PART C:

Preface:

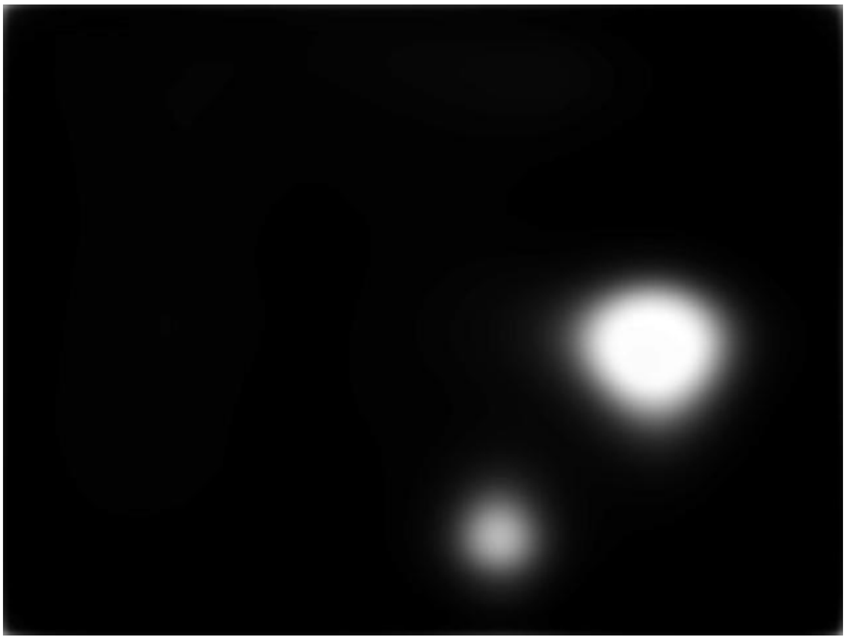
First we need to generate the ROI/Thumbnail using the
generate _roi(input img, salmap, thresh, alpha, beta) function.

```
def generate_roi(salmap, thresh, alpha, beta):

    image=salmap
    # height, width, number of channels in image
    height = image.shape[0]
    width = image.shape[1]
    aspect=width/height
    threshMap = cv2.threshold(image, int(2.55*thresh), 255,cv2.THRESH_BINARY)[1]

    # find contours and get the external one
    contours, hier = cv2.findContours(threshMap, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    roi=[]
    for c in contours:
        # get the bounding rect
        x, y, w, h = cv2.boundingRect(c)
        if w>15 and h>15:
            roi.append([x, y, w, h])
            # cv2.rectangle(image_orig, (x, y), (x+w, y+h), (0, 0, 255), 2)

    # print(roi)
    for r in roi:
        h_opti=round((r[2]/aspect))
        w_opti=round(r[2])
        # draw a red rectangle to visualize the bounding rect
        cv2.rectangle(image_orig, (x, y), (x+w, y+h), (0, 0, 255), 2)
        x1=r[0]
        y1=int(r[1]/4)

        Rs=0
        roi_final=[]
        for j in range(r[3]-h_opti):
            dr=alpha*np.sum(threshMap[r[1]+j:r[1]+j+h_opti, r[0]:r[0]+r[3]])-beta*(h_opti*r[3])
            if dr>Rs:
                Rs=dr
                roi_final=[r[0],r[1]+j,r[3],h_opti]

    return roi_final
```

Examle:

Input Image:

SalMap:



Calculation of Aspect Ratio:

height = image.shape[0]
width = image.shape[1]
aspect=width/height

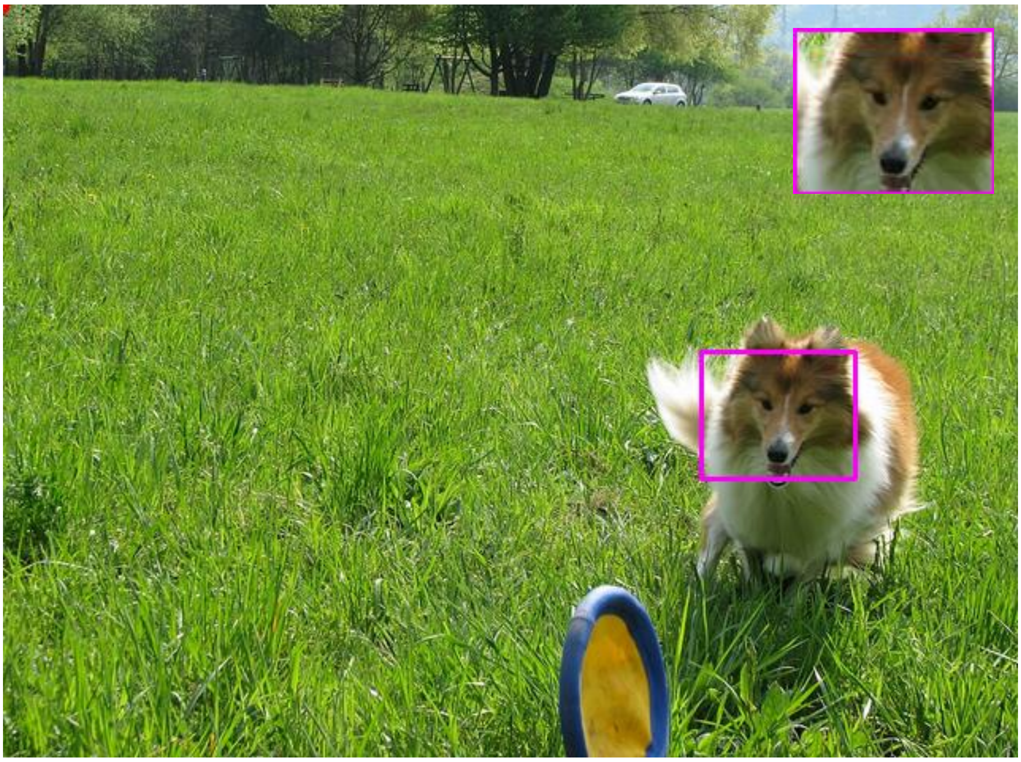
ROI Score:

$$R_s = \alpha \left(\sum_{P \in \mathbf{R}} S_P P \right) - \beta(N)$$

equivalent code:

```
Rs=0
roi_final=[]
for j in range(r[3]-h_opti):
    dr=alpha*np.sum(threshMap[r[1]+j:r[1]+j+h_opti, r[0]:r[0]+r[3]])-beta*(h_opti*r[3])
    if dr>Rs:
        Rs=dr
        roi_final=[r[0],r[1]+j,r[3],h_opti]
```

OUTPUT ROI (With Max Rs)



PARAMETERS:

– α = 0.8, β = 0.3
– α = 1.0, β = 0.1
– α = 0.6, β = 0.4

Precision/Recall Equations:

precision=TP/(TP+FN)
recall=TP/(TP+FP)

where:

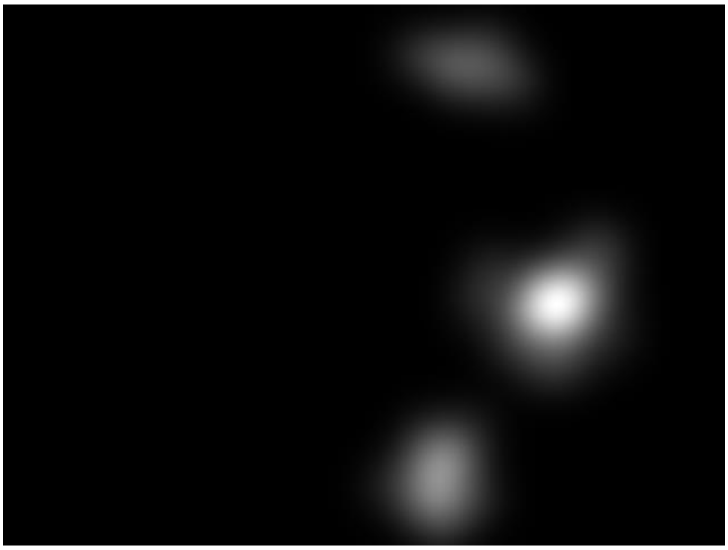
TP=True Positive
FN=False Negative
FP=False Positive

Corresponding Score Move Code :

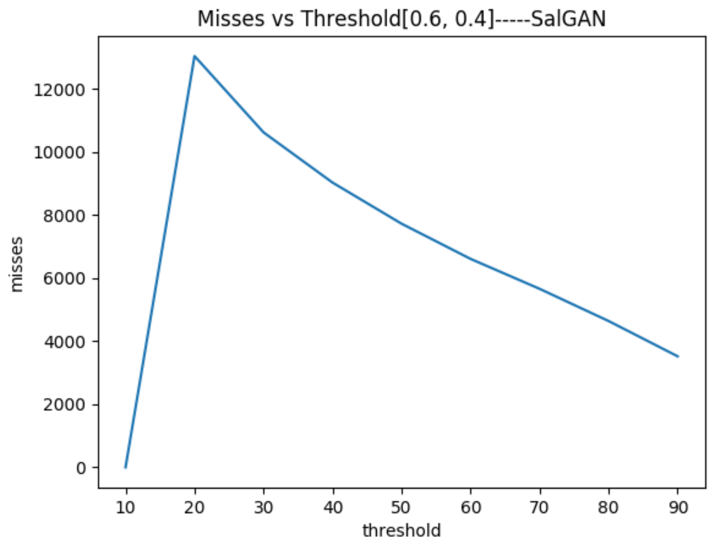
```
def mask_score_roi(mask, roi):

    temp=mask[roi[1]:roi[1]+roi[3],roi[0]:roi[0]+roi[2]]
    TP = np.sum(temp == 255)

    h = mask.shape[0]
    w = mask.shape[1]
    FN = np.sum(temp == 255) -TP

    contours, hier = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    area=0
    for c in contours:
        # get the bounding rect
        x, y, w, h = cv2.boundingRect(c)
        if cv2.contourArea(c)>area:
            r=(x, y, w, h)
```

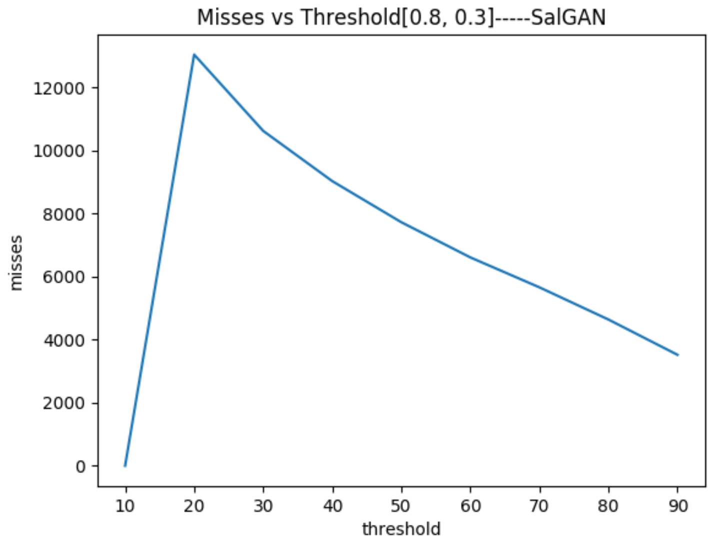
```
            area=cv2.contourArea(c)

    temp=mask[r[1]:r[1]+r[3],r[0]:r[0]+r[2]]
    FP=abs(roi[2]-r[2]) * abs(roi[3]-r[3])

    precision=TP/(TP+FN)
    recall=TP/(TP+FP)

    return precision,recall

def salience_score_roi(map, roi):
    temp=map[roi[1]:roi[1]+roi[3],roi[0]:roi[0]+roi[2]]
    score=np.mean(temp)
    tot=np.sum(temp>0)
    missed=tot-score

    return score,missed

for j in range(3):
    sc=[]
    ms=[]
    th=[]
    for i in range(10,100,10):
        try:
            roi_coords = generate_roi(image_sal, i, param[j][0], param[j][1])
            score,missed = salience_score_roi(image_mask, roi_coords)
            sc.append(score)
            ms.append(missed)
            th.append(i)
        except:
            pass
```

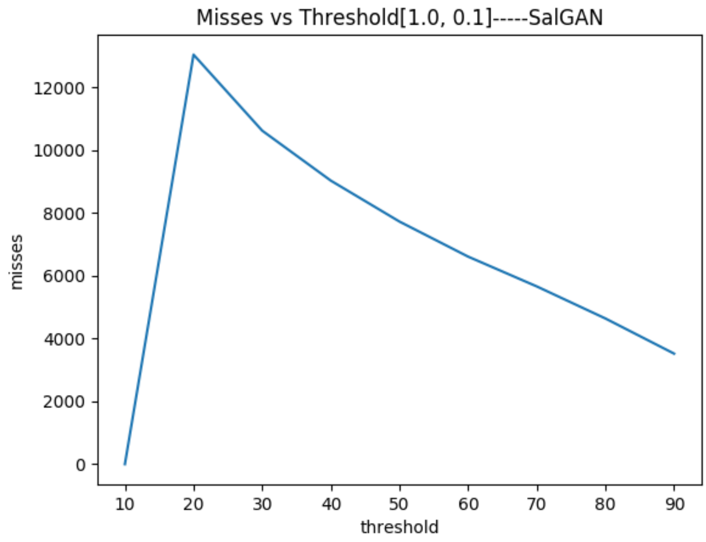
Human Truth:

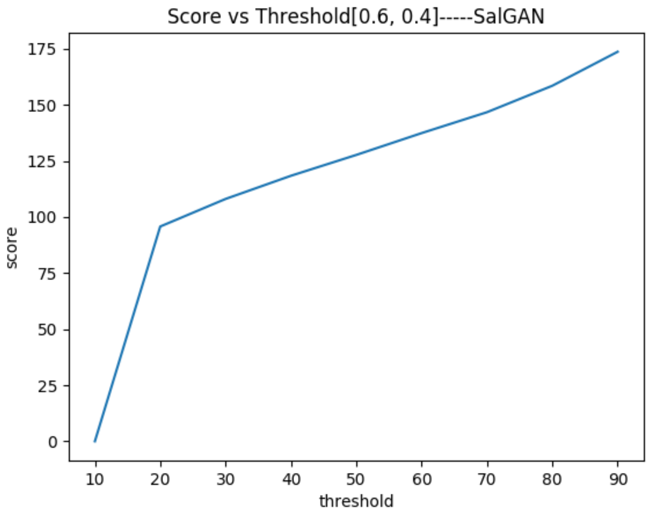


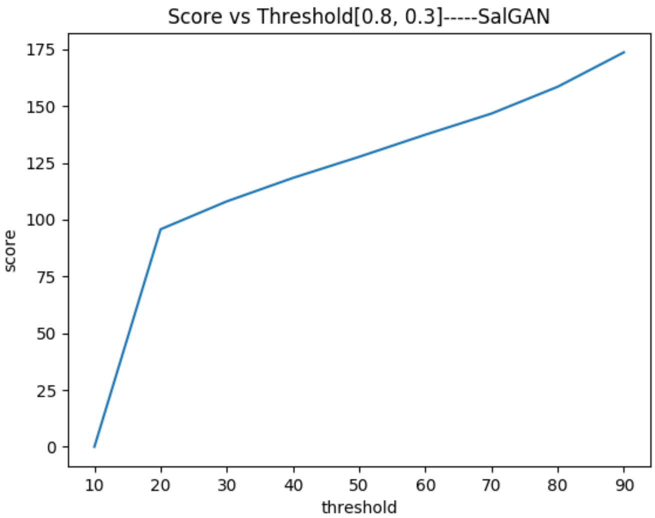
PLOTS (SalGAN):

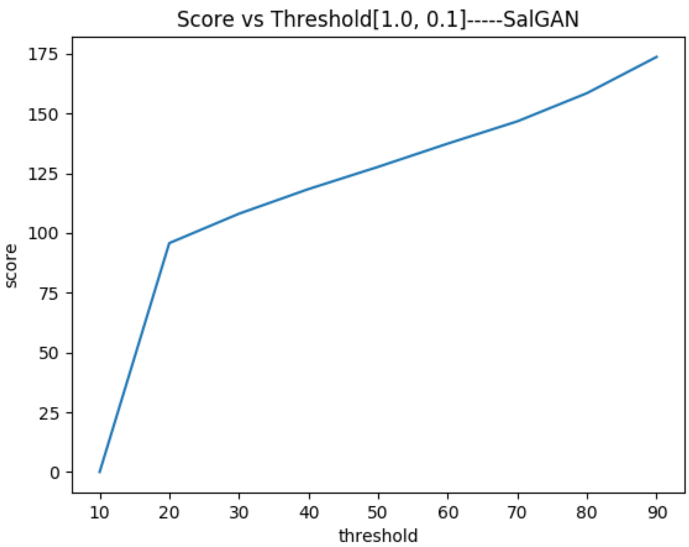




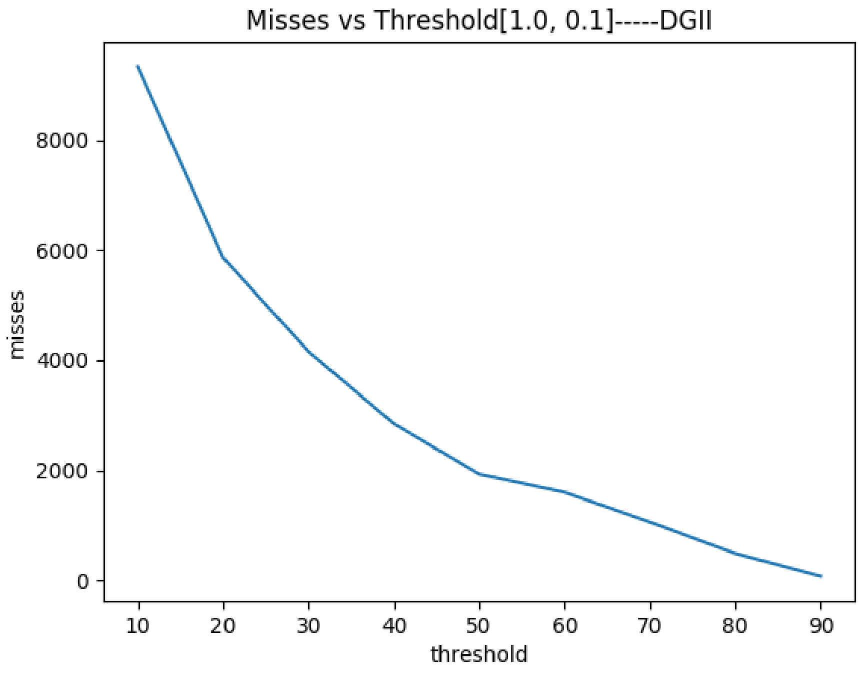



Score vs Threshold[0.6, 0.4]-----SalGAN
score
threshold
175
150
125
100
75
50
25
0
10
20
30
40
50
60
70
80
90

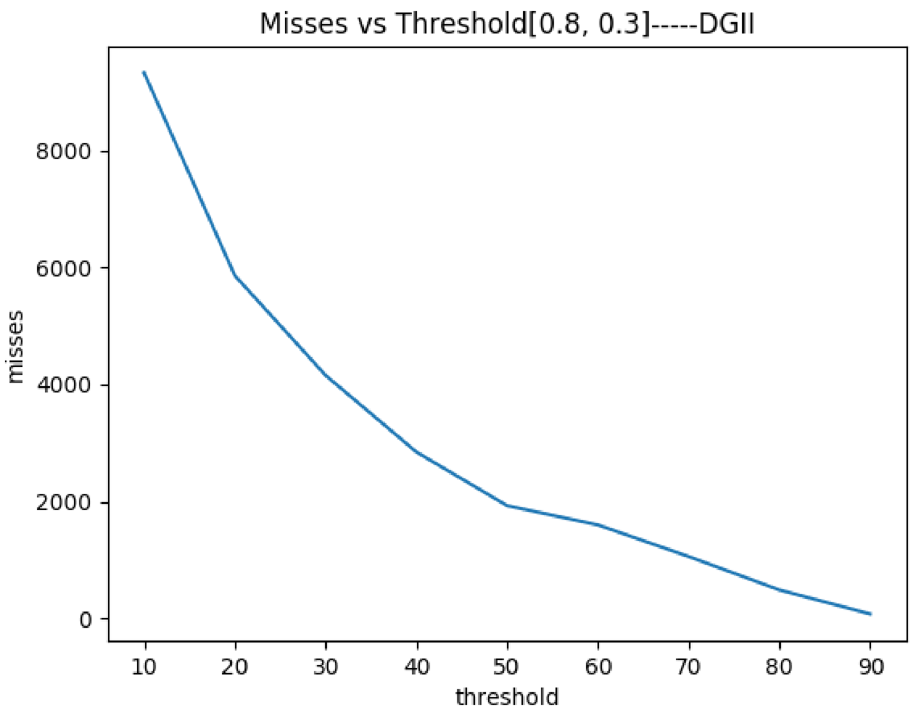

Score vs Threshold[0.8, 0.3]-----SalGAN
score
threshold
175
150
125
100
75
50
25
0
10
20
30
40
50
60
70
80
90


Score vs Threshold[1.0, 0.1]-----SalGAN
score
threshold
175
150
125
100
75
50
25
0
10
20
30
40
50
60
70
80
90

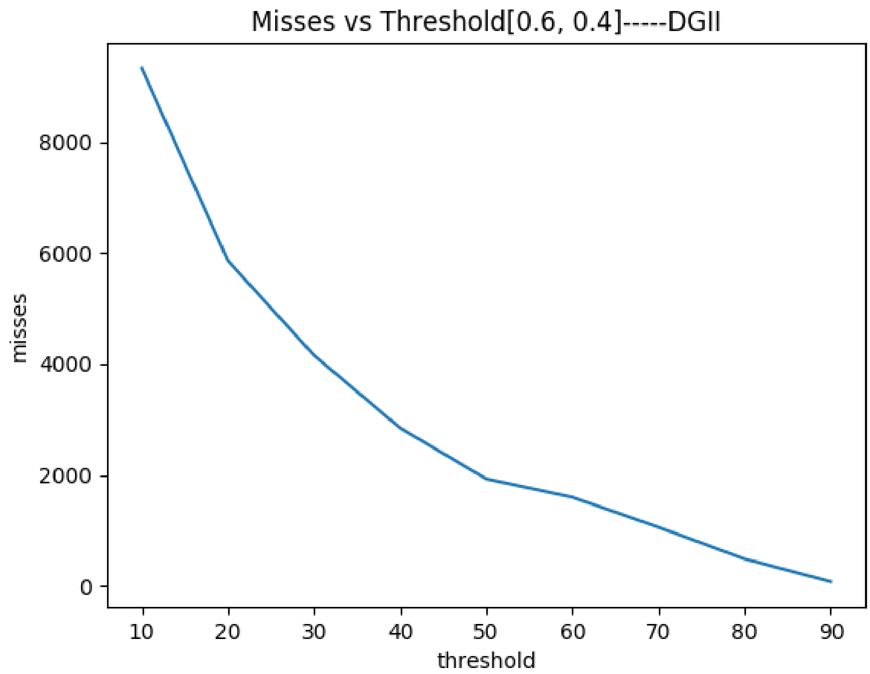
PLOTS (DGII):

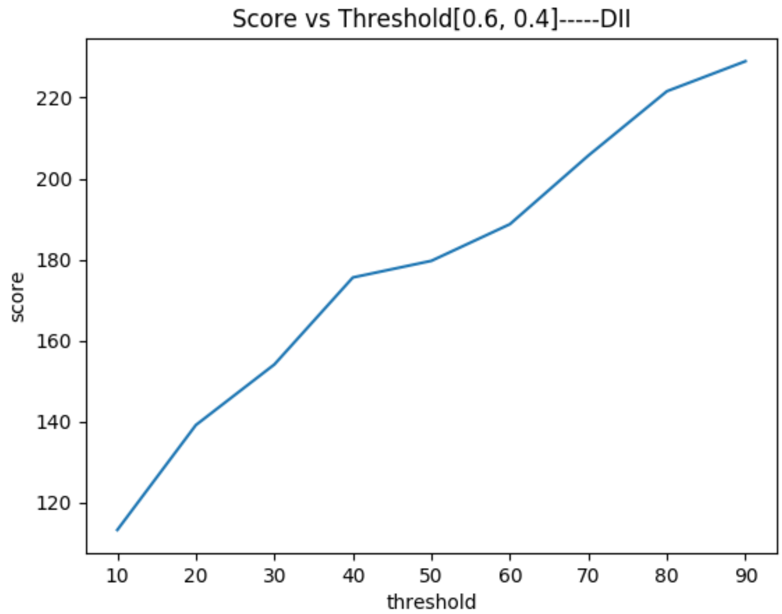


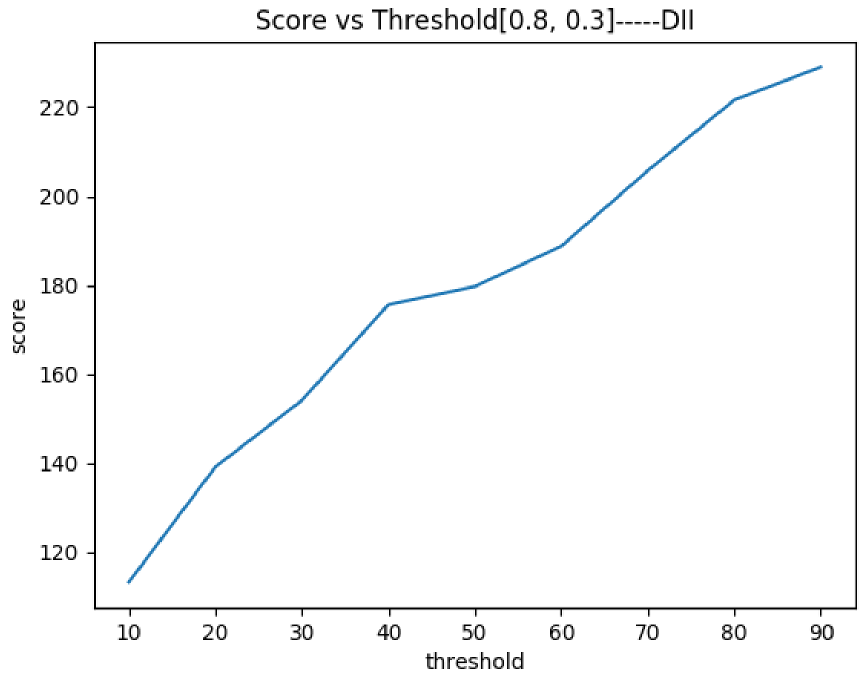
Misses vs Threshold[1.0, 0.1]-----DGII
misses
threshold
0
2000
4000
6000
8000
10
20
30
40
50
60
70
80
90

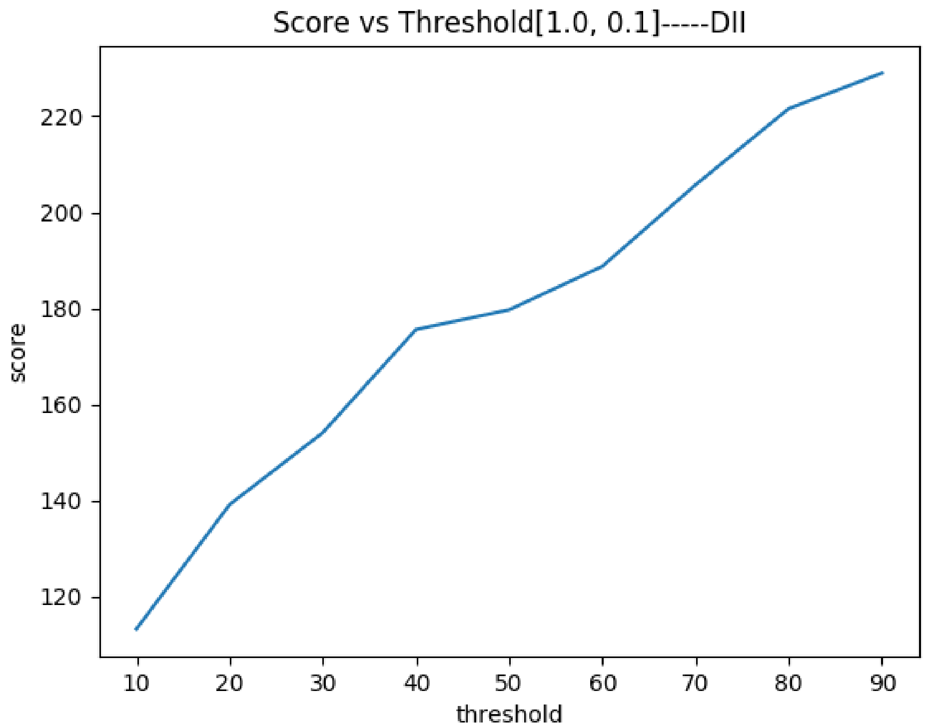


Misses vs Threshold[0.8, 0.3]-----DGII
misses
threshold
0
2000
4000
6000
8000
10
20
30
40
50
60
70
80
90


Misses vs Threshold[0.6, 0.4]-----DGII
misses
0
2000
4000
6000
8000
10
20
30
40
50
60
70
80
90
threshold


Score vs Threshold[0.6, 0.4]-----DII
score
120
140
160
180
200
220
10
20
30
40
50
60
70
80
90
threshold





Discussion:

For SalGAN and DGII, obviously the SalGan has better AUC than DGII as evidentfrom the plots. However, for the other salincy maps, there no visual or numerical conclusion from plots, hence not showed.

The score to threshold maps shows that, for the the DGII odel, the plot is almost linear, howeer for salgan, the plot is almot exponential.It can thus be concluded that for salgan, the increase in threshold has more incremental effect on score than DGII. I would prefer SalGAN for its better AUC and Scor/Threshold values.