TASK 2

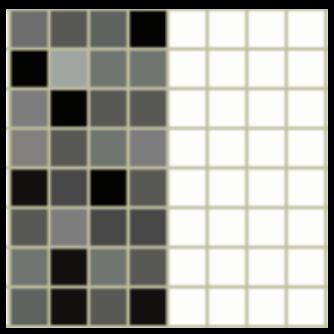
Signal Processing and ML

Machine Learning - CV

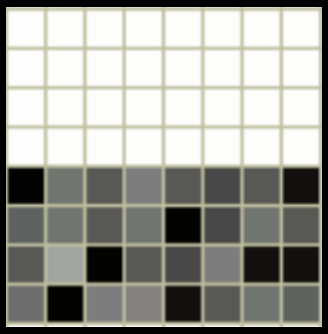
PROBLEM STATEMENT (Normal) – EDGE DETECTION USING CNN:

Edge detection is very useful in image segmentation and identifying objects in an image. In this task you are required to build a model that can classify edges as vertical or horizontal in an image. You need to generate a dataset consisting of 1000 8 x 8 images. 500 of these should have a vertical edge as shown below and the remaining should have a horizontal edge. The “dark” side of every image should be filled with random pixel values from 0-128 and the white side should be filled with all pixel values = 255. The data set must consist of dark and white regions of unequal widths too. Build and train a CNN **using only numpy** to classify these edges as vertical or horizontal.

The network must consist of minimum 2 convolutional layers and 1 FC layer.



Vertical Edge



Horizontal Edge

BONUS TASK (Optional) – CGAN to recreate your dataset:

For the same dataset created above, build a generative adversarial network (GAN) **in Pytorch**, to create an image with the required type of edge (vertical or horizontal edge).

GUIDELINES:

- For Task 2(Normal) only numpy is allowed. DO NOT use libraries like Tensorflow, PyTorch etc. The layers in your model must be built from scratch.
- For Task 2(Bonus) only PyTorch is allowed other libraries like TensorFlow and Keras are not allowed.

EVALUATION METRICS:

- Accuracy of prediction
- Quality of image generated by GAN.
- Efficient implementation of network
- Code quality – use proper classes and functions as necessary

SUBMISSION:

Submit a folder with the following files:

- Normal – A folder containing normal task code
  - train.py – containing the training and evaluation functions
  - layers.py – containing classes of the layers used
  - weights.npy – the final weights of the model after training
  - Other files from task 1 can be reused.
  - load_weights.py – loads the weights into the model
  - main.ipynb – a Jupyter notebook where you import the necessary modules from the files and train and validate the data.
  - A zip file containing your test set images
- Bonus – A folder containing bonus task code
  - discriminator.py – contains the classes for the discriminator
  - generator.py - contains the classes for the generator
  - config.py – hyper-parameters
  - utils.py – any other code that you may require
  - main.ipynb - a Jupyter notebook where you import the necessary modules from the files and train and generate the images.
  - It is recommended to use TensorBoard for graphs of training.
  - A zip file containing your generated images for each type of edge.

After making the folders, compress both folders together into .zip format and upload to your Google Drive, then get the sharing link of the folder from Google Drive and submit that link to induction portal.

RESOURCES:

- https://numpy.org/doc/1.19/
- https://pytorch.org/docs/stable/index.html
- https://www.coursera.org/learn/convolutional-neural-networks
- https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html