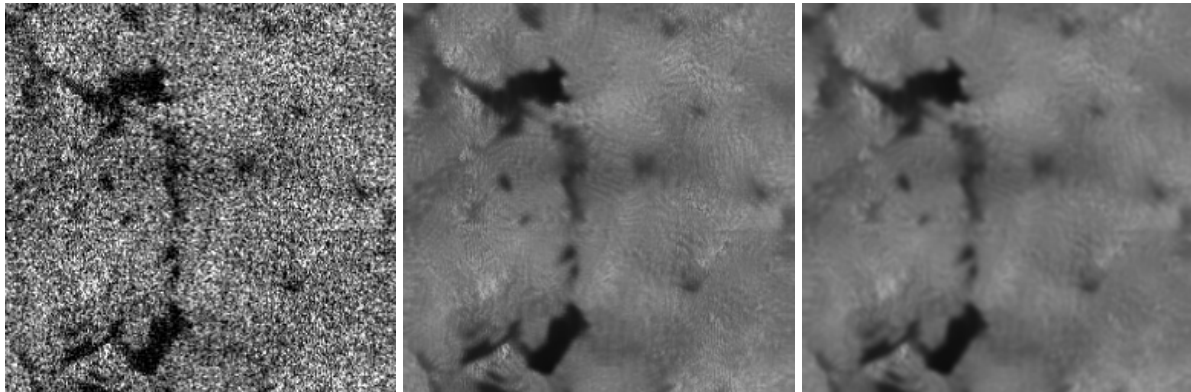
### 3.2 - Preprocessing

La fase di preprocessing mira a migliorare l'accuratezza delle predizioni riducendo la componente di rumore presente nelle immagini del dataset (*denoising*). Gli autori dell'articolo considerato hanno scelto di ricorrere alla *FFDNet*, una CNN preaddestrata per predire le versioni originali (senza rumore) di immagini a cui è stato aggiunto del rumore (queste ultime usate come input per addestrare la rete). Tuttavia, dai risultati forniti nell'articolo osserviamo che l'impatto della *FFDNet* non è stato molto rilevante: usata insieme alla *TransUNet* (2021)<sup>[9]</sup>, nel dataset *Palsar* ne ha peggiorato le prestazioni in tutte le metriche con la sola eccezione della *precision*, mentre nel dataset *Sentinel* ne ha migliorato *accuracy*, *recall* e *F1-score*, ma al contempo peggiorando *precision* e *mIoU*.

Per questo motivo, si sono tentate altre strade: i filtri che sono stati presi in considerazione sono il *Box filter*, il *Gaussian blur*, il *Median blur*, il *Bilateral filter* e il *Fast NL-Means*<sup>[10]</sup>.

Nel *Fast NL-Means*, per ogni pixel dell'immagine si considera una patch, cioè un piccolo blocco quadrato centrato su quel pixel. Intorno al pixel si definisce poi una finestra di ricerca più grande, all'interno della quale si cercano altre patch simili. Il confronto tra patch avviene pixel per pixel, calcolando una distanza pesata tra i valori corrispondenti nelle due patch. Le patch più simili ricevono un peso maggiore, e il nuovo valore del pixel viene ottenuto tramite una media pesata dei pixel centrali delle patch presenti nella finestra.

La combinazione più efficace è risultata essere l'applicazione del Fast NL-Means seguita da un filtro gaussiano. Tuttavia, poiché l'applicazione a runtime del Fast NL-Means a un'immagine richiede un tempo non trascurabile (il tempo di addestramento di una UNet viene più che raddoppiato), l'intero dataset SOS è stato preprocessato secondo tale filtro. Il nuovo dataset ottenuto è quello usato per l'addestramento delle reti.



*Figura 1 - Confronto tra immagine originale (sinistra), immagine dopo il Fast NL Means (centro) e dopo Fast NL Means + Gaussian filter (destra)*

Il resto del preprocessing avviene a runtime durante la generazione dei batch di training e di validazione: per ragioni di efficienza in termini di memoria occupata, vengono caricate soltanto le immagini e le maschere che fanno parte del batch; successivamente, vengono rimossi due dei tre canali (poiché ridondanti) e viene applicato il filtro gaussiano alle immagini; infine le immagini sono normalizzate, mentre le maschere vengono binarizzate.

### 3.3 - Reti

Le reti utilizzate in questo progetto sono UNet, UNet++, TransUNet e LightMUNet. A queste si aggiungono due varianti definite nell'ambito di questo lavoro: si tratta di versioni semplificate di UNet e UNet++, denominate rispettivamente *UNet Light* (UNetL) e *UNet++ Light* (UNet++L). Queste architetture alleggerite sono ottenute rimuovendo un livello sia dall'encoder che dal decoder delle reti originali, riducendo così la profondità complessiva del modello e il numero di parametri.

Le reti sono state definite usando *Tensorflow* e *Keras*, rispettando il più possibile l'implementazione classica. La rete TransUNet è la stessa di quella utilizzata nell'articolo di riferimento, ottenuta dalla libreria *TransUNet-tf*<sup>[11]</sup>. Per quanto riguarda la LightMUNet, l'architettura definita ricalca quella del suo articolo originale, ma è stato necessario alleggerire la rete semplificando il modulo SSM a causa della sua pesantezza in termini sia di memoria occupata che di tempo impiegato per l'addestramento.

UNet++ è un'evoluzione dell'architettura UNet progettata per migliorare l'accuratezza della segmentazione attraverso una più efficace fusione delle

feature tra encoder e decoder. A differenza di UNet, in cui le skip connection collegano direttamente i livelli corrispondenti, UNet++ introduce una serie di connessioni nidificate e blocchi convoluzionali intermedi tra i diversi stadi, che riducono il divario semantico tra feature a bassa e alta risoluzione. L'effetto complessivo è una segmentazione più precisa, ma con lo svantaggio di una maggiore complessità della rete, che si traduce in tempi di addestramento più lunghi.

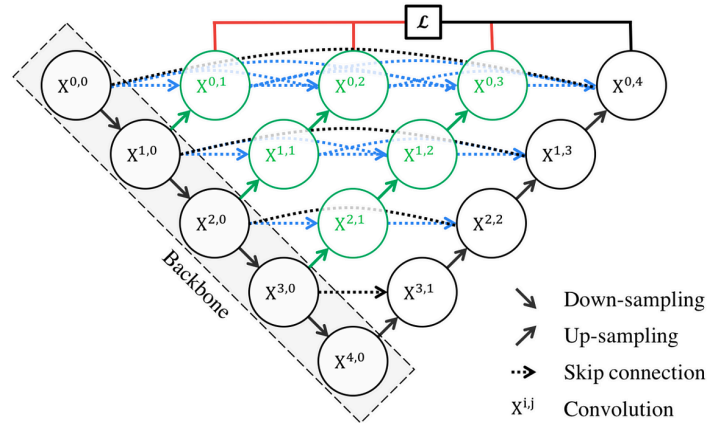


Figura 2 - Architettura della UNet++

TransUNet combina i punti di forza delle CNN e dei Transformer in un'unica architettura per la segmentazione. L'encoder è composto da una ResNet-50 che estrae feature locali e gerarchiche, seguita da un Transformer che rielabora queste feature in forma sequenziale per catturare dipendenze a lungo raggio nell'immagine. Le rappresentazioni così ottenute vengono poi decodificate attraverso un decoder simile a quello della UNet, con upsampling progressivo e skip connection dai livelli convoluzionali iniziali. Questo design ibrido consente di combinare la precisione spaziale delle CNN con la capacità di modellare contesti globali tipica dei Transformer.

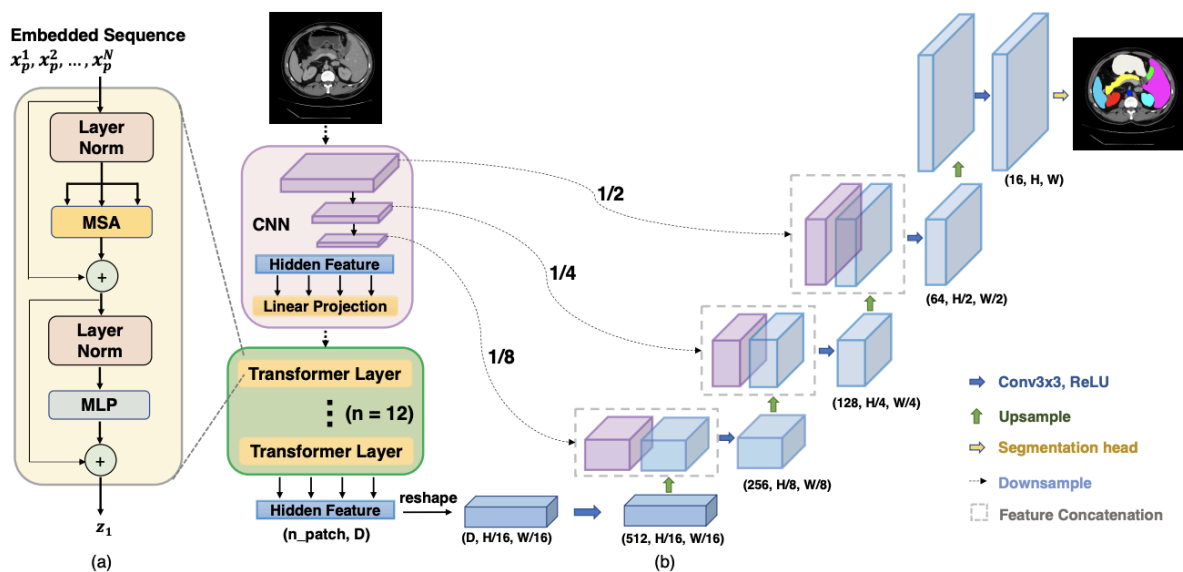


Figura 3 - Architettura della TransUNet

LightMUNet adotta una classica struttura a “U” ma sostituisce le classiche convoluzioni e i transformer con blocchi Mamba – un tipo di State Space Model (SSM) a complessità lineare – per modellare dipendenze spaziali a lungo raggio. Dopo un’estrazione iniziale delle caratteristiche tramite depthwise separable convolution poco costose, l’architettura si sviluppa in un encoder-decoder simmetrico: ogni livello utilizza Residual Vision Mamba Layers per catturare rappresentazioni semantiche profonde, mantenendo connessioni skip per preservare i dettagli spaziali.

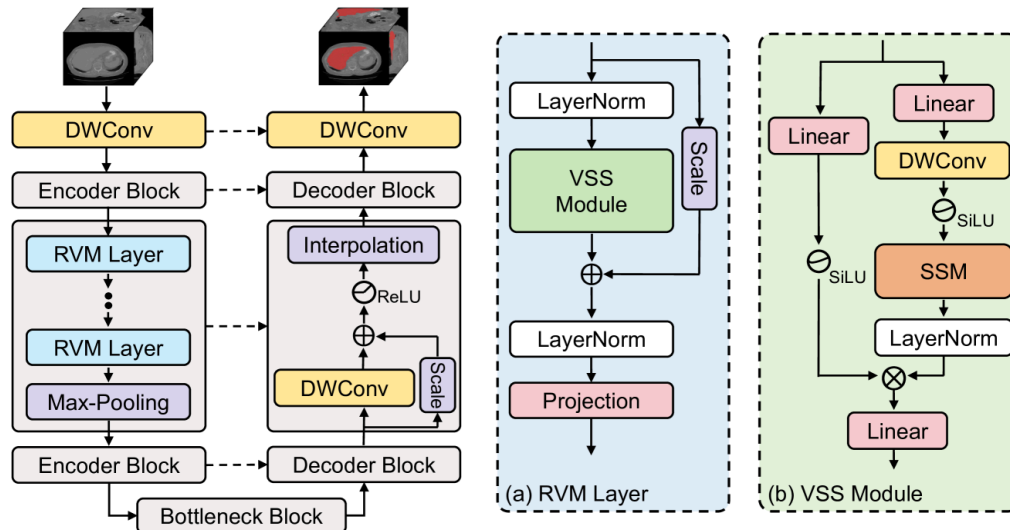


Figura 4 - Architettura della LightMUNet

### 3.4 - Training

Per ciascuna rete, in un primo momento si è cercato un insieme di iperparametri che fornisse dei risultati accettabili. Questa prima fase è stata di tipo trial-and-error, ed aveva l'obiettivo di trovare un punto di partenza per un fine-tuning automatico, basato cioè sull'addestramento consecutivo della rete su una serie di iperparametri entro certi range prefissati. In particolare, il *learning rate* iniziale è l'iperparametro più importante che i tentativi effettuati manualmente puntavano a determinare. Successivamente, per ciascun dataset, le reti sono state addestrate su un totale di 60 configurazioni: 12 per UNet e UNetL, 8 per UNet++ e UNet++L, 16 per TransUNet (di cui metà relative a TransUNet preaddestrata su ImageNet) e 4 per LightMUNet.

Le configurazioni sono ottenute variando il numero delle immagini in un batch e gli iperparametri relativi al decadimento del learning rate, nello specifico la pazienza, ossia il numero di epoche consecutive prive di miglioramento della loss necessarie a causare il decadimento, e il tasso di decadimento.

Altri iperparametri rilevanti sono il limite minimo sotto il quale il learning rate non può scendere e il numero di epoche di addestramento.

Rete	Epoche	LR	Min LR	Pazienza	Tasso dec	Batch size
UNet	50	5e-5	1e-6	3 - 4	0,4 - 0,5	8 - 16 -32
UNetL	50	1e-4	1e-6	3 - 4	0,4 - 0,5	8 - 16 -32
UNet++	40	2,5e-5	1e-7	2 - 3	0,4 - 0,5	8 - 16
UNet++L	40	5e-5	5e-7	2 - 3	0,4 - 0,5	8 - 16
TransUNet	40	1e-4	5e-7	2 - 3	0,3 - 0,5	8 - 16
LightMUNet	40	1e-5	1e-7	2 - 3	0,3 - 0,5	4

*Tabella 1 - Configurazioni testate per ciascuna rete*

L'ottimizzatore usato è Adam. La scelta per la funzione di loss è ricaduta sulla *binary cross-entropy* (BCE). Altre funzioni considerate inizialmente sono la *dice loss* e una composizione di dice loss e BCE, tuttavia non vi è differenza rilevante nelle prestazioni delle reti addestrate.

Ad ogni addestramento vengono salvati, oltre al modello, anche un log delle epoche e i valori degli iperparametri.

### 3.5 - Evaluation

La valutazione dei modelli addestrati consiste nel calcolo di un insieme di metriche che hanno lo scopo di misurarne le prestazioni. Le metriche scelte sono: accuracy, precision, recall, F1-score e IoU. L'accuracy è l'unica metrica globale, mentre tutte le altre sono state valutate sia separatamente per la classe di oil spill e quella di non oil spill (background), sia globalmente effettuando una media tra i due valori ottenuti. Il calcolo dei valori medi delle metriche consente inoltre di effettuare un confronto con i modelli presentati nell'articolo scientifico di riferimento, in quanto non sono riportati tutti i valori per singole classi.

Sono state calcolate e salvate anche delle metriche temporali, tra cui il tempo complessivo di addestramento e di valutazione del modello, il tempo di addestramento per singola epoca e il tempo di inferenza.

Sono state infine misurate le emissioni di CO<sub>2</sub> generate durante l'addestramento dei modelli usando la libreria *codecarbon*.



## 3 - Risultati e discussione

### 3.1 - Preprocessing

Per determinare la miglior impostazione del preprocessing sono stati eseguiti test con i filtri usando il dataset Palsar e una UNetL.

Filtro	Accuracy	mPrecision	mRecall	mF1-score	mIoU
-	93,1%	87,0%	88,9%	87,9%	79,3%
Gaussian	93,1%	87,0%	<b>89,3%</b>	88,1%	79,6%
Median	93,3%	87,8%	88,2%	88,0%	79,5%
Box	93,2%	87,3%	88,9%	88,1%	79,6%
Bilateral	93,3%	87,7%	88,4%	88,1%	79,6%
Fast NL Means	93,5%	88,3%	88,3%	88,3%	79,9%
Fast NL Means + Gaussian	<b>93,7%</b>	<b>89,0%</b>	88,2%	<b>88,6%</b>	<b>80,4%</b>
Fast NL Means + Median	93,6%	88,9%	88,1%	88,5%	80,2%
Fast NL Means + Box	93,6%	88,5%	88,6%	<b>88,6%</b>	80,3%
Fast NL Means + Bilateral	93,6%	88,9%	87,8%	88,3%	80,0%

*Tabella 2 - Metriche ottenute dai test sul denoising*

Dai risultati ottenuti osserviamo che l'applicazione di un filtro base (Gaussian, Median, Box, Bilateral) comporta un miglioramento medio dello 0,2% per mF1-score e dello 0,3% per mIoU; il Fast NL-Means consente invece di ottenere un miglioramento dello 0,4% per mF1-score e dello 0,6% per mIoU. I risultati migliori si ottengono però applicando un filtro base successivamente al Fast NL-Means: la combinazione che assicura le prestazioni più alte è il Fast NL-Means seguito da filtro gaussiano, con un +0,7% per mF1-score e un +1,1% per mIoU.

### 3.2 - Training e performance

L'addestramento delle reti è avvenuto in locale con una scheda video NVIDIA GeForce RTX 4070.

I risultati presentati sono divisi per dataset. Per ogni dataset, mostriamo tre tabelle, una per classe e una con i valori medi delle metriche di valutazione. Le metriche riportate sono quelle relative alla miglior configurazione.

### 3.2.1 - Palsar

Rete	Precision	Recall	F1-score	IoU
UNetL	<b>82,0%</b>	80,0%	81,0%	68,0%
UNet	80,1%	82,2%	81,1%	68,3%
UNet++L	80,6%	81,3%	80,9%	68,0%
UNet++	80,7%	82,0%	<b>81,3%</b>	<b>68,5%</b>
TransUNet	78,6%	82,6%	80,6%	67,5%
TransUNet preaddestrata	79,4%	<b>82,9%</b>	81,1%	68,3%
LightMUNet	82,2%	76,5%	79,2%	65,6%

*Tabella 3 - Metriche di valutazione per dataset Palsar sulla classe Oil Spill*

Rete	Precision	Recall	F1-score	IoU
UNetL	96,0%	<b>96,5%</b>	<b>96,2%</b>	<b>92,7%</b>
UNet	96,4%	95,6%	96,1%	92,6%
UNet++L	96,2%	96,1%	96,1%	92,6%
UNet++	96,4%	96,0%	<b>96,2%</b>	<b>92,7%</b>
TransUNet	<b>96,5%</b>	95,5%	96,0%	92,2%
TransUNet preaddestrata	<b>96,5%</b>	95,7%	96,1%	92,5%
LightMUNet	95,3%	96,7%	96,0%	92,3%

*Tabella 4 - Metriche di valutazione per dataset Palsar sulla classe Background*

Rete	Accuracy	mPrecision	mRecall	mF1-score	mIoU
UNetL	<b>93,7%</b>	<b>89,0%</b>	88,2%	88,6%	80,4%
UNet	93,6%	88,2%	89,0%	88,6%	80,4%
UNet++L	93,6%	88,4%	88,7%	88,5%	80,3%
UNet++	<b>93,7%</b>	88,5%	89,0%	<b>88,8%</b>	<b>80,6%</b>

TransUNet	93,3%	87,5%	89,1%	88,3%	79,9%
TransUNet preaddestrata	93,5%	88,0%	<b>89,3%</b>	88,6%	80,4%
LightMUNet	93,3%	88,8%	86,6%	87,6%	78,9%

*Tabella 5 - Metriche complessive di valutazione per dataset Palsar*

La rete con le performance più alte sul dataset Palsar è la UNet++, seguita dalla TransUNet preaddestrata e dalle UNet. La TransUNet preaddestrata performa meglio in tutte le metriche rispetto alla versione non preaddestrata. È interessante notare che tutte le reti ottengono punteggi migliori sulla classe di background rispetto a quella di oil spill, a indicare che la tendenza è quella di assegnare più spesso pixel di oil spill alla classe background che viceversa. Si ritiene che ciò sia dovuto al fatto che piccole macchie isolate di petrolio vengono etichettate dalle reti come background a causa della forte componente di rumore.

### 3.2.2 - Sentinel

Rete	Precision	Recall	F1-score	IoU
UNetL	85,5%	81,5%	83,5%	71,6%
UNet	84,2%	<b>83,7%</b>	83,9%	72,3%
UNet++L	84,5%	82,0%	83,3%	71,3%
UNet++	84,7%	83,6%	84,1%	72,6%
TransUNet	85,6%	81,3%	83,4%	71,6%
TransUNet preaddestrata	85,7%	83,4%	<b>84,5%</b>	<b>73,2%</b>
LightMUNet	<b>85,7%</b>	76,8%	81,0%	68,0%

*Tabella 6 - Metriche di valutazione per dataset Sentinel sulla classe Oil Spill*

Rete	Precision	Recall	F1-score	IoU
UNetL	90,3%	92,6%	91,5%	84,3%
UNet	<b>91,3%</b>	91,6%	91,4%	84,2%
UNet++L	90,5%	92,0%	91,2%	83,9%

UNet++	91,2%	91,9%	91,6%	84,5%
TransUNet	90,3%	<b>92,7%</b>	91,5%	84,3%
TransUNet preaddestrata	91,2%	92,5%	<b>91,9%</b>	<b>85,0%</b>
LightMUNet	88,2%	93,1%	90,6%	82,8%

*Tabella 7 - Metriche di valutazione per dataset Sentinel sulla classe Background*

Rete	Accuracy	mPrecision	mRecall	mF1-score	mIoU
UNetL	88,7%	87,9%	87,1%	87,5%	78,0%
UNet	88,8%	87,7%	87,6%	87,7%	78,3%
UNet++L	88,5%	87,5%	87,0%	87,2%	77,6%
UNet++	89,0%	88,0%	87,7%	87,9%	78,6%
TransUNet	88,7%	87,9%	87,0%	87,4%	77,9%
TransUNet preaddestrata	<b>89,3%</b>	<b>88,5%</b>	<b>88,0%</b>	<b>88,2%</b>	<b>79,1%</b>
LightMUNet	87,4%	86,9%	85,0%	85,8%	75,4%

*Tabella 8 - Metriche complessive di valutazione per dataset Sentinel*

La rete con le performance più alte sul dataset Sentinel è la TransUNet preaddestrata, seguita dalla UNet++ e dalla UNet. Anche in questo caso la TransUNet preaddestrata performa meglio in tutte le metriche rispetto alla versione non preaddestrata. Come nel dataset Palsar, le reti ottengono punteggi migliori sulla classe di background rispetto a quella di oil spill, ma con scarto minore.

Confrontando le performance ottenute con quelle dell'articolo su cui ci si è basati, si può osservare che il lavoro presente è riuscito a ottenere un miglioramento di circa un punto percentuale in entrambi i dataset. Tale risultato è dovuto in gran parte al preprocessing più avanzato, e secondariamente ad altri aspetti, come l'adozione di un algoritmo di decadimento del learning rate, che previene l'overfitting, e la normalizzazione dell'input; questi due accorgimenti non sono stati impiegati nell'articolo di riferimento.

In particolare, usando la mIoU come metrica di confronto tra i rispettivi migliori modelli, si è ottenuto un miglioramento dello 0,8% sul dataset Palsar e del 1,0% sul dataset Sentinel. Confrontando gli stessi modelli sull'accuracy, si può invece

osservare un miglioramento dello 0,2% sul dataset Palsar e dello 0,8% sul dataset Sentinel.

Per quanto riguarda le altre metriche, non è possibile effettuare un confronto adeguato in quanto l'articolo riporta solo tali metriche per la classe background.

### 3.3 - Statistiche temporali

Durante l'addestramento sono state misurate anche delle metriche temporali, presentate nelle tabelle seguenti. I valori sono misurati in secondi.

Rete	Epoch	Training	Evaluation	Inference
UNetL	17,467	873,344	20,997	0,054
UNet	19,796	989,812	25,001	0,064
UNet++L	36,975	1478,998	24,807	0,064
UNet++	56,640	2265,591	26,097	0,067
TransUNet	80,659	3226,372	43,342	0,112
LightMUNet	217,646	8705,848	45,816	0,118

*Tabella 9 - Statistiche temporali in secondi per il dataset Palsar*

Rete	Epoch	Training	Evaluation	Inference
UNetL	17,947	897,325	25,327	0,060
UNet	19,980	999,014	26,992	0,064
UNet++L	55,494	2219,777	51,860	0,124
UNet++	59,959	2398,357	29,269	0,070
TransUNet	90,643	3625,714	44,248	0,105
LightMUNet	238,036	9521,456	56,949	0,136

*Tabella 10 - Statistiche temporali in secondi per il dataset Sentinel*

Le misurazioni temporali riportate sono indicative poiché l'addestramento è avvenuto anche durante l'utilizzo dell'hardware per altri scopi. Come ci si aspettava, le reti più complesse presentano tempi di addestramento e inferenza maggiori. Inoltre, il tempo di inferenza è misurato sul dataset preprocessato con il Fast NL-Means. Sul dataset originale, si stima che l'inferenza di un'immagine necessiti di un tempo maggiore di 0,070 - 0,100 rispetto ai valori indicati.

### 3.4 - Monitoraggio delle emissioni

Di seguito sono presentati i risultati del monitoraggio delle emissioni durante l'addestramento automatico. Ciò non comprende le emissioni durante gli addestramenti di prova. Le misurazioni sono in grammi di CO<sub>2</sub>eq. Le prime quattro colonne si riferiscono alla singola rete, l'ultima a tutte le configurazioni.

Rete	Epoch	Training	Evaluation	Train+Eval	Total
UNetL	0,35	17,30	0,14	17,44	209,23
UNet	0,40	20,13	0,15	20,28	243,27
UNet++L	0,77	30,91	0,20	31,11	248,84
UNet++	1,19	47,47	0,20	47,67	381,32
TransUNet	1,63	65,14	0,32	65,46	523,66
LightMUNet	3,79	151,48	0,35	151,83	607,32
Totale					2213,64

*Tabella 11 - Emissioni stimate di CO<sub>2</sub> in grammi per il dataset Palsar*

Rete	Epoch	Training	Evaluation	Train+Eval	Total
UNetL	0,42	20,99	0,20	21,19	254,31
UNet	0,48	24,09	0,21	24,30	291,54
UNet++L	0,85	34,05	0,18	34,23	273,89
UNet++	1,27	50,83	0,20	51,03	408,20
TransUNet	1,71	68,52	0,30	68,82	550,58
LightMUNet	3,52	140,92	0,31	141,22	564,92
Totale					2343,44

*Tabella 12 - Emissioni stimate di CO<sub>2</sub> in grammi per il dataset Sentinel*

L'addestramento automatico ha prodotto una quantità stimata di 4,56 kg di CO<sub>2</sub>. Le emissioni associate al dataset Sentinel sono più alte poiché in esso è presente un numero maggiore di immagini (circa 5% in più).

## 4 - Conclusioni

Il presente lavoro ha esplorato l'efficacia di diverse architetture di deep learning per il rilevamento di oil spill in immagini SAR di bassa qualità, ponendo particolare attenzione alle reti basate su U-Net e alle loro varianti. I risultati ottenuti evidenziano come la TransUNet preaddestrata rappresenti la soluzione più performante sul dataset Sentinel, mentre UNet++ si afferma come la rete più efficace sul dataset Palsar, seguita dalla TransUNet preaddestrata. In generale, i modelli più complessi, sebbene più onerosi in termini computazionali ed energetici, forniscono segmentazioni più accurate e affidabili.

Dal punto di vista del preprocessing, l'applicazione combinata di Fast NL-Means e filtro gaussiano ha permesso un miglioramento significativo della qualità delle immagini, aumentando la robustezza dei modelli alla presenza di rumore speckle, tipico delle immagini SAR. Questo passaggio si è rivelato essenziale per ottimizzare le prestazioni.

Tuttavia, l'analisi delle metriche ha anche mostrato una difficoltà comune tra i modelli: la tendenza a confondere piccole aree di oil spill con il background, penalizzando le metriche specifiche per la classe minoritaria. Ciò suggerisce che una futura direzione possa essere l'adozione di una funzione di loss personalizzata, che penalizzi maggiormente l'errata classificazione dell'oil spill come background, migliorando il bilanciamento nelle predizioni.

Un ulteriore ambito di interesse riguarda le architetture basate su State Space Models, in particolare i modelli della famiglia Mamba, rappresentati in questo studio dalla LightMUNet. Sebbene i risultati della versione semplificata di quest'ultima siano inferiori rispetto a TransUNet e UNet++, l'approccio SSM costituisce un'interessante alternativa ai modelli basati su Transformers. Un'analisi più approfondita su una versione di LightMUNet più vicina al modello originale, insieme a test su altre varianti della famiglia Mamba, potrebbe portare a risultati più competitivi. Negli ultimi anni, infatti, questi modelli stanno dimostrando un grande potenziale nelle applicazioni di segmentazione semantica e meritano di essere esplorati più a fondo.

La valutazione delle emissioni di CO<sub>2</sub> associate all'addestramento dei modelli ha permesso di quantificare l'impatto ambientale delle diverse configurazioni. Questi dati offrono uno spunto di riflessione sul costo energetico degli esperimenti di deep learning, soprattutto quando condotti su larga scala.

In sintesi, questo progetto ha mostrato come un approccio sistematico al preprocessing, alla scelta architetturale e alla valutazione ambientale possa migliorare l'affidabilità del rilevamento automatico di oil spill, aprendo la strada a soluzioni più accurate, efficienti e sostenibili per il monitoraggio marittimo.

## Bibliografia

- [1] Xiaorui Dong, Jiansheng Li, Bing Li, Yueqin Jin, Shufeng Miao. *Marine Oil Spill Detection from Low-Quality SAR Remote Sensing Images*. **Journal of Marine Science and Engineering**, 2023. <https://www.mdpi.com/2077-1312/11/8/1552>
- [2] Xiaorui Dong, Jiansheng Li, Bing Li, Yueqin Jin, Shufeng Miao. *Deep-SAR Oil Spill (SOS) dataset (corrected)*, 2023. [https://drive.google.com/file/d/12grU\\_EAPbW75eyyHj-U5pOfnwQzm0MFw/view](https://drive.google.com/file/d/12grU_EAPbW75eyyHj-U5pOfnwQzm0MFw/view)
- [3] Rubicel Trujillo-Acatitla, José Tuxpan-Vargas, Cesaré Ovando-Vázquez, Erandi Monterrubio-Martínez. *Marine oil spill detection and segmentation in SAR data with two steps Deep Learning framework*. **Marine Pollution Bulletin**, 2024. <https://www.sciencedirect.com/science/article/pii/S0025326X24005265>
- [4] Rubicel Trujillo-Acatitla, José Tuxpan-Vargas, Cesaré Ovando-Vázquez, Erandi Monterrubio-Martínez. *Sentinel-1 SAR Oil spill image dataset for train, validate, and test deep learning models*, 2024. <https://zenodo.org/records/8346860> (part 1)  
<https://zenodo.org/records/8253899> (part 2)  
<https://zenodo.org/records/13761290> (part 3)
- [5] Jin Zhang, Hao Feng, Qingli Luo, Yu Li, Yu Zhang, Jian Li, Zhoumo Zeng. *Oil Spill Detection with Dual-Polarimetric Sentinel-1 SAR Using Superpixel-Level Image Stretching and Deep Convolutional Neural Network*. **Remote Sensing**, 2022. <https://www.mdpi.com/2072-4292/14/16/3900>
- [6] Fahim Abdul Gafoor, Maryam R. Al Shehhi. *Oil Spill Detection and Mapping Using Sentinel-1 and Sentinel-2 in the Arabian Gulf Coastal Waters*. **IEEE Sensors Journal**, 2022. <https://ieeexplore.ieee.org/document/9883723>
- [7] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, Jianming Liang. *UNet++: A Nested U-Net Architecture for Medical Image Segmentation*. **arXiv preprint**, 2018. <https://arxiv.org/abs/1807.10165>
- [8] Kai Zhang, Chenglin Li, Wenxuan Wu, Mingjian He, Xingguang Zhou, Ziwei Liu. *LightM-UNet: Mamba Assists in Lightweight UNet for Medical Image Segmentation*. **arXiv preprint**, 2024. <https://arxiv.org/abs/2403.05246>
- [9] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, Yuyin Zhou. *TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation*. **arXiv preprint**, 2021. <https://www.cs.jhu.edu/~alanlab/Pubs21/chen2021transunet.pdf>



**[10] Buades Antoni, Coll Bartomeu, Morel Jean-Michel.** *Non-local Means Denoising*, 2011. [https://www.ipol.im/pub/art/2011/bcm\\_nlm/](https://www.ipol.im/pub/art/2011/bcm_nlm/)

**[11] Rahman Md Awsafkur,** *TransUNet-tf*, 2022.  
<https://github.com/awsaf49/TransUNet-tf>