

4. Evaluate the classifier

Load a classifier that you have trained (specify the classifiers filename) and evaluate it on the satellite images.

This section will save the output of the classification for each site in a directory named levaluation.

In [13]:

```

1 # Load and evaluate a classifier
2 %matplotlib qt
3 classifier = joblib.load(os.path.join(filepath_models, 'NN_4classes_Landsat_test2.pkl'))
4 settings['output_epsg'] = 3857
5 settings['min_beach_area'] = 4500
6 settings['buffer_size'] = 200
7 settings['min_length_sl'] = 200
8 settings['cloud_thresh'] = 0.5
9 # visualise the classified images
10 for site in train_sites:
11     settings['inputs']['sitename'] = site[:site.find('.')]
12     # Load metadata
13     metadata = SDS_download.get_metadata(settings['inputs'])
14     # plot the classified images
15     SDS_classify.evaluate_classifier(classifier,metadata,settings)

```

Exception

Traceback (most recent call last)

<ipython-input-13-707707ff7fe5> in <module>

```

13     metadata = SDS_download.get_metadata(settings['inputs'])
14     # plot the classified images
--> 15     SDS_classify.evaluate_classifier(classifier,metadata,settings)

```

~\CoastSat-master\coastsat\SDS_classify.py in evaluate_classifier(classifier, metadata, settings)

```

589         continue
590         # process the water contours into a shoreline
--> 591     shoreline = SDS_shoreline.process_shoreline(contours_mwi
, cloud_mask, georef, image_epsg, settings)
592         try:
593             sl_pix = SDS_tools.convert_world2pix(SDS_tools.conve
rt_epsg(shoreline,

```

~\CoastSat-master\coastsat\SDS_shoreline.py in process_shoreline(contours, cloud_mask, georef, image_epsg, settings)

```

632
633     # convert pixel coordinates to world coordinates
--> 634     contours_world = SDS_tools.convert_pix2world(contours, georef)
635     # convert world coordinates to desired spatial reference system
636     contours_epsg = SDS_tools.convert_epsg(contours_world, image_eps
g, settings['output_epsg'])

```

~\CoastSat-master\coastsat\SDS_tools.py in convert_pix2world(points, georef)

```

66
67     else:
--> 68         raise Exception('invalid input type')
69
70     return points_converted

```

Exception: invalid input type