**Report: Assignment 2**
**COL775: Deep Learning. Semester II, 2023-2024**
**2023AIB2079, 2023AIB2074**

## PART 1: Object-Centric Learning With Slot Attention

**Code flow/description**:

- Image preprocessing: Input Image of size H×W×3 resized to 128x128x3.
- CNN encoder:
    - 4 convolution layers each of kernel=5 & padding=2 (each conv layer followed by ReLU)
    - 1st conv layer: stride=1, in_channel=3, out_channel=32
    - 2nd conv layer: stride=2, in_channel=32, out_channel=32
    - 3rd conv layer: stride=2, in_channel=32, out_channel=32
    - 4th conv layer: stride=1, in_channel=32, out_channel=32
- 2D positional embedding: Embedding of dim(batch_size, height, width, 4) is initialized. The last dimension of size 4 encodes positional information for each pixel, containing values for left, top, right, and bottom positions normalized between 0 and 1. Then positional embeddings tensor is passed through a linear layer to project it to the same dimensionality as the input feature vectors. Position embedding added to input.
- Layer norm
- 2 linear layers of 32x32
- Slot Attention module returned slot of shape batchxK(=11)×Dslots(=64)
- Slots sent to Spatial Broadcast Decoder.
- From decoder output, final image and K masks are generated one for each slot by taking softmax for each pixel of the first channel, across slots.

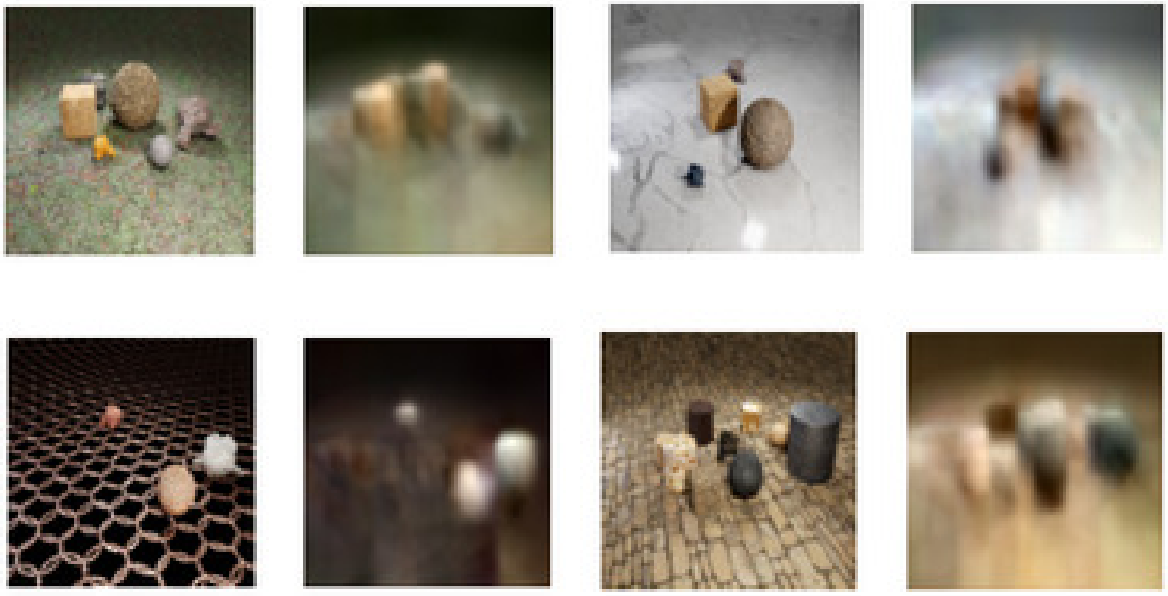
**Training parameters:**

- Optimizer = Adam with learning rate=1e-4
- Scheduler = ReduceLROnPlateau with  factor=0.1, patience=3, min learning rate=1e-7
- Criterion=nn.MSELoss()
- Batch size=32
- epochs=45

**Experiments: visualization of the generated masks and the reconstructed images**

**generated masks**

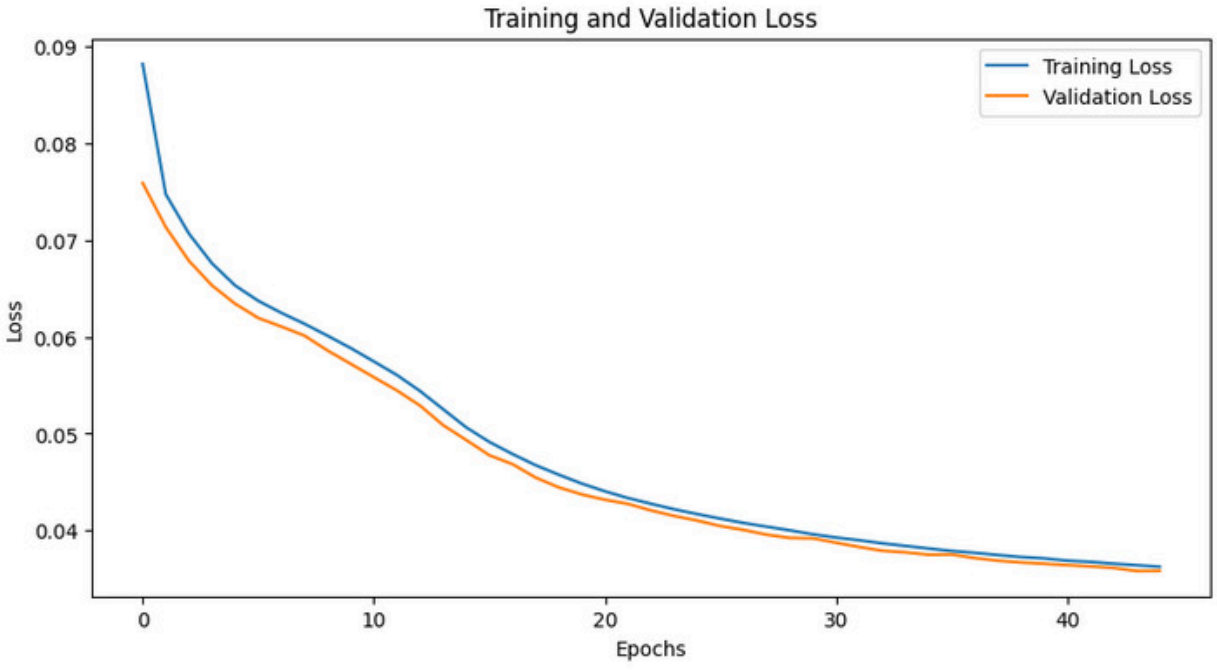
**Model Prediction on Validation dataset**:



Val_img Pred_img Val_img Pred_img

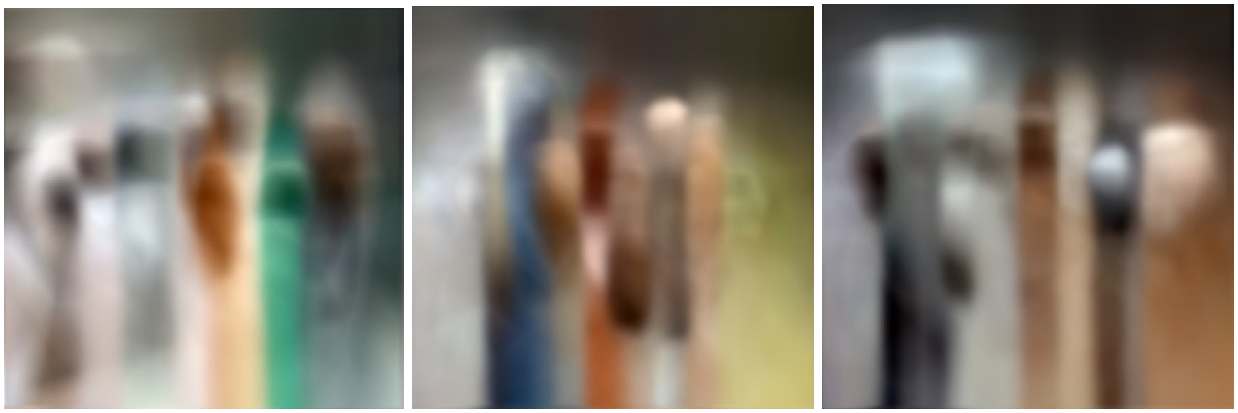
**Adjusted Rand Index (ARI)** score between the ground-truth and predicted object segmentation masks on the val split: **0.1237**

**Train and Val image reconstruction loss vs epochs**:



**Compositional Generation**:

**Reconstructed images after k means**:



**Clean-fid metric** using the validation images as ground truth: **303.407155281178**

**Trained model Google Drive link for Part1:**

**https://drive.google.com/file/d/1n90n03FDB_gemNqXDV7PpUp019AuPlgx/view?usp=sharing**

## PART 2: Slot Learning using Diffusion based Decoder

**Code flow/description**:

- Image preprocessing: Input Image of size H×W×3 resized to 128x128x3.
- Same CNN encoder and Slot Attention module used in Part-2 to generate slots.
- Input of shape 128x128x3 sent to VAE to get 32x32x3 sized output.
- Time sampled from uniform distribution and alpha_t computed with this time.
- Time embedded using sinusoidal embedding.
- Noise sampled from random distribution with mean 0 and unit var.
- Using output of VAE, noise and alphas, input_UNET to UNET computed by following formula: $\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$
- input_UNET, slots and time embedding sent to UNET.
- Model is trained using UNET output and sampled noise.

**Training parameters:**

- Optimizer = Adam with learning rate=4e-4
- Scheduler = StepLR with  step size=1, Gamma=0.95
- Criterion=nn.MSELoss()
- Batch size=32
- epochs=38

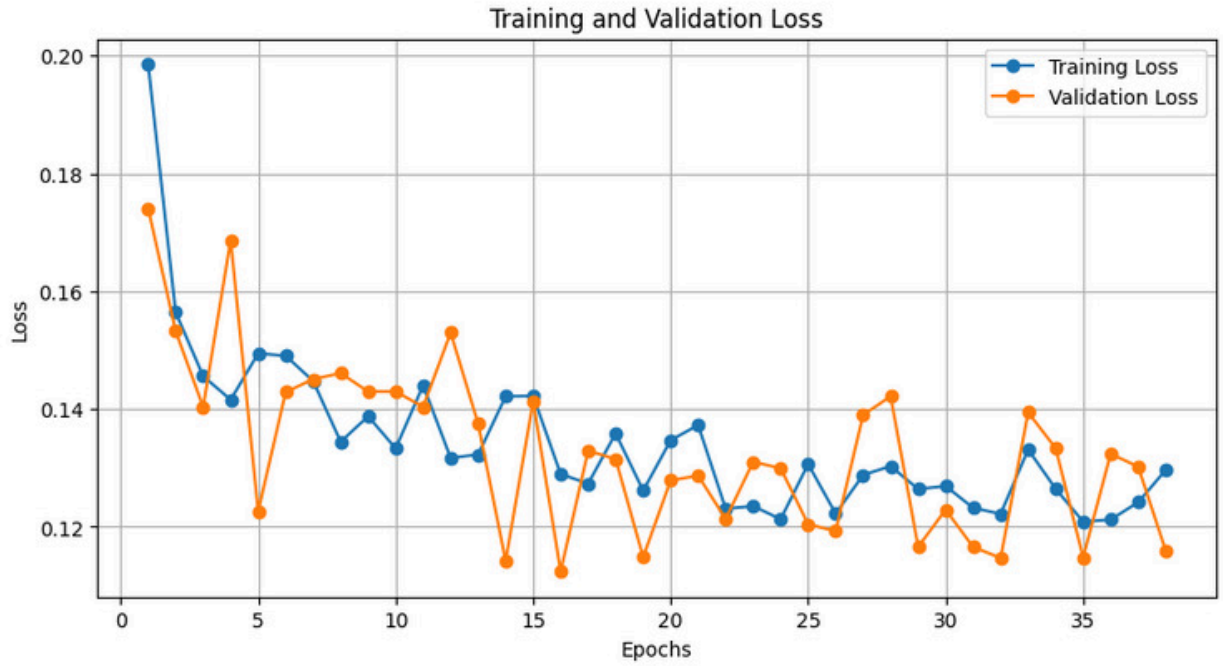
**Model Prediction on Validation dataset:**

(These images are generated by giving a random noise and slots of the input image. Algorithm 2 Sampling is used for image  generation)

Val_img Pred_img Val_img Pred_img

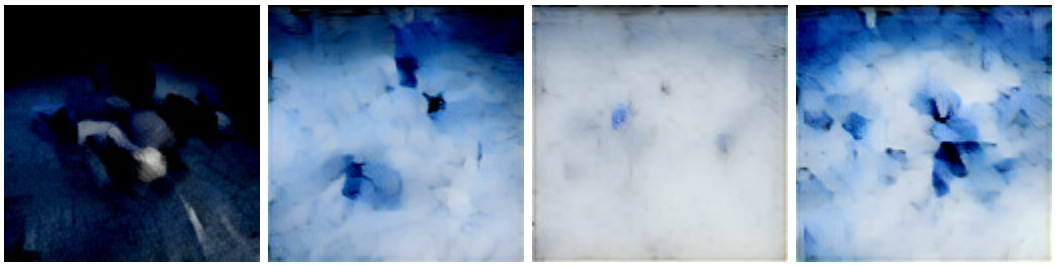
**Adjusted Rand Index (ARI) score** between the ground-truth and predicted object segmentation masks on the val split: **0.0529**

**Train and Val image reconstruction loss vs epochs**:




**Decoding (Generation) Ancestral Sampling**:

(These images are generated by giving a random noise and 11 slots randomly picked from clusters generated after k-means. Algorithm 2 Sampling is used for image generation)



**Clean-fid metric** using the validation images as ground truth: **211.49077285503185**

**Trained model Google Drive link for Part2:**

https://drive.google.com/file/d/1vrt2YbLegPMw868p1PATIyynT_2NnN_q/view?usp=sharing