# Practical 4 - Part 1

**Q1.** Where in the code do you decide the amount of objects to be segmented?

We decide the amount of objects to be segmented in the loop where you give your interested coordinates ‘d’ and ask to find rows and columns from the connected components ‘L’.

```
for d=17:28
    [r,c]=find(L==d); …
```

**Q2.** What happens in the following rows?

```
% Measure all region areas.
regionMeasurements = regionprops(L, 'basic');
allAreas = [regionMeasurements.Area];
```

In the first row, regionprops calculates the 'Area', 'Centroid', and 'BoundingBox' measurements and stores them in variable ‘regionMeasurements’.

The second row calls the ‘Area’ measurement via the stored variable ‘‘regionMeasurements’ and stores this measurement in a separate variable ‘allAreas’.

**Q3.** What is stored in the variable ‘new_Biggest’?
‘new_Biggest’ variable stores the logical values 0 or 1 by comparing the connected component image L’s coordinates with the coordinate containing the maximum area.

**Q6.** What is stored in variable matches?
Variable matches stores the regionprops (Area, Centroid and BoundingBox) of the target image when it is compared to template image on the 7 statistical moments and these moments match (within a tolerance limit of 0.05).

In our case, 2 out of the 5 components in the target image match the template image within a tolerance of 5%. The matched components of the target image are shown by the red marked rectangle in the target image.

**Q7.** Present results both as code and a screenshot

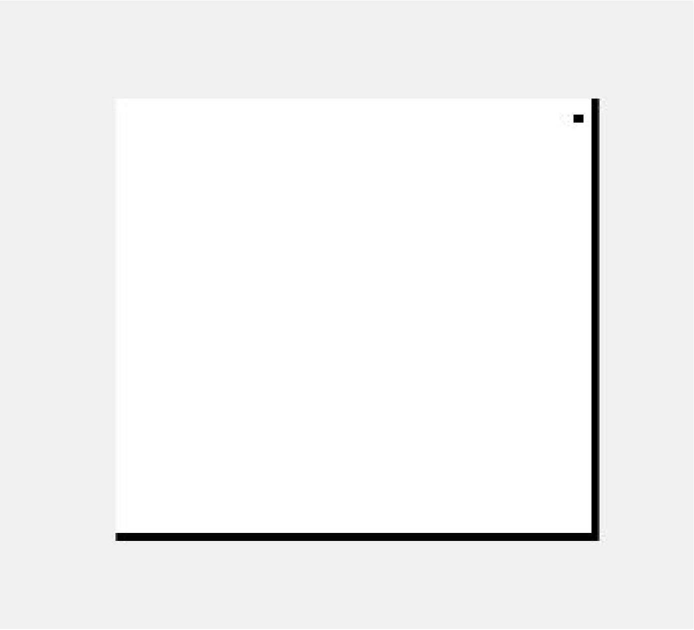
Code as below

```
clear all
im=imread('shapes.jpeg');
template=imread('ss.jpeg');
im=rgb2gray(im);
template=rgb2gray(template);
bwim=imbinarize(im);
bwtemp=imbinarize(template);
```

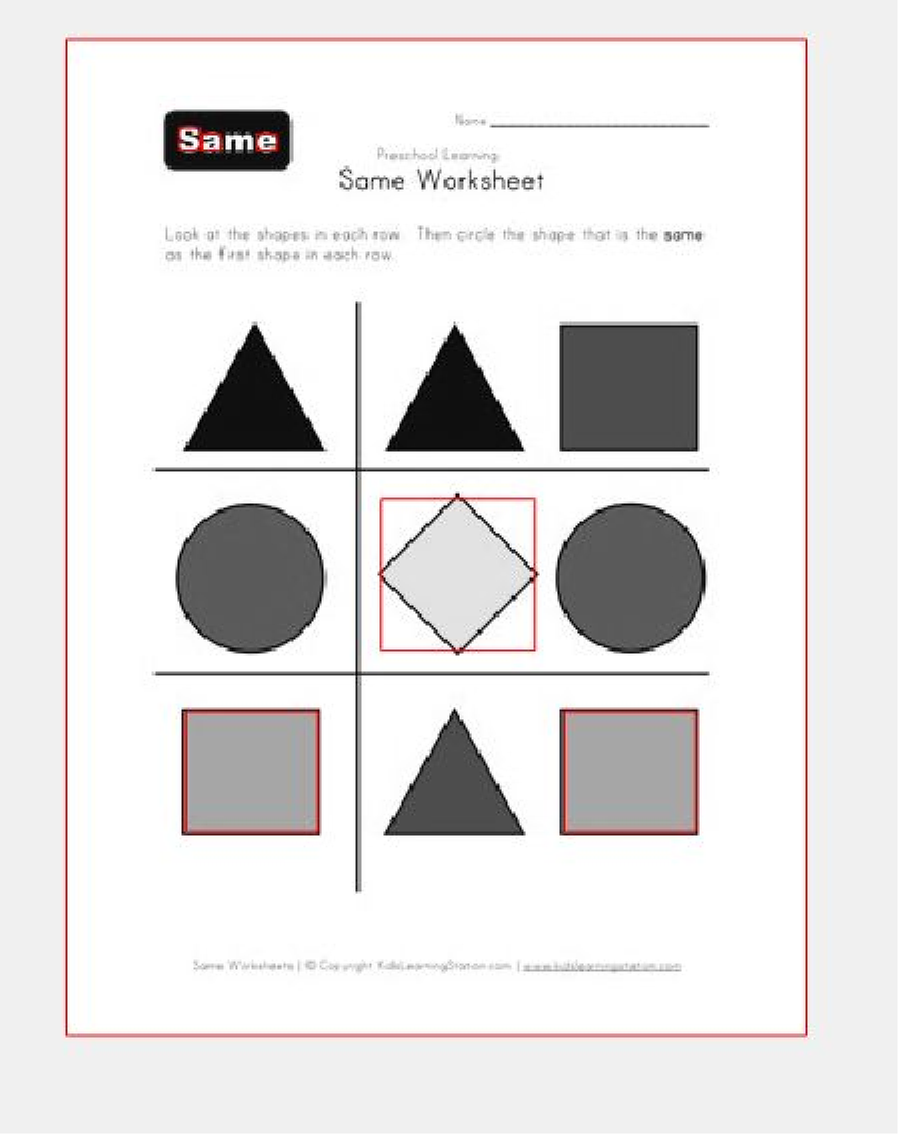
```
template_hu=invmoments(bwtemp);

[labeled, num]=bwlabel(bwim,8);
regions=regionprops(labeled);
regionSize = size(regions);
matchIndex = 1;
for i = 1:regionSize(1)
    croppedImage = imcrop(bwim, [regions(i).BoundingBox]);
    mask = bwareaopen(croppedImage, regions(i).Area - 1);
    figure(1),
    imshow(mask);

    hu = invmoments(mask);
      tol = 0.05;
      LIA = ismembertol(hu,template_hu,tol)
      if(mean(LIA) > 0.85)
          matches(matchIndex) = regions(i);
          matchIndex =matchIndex + 1;
      end
end
figure(2),imshow(im)
hold on
for j=1:length(matches)
rectangle('Position',matches(j).BoundingBox,'EdgeColor','r')
End
```

Results as below

figure(1)



figure(2)



Same

Name ____________________

Preschool Learning

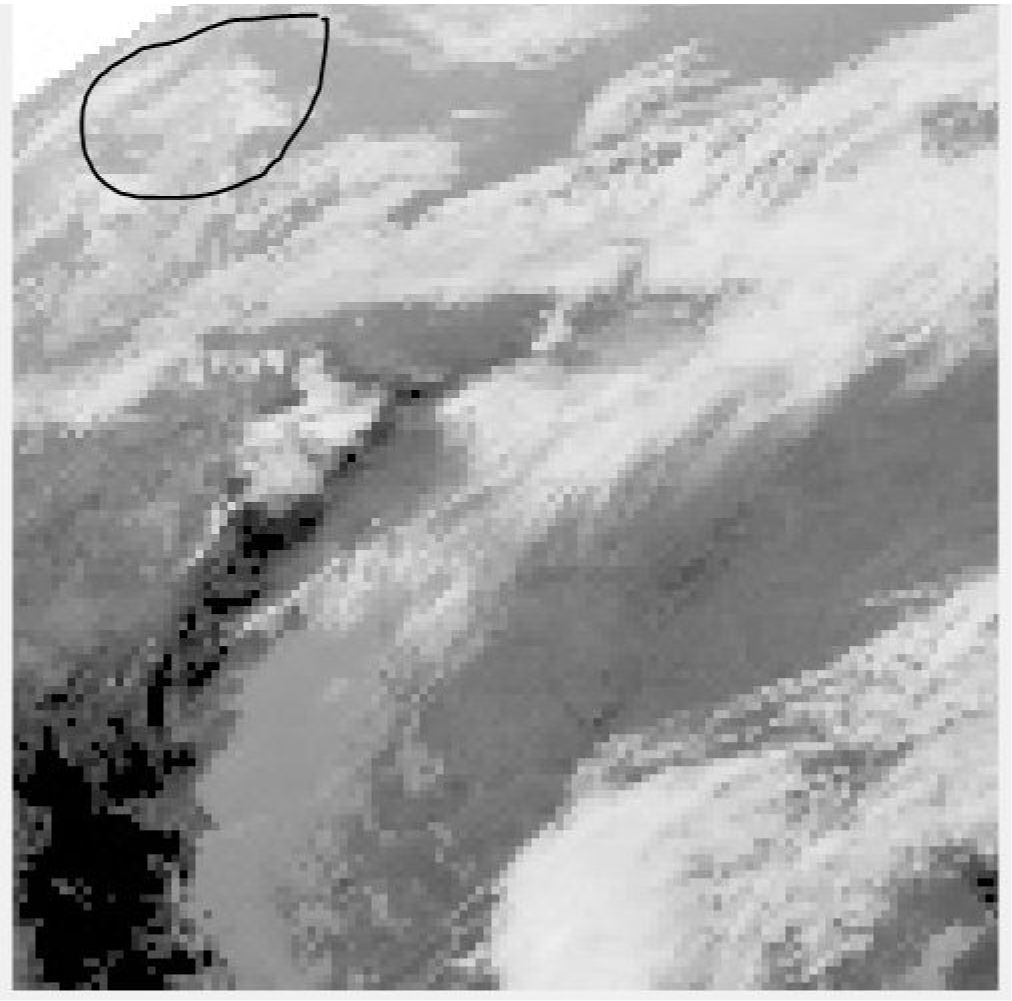
Same Worksheet

Look at the shapes in each row. Then circle the shape that is the **same** as the first shape in each row.

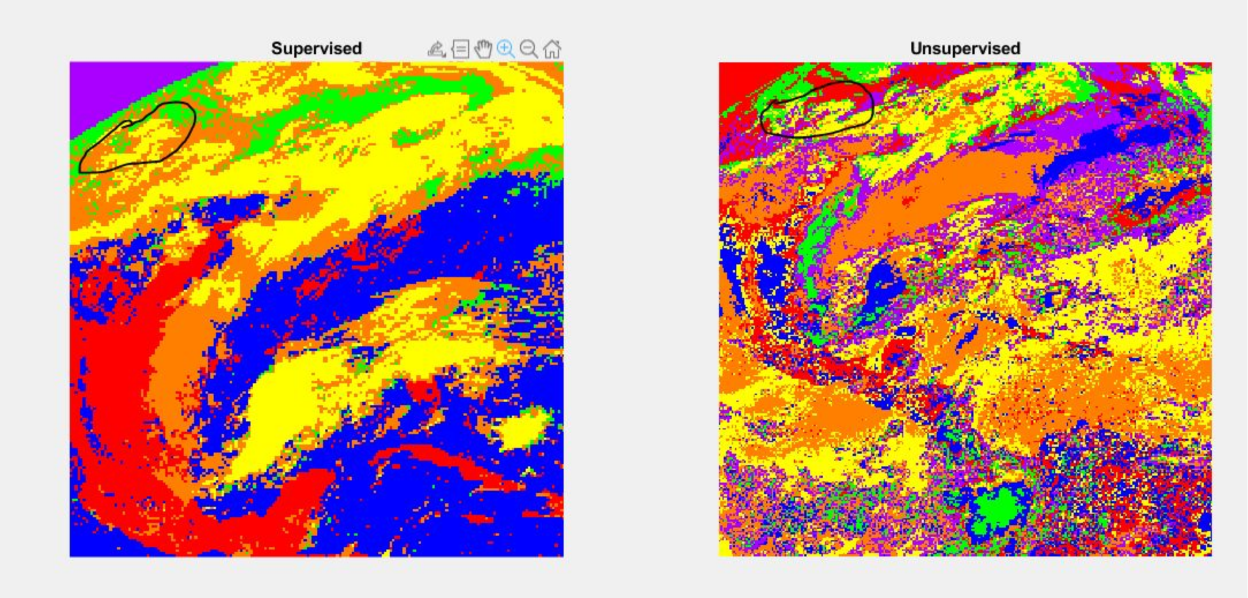
Same Worksheets | © Copyright KidsLearningStation.com | www.kidslearningstation.com

**Q8.** Can you detect any differences in the results based on the different approaches? Did supervised perform better in classifying the objects?

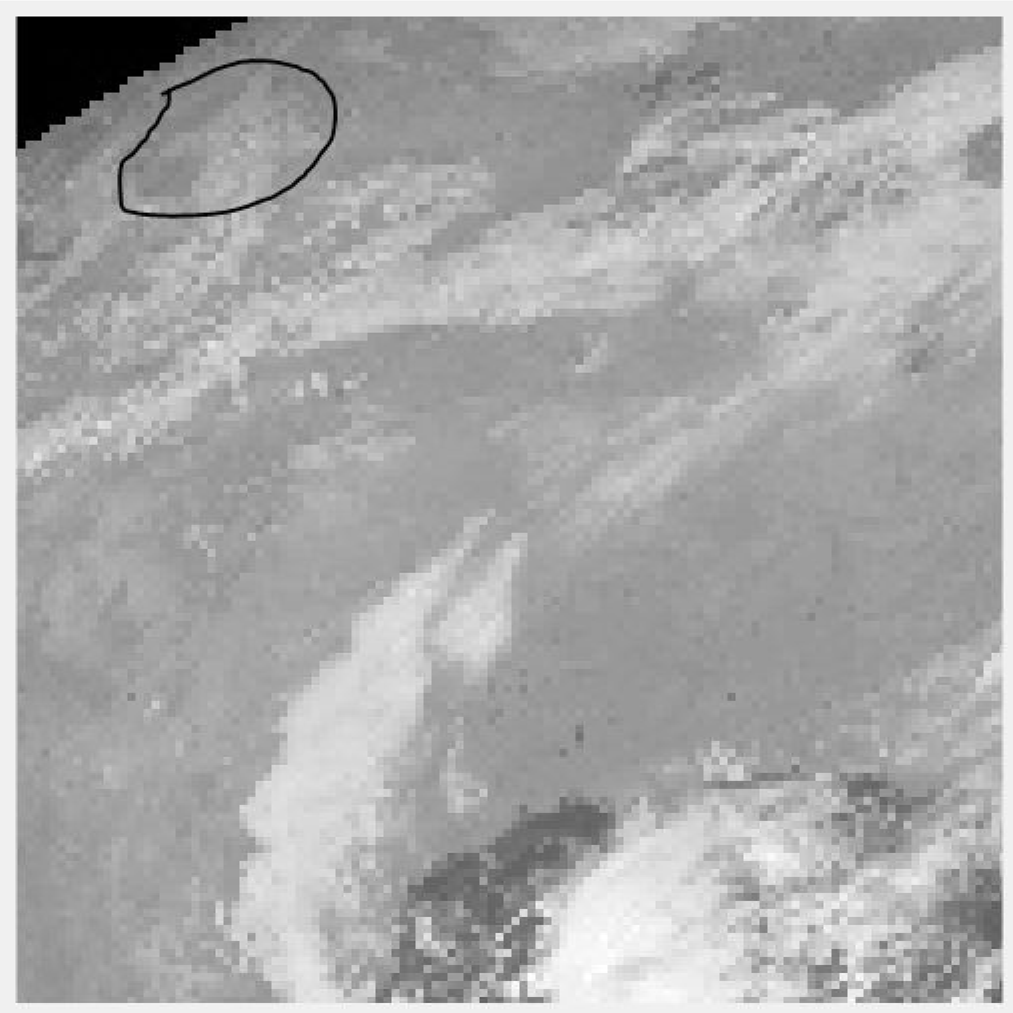
Let us zoom into the IR image at pixel value (191, 59)  197 shown below



Around the same pixel value look at the supervised and unsupervised zoomed images



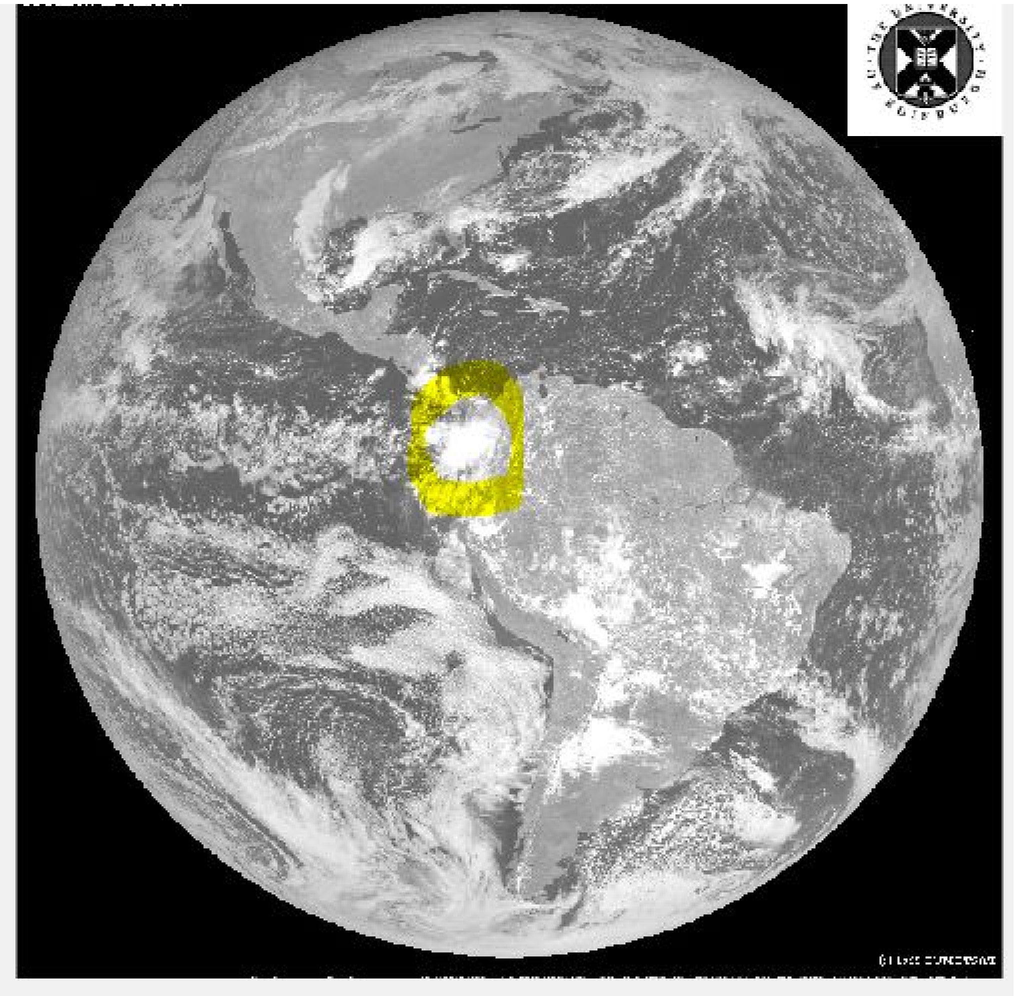
And the zoomed visible image below



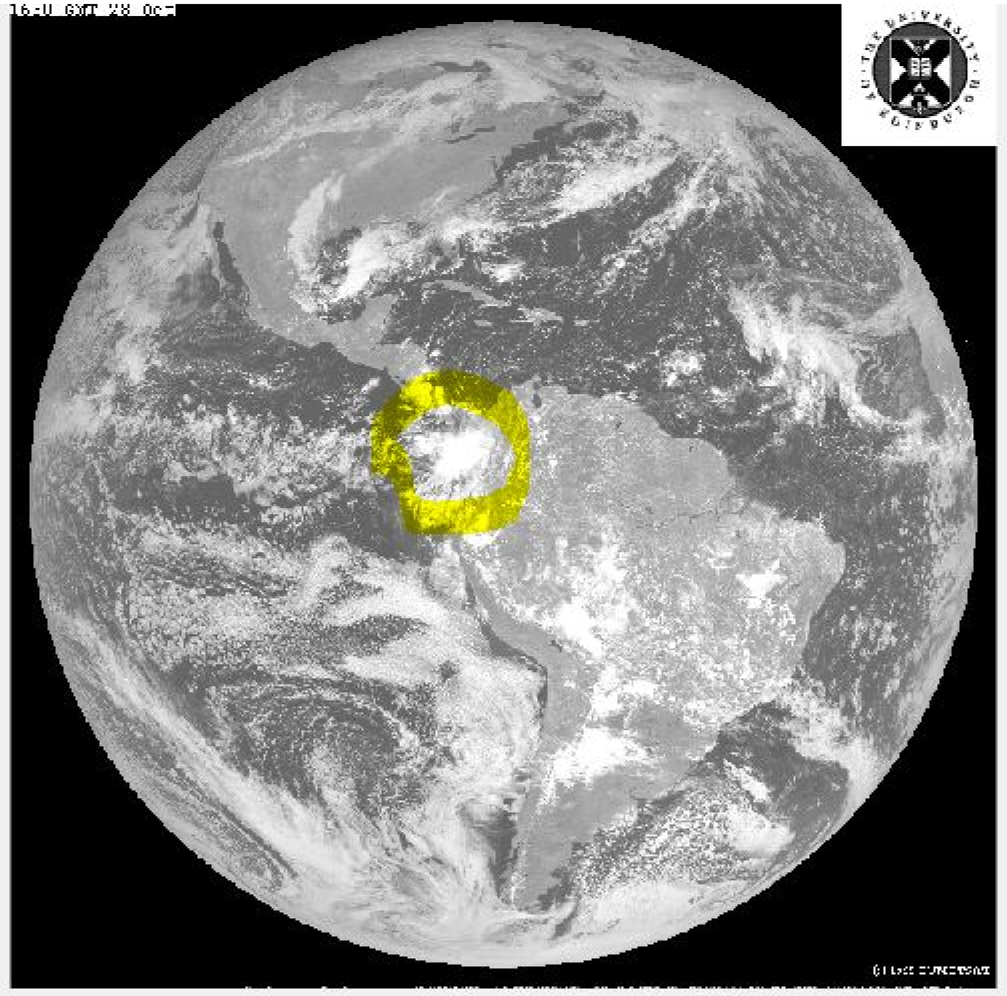
Looks like the supervised image classifies more close to the original IR and visible earth images.

One more example of the snow like area in the north of south american continent.

Below from the visible image in yellow circle

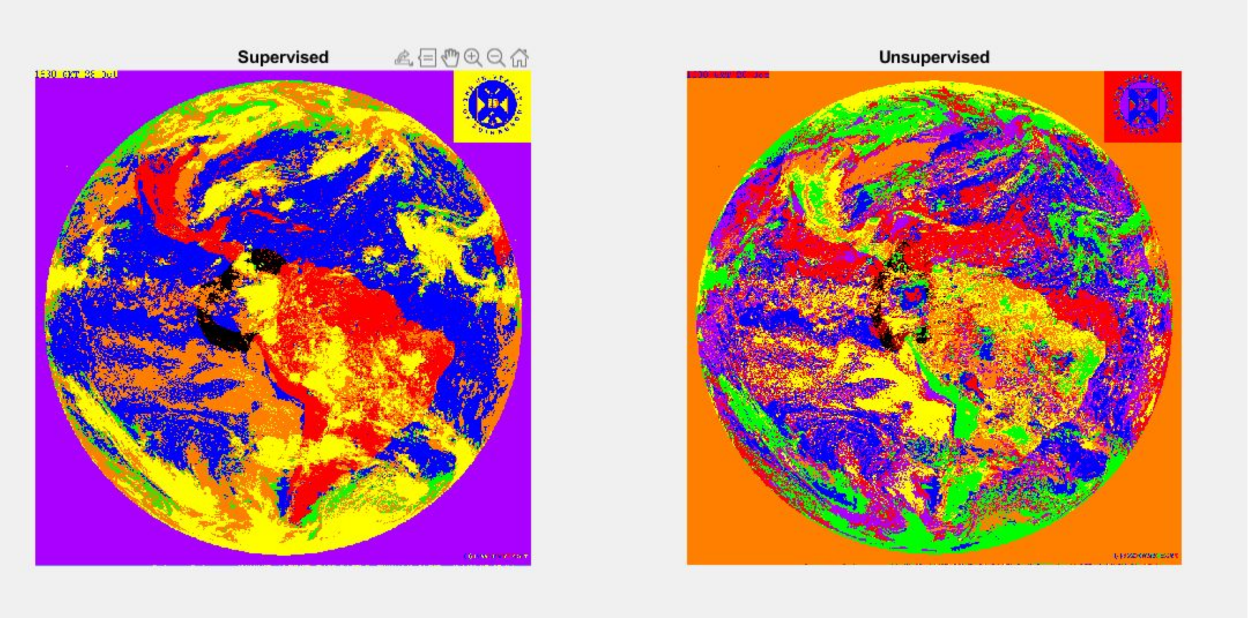


From the IR image in yellow circle



And the same area in the supervised and unsupervised images marked below (in black circle)

I think the supervised algorithm performed better in classifying the objects.



The area from the supervised classification looks closer to the original images.