**Tiles HW**

I have decided to use Santa image from lectures for this homework. Its resized to 100x140 pixels and recolored to black/white;

A person wearing a mask

Description automatically generated with medium confidence

Note that while construction/reconstruction of image, tile size is 10x10.

As we know, there are 3 subtasks:

1. Simple Training
2. Using Rotation
3. Using Prefabricateds

**Subtask 1 – Simple Training**

For this subtask, we just train on the original tiles of the Santa image, then try to reconstruct the image by network.

Network takes the single tile of the Santa image, and outputs single tile.

Network Specifications:

100 –7 – 4 – 100

100 input layer neurons, 7 first hidden layer neurons, 4 second hidden layer neurons, 100 output layer neurons

We use tiles of shape 10x10 for traversing the image, therefore input and output shape is 10x10 = 100

Solver: Adam

Learning Rate: 0.001

Iterations: 200

We have already trained the network and saved the best network that produce the best results among others. Saved network of Subtask 1 is file: **/Tiles/models/1/best.pth**

In order to run saved network and produce results, please run this in console:

python source\_code.py --subtask=1 --model=best.pth

If we want to train new network rather than testing the best network:

python source\_code.py --subtask=1 --train=True

**Subtask 2 – Using Rotation**

For this subtask, we also use rotation of tiles during training of network.

We first train on original tiles for 0.15 portion of iterations, just for network to recognize the original tiles, then we train on rotations of tiles for the rest of iterations.

Network takes the single tile of the Santa image, and outputs single tile.

Network Specifications:

100 –8 – 4 – 100

100 input layer neurons, 8 first hidden layer neurons, 4 second hidden layer neurons, 100 output layer neurons

We use tiles of shape 10x10 for traversing the image, therefore input and output shape is 10x10 = 100

Solver: Adam

Pretraining ratio for Recognizing Tiles: 0.15

Learning Rate: 0.0007

Iterations: 250

We have already trained the network and saved the best network that produce the best results among others. Saved network of Subtask 1 is file: **/Tiles/models/2/best.pth**

In order to run saved network and produce results, please run this in console:

python source\_code.py --subtask=2 --model=best.pth

If we want to train new network rather than testing the best network:

python source\_code.py --subtask=2 --train=True

**Subtask 3 – Using Prefabricates**

For this subtask, in addition to the subtask 2, we also incorporate the prefabricated tiles which help us to reduce the expenses.

We use prefabricated tiles in 2 cases:

1. When some input tile is similar to some custom tile.
2. When some tile produced by network is almost similar to some custom tile.

Rather than training any new network for this subtask, we use the saved network from subtask 2. We only use the prefabricated tiles by checking the input and produced tiles and if any of them are similar to prefabricated tile, we replace them.

Prefabricated tiles are under directory: **/Tiles/prefab/**

 ****      

Saved network of Subtask 3 is file: **/Tiles/models/3/best.pth**

For seeing result of this subtask, please run this in console:

python source\_code.py --subtask=3 --model=best.pth