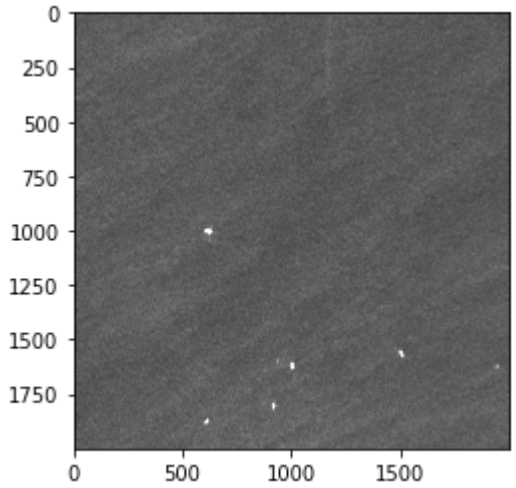
Importing Modules

In [23]:
```
import import_ipynb
import CFAR as cfar
import CFAR_v2 as cfarv2
import GeoProcess as gp
import numpy as np
import imageResizing as ir

DATA_PATH = 'Dataset_963A/'
```

In [24]:
```
#Computing rows and columns
band_data_arr = gp.readGeoTiff(DATA_PATH+'LandMasked_Amplitude_VV.tif')
#subset_img = gp.readGeoTiff('Dataset_963A/CFARResults/meanfiltered_img.tif')
rows,col = band_data_arr.shape
#print(rows,col)
#gp.visualizeImg(band_data_arr)
```

In [25]:
```
subset_img = (gp.subsetImg(band_data_arr,4000,6000)) #single ship
#subset_img = gp.subsetImg(band_data_arr,4950,5700) #multiple ship.

#subset_img = band_data_arr[5853:5888,4594:4621]
#subset_img = (ir.adjustImg(np.array(subset_img)))
print(subset_img.shape)
gp.visualizeImg(subset_img)
gp.save_img2Geotiff(subset_img,DATA_PATH+'CFARResults/Input.tif')
```

```
(2000, 2000)
Image Saved Succesfully.
```



In [26]:
```
#arr = np.array(np.arange(36))
#arr = arr.reshape(6,6)
backgroundWindow_size = 7
guardWindow_size = 5
targetWindow_size = 3
kernel = 3
pfa = 0.9973
```

In [27]:
```
print("CFAR Version 2")
cfar_version2 = cfarv2.CFAR_v2(subset_img,targetWindow_size,guardWindow_size,backgroundWindow_size,pf
finalImg_version2, DV_img,T_img = cfar_version2.shipDetection()

# print("CFAR version 1")
# cfar_filter = cfar.CFAR(subset_img,backgroundWindow_size,guardWindow_size,targetWindow_size,pfa,ke
# finalImg = cfar_filter.Shipdetection()
```

```
CFAR Version 2
Kernel Ready.
Computing P...
```

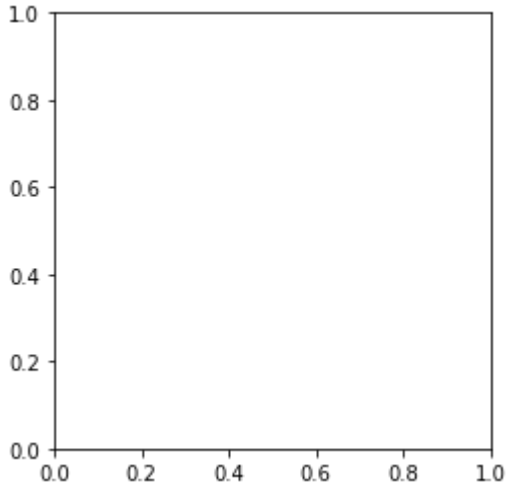
A Jupyter widget could not be displayed because the widget state could not be found. This could happen if the kernel storing the widget is no longer available, or if the widget state was not saved in the notebook. You may be able to create the widget by running the appropriate cells.

```
Process Completed, P image succesfully computed.

Binary Image of Ships is Succesfully Generated.
```

In [28]:
```
gp.visualizeBinaryImg(DV_img)
gp.save_img2Geotiff(DV_img,'Dataset_963A/CFARResults/Output_DVCFARversion2'+str(backgroundWindow_size
```

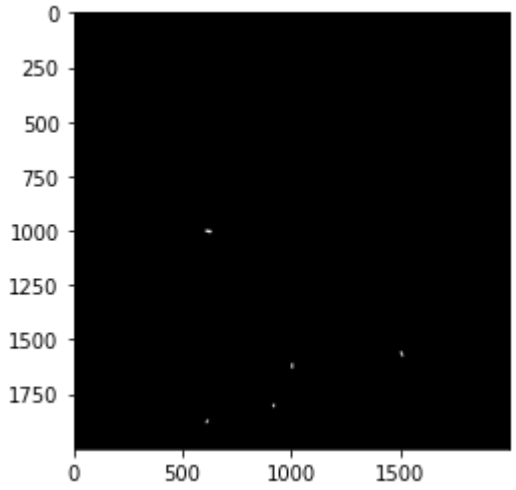
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-28-054d6064bc22> in <module>
----> 1 gp.visualizeBinaryImg(DV_img)
      2 gp.save_img2Geotiff(DV_img,'Dataset_963A/CFARResults/Output_DVCFARversion2'+str(backgroun
dWindow_size)+str(guardWindow_size)+str(targetWindow_size)+'.tif')

/media/h_mittal/My Data 2/Dissertation/ASDUSI/processing_outputs/snappy/GeoProcess.ipynb in visua
lizeBinaryImg(img)

~/.local/lib/python3.6/site-packages/matplotlib/pyplot.py in imshow(X, cmap, norm, aspect, interp
olation, alpha, vmin, vmax, origin, extent, filternorm, filterrad, resample, url, data, **kwargs)
   2728         filternorm=filternorm, filterrad=filterrad, resample=resample,
   2729         url=url, **({"data": data} if data is not None else {}),
-> 2730         **kwargs)
   2731     sci(__ret)
   2732     return __ret

~/.local/lib/python3.6/site-packages/matplotlib/__init__.py in inner(ax, data, *args, **kwargs)
   1436     def inner(ax, *args, data=None, **kwargs):
   1437         if data is None:
-> 1438             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1439
   1440         bound = new_sig.bind(ax, *args, **kwargs)

~/.local/lib/python3.6/site-packages/matplotlib/axes/_axes.py in imshow(self, X, cmap, norm, aspe
ct, interpolation, alpha, vmin, vmax, origin, extent, filternorm, filterrad, resample, url, **kwa
rgs)
   5521                               resample=resample, **kwargs)
   5522
-> 5523         im.set_data(X)
   5524         im.set_alpha(alpha)
   5525         if im.get_clip_path() is None:

~/.local/lib/python3.6/site-packages/matplotlib/image.py in set_data(self, A)
    708                 or self._A.ndim == 3 and self._A.shape[-1] in [3, 4]):
    709             raise TypeError("Invalid shape {} for image data"
--> 710                             .format(self._A.shape))
```

```
    711
    712         if self._A.ndim == 3:

TypeError: Invalid shape () for image data
```



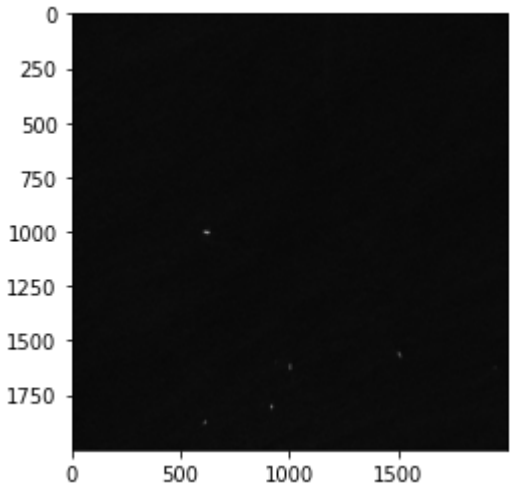
In [29]:
```
gp.visualizeBinaryImg(finalImg_version2)
gp.save_img2Geotiff(finalImg_version2,'Dataset_963A/CFARResults/Output_SHIPCFARversion2'+str(backgrou
```

```
Image Saved Succesfully.
```



In [30]:
```
gp.visualizeBinaryImg(T_img)
gp.save_img2Geotiff(T_img,'Dataset_963A/CFARResults/Output_TCFARversion2'+str(backgroundWindow_size)+
```

```
Image Saved Succesfully.
```



In [ ]:
```
# gp.visualizeBinaryImg(finalImg)
# gp.save_img2Geotiff(finalImg,'Dataset_963A/CFARResults/Output_'+str(backgroundWindow_size)+str(gua
```

In [ ]: