

ABSTRACTION AND REASONING CHALLENGE

Using Convolutional Autoencoder to solve abstract problems

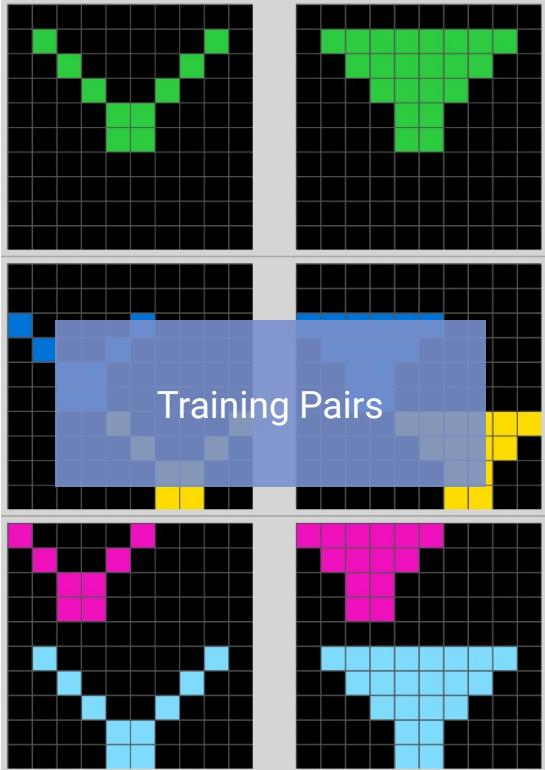
Monique Hübsch

13th July 2021

THE PROBLEM

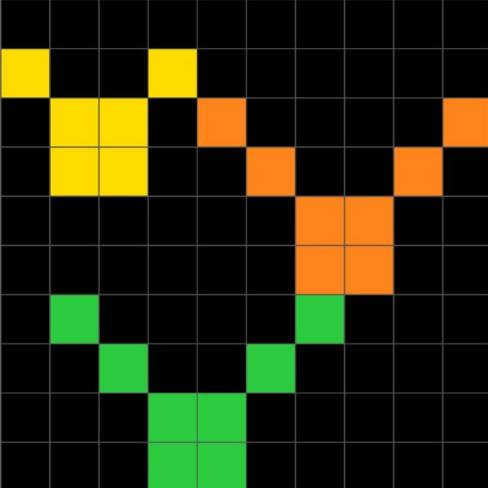
CREATE AN AI CAPABLE OF SOLVING REASONING TASKS IT HAS NEVER
SEEN BEFORE

Task demonstration



Test input grid 1/1

[Next test input](#)



Load task JSON:

[Browse...](#)

[Random...](#)

Task name: 22168020.json

Change grid size:

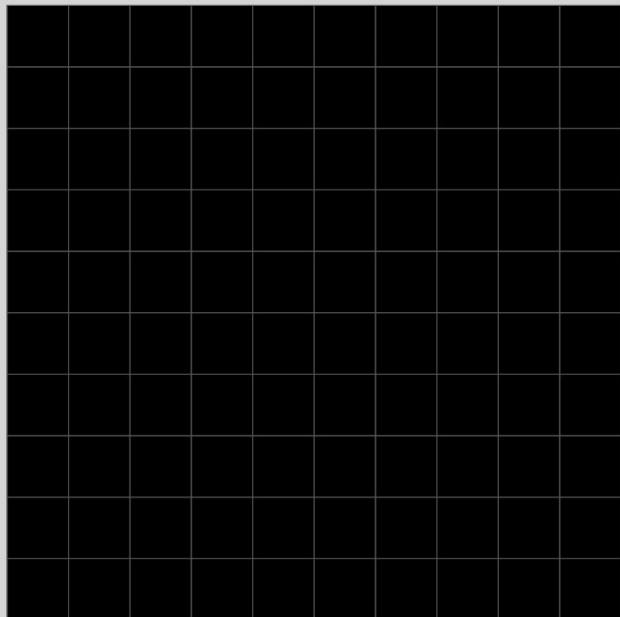
10x10

[Resize](#)

[Copy from input](#)

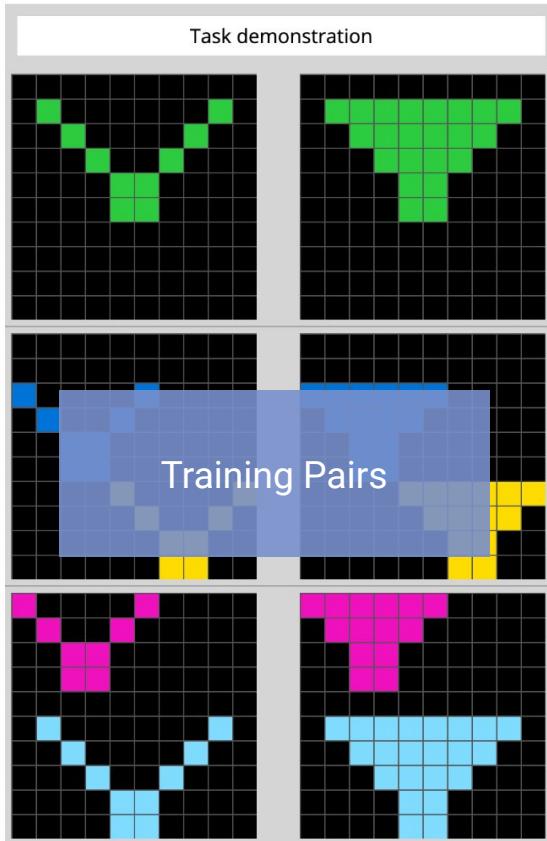
[Reset grid](#)

[Submit!](#)



Edit Select Flood fill





Test input grid 1/1 [Next test input](#)

Load task JSON: [Browse...](#) [Random...](#)
Task name: 22168020.json

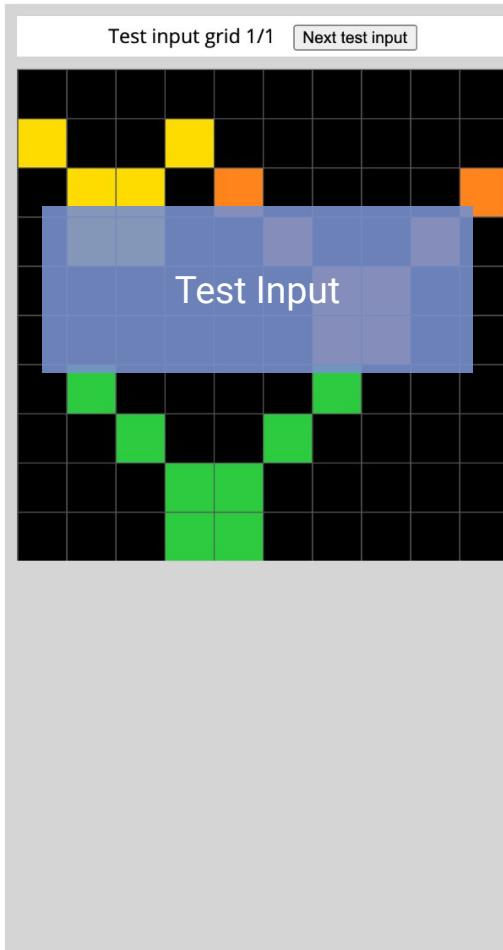
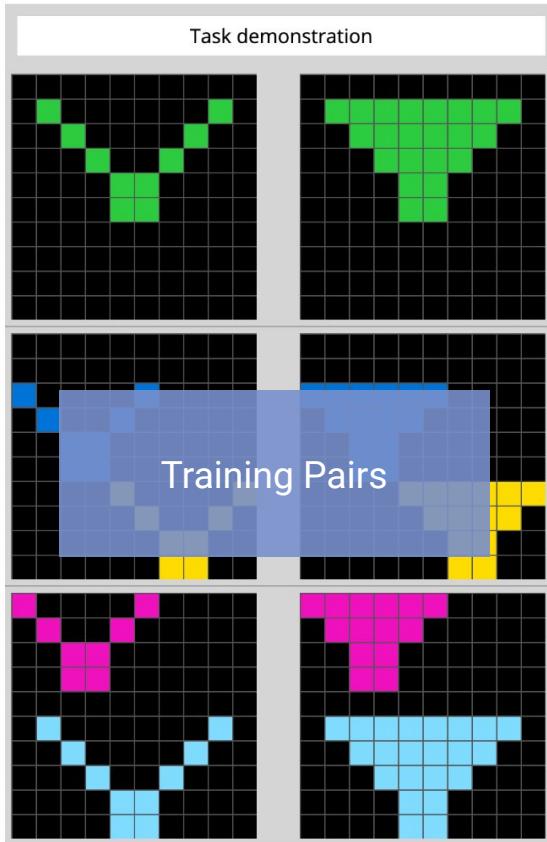
Change grid size: 10x10 [Resize](#)
[Copy from input](#) [Reset grid](#) [Submit!](#)

Test Input

The main area is a 10x10 grid editor. It contains a pattern of yellow and orange pixels at the top left, followed by a large rectangular area filled with a blue-to-white gradient. Below this is a smaller pattern of green and black pixels. The word "Test Input" is centered in the blue gradient area. At the bottom of the grid, there is a row of green and black pixels. The interface includes various controls for loading tasks, changing grid size, and saving changes.

Edit Select Flood fill

A color palette bar at the bottom shows nine colored squares: black, blue, red, green, yellow, grey, magenta, cyan, and dark red. The "Edit" radio button is selected.



Load task JSON: [Browse...](#) [Random...](#)

Task name: 22168020.json

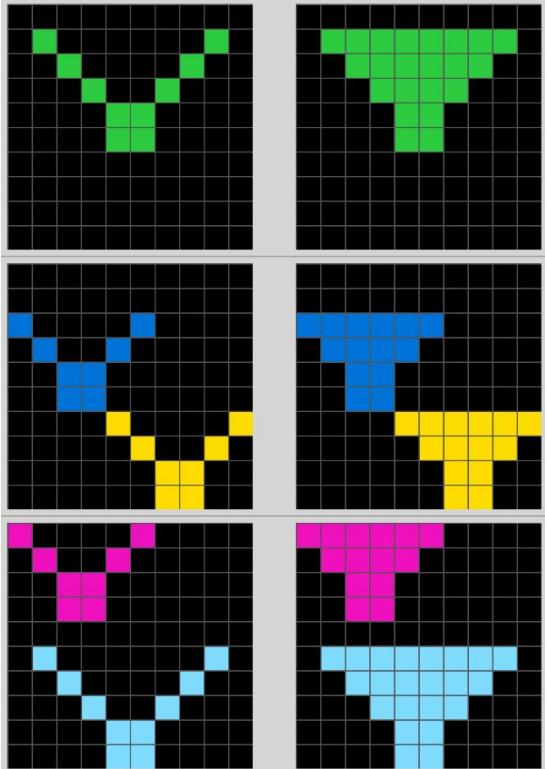
Change grid size: 10x10 [Resize](#)

[Copy from input](#) [Reset grid](#) **Submit!**

Test Output

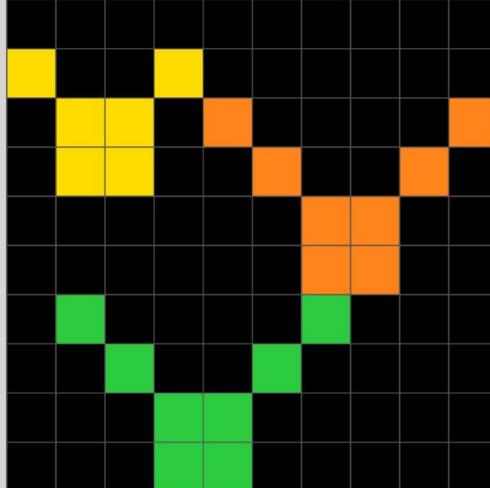
Edit Select Flood fill

Task demonstration



Test input grid 1/1

[Next test input](#)



Load task JSON:

[Browse...](#)

[Random...](#)

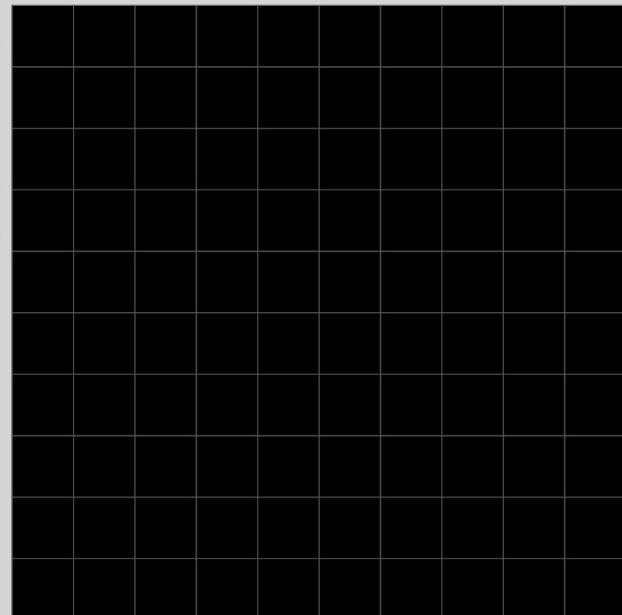
Task name: 22168020.json

Change grid size: 10x10 [Resize](#)

[Copy from input](#)

[Reset grid](#)

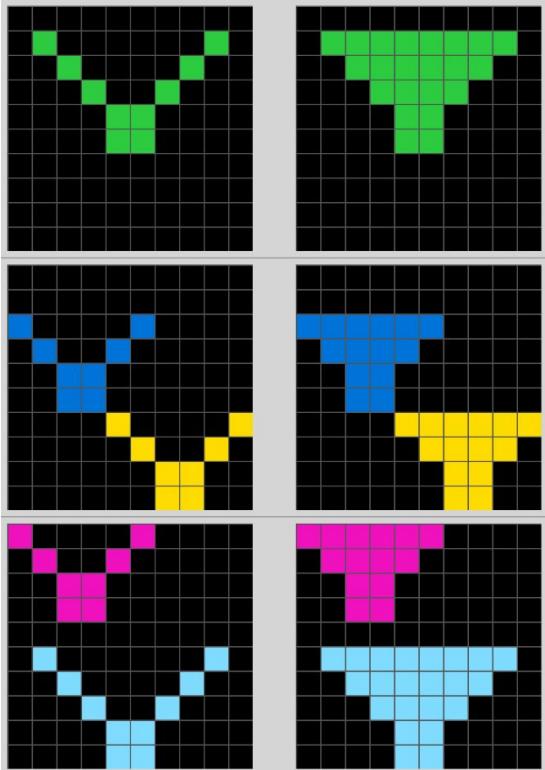
Submit!



Edit Select Flood fill

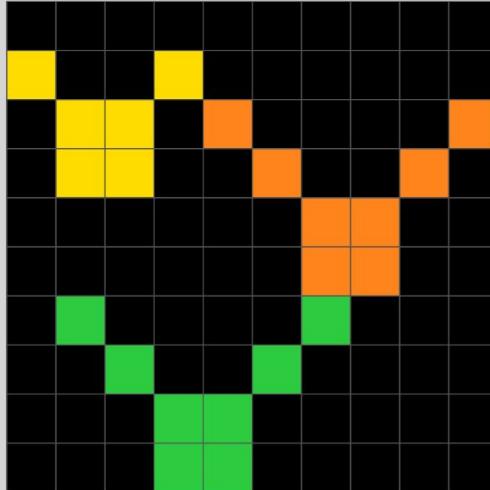


Task demonstration



Test input grid 1/1

[Next test input](#)



Load task JSON:

[Browse...](#)

[Random...](#)

Task name: 22168020.json

Change grid size:

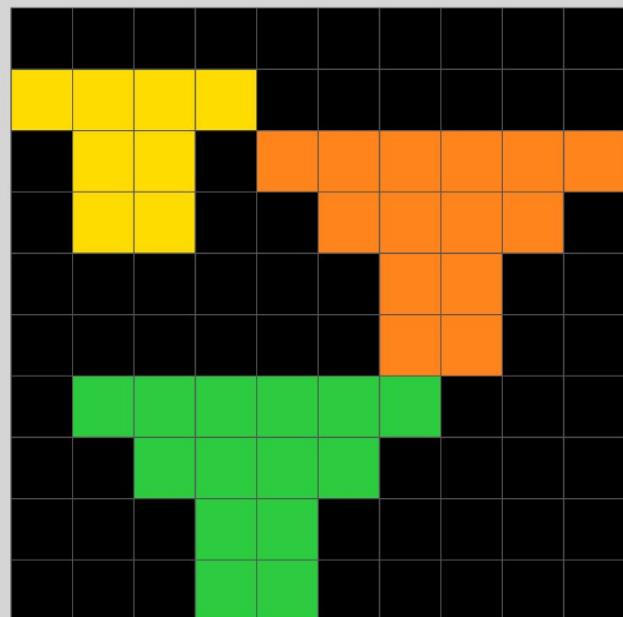
10x10

[Resize](#)

[Copy from input](#)

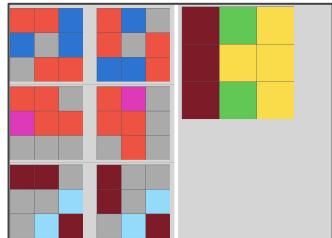
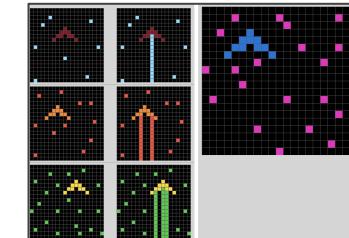
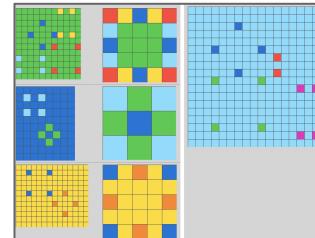
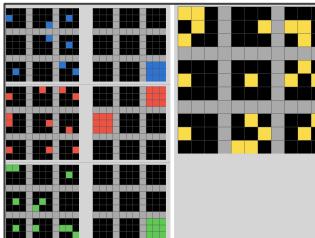
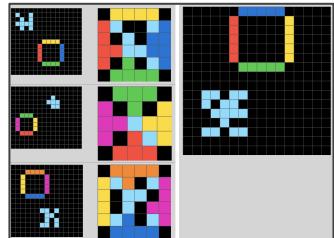
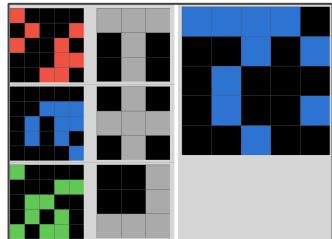
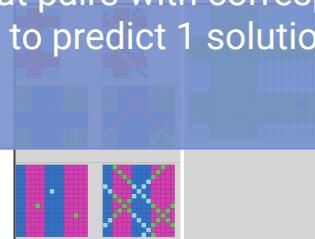
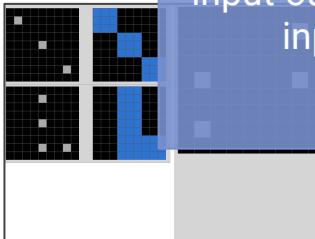
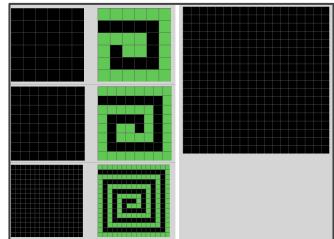
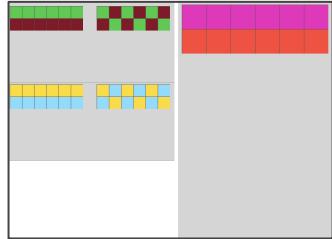
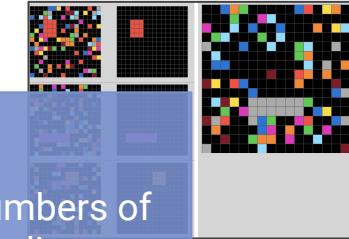
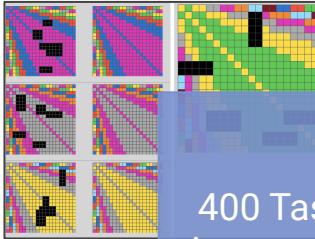
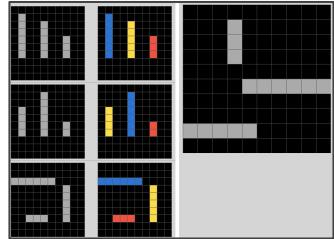
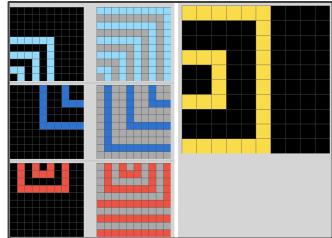
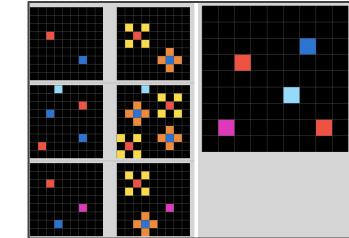
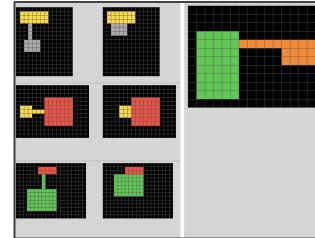
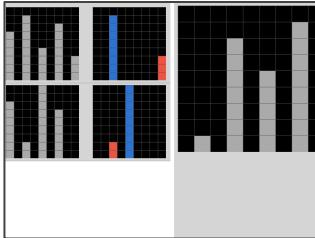
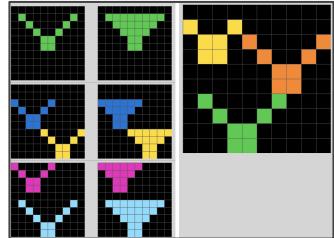
[Reset grid](#)

Submit!



Edit Select Flood fill



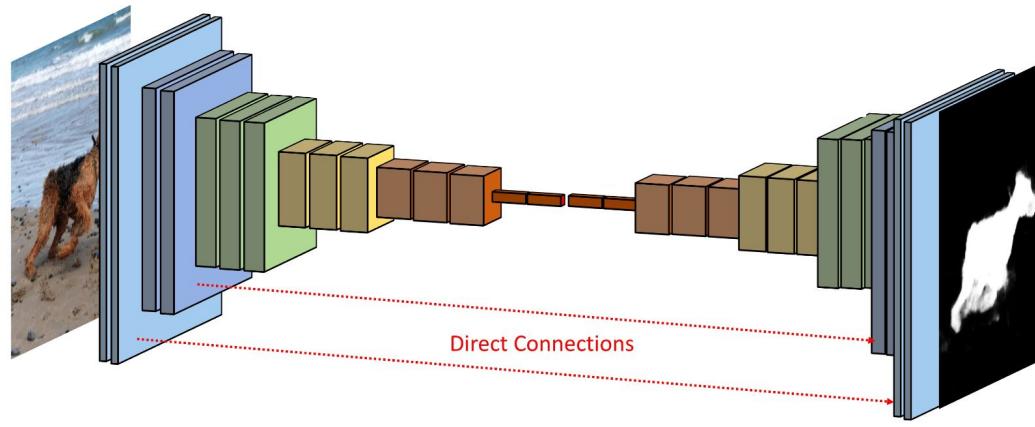


400 Tasks containing varying numbers of
input-output pairs with corresponding test
input to predict 1 solution each

MY APPROACH

CONVOLUTIONAL AUTOENCODER WITH DATA AUGMENTATION

Convolutional Autoencoder

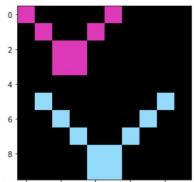
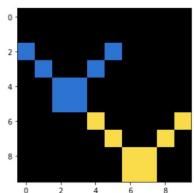
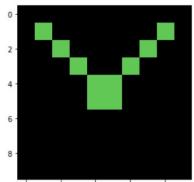


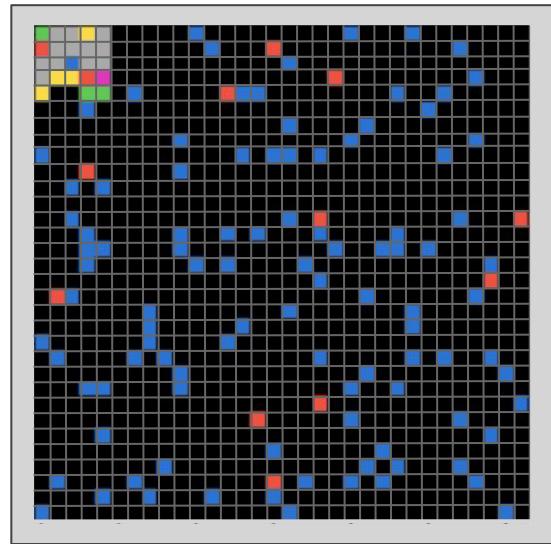
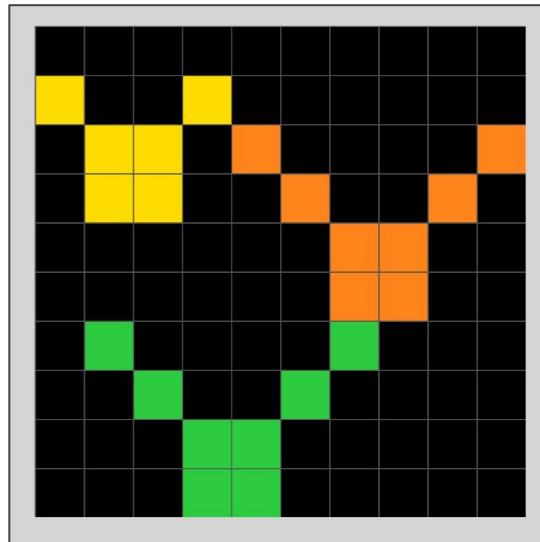
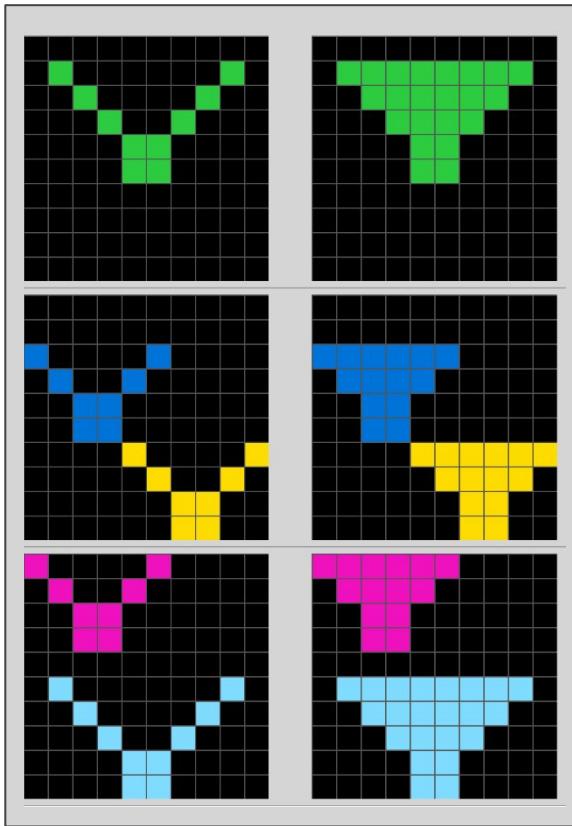
Convolutional Part (Encoder)

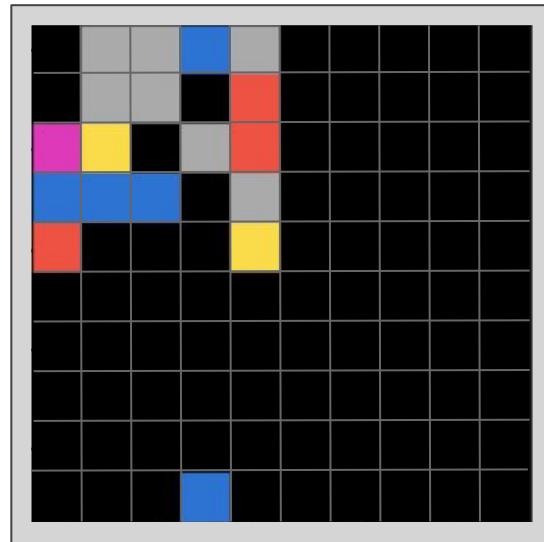
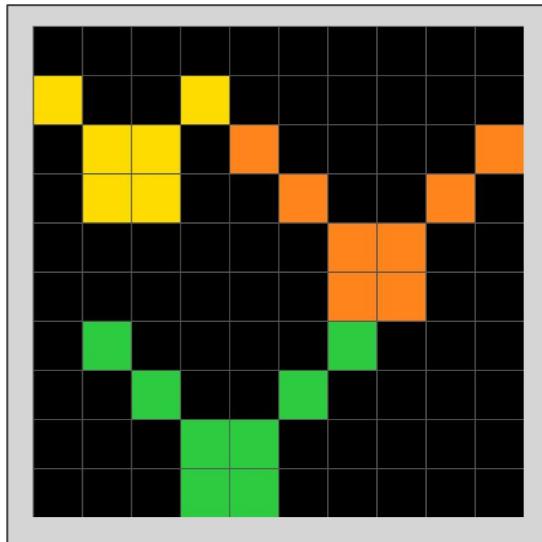
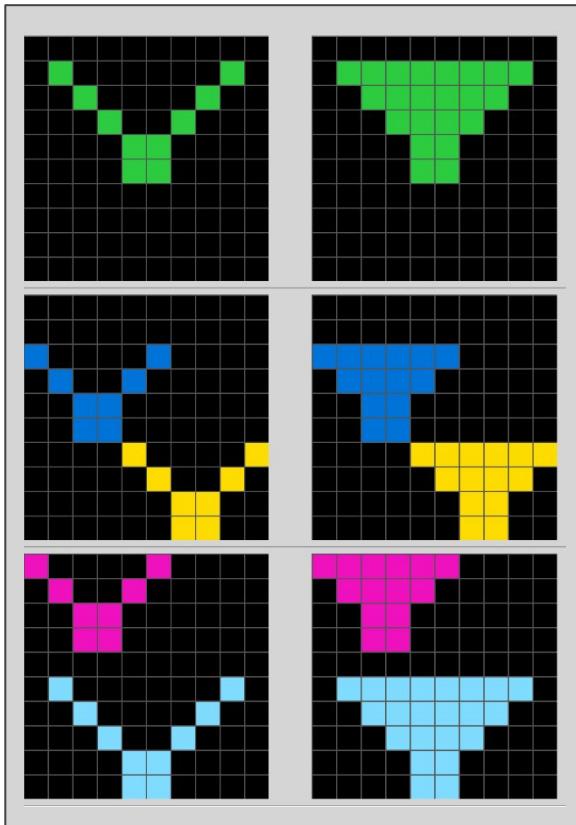
Deconvolutional Part (decoder)

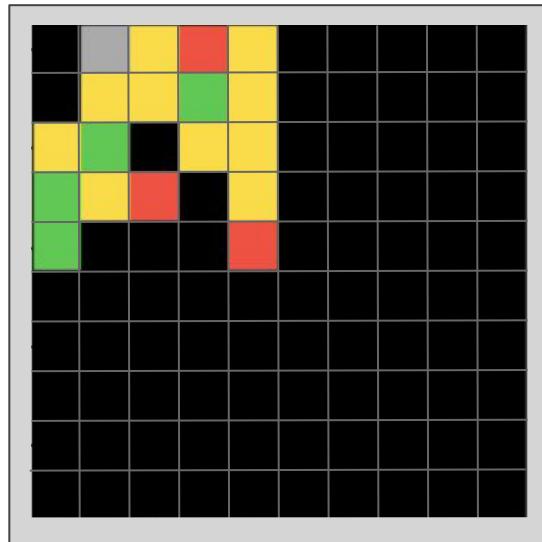
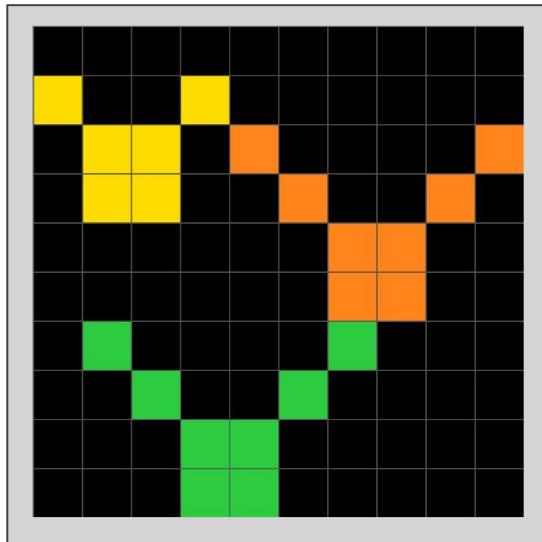
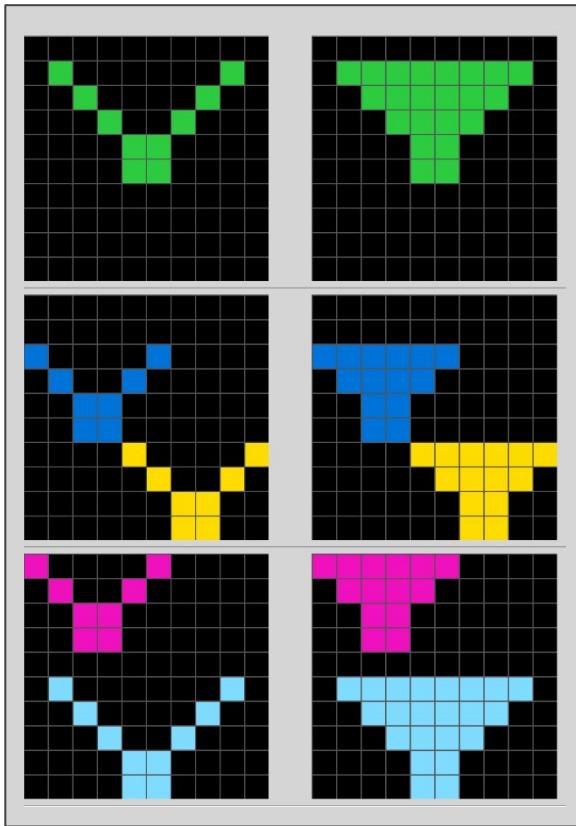
Data Augmentation

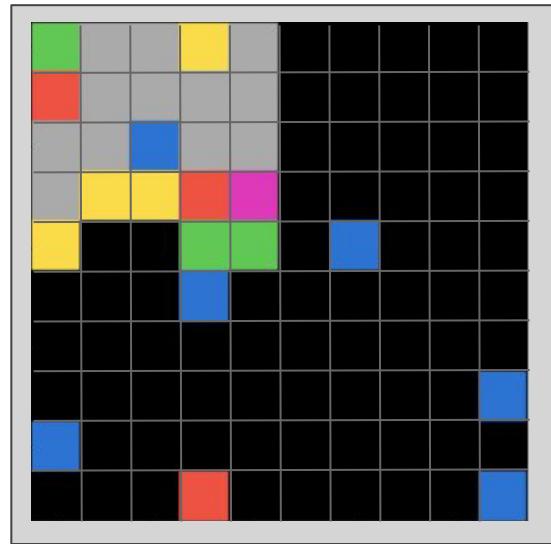
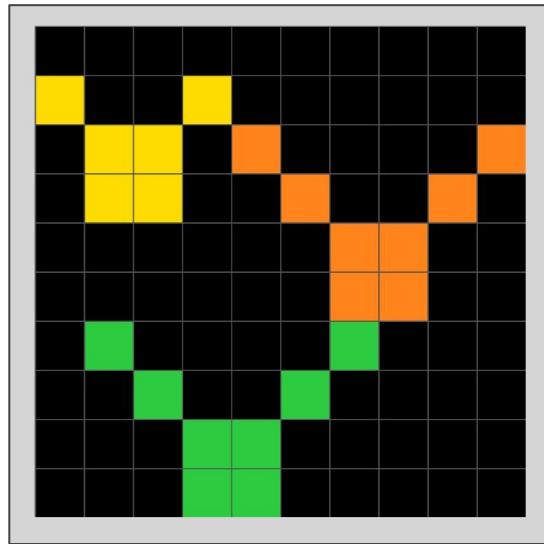
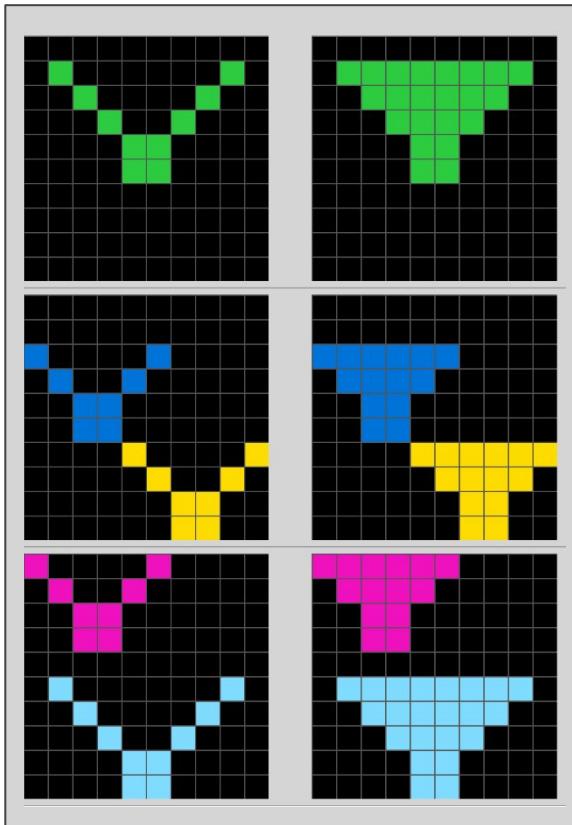
- Each task presents with varying input and output dimensions
- They come with varying numbers of training input-output pairs; most of the time it's one test input
- Each Task itself doesn't provide a whole lot of data for the algorithm to learn from

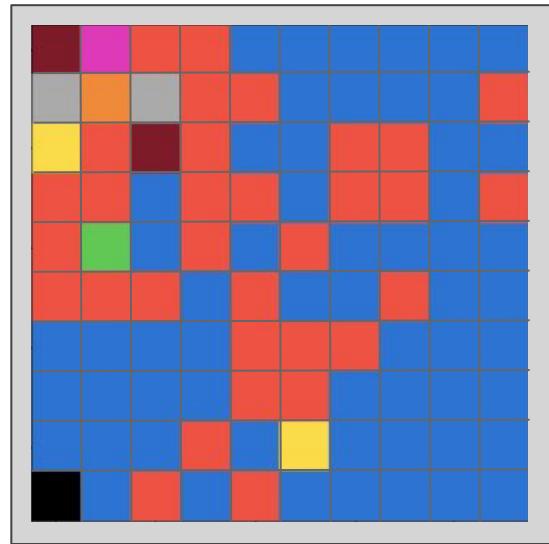
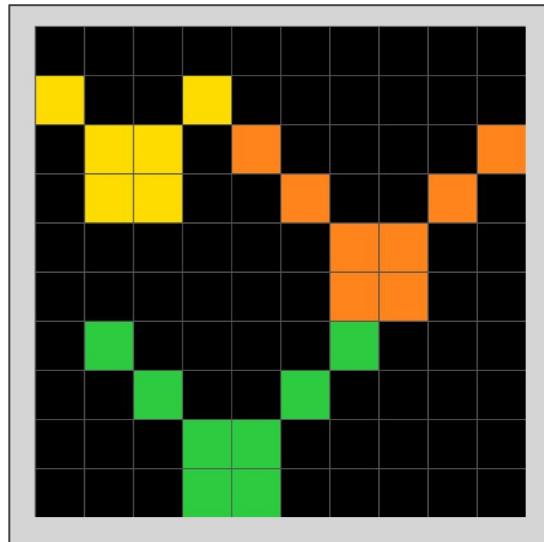
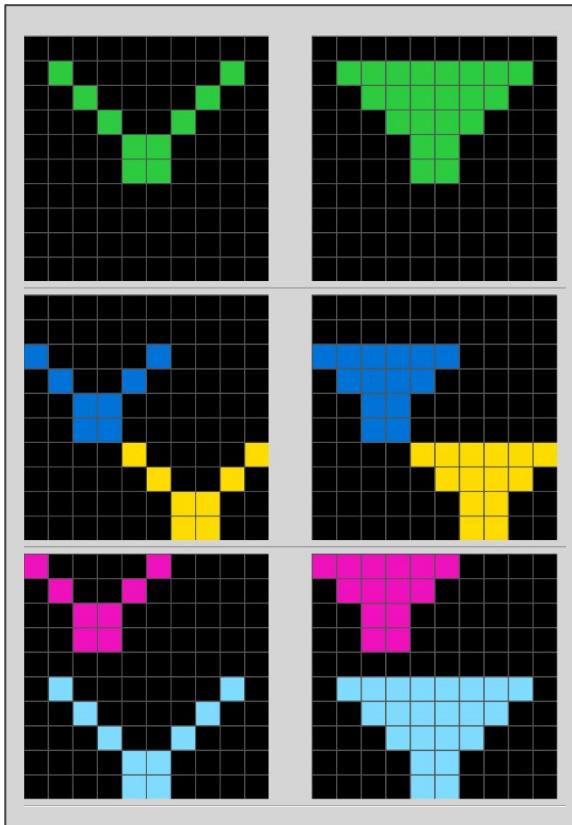


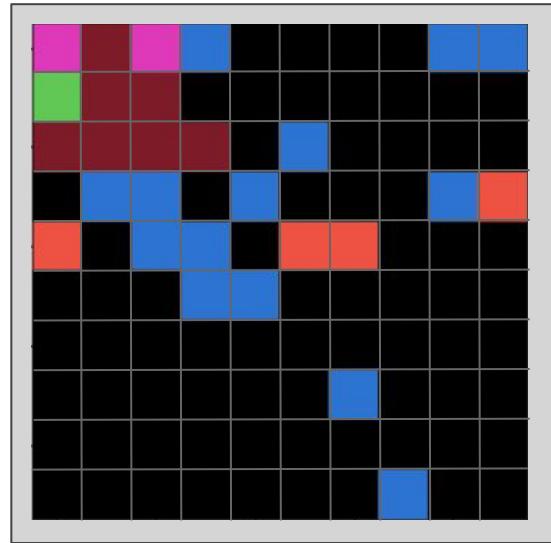
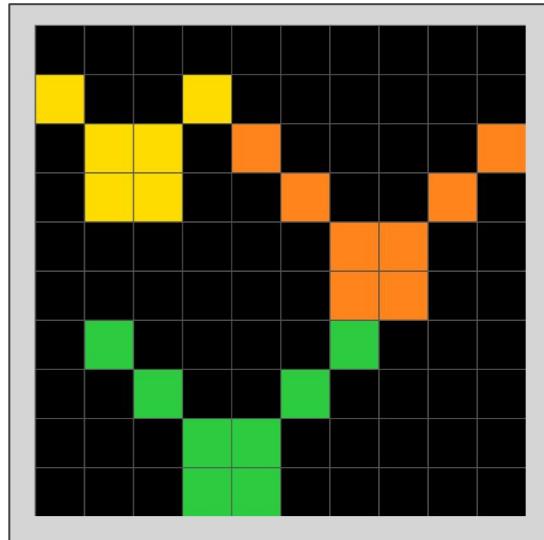
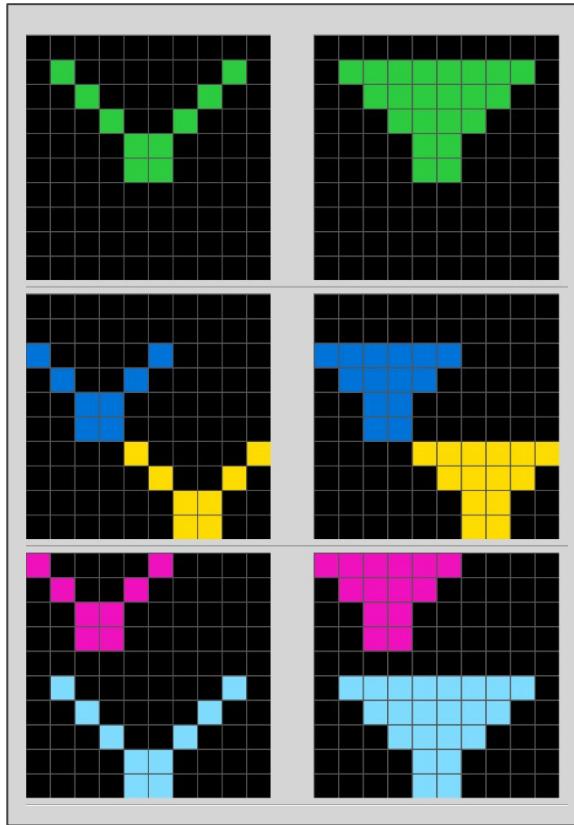


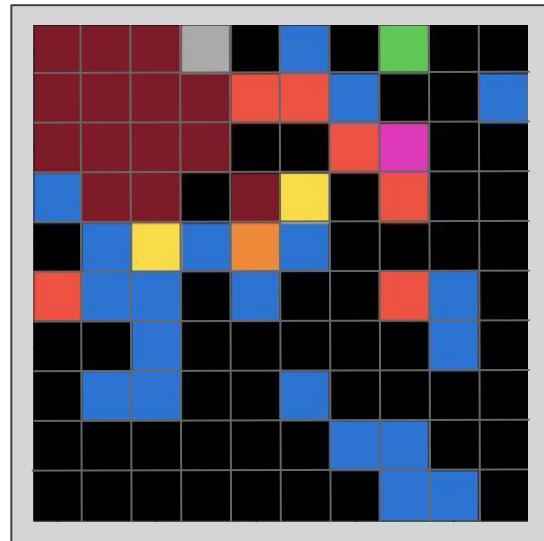
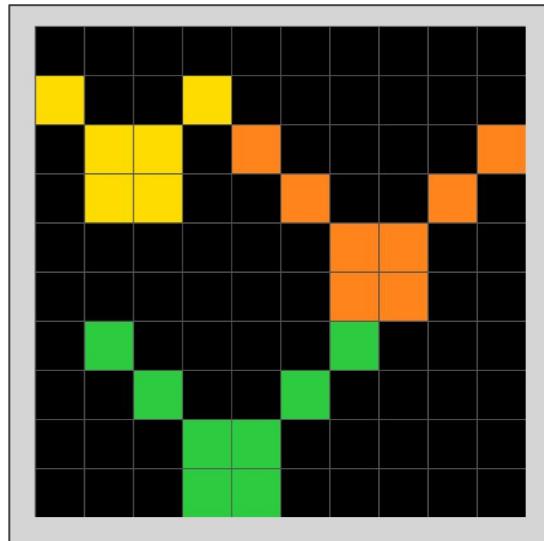
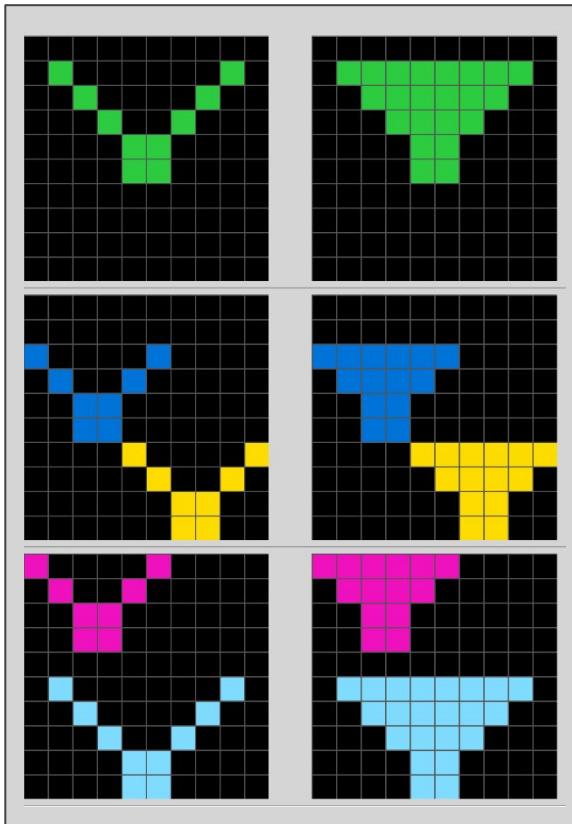


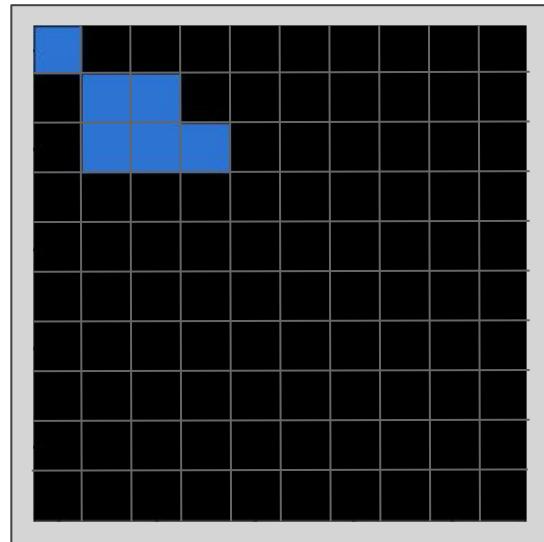
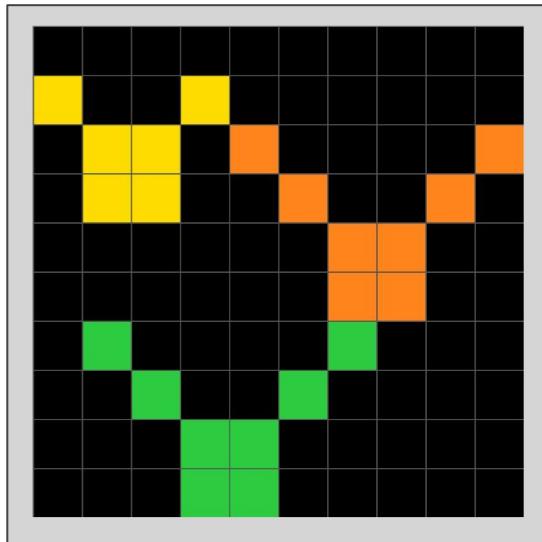
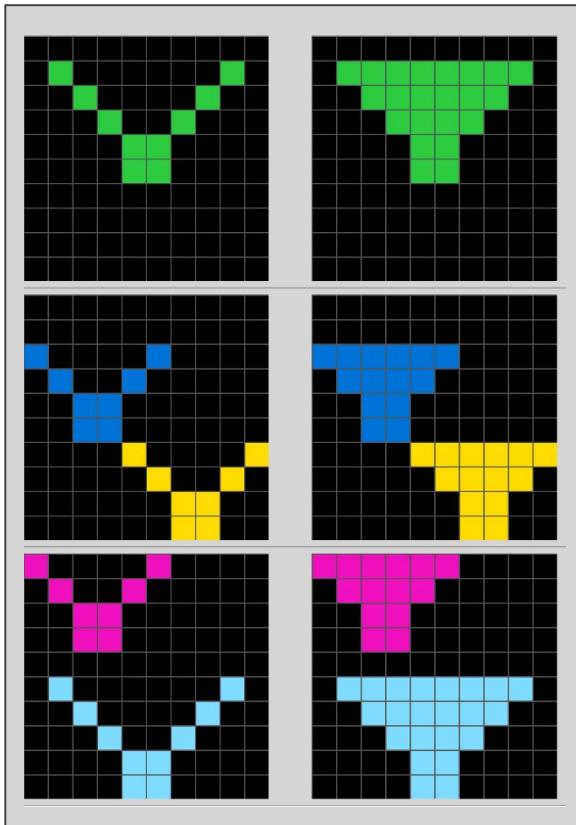


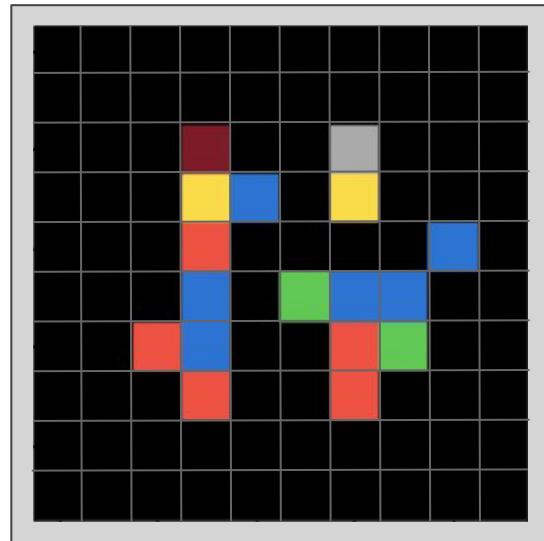
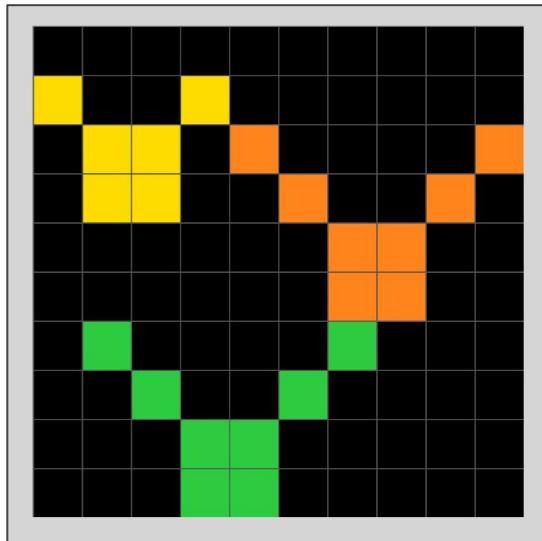
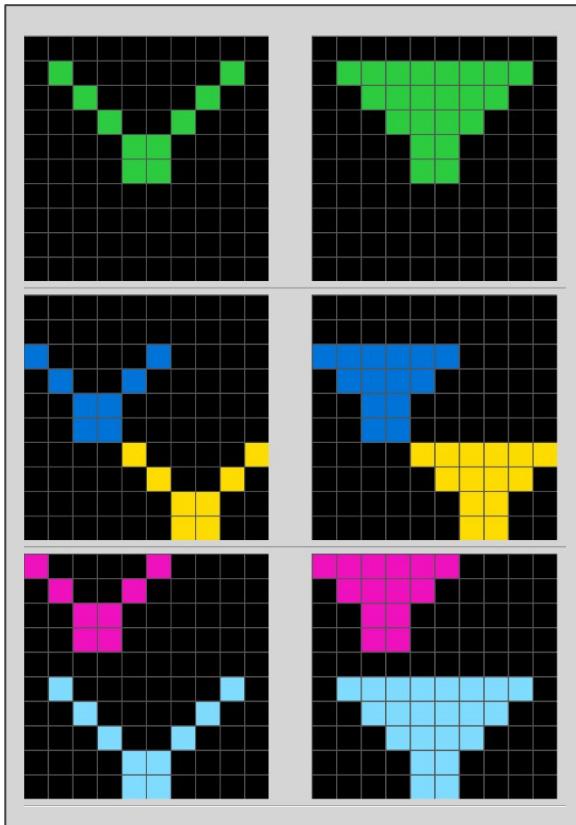


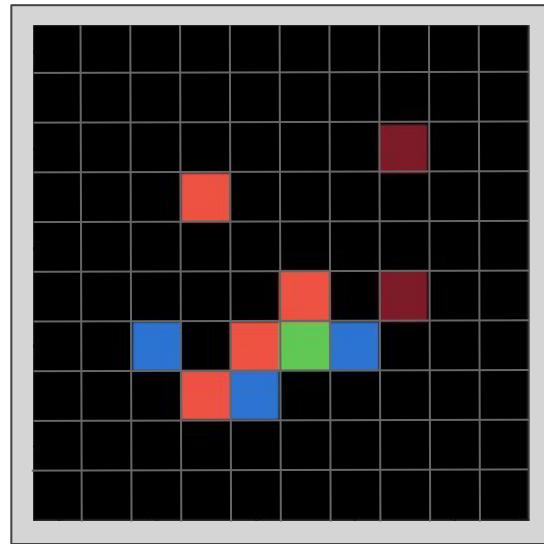
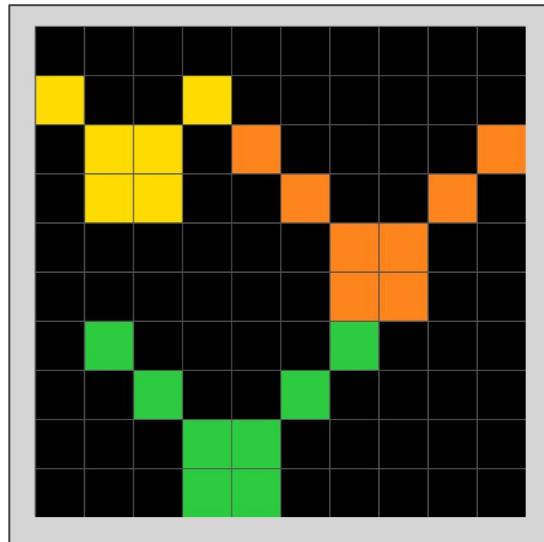
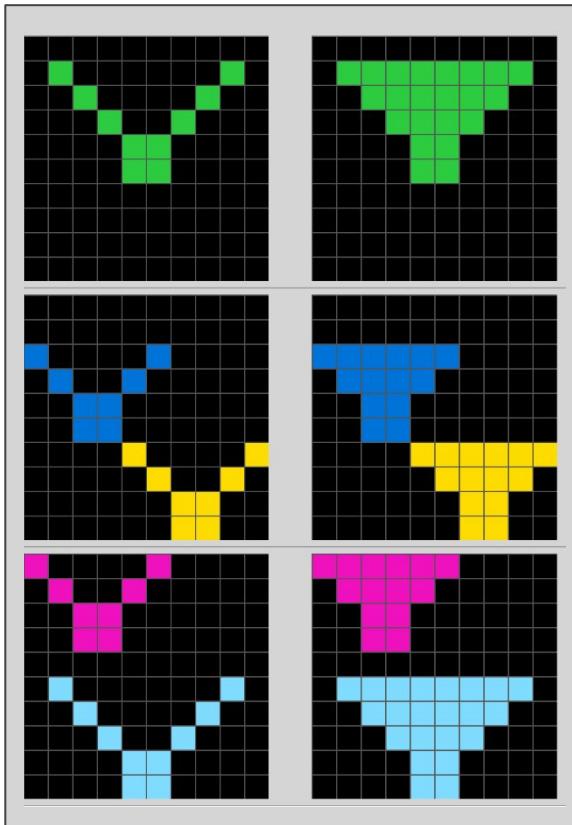


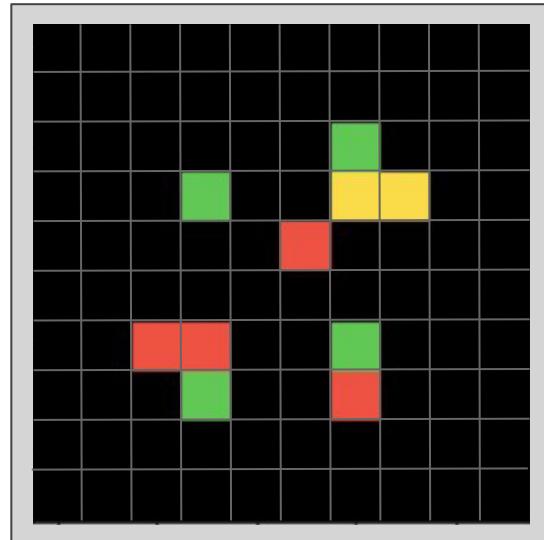
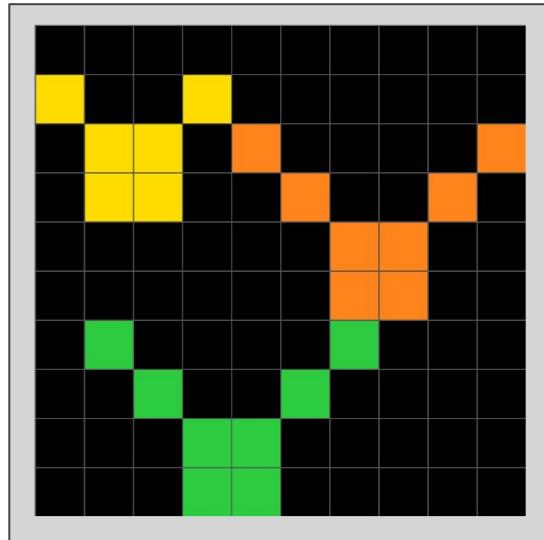
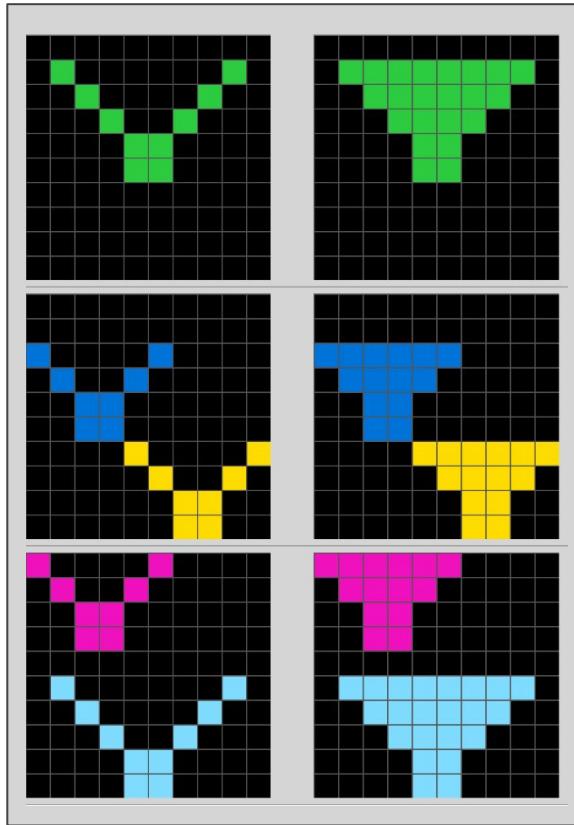


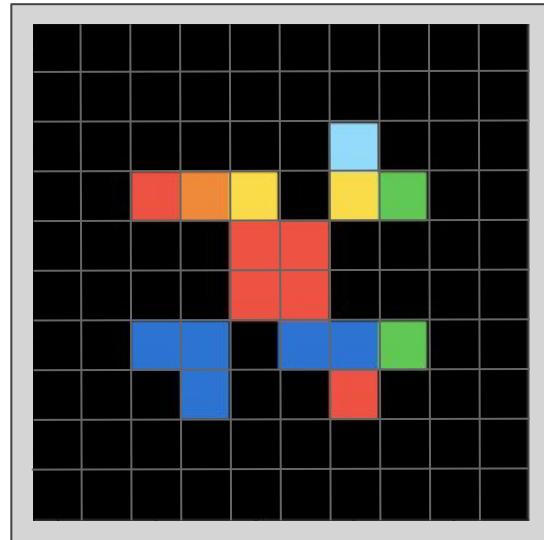
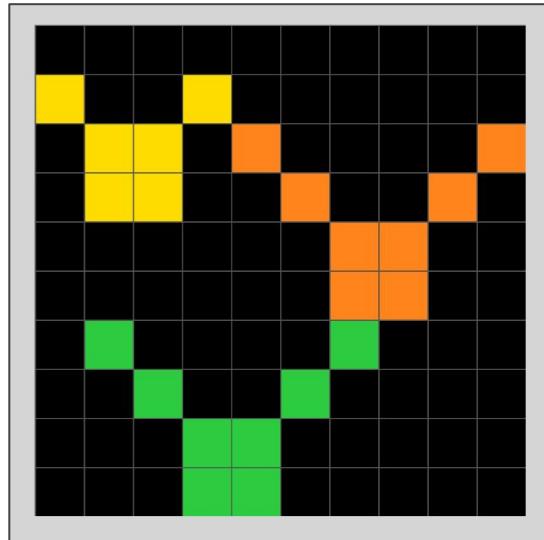
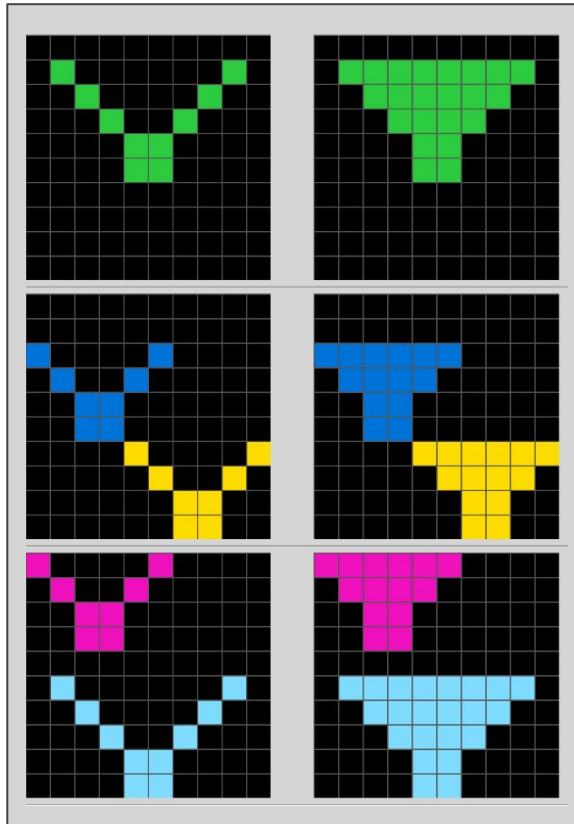










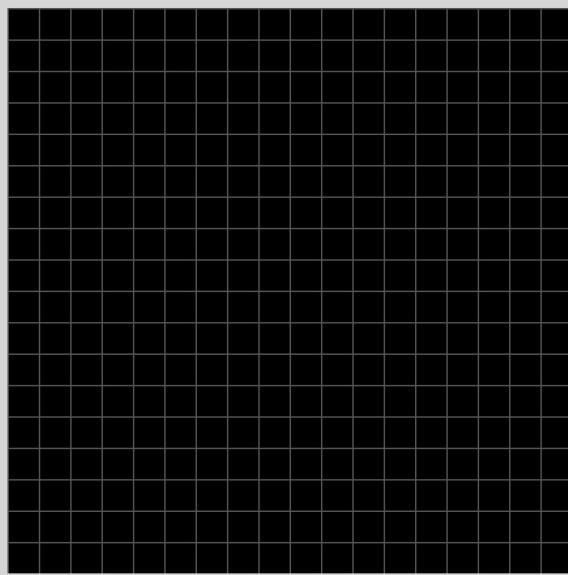
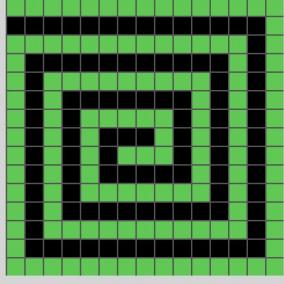
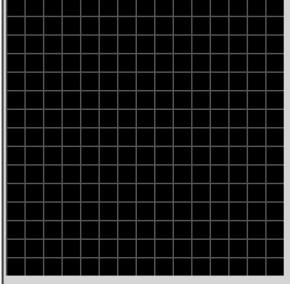
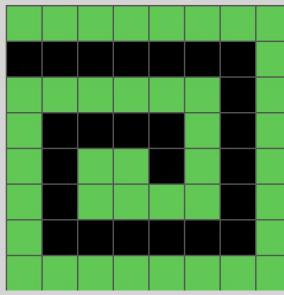
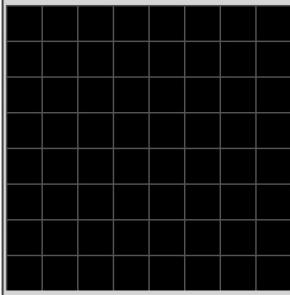
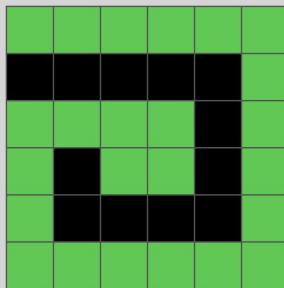
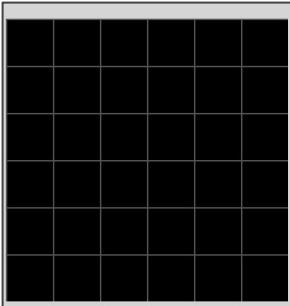


Takeaways

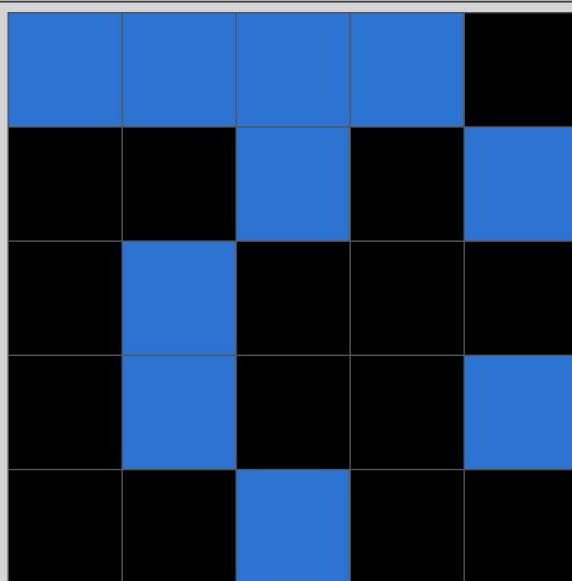
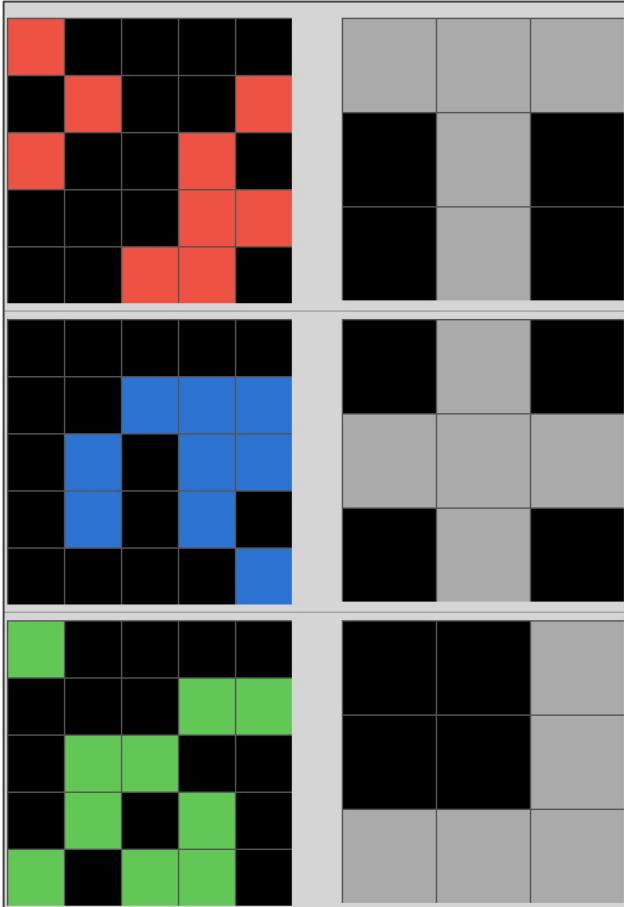
- CNN is not the optimal approach to solve the challenge
- Other approaches could include
 - DSL
 - Cellular automata
- Reasoning and abstraction are at the core of self-thinking AI



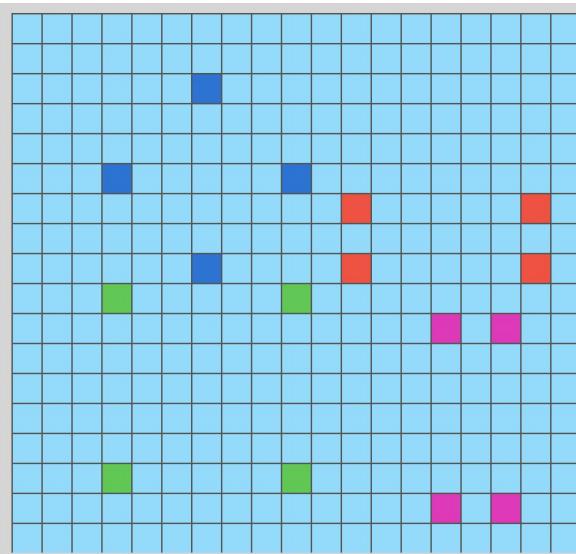
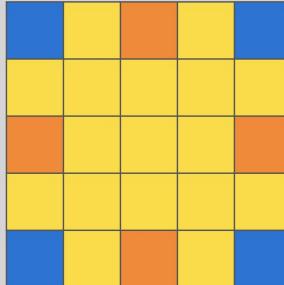
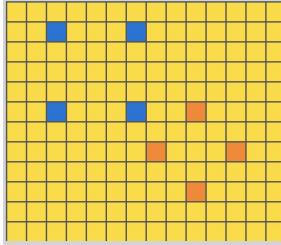
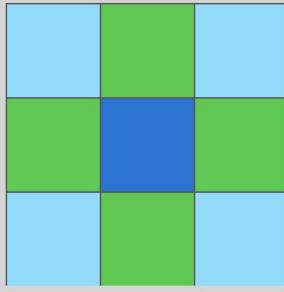
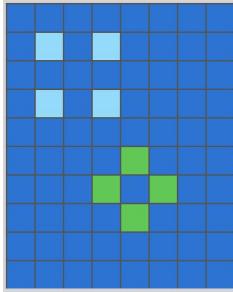
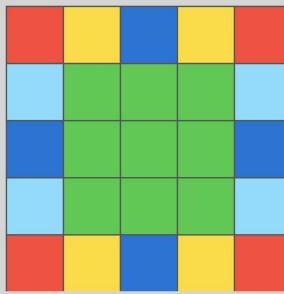
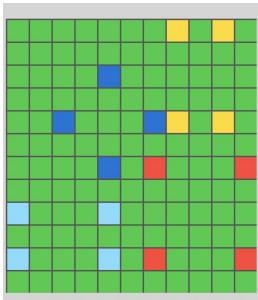
Thank you!



Go back



Go back



[Go back](#)