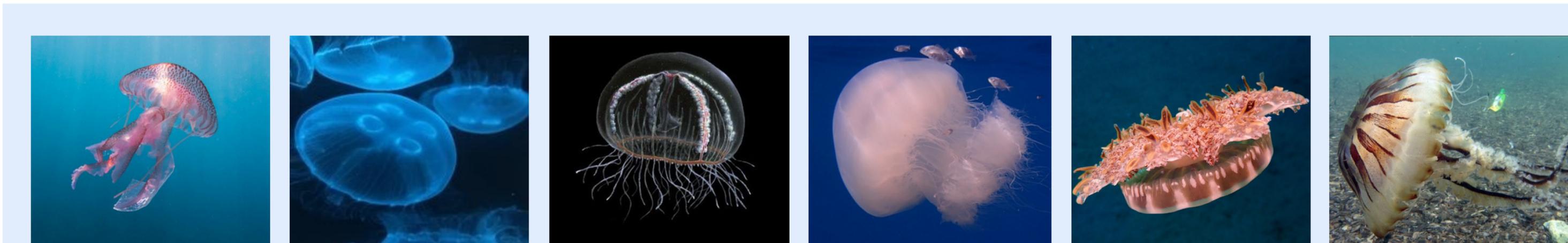


# Classificazione delle meduse nel Mar Mediterraneo

Sara Marchetti



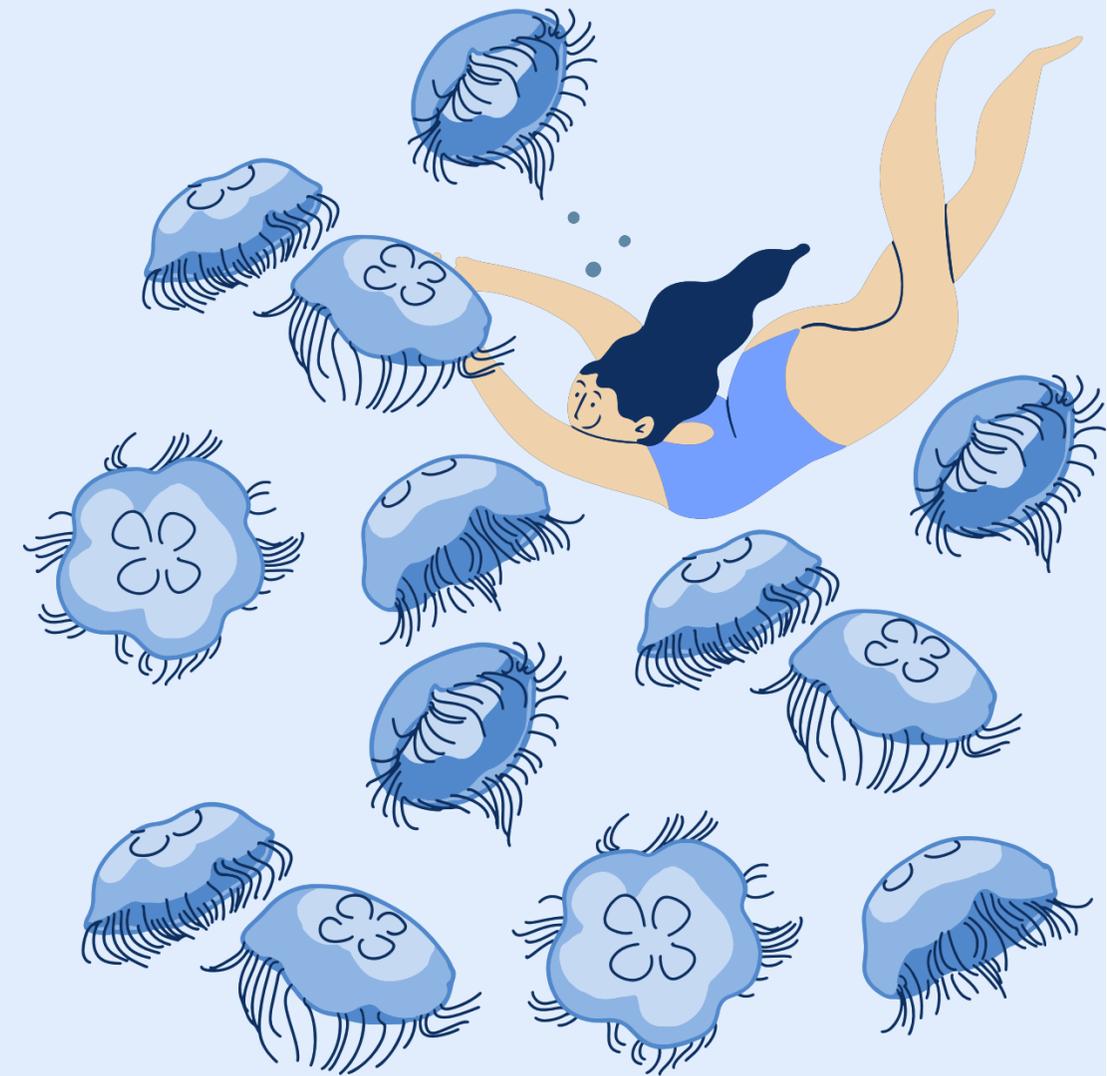
# Abstract

L'aumento della presenza di meduse lungo le coste del Mar Mediterraneo rappresenta una sfida per la sicurezza dei bagnanti e l'ecosistema marino. Questo progetto si propone di classificare le specie di meduse attraverso modelli di deep learning, utilizzando inizialmente una **Convolutional Neural Network** e successivamente una **ResNet**. I dati utilizzati sono una combinazione di un dataset disponibile su Kaggle e di immagini scaricate da Google. I risultati hanno evidenziato una classificazione non ottimale con la CNN, mentre la ResNet ha dimostrato prestazioni migliori, garantendo un'elevata accuratezza. Questa ricerca apre la strada a strumenti di riconoscimento più efficaci e applicativi nell'ambito della sicurezza delle spiagge.



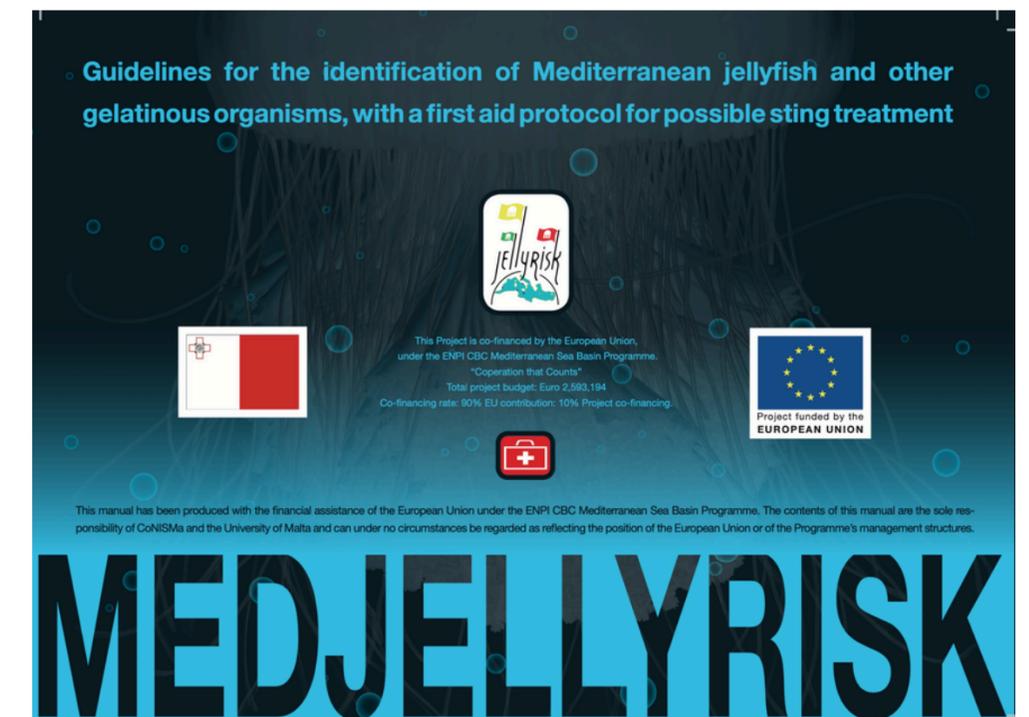
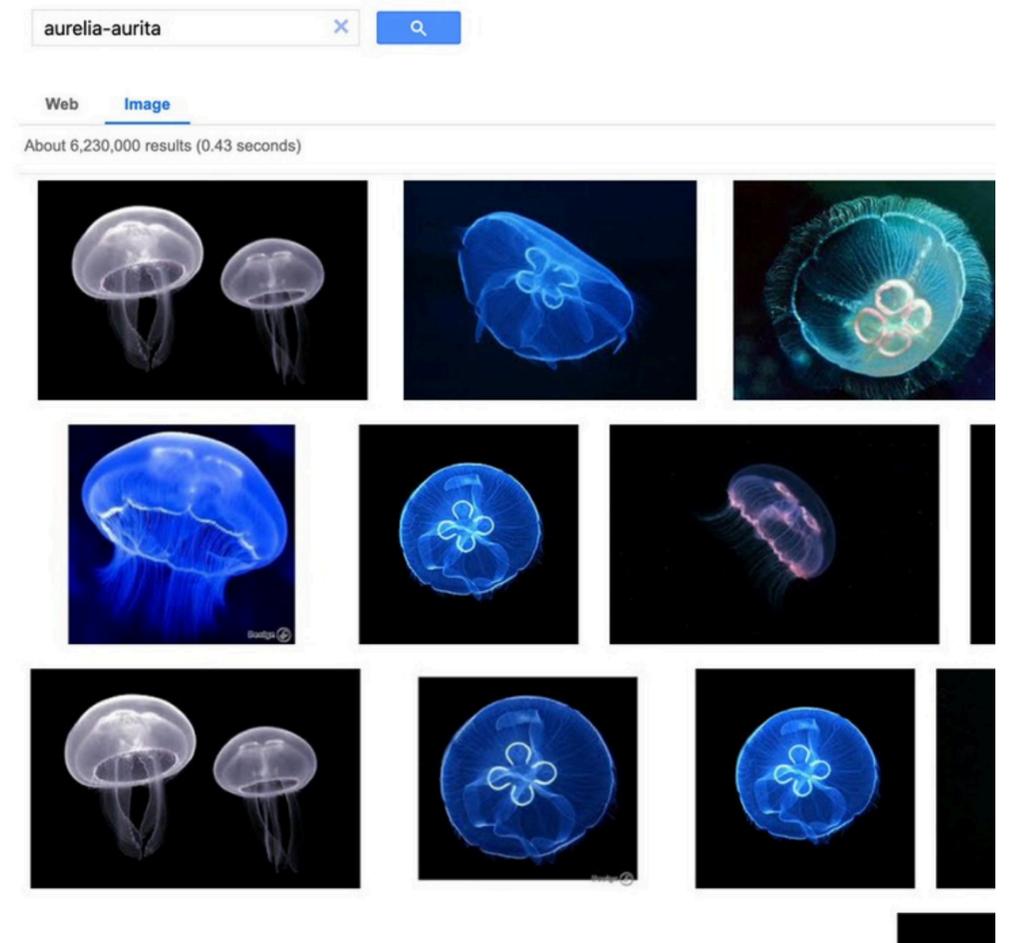
# Motivazioni

- **Sicurezza dei bagnanti:** identificare rapidamente le specie di meduse per prevenire incontri pericolosi e ridurre il rischio di punture.
- Supporto alla **ricerca biologica**
- **Monitoraggio** ambientale e avvistamenti
- Valutazione della **pericolosità:** distinguere tra specie innocue e potenzialmente pericolose per l'uomo.
- **Limitazioni dei dataset:** necessità di migliorare la raccolta dati per modelli più precisi

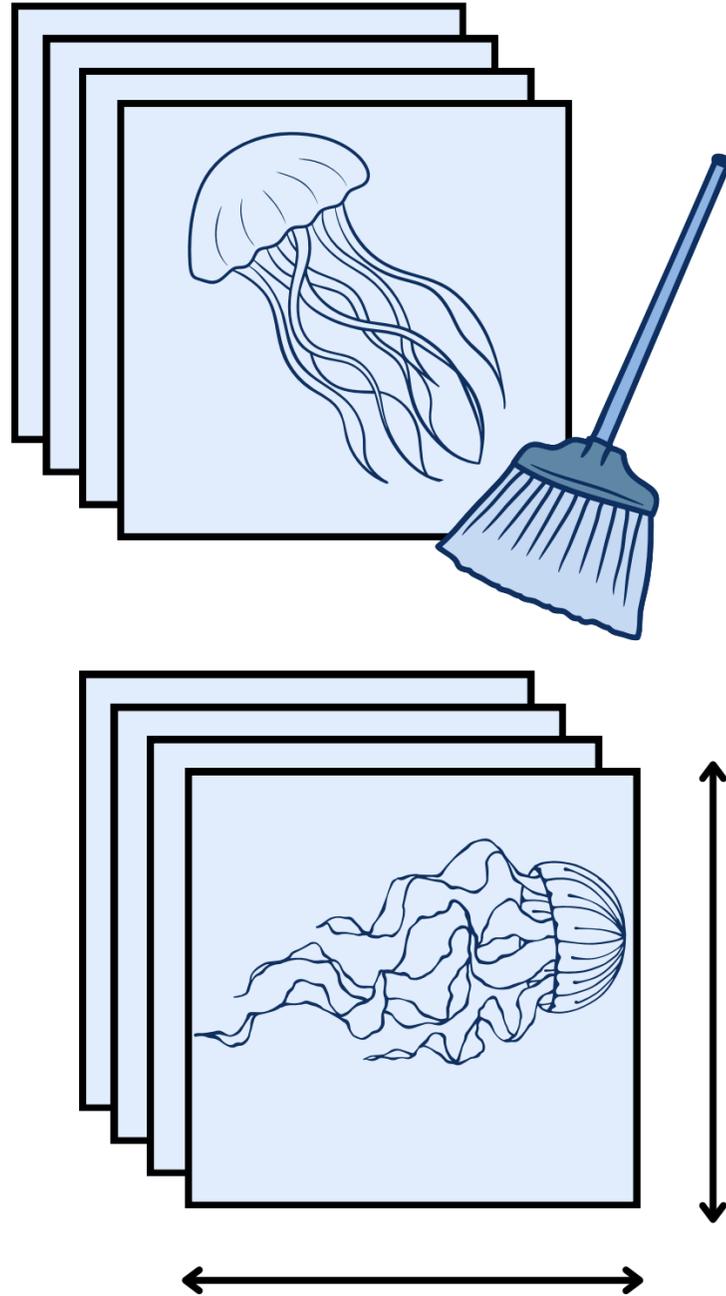


# Ricerca e raccolta dei dati

- **Dataset Kaggle:**
  - 6 categorie di meduse del Mare del Nord
  - 150 immagini per categoria
- **Espansione del dataset** per il Mar Mediterraneo:
  - Selezionate 12 categorie di meduse presenti (4 dal dataset originale)
- **Conferma delle specie** selezionate basata sul contributo di una laureanda magistrale in biologia marina
- **Raccolta dati aggiuntivi:**
  - Utilizzo di **API** e **Google Custom Search**
  - Acquisizione di **100** immagini

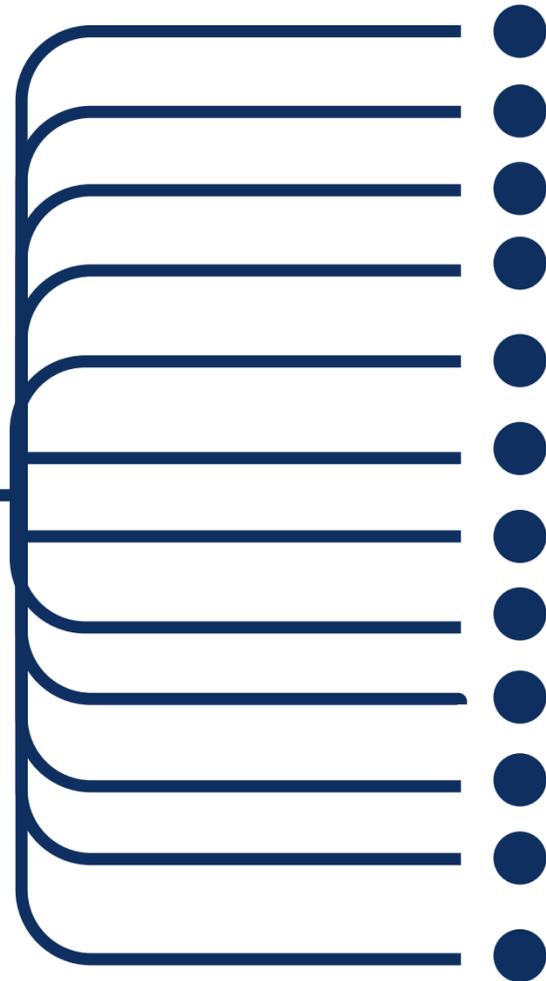
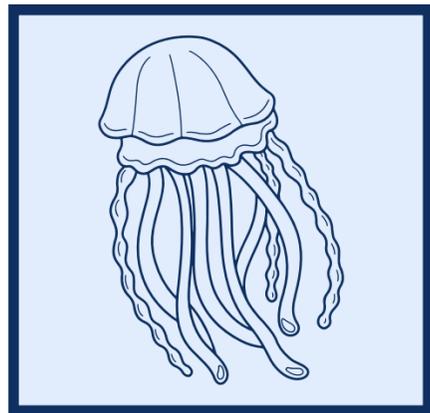


# Preprocessing



- **Filtraggio** delle immagini e rimozione di duplicati
- **Ridimensionamento** (Resize): adattamento delle immagini alla dimensione scelta
- **Standardizzazione**
- Conversione in **Tensori**: trasformazione delle immagini in un formato compatibile con il deep learning
- **Data Augmentation**: applicazione di tecniche di aumento dei dati (rotazioni e flip) per migliorare la generalizzazione del modello
- Analisi della **rappresentazione**: verifica della distribuzione delle classi per evitare squilibri nel dataset

# Obiettivi del progetto



- E' possibile creare un dataset con immagini scaricate dal web?
- E' possibile distinguere le specie di meduse tramite la **classificazione per immagini** con Convolutional Neural Network ? Con quale accuratezza?

# Prima metodologia

Per la classificazione delle meduse, ho scelto di utilizzare una **Convolutional Neural Network**, particolarmente efficace nel riconoscimento di immagini. L'architettura si compone di due parti principali:

- **Parte Convolutionale:**

- Cinque blocchi convoluzionali con un numero crescente di filtri (16, 32, 64, 128, 256).
- Ogni blocco include Convoluzione 3x3, Batch normalization, ReLU e Max pooling 2x2.

- **Parte Fully Connected:**

- i tensori vengono appiattiti (Flatten) e passano attraverso tre strati fully connected, attivati con la ReLU
- I livelli sono composti da 1024, 256 e 12 neuroni
- Sono presenti dropout (0.5) per ridurre l'overfitting.





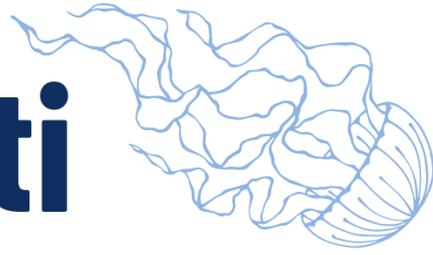
# Problematiche e tentativi di miglioramento

- **Problemi:**
  - Scarsità di dati
  - Complessità del problema che richiede una rete di medie dimensioni
  - Limitata potenza di calcolo
- **Risultato:** Accuratezza < 70%
- **Tentativi di miglioramento non sufficienti:**
  - aumentare il numero di blocchi
  - Cambiare il numero dei parametri
  - Cambiare metodo di ottimizzazione (AdamW)
  - Ottimizzare gli iperparametri (learning rate, regolarizzazione, batch size)
  - Data augmentation

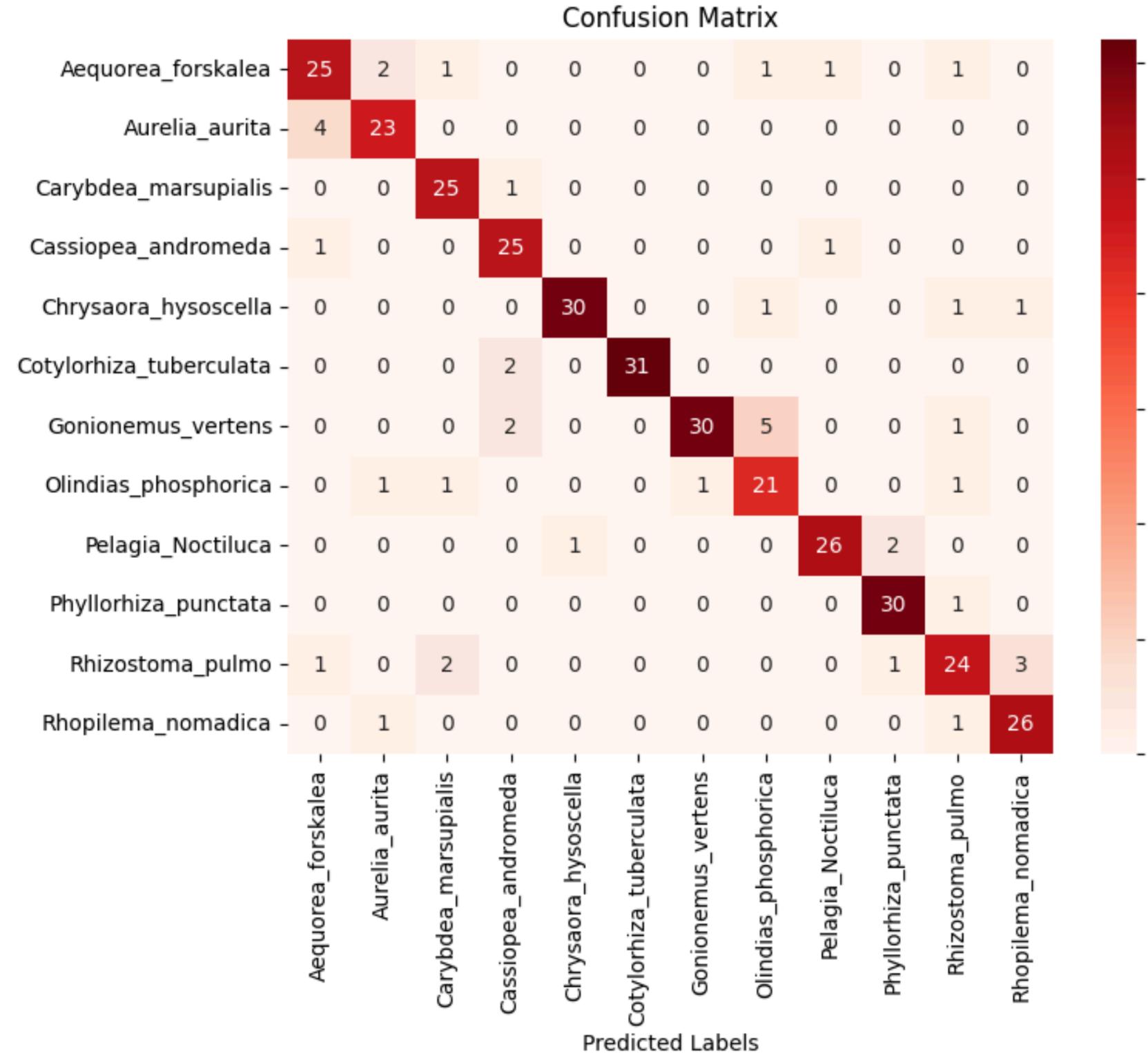
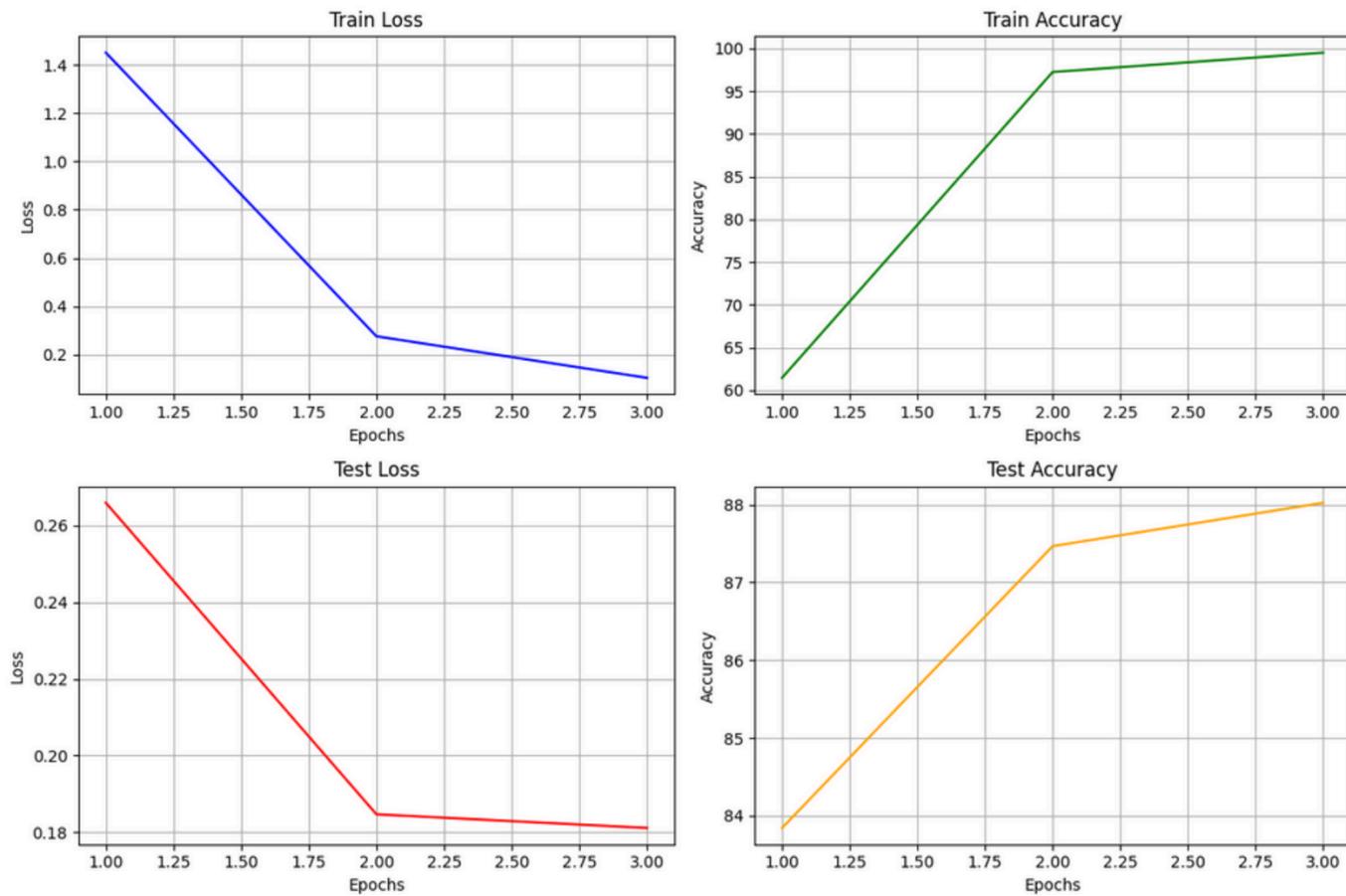
# Soluzione: seconda metodologia

Nel tentativo di migliorare le performance del modello, ho deciso di passare dalla rete progettata inizialmente a un modello **ResNet18 pre-addestrato**. La ResNet, grazie alla sua architettura con **residual connections**, è in grado di gestire meglio il problema del vanishing gradient, che può verificarsi in reti profonde, migliorando così l'efficacia nell'addestramento. Utilizzando un modello pre-addestrato possiamo sfruttare le caratteristiche già apprese su un ampio dataset, come ImageNet, e adattarle alla classificazione delle meduse, **riducendo il tempo di addestramento e migliorando l'accuratezza** (utile quando i dati a disposizione sono limitati). Il modello è stato adattato per il nostro problema per ottenere le 12 classi di meduse.

# Risultati



- Epoche: 3
- Accuratezza sul test set: 88%
- Loss sul test set: 0.4

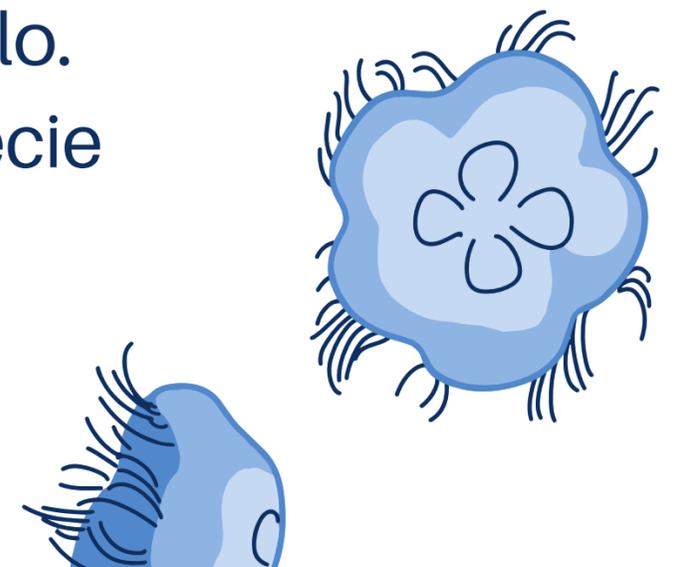


# Conclusioni e lavori futuri



Il progetto ha permesso di sviluppare un sistema di classificazione delle meduse presenti nel Mar Mediterraneo, utilizzando tecniche di deep learning con due tipologie di reti neurali convoluzionali, permettendo di osservare e confrontare i risultati ottenuti. Nonostante alcune limitazioni, come la scarsità di dati e la somiglianza visiva tra alcune specie, i risultati ottenuti sono soddisfacenti e offrono un buon punto di partenza per miglioramenti futuri come ad esempio:

- **Ampliamento e diversificazione** del dataset con l'eventuale aggiunta di altre specie.
- **Implementazione di tecniche avanzate di data augmentation** per aumentare la varietà delle immagini e migliorare la generalizzazione del modello.
- **Creazione di un applicativo** per l'identificazione tempestiva di specie pericolose e il monitoraggio delle popolazioni di meduse.



# Bibliografia

## Dataset Kaggle:

- <https://www.kaggle.com/datasets/anshtanwar/jellyfish-types>

## Download immagini:

- <https://developers.google.com/custom-search/v1/overview>
- <https://programmablesearchengine.google.com/about/>

## Documentazione PyTorch:

- [https://pytorch.org/tutorials/beginner/blitz/cifar10\\_tutorial.html](https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html)
- [https://pytorch.org/hub/pytorch\\_vision\\_resnet/](https://pytorch.org/hub/pytorch_vision_resnet/)
- <https://pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html>

## ResNet:

- [https://en.wikipedia.org/wiki/Residual\\_neural\\_network](https://en.wikipedia.org/wiki/Residual_neural_network)
- Deep Residual Learning for Image Recognition: <https://arxiv.org/abs/1512.03385>

