

**Assignment 3:**

**Due date: 28th March 2019**

**Total Marks: 100**

---

**Problem 1:** An image patch  $P$  and 2D Gaussian filter  $F$  are given as bellow:

$$P = \begin{bmatrix} 142 & 155 & 238 & 246 & 242 \\ 143 & 165 & 244 & 246 & 242 \\ 164 & 177 & 243 & 243 & 239 \\ 185 & 173 & 238 & 246 & 237 \\ 175 & 183 & 245 & 243 & 239 \end{bmatrix} \text{ and } F = \begin{bmatrix} 0.0751 & 0.1238 & 0.0751 \\ 0.1238 & 0.2042 & 0.1238 \\ 0.0751 & 0.1238 & 0.0751 \end{bmatrix}$$

(a) Compute the output of convolutional layer with input  $P$  and filter  $F$  (10 marks) with zero padding and stride = 2 (10 marks).

(b) Compute the output of transposed-convolutional layer with input  $P$  and filter  $F$  (10 marks) with zero padding and stride = 2 (15 marks).

**Problem 2:** Use deepNetworkDesigner of Matlab to design a simple semantic CNN with two paths as:

1. Input image size: [90 90 3] (30 points).
2. Two 3 convolutional layers each of them with 48 channels, stride =2, and same (half) padding.
3. First convolutional layer must be followed by a 3 max pooling layer with stride =2.
4. The designed CNN has to classify two objects from the background.

**Problem 3:** A robot randomly walks (i.e., moves only one block) either to right or up directions in the environment shown in Figure 1 and stops moving at states (3,4) and (4,4) when it is rewarded.

(a) Draw the Markov model for this problem (10 marks: marks for the diagram and 5 marks for the transition matrix).

(b) Find all possible paths starting from state (1,1) and then compute the utility for each state using LMS algorithm (20 marks: 10 marks for correct set of routs and 10 points for the computed utilities).

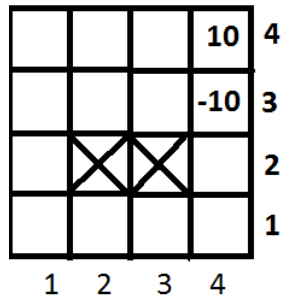


Figure 1: The environment for the robot in problem 3.

(c) Assume the same robot can also jump i.e., moves two blocks either to right or up directions if it is still inside the environment. Draw the new Markov model for this active learning problem and show the transition matrices (15 marks: 5 marks for the diagram, and 5 marks for each transition matrix).