

Contents

- [Upload excel file](#)
- [Plot products](#)
- [Boxplot for different products](#)
- [Getting values to box plot for different values of bottles in the bin.](#)
- [Boxplot](#)
- [IoU plot different weights](#)
- [Splitting items](#)

Upload excel file

```
close all
clc
clear all
m = readtable('C:\Users\Aron Gauti\Dropbox\HR\Meistaraverkefni\Results\toExcel2175.csv');
% m = readtable('toExcel2175.csv');
imnr = table2array(m(:,1));
filename = table2array(m(:,2));
bottles = table2array(m(:,3));
detections = table2array(m(:,4));
truepos = table2array(m(:,5));
falsepos = table2array(m(:,6));
iou = table2array(m(:,7));
mind = table2array(m(:,8));
maxd = table2array(m(:,9));
aved = table2array(m(:,10));
% Precision and Recall
precision = truepos./(falsepos+truepos); nanpre = isnan(precision); precision(nanpre) = 0;
recall = truepos./(truepos+(bottles-truepos)); nanrec = isnan(recall); recall(nanrec) = 0;
f1 = 2*((precision.*recall)./(precision+recall)); f1(isnan(f1)) = 0;
m('Precision') = precision;
m('Recall') = recall;
m('F1') = f1;
```

Warning: Column headers from the file were modified to make them valid MATLAB identifiers before creating variable names for the table. The original column headers are saved in the VariableDescriptions property. Set 'VariableNamingRule' to 'preserve' to use the original column headers as table variable names.

Plot products

```
close all
clc
a = 1;
maxY = 0;
minY = 1;
prnames = {'1' '2' '3' '4' '5' '6' '7' '8' '9' '10' '11' '12' '13' '14' '15'};
pr = NaN(145, length(prnames));
prf1 = NaN(145, length(prnames));
counter = 1;
numbersname = zeros(1,2175);
for i = 1:145:length(imnr)
    for j = 1:145
        y(j) = iou(j+(i-1));
        f1curr(j) = f1(j+(i-1));
        numbersname(j+(i-1)) = counter;
    end
    pr(1:145,counter) = y;
    prf1(1:145,counter) = f1curr;
    counter = counter +1;
    x = 1:1:145;
    length(y);
    averageY = mean(y);
    if averageY > maxY
        maxY = averageY;
        maxIndex = a;
    end
    if averageY < minY
        minY = averageY;
        minIndex = a;
    end
    figure(a)
    scatter(x, y)
    hold on

    plot(x,ones(length(x),1)*averageY,"-r")
    title(['IOU for item nr.' num2str(a)])
    xlabel('IMG number')
    ylabel('Intersection over Union')
    legend('Intersection over Union', 'Average')
    name = "C:\Users\Aron Gauti\Dropbox\HR\Meistaraverkefni\Results\item"+ num2str(a) + ".png";
```

```

saveas(gcf,name,'png')

a = a+1;
end
% Max averages IOU
maxY
maxIndex

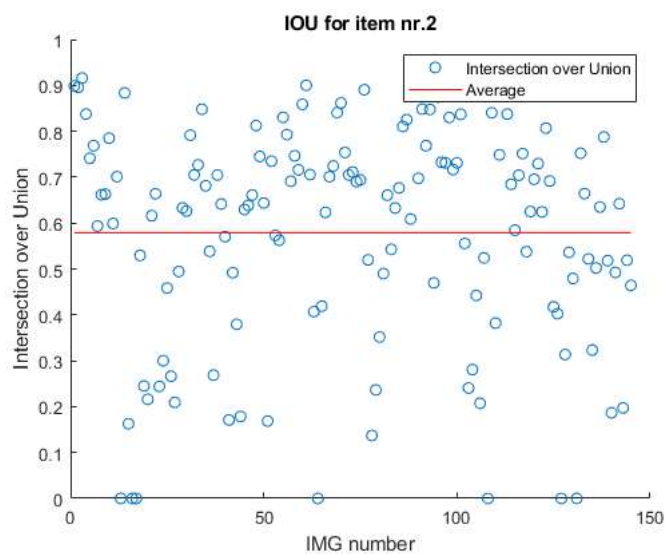
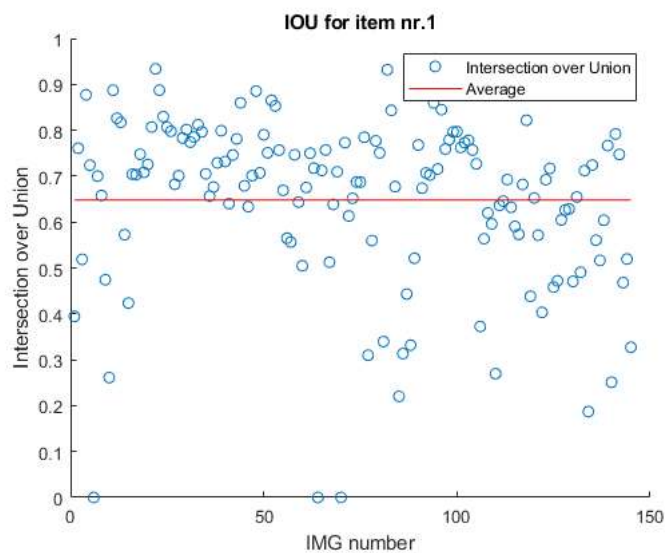
```

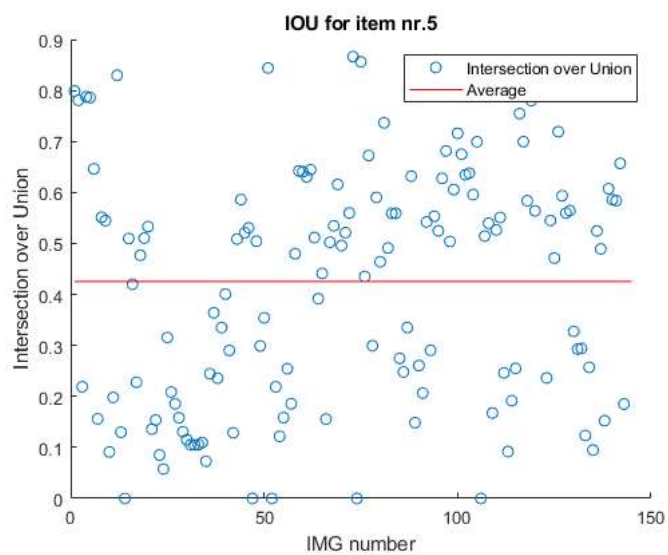
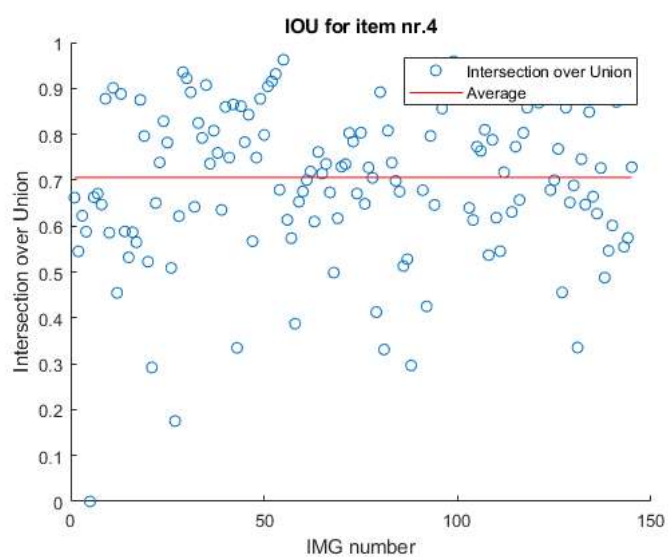
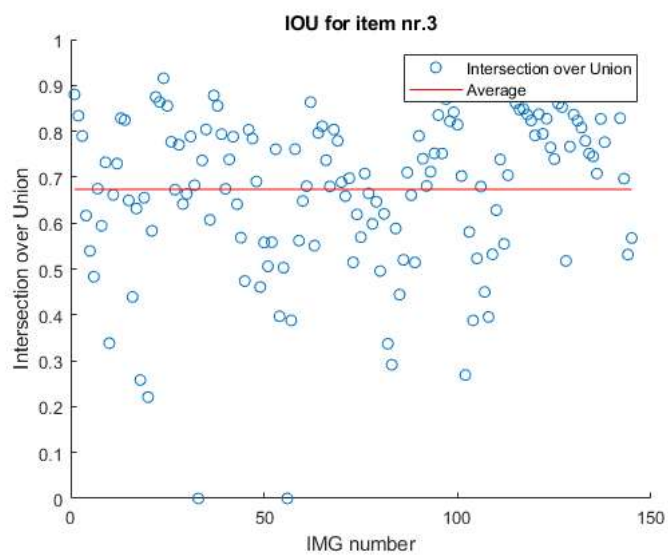
maxY =

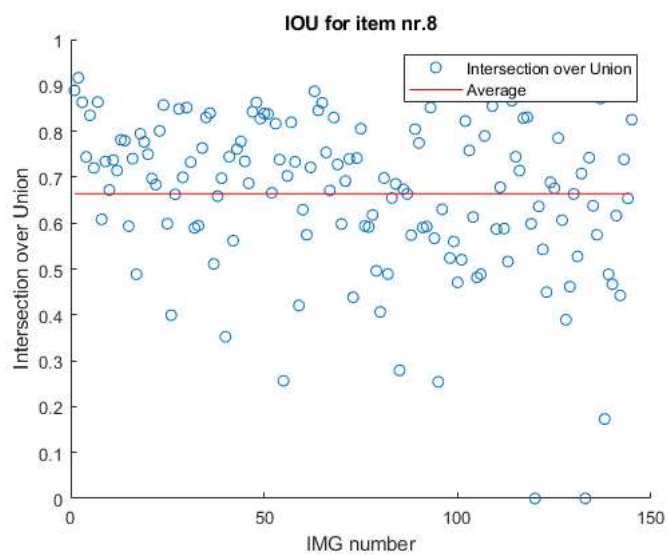
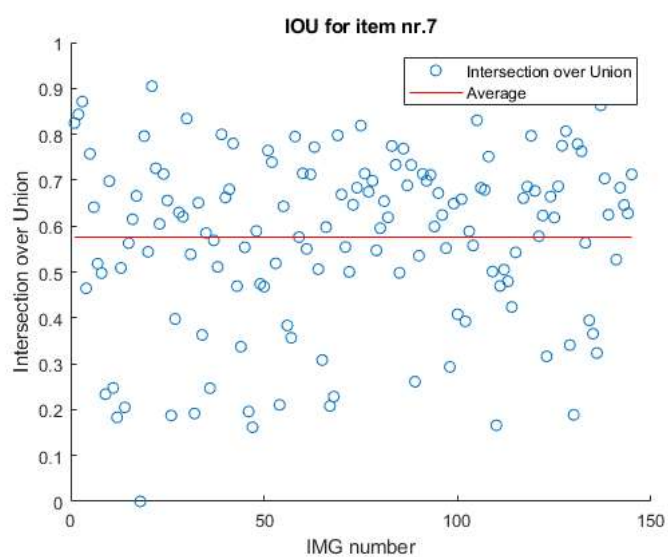
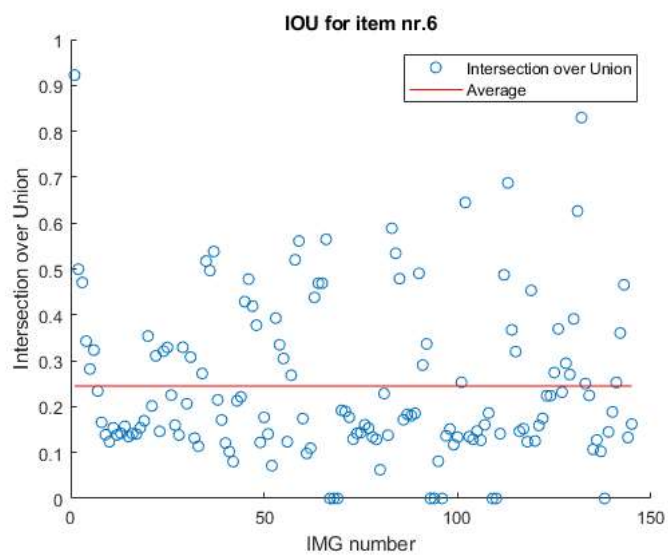
0.72452

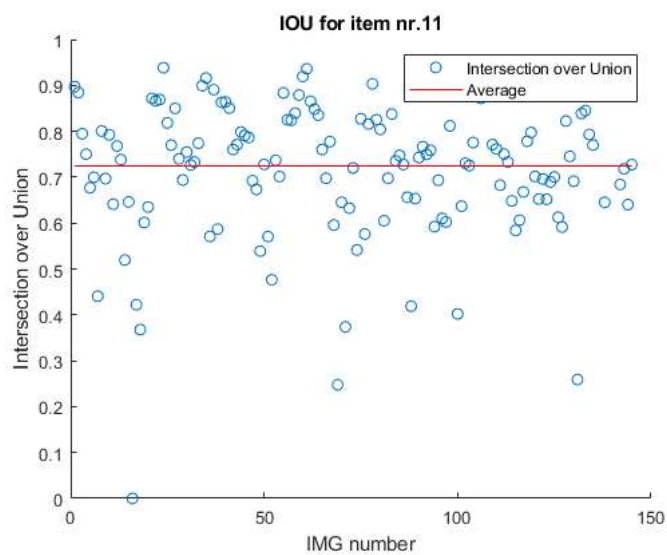
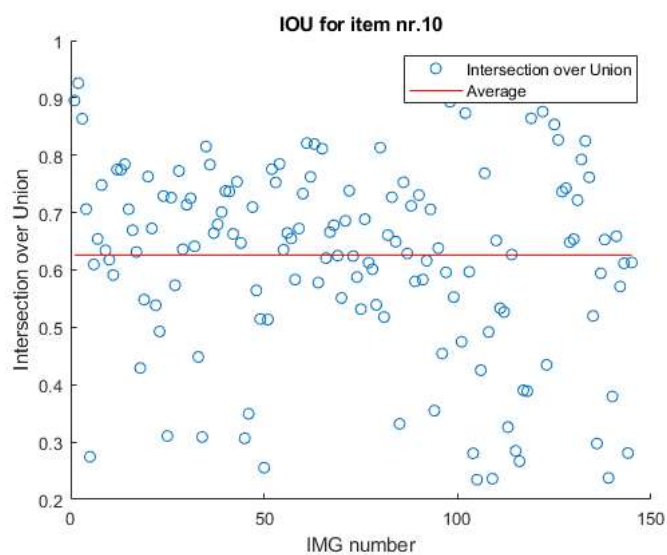
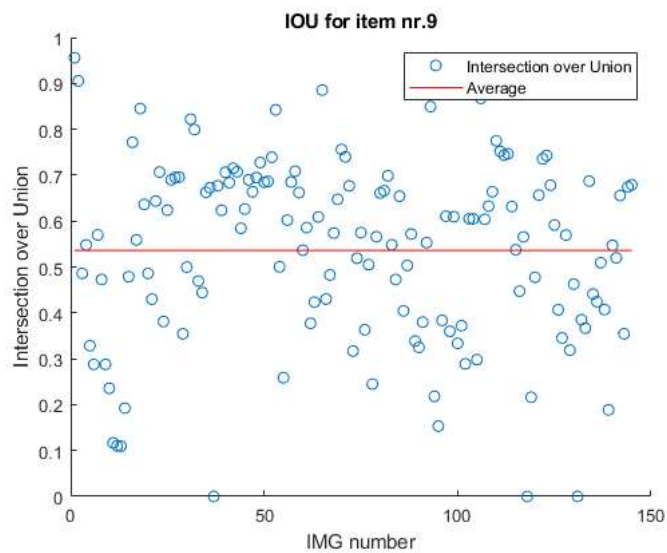
maxIndex =

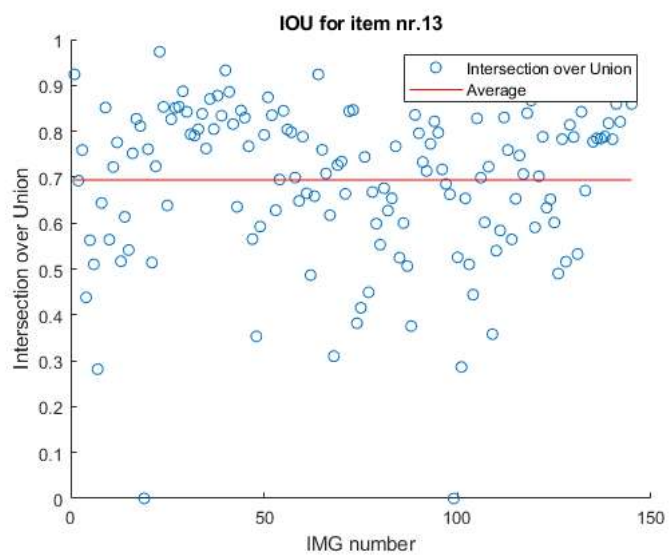
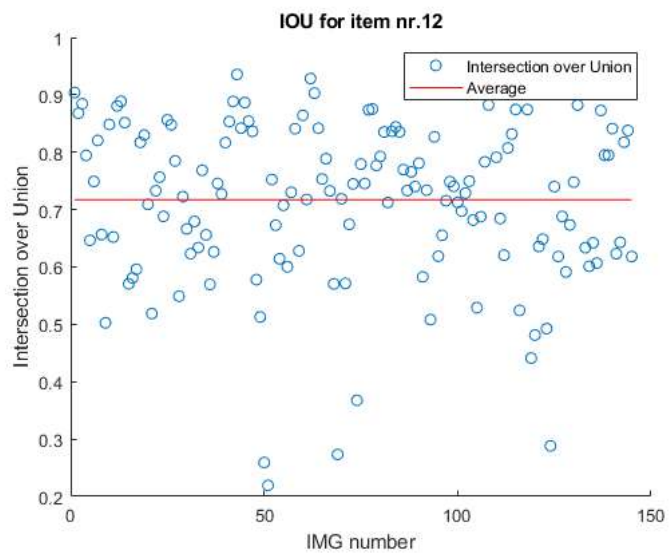
11

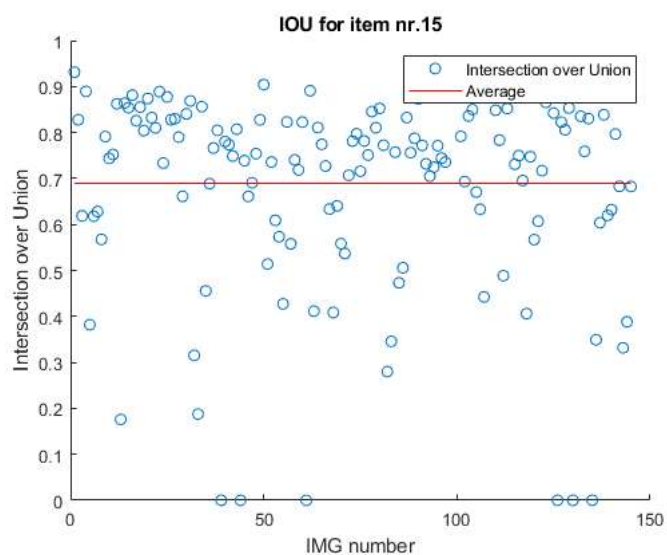
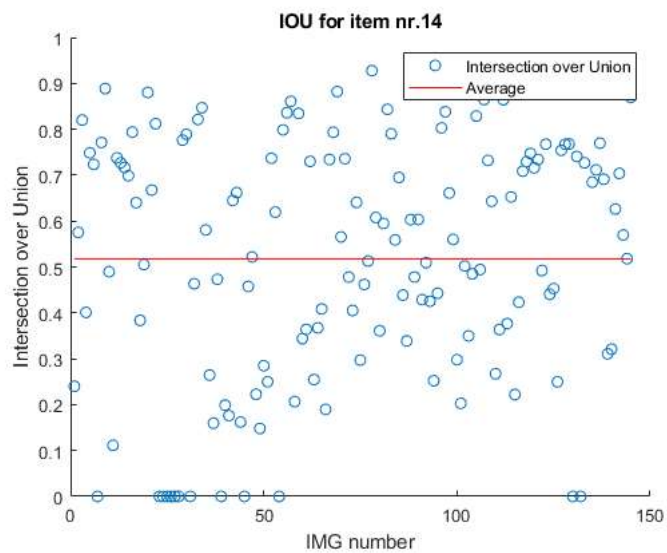








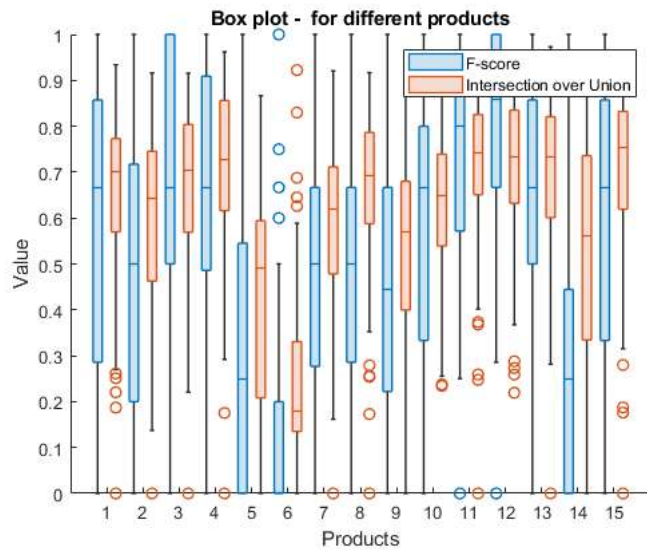




Boxplot for different products

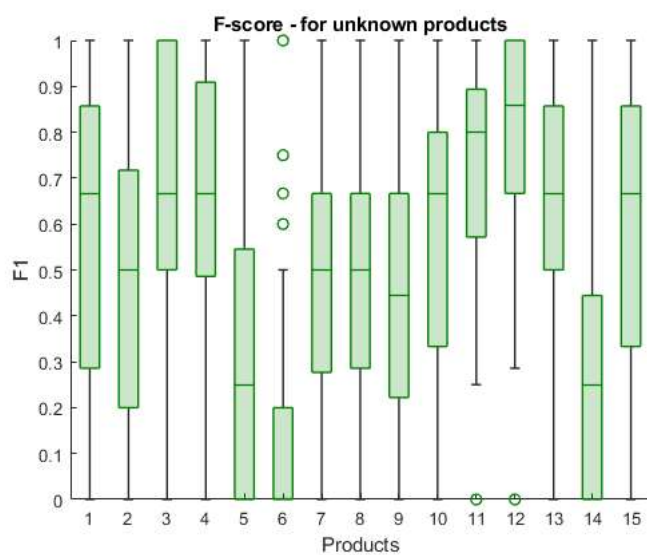
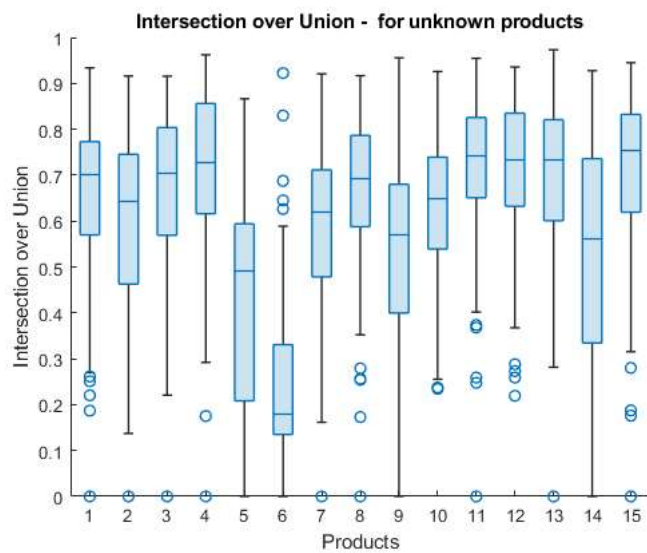
```
iouname = strings(1,2175); iouname(:) = 'IoU';
fname = strings(1,2175); fname(:) = 'F1';
name = [iouname,fname]';
blend = [iou; f1];
blendnum = [numbersname';numbersname'];
```

```
figure()
boxchart(blendnum, blend, 'GroupByColor', name)
title(['Box plot - for different products'])
xlabel('Products')
ylabel('Value')
xticks([1 2 3 4 5 6 7 8 9 10 11 12 13 14 15])
axis([0 16 0 1])
legend('F-score','Intersection over Union')
```



```
figure()
%subplot(1,2,1)
boxchart(pr)
title(['Intersection over Union - for unknown products'])
xlabel('Products')
ylabel('Intersection over Union')
%subplot(1,2,2)
figure()
boxchart(prf1,'BoxFaceColor',[0 0.5 0],'MarkerColor',[0 0.5 0])
title(['F-score - for unknown products'])
xlabel('Products')
ylabel('F1')

name = "C:\Users\Aron Gauti\Dropbox\HR\Meistaraverkefni\Results\boxplotForProducts.png";
saveas(gcf,name,'png')
```

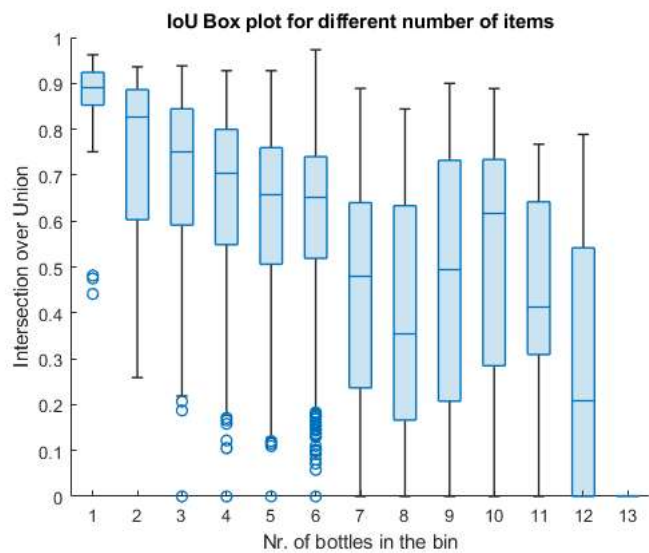



Getting values to box plot for different values of bottles in the bin.

```
names = {'one' 'two' 'three' 'four' 'five' 'six' 'seven' 'eight' 'nine' 'ten' 'eleven' 'twelve' 'thirteen'};
M = NaN(787, length(names));
for i=1:length(names)
    M(1:length(iou(find(bottles == i))),i) = iou(find(bottles == i));
end
```

Boxplot

```
figure()
boxchart(M)
title(['IoU Box plot for different number of items'])
xlabel('Nr. of bottles in the bin')
ylabel('Intersection over Union')
name = "C:\Users\Aron Gauti\Dropbox\HR\Meistaraverkefni\Results\boxplotBottles.png";
saveas(gcf,name,'png')
```



IoU plot different weights

```
close all
epochs = 0:1000:40000;

iouw = [0 5178 8403 9002 8867 9163 9184 9206 9288 9286 9163 8279 9410 9401 8815 9248 9352 9357 9439 9421 9416 9476 9458 9495 9427 9429 9455 9470 9536 9504 9494 9506];
hi = max(iouw)
hiidx = find(iouw==hi)*1000
hiarr = "(" +string(hiidx)+ "," +string(hi)+ ")"
figure()
plot(epochs, iouw, 'LineWidth',6)
hold on
plot(hiidx,hi, 'r*', 'LineWidth',10)
text(hiidx-2000,0.93,hiarr,'Color','r')
title(['Neural network - IoU every 1000 epochs'])
xlabel('Nr. of epochs')
ylabel('Average IoU on test set')
legend('IoU', 'Highest IoU')
```

```
hi =

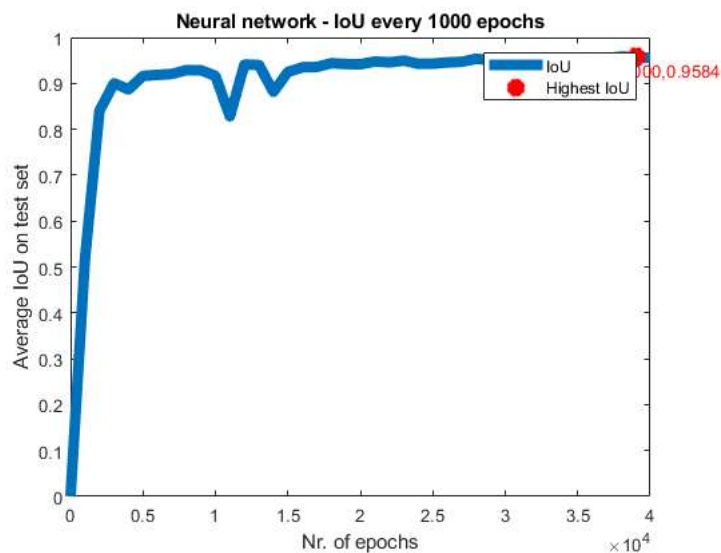
    0.9584

hiidx =

    39000

hiarr =

    "(39000,0.9584)"
```



Splitting items

```
format short g
i1 = m(m.ImageNr <= 145, :); i2 = m((m.ImageNr <= 145*2 & m.ImageNr > 145), :);
i3 = m((m.ImageNr <= 145*3 & m.ImageNr > 145*2), :); i4 = m((m.ImageNr <= 145*4 & m.ImageNr > 145*3), :);
i5 = m(m.ImageNr <= 145*5 & m.ImageNr > 145*4, :); i6 = m(m.ImageNr <= 145*6 & m.ImageNr > 145*5, :);
i7 = m(m.ImageNr <= 145*7 & m.ImageNr > 145*6, :); i8 = m(m.ImageNr <= 145*8 & m.ImageNr > 145*7, :);
i9 = m(m.ImageNr <= 145*9 & m.ImageNr > 145*8, :); i10 = m(m.ImageNr <= 145*10 & m.ImageNr > 145*9, :);
i11 = m(m.ImageNr <= 145*11 & m.ImageNr > 145*10, :); i12 = m(m.ImageNr <= 145*12 & m.ImageNr > 145*11, :);
i13 = m(m.ImageNr <= 145*13 & m.ImageNr > 145*12, :); i14 = m(m.ImageNr <= 145*14 & m.ImageNr > 145*13, :); i15 = m(m.ImageNr <= 145*15 & m.ImageNr > 145*14, :);
for i=1:15
    itemprod(i) = sum(eval(['i' num2str(i) '.Bottles']));
    itemdet(i) = sum(eval(['i' num2str(i) '.Detections']));
    itemTP(i) = sum(eval(['i' num2str(i) '.TruePositive']));
    itemFP(i) = sum(eval(['i' num2str(i) '.FalsePositive']));
    itemIoU(i) = round(mean(eval(['i' num2str(i) '.IOU'])),4);
    itemPr(i) = round(mean(eval(['i' num2str(i) '.Precision'])),4);
    itemRe(i) = round(mean(eval(['i' num2str(i) '.Recall'])),4);
    itemF1(i) = round(mean(eval(['i' num2str(i) '.F1'])),4);
end
itemnr = 1:15;

avgtable = table(itemnr', itemprod', itemdet', itemTP', itemFP', itemIoU', itemPr', itemRe', itemF1');
avgtable.Properties.VariableNames = {'Item' 'All-Products' 'All-Detections' 'All-True Positive' 'All-False Positive' 'Avg-IoU' 'Avg-Precision' 'Avg-Recall' 'Avg-F1'}
writetable(avgtable,'table.csv', 'Delimiter',';')
```

avgtable =

15x9 table

Item	All-Products	All-Detections	All-True Positive	All-False Positive	Avg-IoU	Avg-Precision	Avg-Recall	Avg-F1
1	684	332	300	29	0.648	0.8299	0.5032	0.595
2	1029	392	330	55	0.579	0.7327	0.3862	0.471
3	667	368	337	29	0.6734	0.8523	0.569	0.6542
4	683	350	325	24	0.7058	0.8891	0.5426	0.6394
5	892	317	202	110	0.4253	0.5205	0.2418	0.3194
6	918	195	57	129	0.2449	0.1833	0.0682	0.0956
7	851	405	326	78	0.5757	0.7314	0.3907	0.4956
8	788	292	264	26	0.664	0.8615	0.3707	0.4895
9	887	333	270	60	0.536	0.7218	0.3308	0.4349
10	665	344	293	51	0.626	0.7736	0.4761	0.5667
11	627	385	361	23	0.7245	0.8994	0.623	0.7106
12	574	417	394	23	0.7171	0.9268	0.7376	0.7971
13	618	307	291	14	0.694	0.9149	0.5309	0.6403
14	1031	245	180	52	0.5178	0.6299	0.2211	0.3012
15	616	302	273	23	0.6897	0.8368	0.4918	0.5858