CGS786 Assignment 3

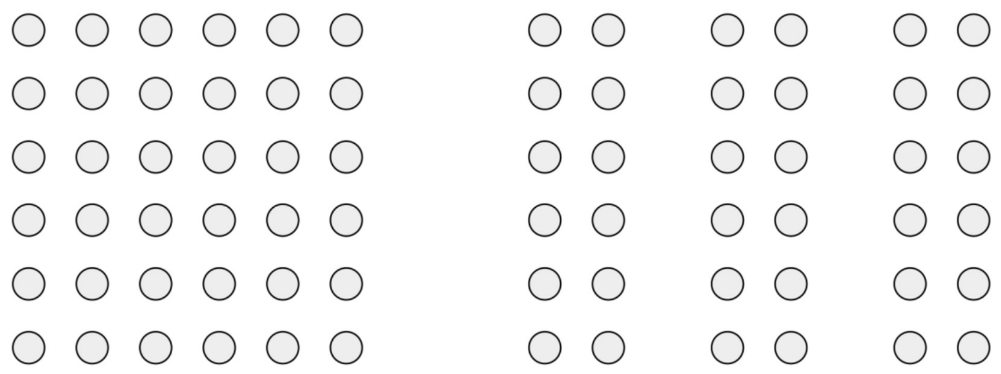
Part 1: Gestalt Principles and CNNs (70 points)

Throughout the first half of this assignment, 'paper' means the Yan & Zhou (2017) paper enclosed with this assignment as a pdf. The code and datasets used in the paper are available here.

Q1(a). The paper uses inception v3 networks, but those are outdated. Can you reproduce the Study Design steps in the paper for inception v3 on inception v4 networks? [20 points]

Q1(b). The paper studies whether CNNs can learn the concept of symmetry. Reproduce their results for global symmetry on synthetic data; you're free to use their data and their code. [20 points]

Q1(c). We discussed in class that there are a number of other gestalt principles beyond symmetry. Here is a fairly sensible list of five such principles. Test for one such principle – the principle of proximity. You basically need to figure out how to generate macro-objects that are made up of multiple micro-objects and find a way for the CNN to classify the macro-objects without being confused by the presence of the micro-objects. *For this question, I will grade on effort, not results.*



For example, a CNN trained to recognize the letter 'I' should interpret the display on the right in the figure above as three 'I's, generalizing across the empty space between the constituent circles (micro-objects) that make up each I (macro-object). I would design training sets that densely pack micro-objects to make macro-objects that are given class labels. I would also ensure that the micro-objects don't look anything like the macro-objects. Once the model is successfully detecting the macro-objects for this training set, I would take this trained model and retrain on training data with relatively looser packed micro-objects and repeat the whole process for 3-4 rounds, increasing the gap between the micro-objects in each round until you get human-like proximity generalization behavior. [30 points]

Part 2: RL basics

Q2. **(30 points)** Without using any third-party libraries or toolboxes, write code to

(a) generate a random instance of the frozen lake scenario given two inputs - the size of the lake (N) assuming its square, and the number of holes (M).

(b) implement a Q-learning agent to find a path through the lake to the goal. Report the results of the learning algorithm by plotting episode count on the x-axis and total reward received by the agent within an episode on the y-axis

(c) identify how learning performance changes when you change the parameters α and λ in your algorithm

(d) identify how learning performance changes with respect to changes in N and M.