In [24]:

```
import os
import PIL
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from skimage.data import imread
```

In [25]:

```
#### reference:https://www.kaggle.com/paulorzp/run-length-encode-and-decode
#### https://www.kaggle.com/inversion/run-length-decoding-quick-start


def rle_encode(img):
    '''
    img: numpy array, 1 - mask, 0 - background
    Returns run length as string formated
    '''
    pixels = img.flatten()
    pixels = np.concatenate([[0], pixels, [0]])
    runs = np.where(pixels[1:] != pixels[:-1])[0] + 1
    runs[1::2] -= runs[::2]
    return ' '.join(str(x) for x in runs)
def rle_decode(mask_rle, shape=(768, 768)):
    '''
    mask_rle: run-length as string formated (start length)
    shape: (height,width) of array to return
    Returns numpy array, 1 - mask, 0 - background

    '''
    s = mask_rle.split()
    starts, lengths = [np.asarray(x, dtype=int) for x in (s[0:][::2], s[1:][::2])]
    starts -= 1
    ends = starts + lengths
    img = np.zeros(shape[0]*shape[1], dtype=np.uint8)
    for lo, hi in zip(starts, ends):
        img[lo:hi] = 1
    return img.reshape(shape).T
```

In [26]:

```
masks = pd.read_csv("./train_ship_segmentations_v2.csv")
display(masks.head())
```

	ImageId	EncodedPixels
0	00003e153.jpg	NaN
1	0001124c7.jpg	NaN
2	000155de5.jpg	264661 17 265429 33 266197 33 266965 33 267733...
3	000194a2d.jpg	360486 1 361252 4 362019 5 362785 8 363552 10 ...
4	000194a2d.jpg	51834 9 52602 9 53370 9 54138 9 54906 9 55674 ...

In [35]:

```
masks['ships'] = masks['EncodedPixels'].map(lambda c_row: 1 if isinstance(c_row, str
unique_img_ids = masks.groupby('ImageId').agg({'ships': 'sum'}).reset_index()
unique_img_ids['file_size_kb'] = unique_img_ids['ImageId'].map(lambda c_img_id: os.s
display(unique_img_ids.head())
```

	ImageId	ships	file_size_kb
0	00003e153.jpg	0	128.944336
1	0001124c7.jpg	0	76.059570
2	000155de5.jpg	1	147.625977
3	000194a2d.jpg	5	75.221680
4	0001b1832.jpg	0	95.627930

In [36]:

```
# unique_img_ids = unique_img_ids[unique_img_ids['ships']>10]
# display(unique_img_ids.head())
# display(unique_img_ids['ships'].hist())
```

In [51]:

```
max_ship = unique_img_ids['ships'].max()
print(max_ship)
```

```
15
```

In [52]:

```
max_file = unique_img_ids['file_size_kb'].max()
print(max_file)
```

```
511.942382812
```

In [63]:

```
min_ship = unique_img_ids['ships'].min()
print(min_ship)
```

```
0
```

In [64]:

```
min_file = unique_img_ids['file_size_kb'].min()
print(min_file)
```

```
9.6123046875
```

In [62]:

```
maxship_data_frame = unique_img_ids.loc[unique_img_ids['ships']== 15]
unique_img_ids.loc[unique_img_ids['ships']== 15].sample(5)
```

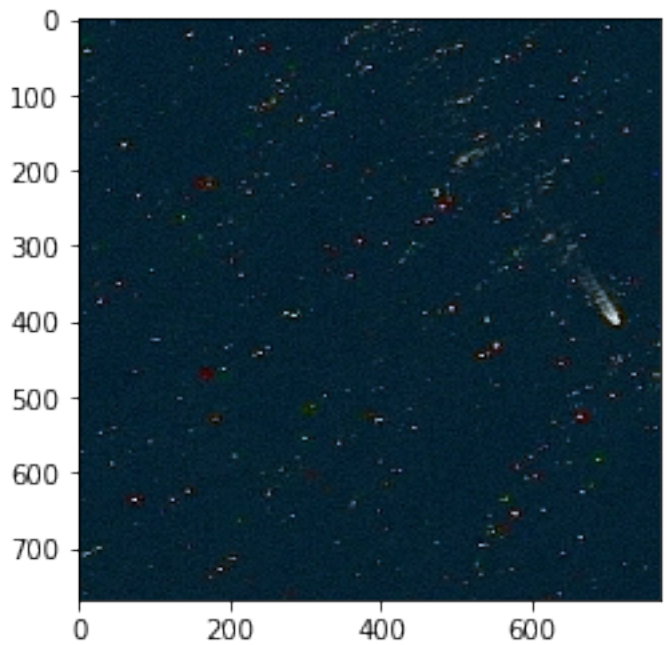
Out[62]:

	ImageId	ships	file_size_kb
2579	0368beab8.jpg	15	166.166992
129966	accdf6c3e.jpg	15	332.364258
58420	4de149bd9.jpg	15	138.797852
49351	41bdd5164.jpg	15	299.685547
65198	56d23b600.jpg	15	144.694336

In [61]:

```
maxship_image_list = []
for index in maxship_data_frame.index:
    maxship_image_list.append(masks.loc[index,'ImageId'])
print(len(maxship_image_list))
for imageId in maxship_image_list:
    img = imread('./train_v2/'+ imageId)
    plt.imshow(img)
    plt.show()
```

66



In [55]:

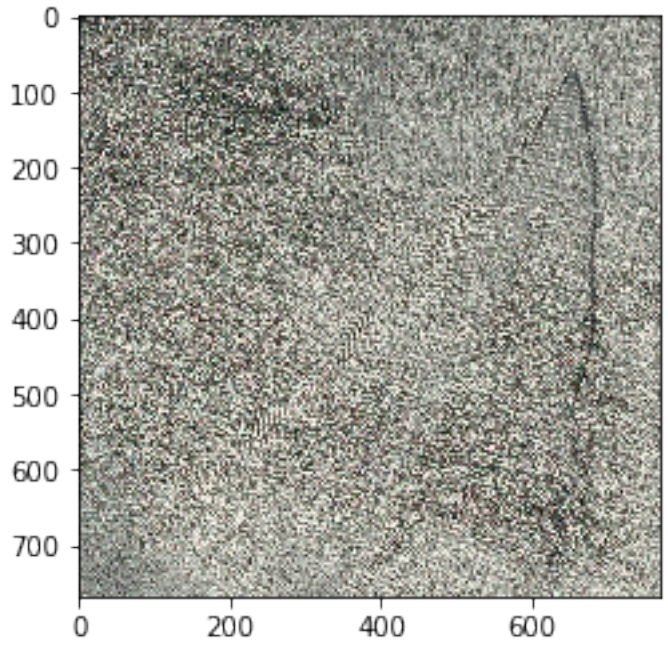
```
unique_img_ids.loc[unique_img_ids['file_size_kb']== max_file]
```

Out[55]:

	ImageId	ships	file_size_kb
41654	378562135.jpg	1	511.942383

In [54]:

```
img = imread('./train_v2/'+'378562135.jpg')
plt.imshow(img)
plt.show()
```



In [65]:

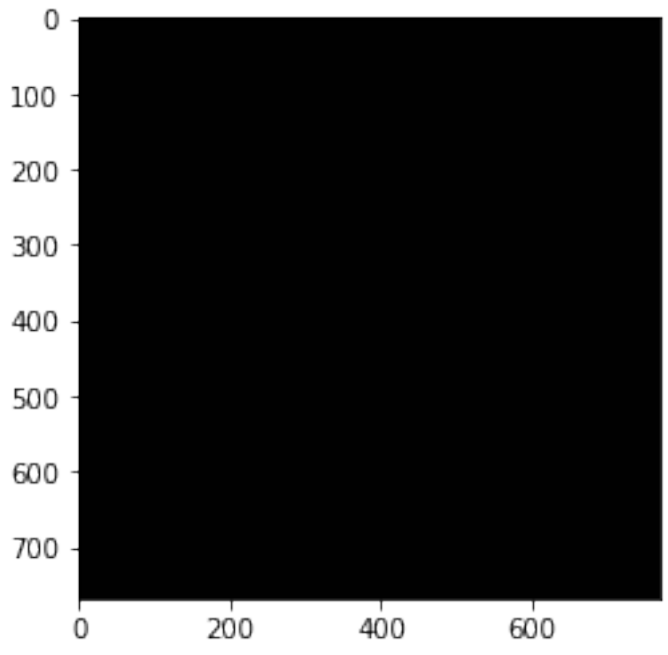
```
unique_img_ids.loc[unique_img_ids['file_size_kb']== min_file]
```

Out[65]:

	ImageId	ships	file_size_kb
87342	73fec0637.jpg	0	9.612305

In [66]:

```
img = imread('./train_v2/'+'73fec0637.jpg')
plt.imshow(img)
plt.show()
```



In [ ]:

```
```