Measuring Objectness of Image Windows

-Poonam Warade(MT2017511)

Platform:

Linux
Opencv

Steps to Run Code:

1.Download the dataset from given linnks and extract those in respective folders.
2. Move to bin folder
3. Run python test_object_detector.py (To get results without measuring objectness of image windows)
4. Run python test_objectness.py (To get results WITH measuring objectness of image windows)

Github Link:

https://github.com/Poonam0602/DigitalImageProcessing

Abstract:

Object is defined as something which has,

- A different appearance from its surroundings
- Sometimes it is unique within the image and stands out as salient
- A well defined closed boundry in space

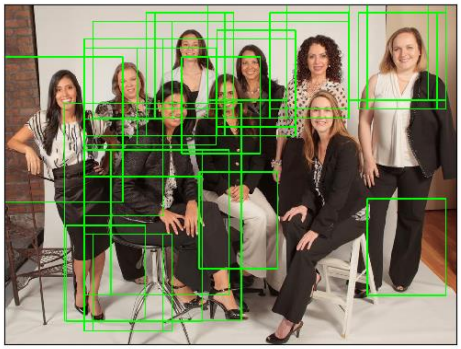
In this paper, they had claimed that including objectness measure will reduce the computation requirement and time for getting results in variety of applications such as object detection, object tracking, content aware image resizing, etc.

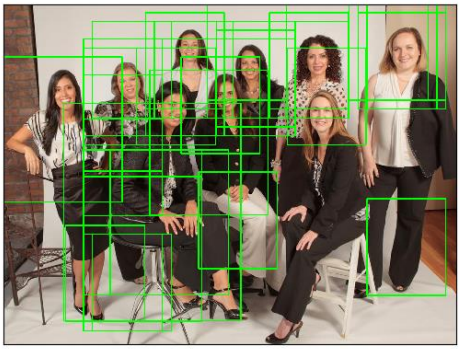
As a part of my work on this paper, I showed how this paper has reduced number of image windows required to be processed to get same results as of without using objectness measure.

I used HoG descriptor based human detection paper. The file test_object_detector.py demonstrates the paper "Histogram of Oriented Gradients (By Dalal)". With my extension to the code, objectness measure (partial though), results can be observed from test_objectness.py

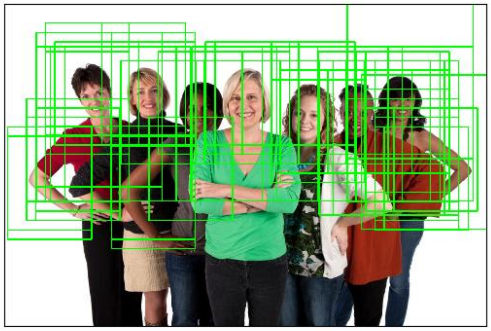
Results:

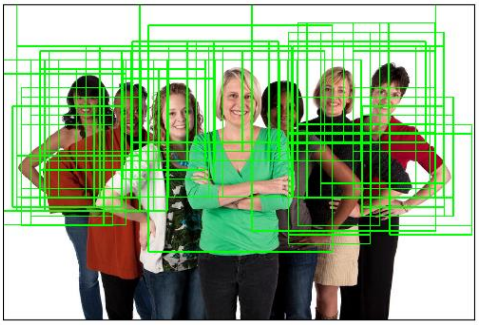
(Figures: With Objectness || Without Objectness)



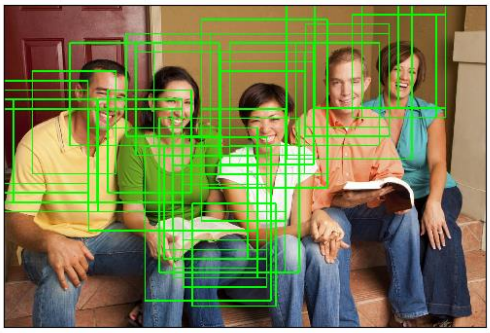


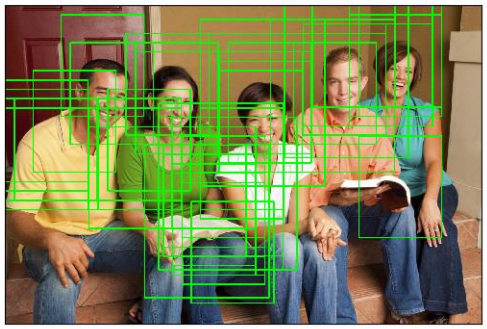
# of Windows reduced from 1699 to 31



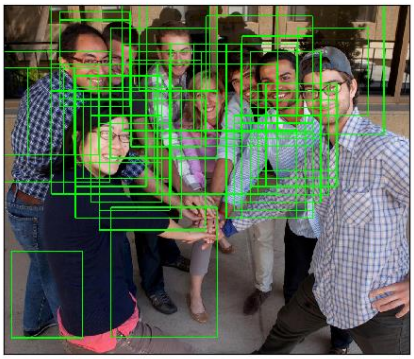


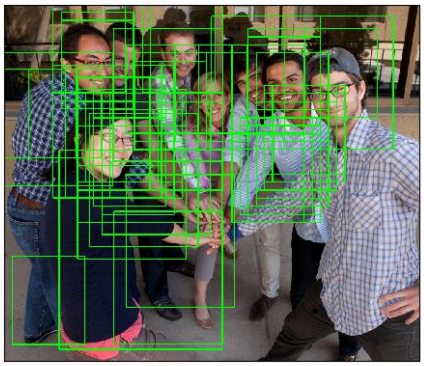
# of Windows reduced from 1414 to 64



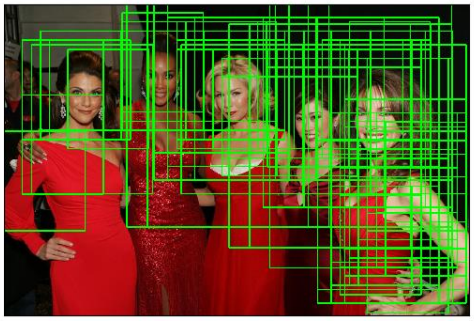


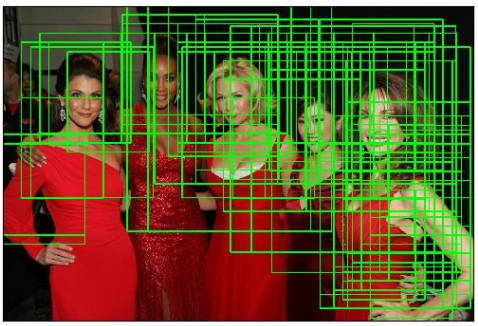
# of Windows reduced from 1414 to 47



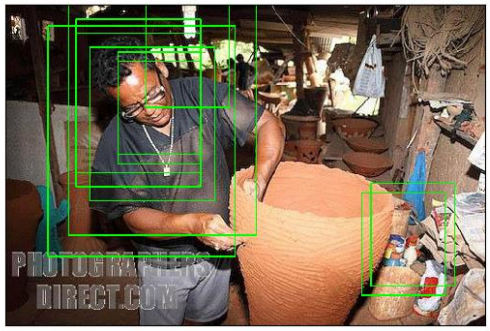


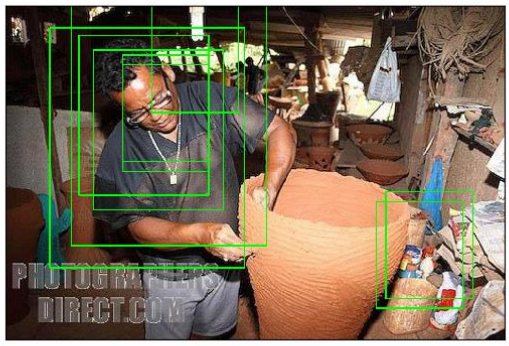
# of Windows reduced from 2103 to 51





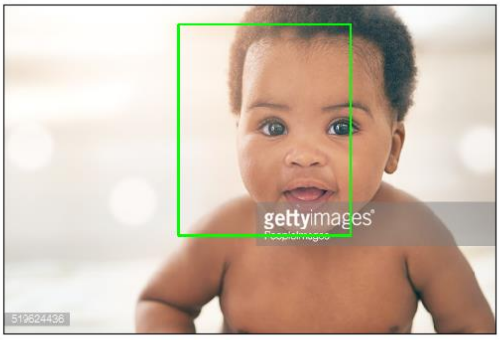
# of Windows reduced from 1414 to 114

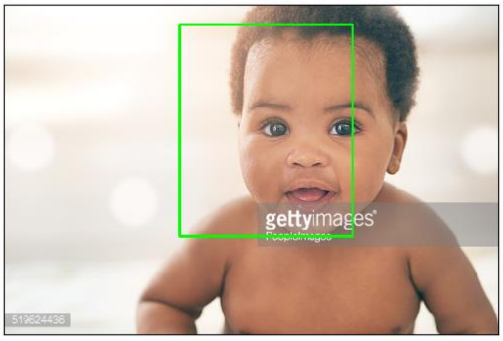




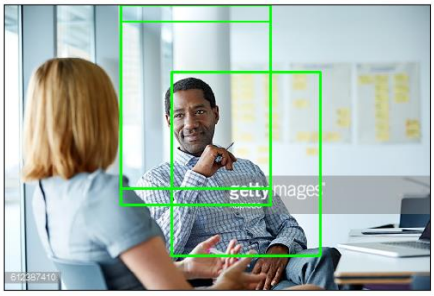
# of Windows reduced from 1414 to 10

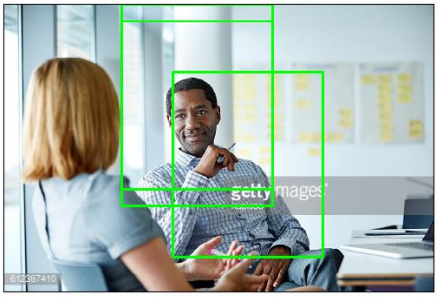
# of Windows reduced from 120 to 1



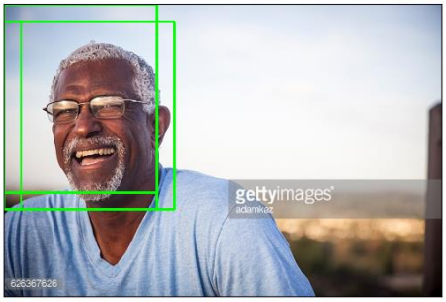


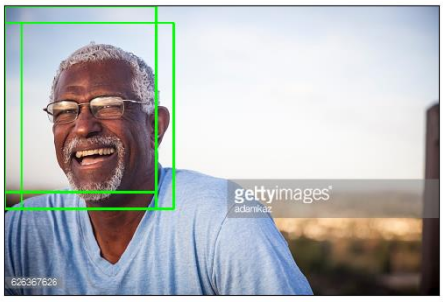
# of Windows reduced from 128 to 1





# of Windows reduced from 128 to 3





# of Windows reduced from 128 to 3

Observations:

There are five cues which author has used to measure objectness. These are:
1. Multiscale Saliency
2. Color Contrast
3. Edge Density
4. Superpixel Straddling
5. Location and Size

Each cue has it's own advantage, dependencies and disadvantages. So, author has used Bayesian of all the cues to get more to accurate results.

I started with multiscale saliency. There is a library called objectness_saliency already implemented in opencv. I used the same to generate saliency map. But the dimensions of feature extracted could not match with the one I got from HoG. So, I kept it aside.

Then I started with color contrast implementation. The color contrast is between the window and it's surrounding. So, I considered the window and the surrounding of 50 pixels from each side of image for color contrast computation. Basically, we had to calculate chi squared distance between LAB histogram of theses two window which will contribute to CC factor of image window.

$$\frac{|\mathrm{Surr}(w,\theta_{CC})|}{|w|} = \theta_{CC}^2 - 1.$$

$$\mathrm{CC}(w,\theta_{CC}) = \chi^2(h(w),\ h(\mathrm{Surr}(w,\theta_{CC}))).$$

After that I had implemented SS cue i.e. superpixel straddling cue for measuring objectness of image window.
Superpixel is the area of same color or texture. SS is highest for windows fitting tightly around the windows. In this paper, they have mentioned a method with which SS cue can be calculated with only four steps.

$$\mathrm{SS}(w,\theta_{SS}) = 1 - \sum_{s\in\mathcal{S}(\theta_{SS})} \frac{\min(|s\setminus w|, |s\bigcap w|)}{|w|},$$

$$|s\setminus w| = |s| - \left|s\bigcap w\right|.$$

Basically, we must compute number of pixels having same value in the image window and outside the window (|s\w|). Using the above formula, SS is calculated.

Instead of using the approach which is used by author for SS calculation, I used my own method to get this parameter. At first, I got histogram of image window as well as the image. So, now I have stats of pixels. Then I have used this data for SS calculation.

Summary:

Implementing just two cues has given such a drastic performance with the computation. So, overall, measuring objectness improves the results of many object-based algorithms such as object detection, object recognition, etc.

Model can be trained to get better and optimised windows for further algorithms. This experiment was done on fixed window size but can be extended to variable window size as well.