Computer Vision **(25 /03/22)**

Mid-Semester Exam - Solutions

Problem 1

We have an image I, with three channels I_1, I_2, I_3 as follows:

$$I_1 = \begin{array}{|c|c|c|}\hline 1 & 2 & 5 \\ \hline 3 & 6 & 7 \\ \hline 9 & 8 & 1 \\ \hline\end{array} \qquad I_2 = \begin{array}{|c|c|c|}\hline 2 & 3 & 5 \\ \hline 7 & 1 & 9 \\ \hline 8 & 2 & 3 \\ \hline\end{array} \qquad I_3 = \begin{array}{|c|c|c|}\hline 6 & 7 & 3 \\ \hline 2 & 3 & 2 \\ \hline 1 & 1 & 5 \\ \hline\end{array}$$

and similarly we have the filter K (same as I)

$$K_1 = \begin{array}{|c|c|c|}\hline 1 & 2 & 5 \\ \hline 3 & 6 & 7 \\ \hline 9 & 8 & 1 \\ \hline\end{array} \qquad K_2 = \begin{array}{|c|c|c|}\hline 2 & 3 & 5 \\ \hline 7 & 1 & 9 \\ \hline 8 & 2 & 3 \\ \hline\end{array} \qquad K_3 = \begin{array}{|c|c|c|}\hline 6 & 7 & 3 \\ \hline 2 & 3 & 2 \\ \hline 1 & 1 & 5 \\ \hline\end{array}$$

The convolution output (O) for would be a vector given by

$$O = [I_1 \cdot K_2 + \ I_2 \cdot K_1 + \ I_3 \cdot K_3]$$

where $\cdot$ represents dot product. On performing the required calculations we get,
$O = [654]$

Problem 2

Given Layer, say L_1 as follows

12	23	34	45
56	67	78	89
90	98	87	76
65	54	43	32

There could be many ways to reduce spatial dimensions in a CNN. Some of them are mentioned below.

1. **Convolution without Padding (zero padding)**

   If we take our filter, say K with dimensions smaller than of the layer, we can reduce the spatial dimensions. For L_1, we can pick a filter K with dimensions (3×3). Let K be defined as follows:

1	1	1
1	1	1
1	1	1

K

   Applying this filter, without padding we get a $[2 \times 2]$ matrix as follows,

545	597
638	624

2. **Max Pooling**

   We can use Max-Pooling. Assuming a (2×2) window, on applying max pooling we get,

67	89
98	87

3. **Average Pooling**

   We can use Avg-Pooling. Assuming a (2×2) window, on applying avg. pooling we get,

39.50	61.50
76.75	59.50

Problem 3

The bandwidth parameter in the Mean Shift Algorithm defines the radius from the center, of the region that is going to be considered for calculating the mean. All the points that lie inside that region are used to calculate the mean for the algorithm.

Given,

Centre: $V9$ (V for Venue), Position: $(72, 173)$, Position is taken as (h5-index, h5-median)

Using Manhattan distance we get,

Venue	Dist	Venue	Dist	Venue	Dist
V1	694	V8	28	V16	134
V2	294	V10	77	V17	112
V3	250	V11	86	V18	124
V4	179	V12	62	V19	133
V5	65	V13	96	V20	137
V6	59	V14	105		
V7	36	V15	102		

As we see no distance is less the 20, the bandwidth parameter, no other point is considered for the mean calculation and the mean stays the same. So the mean after one iteration is $(72, 173)$ itself.

Problem 4

Given

Size of Image $= 200 \times 800$

Number of super-pixels $= 400$

Let edge length of a Super Pixel be e, then the gap between any two super-pixel centers will also be e. (Assuming the super-pixels are square shaped). Then the area of 1 super-pixel is given by $= e^2$.

As we know that we would have 400 super-pixels in total, we can say that,

$$400 = \frac{Total\ Area}{Area\ of\ 1\ super - pixel}$$

Plugging the values we get

$$e = \sqrt{\frac{200 \times 800}{400}}$$

i.e.

$$e = 20$$

So we need to have a gap of 20 between our super-pixel centers.

Problem 5

Algorithm 1 Pseudo-Code for Q5

Input: r: radius, (x, y): center, (h, w): dimensions of the mask
Output: A Binary mask, M.
Initialise: $M[h \times w] \leftarrow 0$
for $i = 1.....h$ **do**
  for $j = 1.....w$ **do**
    $d \leftarrow$ eucl_dist$((x, y), (i, j))$
    if $d \leq r$ **then**
      $M[i, j] \leftarrow 1$
    end if
  end for
end for
return M

Here eucl_dist is a function that takes in two points say (a, b) and (c, d) and returns the euclidean distance between them. The euclidean distance is given by:

$$d_{euclidean} = \sqrt{(a-b)^2 + (c-d)^2}$$