S1 Sensitivity testing of confidence threshold

There are some non-artificial interference methods to determine the value of confidence, such as adding confidence as a parameter to the loss function for model training to obtain higher mAP, but in industry applications, the actual effect of identification is more reliable than these parameters. The confidence of 0.2 was chosen because many eddies below 0.2 were wrong so I chose to keep eddies above the confidence of 0.2. It is preferable to cut some SMEs, but also to improve the reliability of the analytical results generated by the dataset. Additionally, I can upload the full eddies data of the confidence from 0-1. Below Fig. S1 are a few low confidence images.

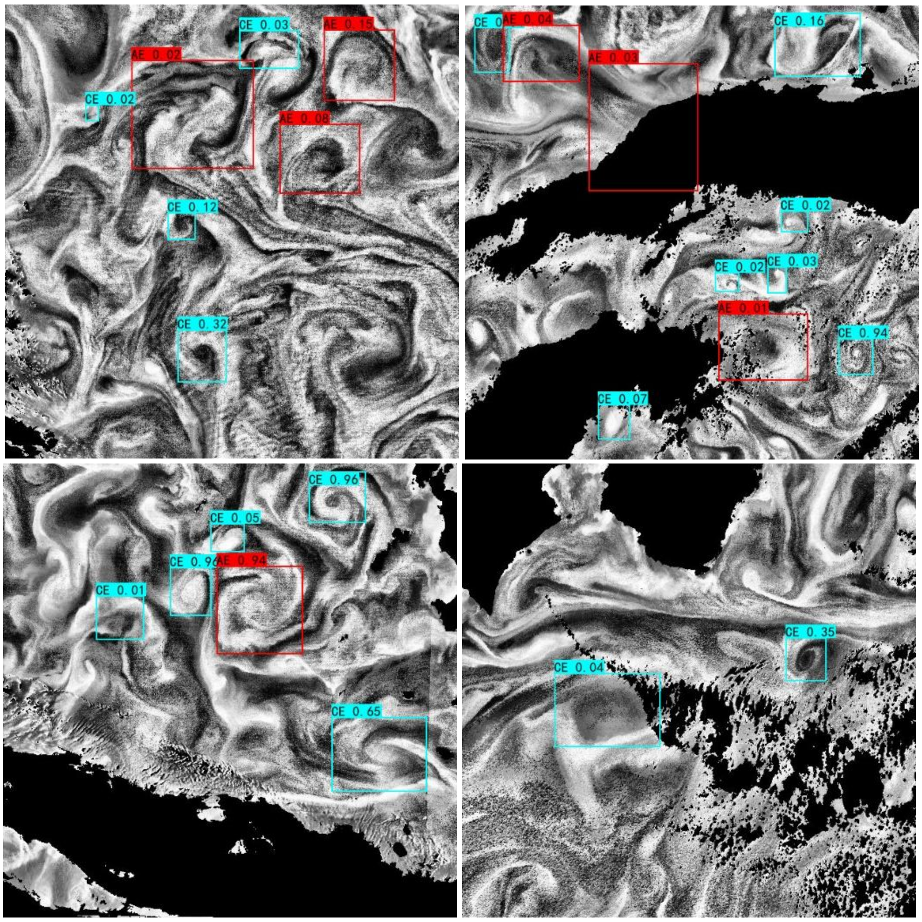

Figure S1: Image identification results of SMEs.

We also verify some of the results with different confidence thresholds. As shown in Fig. S2 below, different confidence levels do not affect the conclusions.

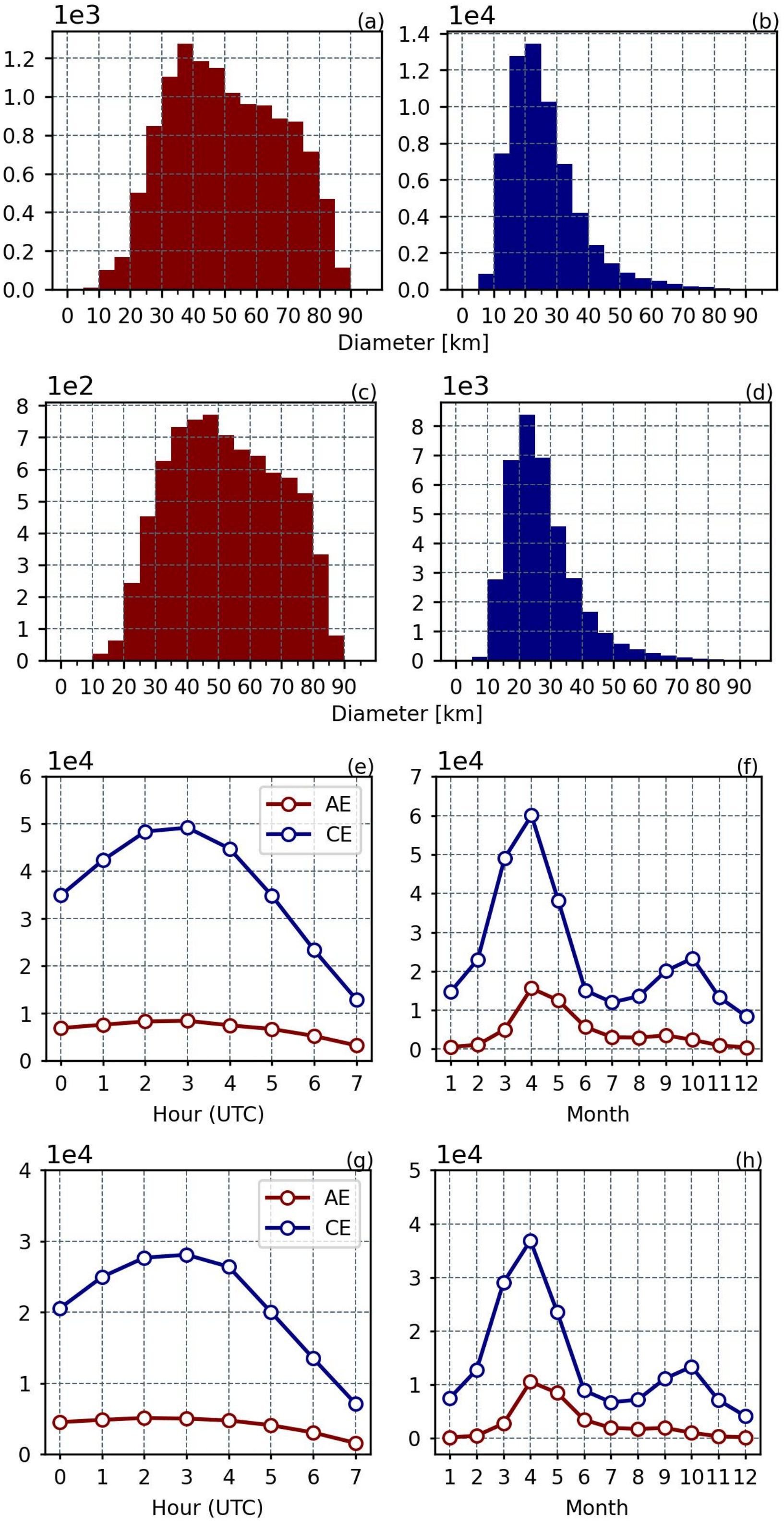

Figure S2: (a)(c) and (b)(d) show the diameter distribution histograms of AE and CE, respectively. (e) and (g): The figure shows the variation in the number of identified eddies over hours. (f) and (h): The figure shows the seasonal variation in the number of identified eddies. (a)(b)(e)(f) is the results with a confidence minimum of 0.5 and the confidence minimum of (c)(d)(g)(h) is 0.8.

S2 A comparison test with or without CLAHE

CLAHE is employed to improve the clarity of chlorophyll spirals, enabling the training and identification of these spirals by AI in sea areas with chlorophyll concentration differences spanning several orders of magnitude using the same training dataset. The absence of CLAHE technology presents two significant challenges, as illustrated in Fig. S3. Firstly, it increases the complexity of establishing the training dataset, and secondly, it diminishes the accuracy of AI-based recognition and the reliability of recognition results.

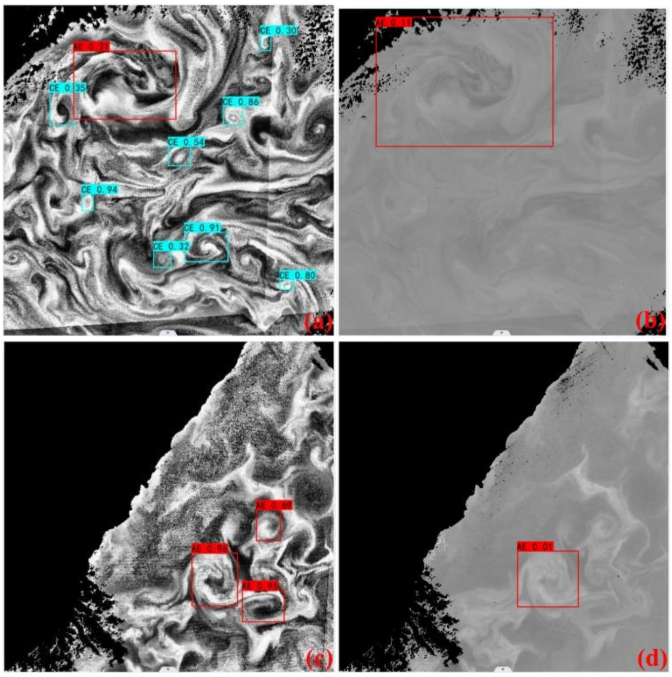

Figure S3: A comparison of eddy identification with and without the application of CLAHE technology. Panels (a) and (c) demonstrate the use of CLAHE, while panels (b) and (d) do not.

We conducted chlorophyll image identification at 0:00 using CLAHE and without it, as demonstrated in Fig. S4. The disparity between the two approaches is striking, underscoring the necessity of employing CLAHE. Without CLAHE, a substantial number of low-confidence eddies would be erroneously identified, leading to an overall underestimation of reliable eddies. However, given the absence of comparable submesoscale eddy datasets, quantitatively assessing the extent to which CLAHE may lead to overestimation or underestimation remains a challenging task.

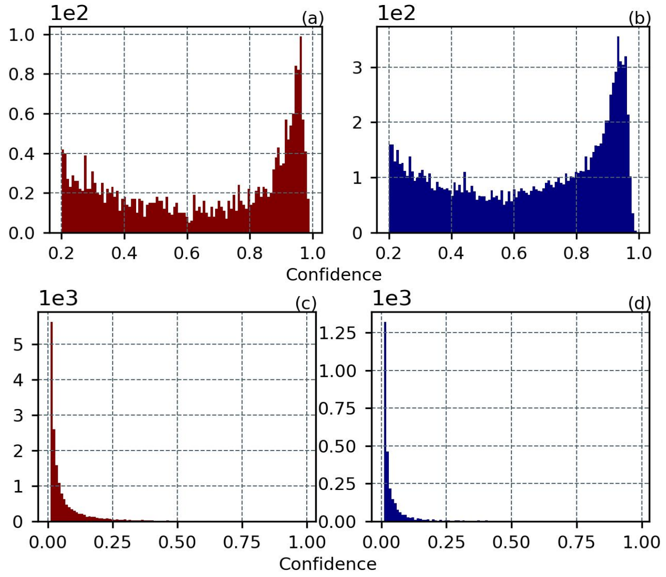

Figure S4: SMEs confidence distribution histogram. (a)(c) and (b)(d) show the confidence distribution histograms of AE and CE, respectively.

S3 Meanings of variables and sample code

The document is used to clarify the meanings of various variables of the dataset (Table S1) and provide an example of processing the data using Python code.

The name of each folder represents the UTC of the files inside.

Table S1: The meanings of various variables of the dataset.

Variable name	Description	Units or Type
time	The time of obtained chlorophyll–a distribution image.	'YYYYMMDD'
AE_sum	The number of anticyclonic	
CE_sum	The number of cyclonic	
predict	Prediction results in an image coordinate system derived from the deep learning model.	Array[n][7]*
eddy_type_AE0_CE1	The type of eddy (0: anticyclonic; 1: cyclonic)	Array[n]
center_lon_lat	The longitude and latitude coordinates of the eddy center pixel.	Array[n][2]
box_min_lon_lat	The longitude and latitude coordinates of the pixel in the upper left corner of the rectangular box.	Array[n][2]
box_max_lon_lat	The longitude and latitude coordinates of the pixel in the bottom right corner of the rectangular box.	Array[n][2]
inradius	The radius of the circle inside the rectangular box	Array[n](meter)
internal_ellipse_area	Area of the internal ellipse of the rectangular box	Array[n](m^2)
confidence	Confidence of each eddy identification. Eddies with confidence levels below 0.2 were considered to be undesirable for data analysis.	Array[n] [0.2,1]

*Array[n][7] represents a two-dimensional array of n rows and 7 columns.
n is the sum of the number of cyclones and anticyclones.

You can perform eddy analysis by Python, or you can download other matching files such as chlorophyll, salinity, and temperature data for matching analysis.

The following example code plots the diameter distribution histograms of anticyclonic and cyclonic by the dataset.

```
import numpy as np
import glob
import pickle
import json
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
from tqdm import tqdm

geo_all = [np.array([]), np.array([])]
file_pre = 'E:\\predict\\'   # The file path needs to be changed
for i in range(8):
    str_i = '0' + str(i) + '/'

    for month in range(12):
        str_month = '0' + str(month + 1) if month < 9 else str(month + 1)
        print(i, ' ' + str_month)
        geo_dis = np.zeros((2, 5685, 5567))
        files_pre = glob.glob(file_pre + str_i + 'dataset\\????' + str_month + '??' + str_i[0:2] + '.json')
        for file in tqdm(files_pre):
            with open(file, 'rb') as f:
                dataset = json.load(f)
            type_index = np.array(dataset['results']['eddy_type_AE0_CE1']) == 0
            a = np.array(dataset['results']['inradius'])[type_index]
            b = np.array(dataset['results']['inradius'])[~type_index]
            geo_all[0] = np.concatenate([geo_all[0], a])
            geo_all[1] = np.concatenate([geo_all[1], b])

fig, (ax, ax2) = plt.subplots(1, 2, figsize=(5, 2), dpi=300)
plt.subplots_adjust(left=None, bottom=0.19, right=None, top=None, wspace=None, hspace=0.2)
ax2.hist(geo_all[1] / 1000 * 2, bins=np.arange(0, 100, 5), color='#000080', label='CE')
ax.hist(geo_all[0] / 1000 * 2, bins=np.arange(0, 100, 5), color='#800000', label='AE')

ax.grid(ls="--", lw=0.5, color="#4E616C")
ax.yaxis.set_major_locator(ticker.MultipleLocator(100 * 3))
ax.xaxis.set_major_locator(ticker.MultipleLocator(10))
ax.xaxis.set_minor_locator(ticker.MultipleLocator(5))
ax.xaxis.set_tick_params(length=2, labelsize=6, which='minor')
ax.xaxis.set_tick_params(length=3, labelsize=8, which='major')
ax.yaxis.set_tick_params(length=3, labelsize=8)
ax.ticklabel_format(style='sci', scilimits=(0, 1), axis='y')

ax2.grid(ls="--", lw=0.5, color="#4E616C")
ax2.yaxis.set_major_locator(ticker.MultipleLocator(1000 * 3))
ax2.xaxis.set_major_locator(ticker.MultipleLocator(10))
ax2.xaxis.set_minor_locator(ticker.MultipleLocator(5))
ax2.xaxis.set_tick_params(length=2, labelsize=6, which='minor')
ax2.xaxis.set_tick_params(length=3, labelsize=8, which='major')
ax2.yaxis.set_tick_params(length=3, labelsize=8)
ax2.ticklabel_format(style='sci', scilimits=(0, 1), axis='y')

fig.text(0.43, 0.03, 'Diameter [km]', fontsize=8)
fig.text(0.45, 0.888, '(a)', fontsize=8)
fig.text(0.872, 0.888, '(b)', fontsize=8)
plt.show()
```