Homework 1

Divesh Badod

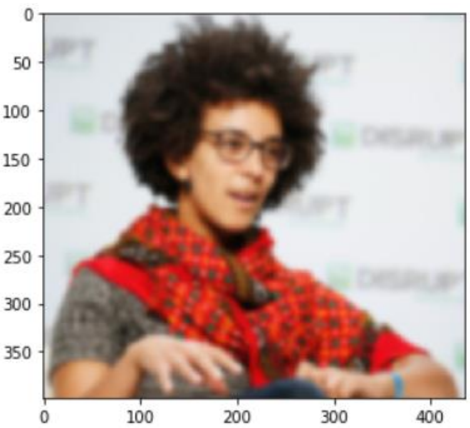
Foundations of CV

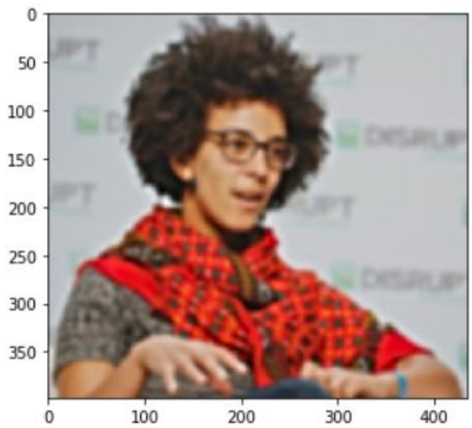
Prof. Ifeoma Nwogu

Problem 1: Image Sharpening:

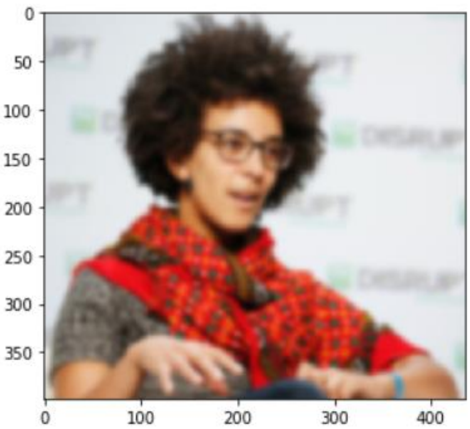
When applying Gaussian filter, the variable which is of most significance is the standard deviation (σ). In the kernel this variable is the units of the interpixel spaces. This technique is used to smoothen the image to reduce noise by enforcing the requirement that pixels should look like their neighbours. Hence the Gaussian kernel forms a weighted average that weights pixels at its centre much more strongly than that at its boundaries. Now the value of σ will determine how much contributions the distant pixels have to the smoothening effects of the kernel. If σ is very small the smoothing effect will have little to no effect because the weights for all the pixels off the centre will be very small. And for a much larger σ, the neighbouring weights will have larger weights in the weighted average, which means the average will be strongly biased towards a consensus of the neighbours. This will be a good estimate of a pixel's value and the noise will largely disappear at the cost of some blurring. Finally, a kernel with large σ will cause much of the image detail to disappear along with the noise. If σ is too large, then the kernel size (k) must be large otherwise we are ignoring the contributions from pixels that should contribute with substantial weight. Below are the examples of the given image with different kernel size (k) and standard deviation (σ)

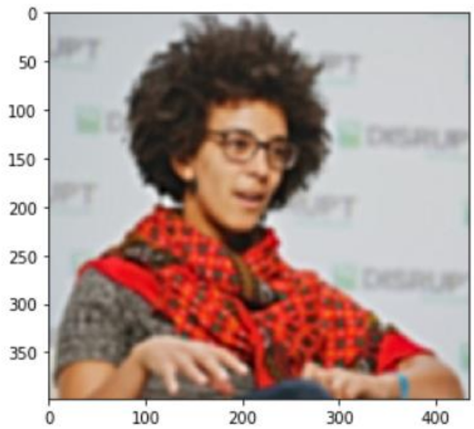
k = 50 σ = 5



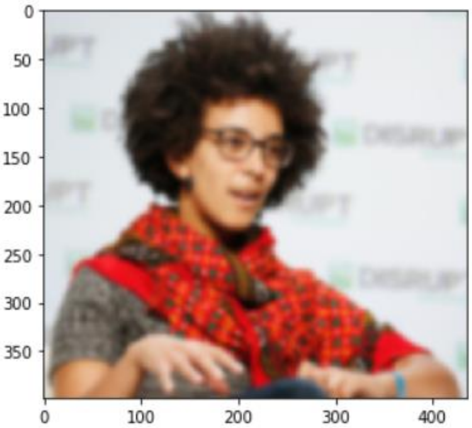


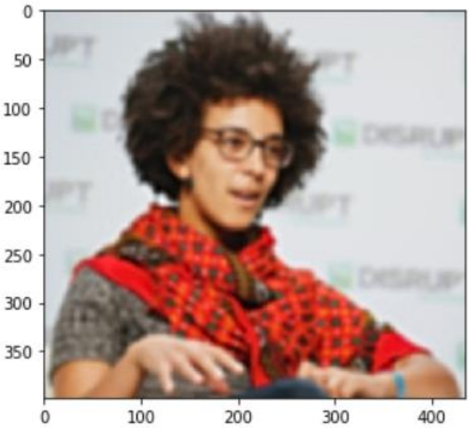
k = 45 σ = 4



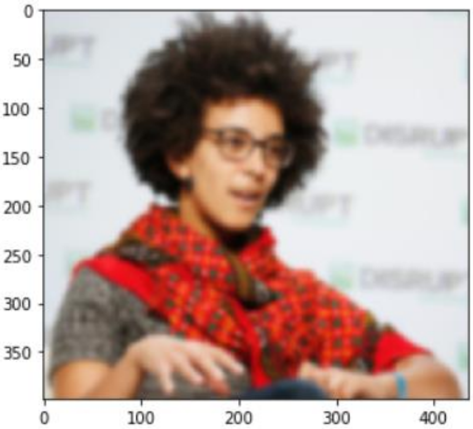


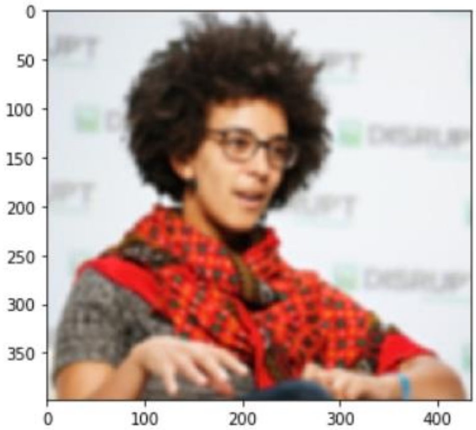
k = 40 σ = 3



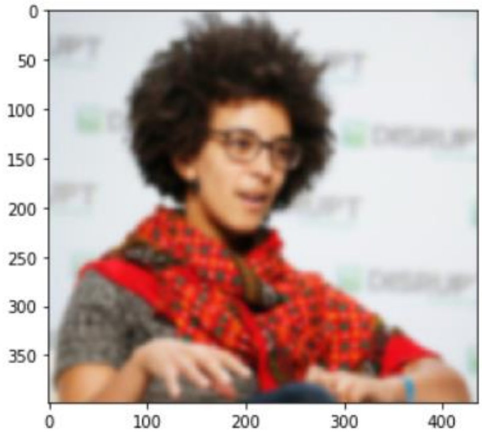


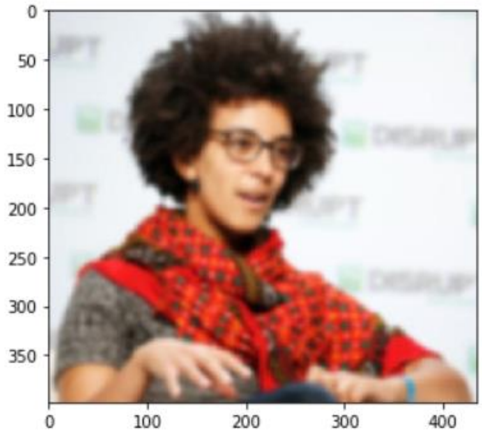
k = 35 σ = 2





k = 30 σ = 1



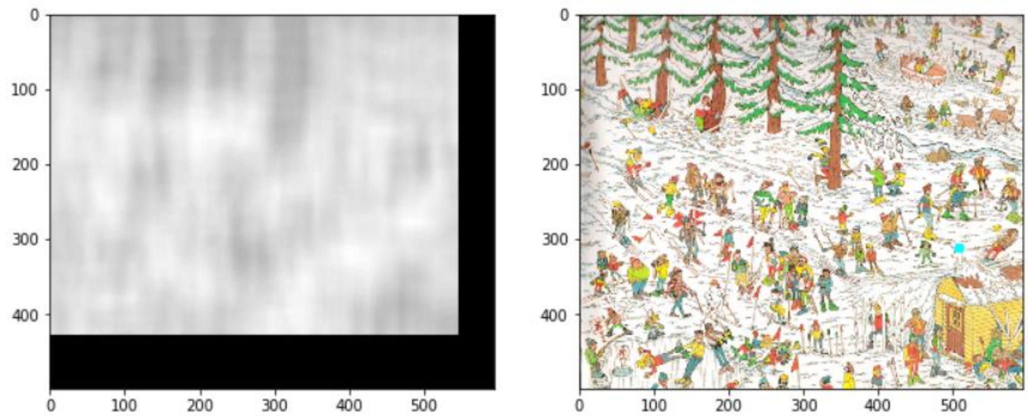


Problem 2: Template Matching – Finding Waldo

Cross-Correlation:

Cross-Correlation is a process to compare a filter with a patch of image centred at the point whose response we look at. In this, view the image neighbourhood corresponding to the filter kernel is scanned into a vector that is compared with the filter kernel. Hence by itself the dot product is a poor way to find features because the value might be large simply because of brightness. Results are based on the brightness of underlying image. Hence in this process finding waldo template doesn't match the correct location of the image because we are looking at the maximum intensity point i.e. the brightest point. The small cyan circle is the output of the matched coordinates using cross correlation in this example

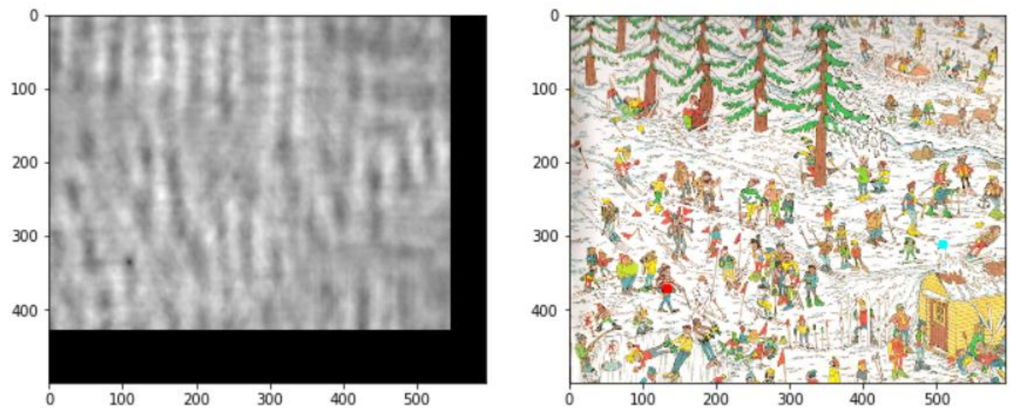
```
Matched Co-ordinates: 311.0 507.5
```



SSD:

SSD is one of the measures of match that is based on pixel by pixel intensity differences between two images. It calculates the summation of squared for the product of pixels subtraction between two images. With this similarity measure, matching point can be determined by considering the location of minimum value in image matrices. The red circle is the output of the matched coordinates using the SSD method.

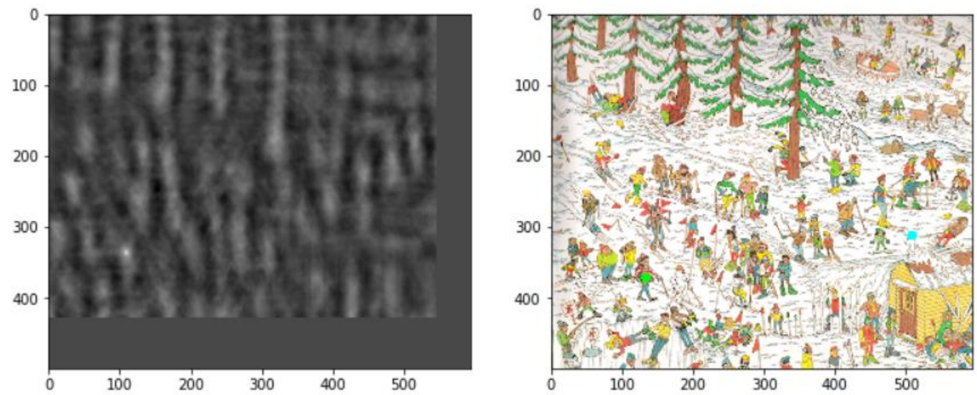
```
371.0 132.5
```



Normalized Cross-Correlation(Handling Intensity Changes):

When a scene is captured by different sensors or under different illumination intensities, both the SSD and cross correlation can be large for windows representing the same area in the scene. A solution is to normalize the pixels in the windows before comparing them by subtracting the mean of the patch intensities and dividing by the standard deviation. The score values range from 1(Perfect match) to -1(completely unrelated). The green circle is the output coordinates after using the Normalized correlation technique.

```
371.0 132.5
```



Problem 3: Canny Edge Detection

Primarily the algorithm consists of 5 steps:

1. Noise Reduction:

Done by applying Gaussian filter with small kernel.

2. Gradient Calculation:

Edges correspond to a change of pixels' intensity. The easiest way to detect is to apply filters that highlight this intensity change in x and y direction. This is done by calculating x and y derivatives, the derivatives are called edge detection operators and the results of these operators is the gradient of the image.

3. Non-Maximum Suppression:

Ideal image should have thin edges and in the previous step the image created will have both thin and thick edges. Non-Maximum suppression is used to thin out the edges. The simple principle is to go through all the points in the matrix and find the pixels with the maximum value in the edge direction.

4. Double Threshold:

This step is used to identify 3 kinds of pixel

- Strong: Pixels with high intensity which surely contributes to the final edge.
- Weak: Pixels with intensity which are not considered as strong edge and neither as non-relevant pixels.
- Non-relevant: Pixels with intensity which cannot be considered relevant for edges.

High threshold is used to identify strong pixels and low threshold is used to identify non-relevant pixels and pixels with intensity within the range of high and low threshold are flagged as weak and the next step will help us in identifying strong and non-relevant pixels among the weaker pixels.

5. Edge Tracking by Hysteresis:

This process transforms weak pixels into strong ones if and only if, at least one of the pixels around the pixel being processed is a strong one.