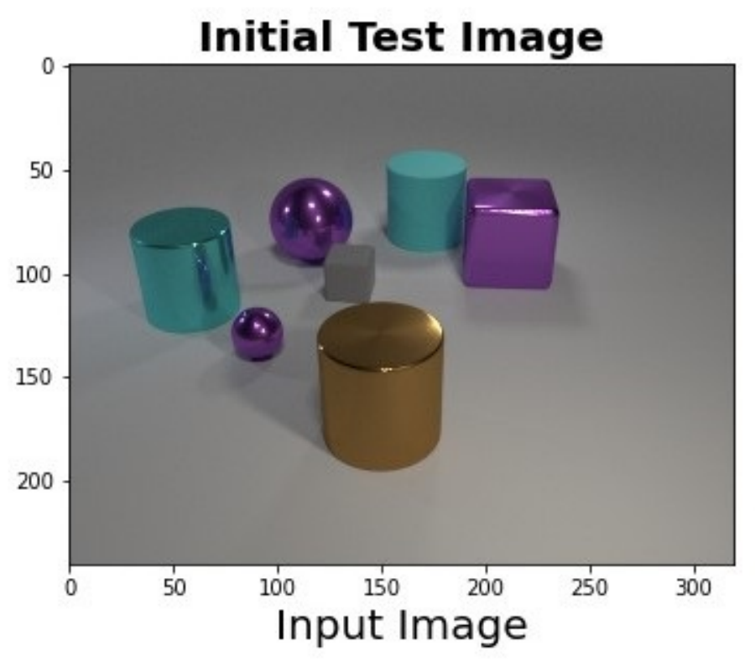
Segmentation as Clusterization

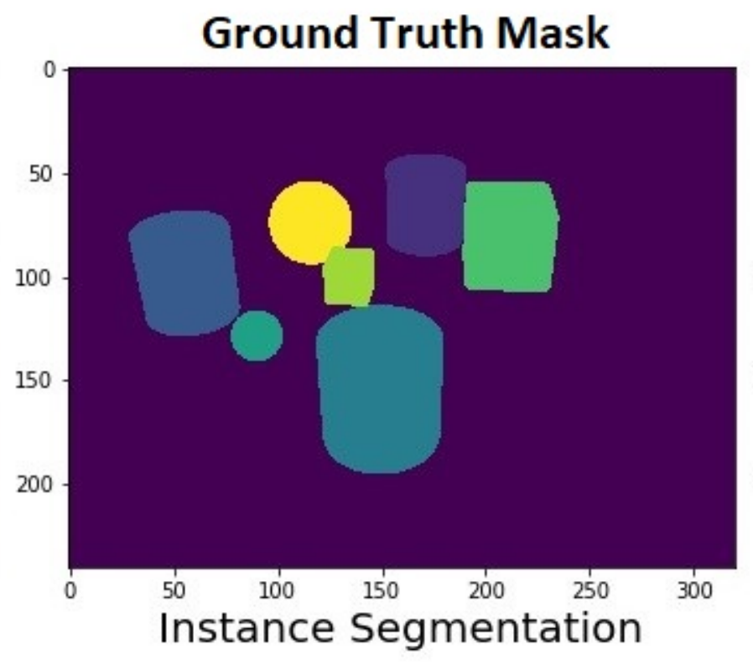
Students: Ionescu Andrei and
Nicolau George-Alexandru
Coordinator: Duţă Iulia

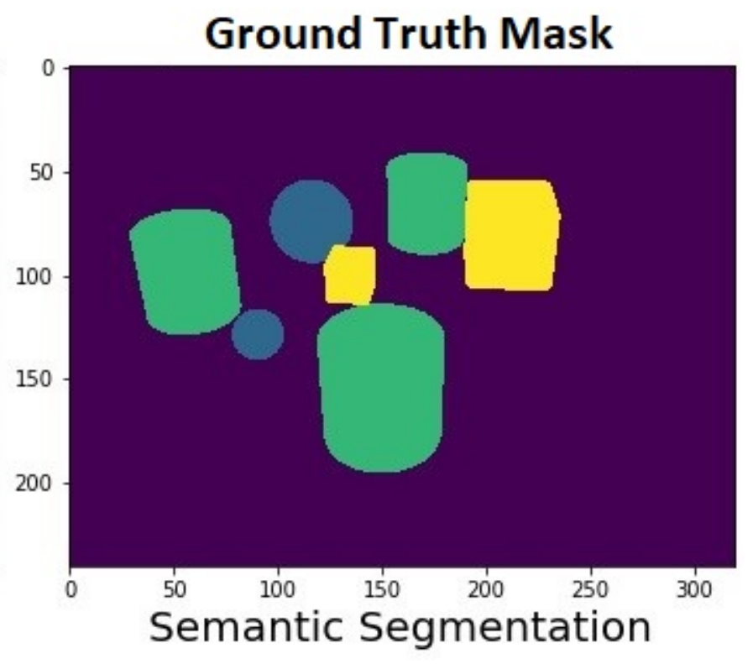
University of Bucharest
Faculty of Mathematics and Informatics

Project Goal

Our goal is to train a model in unsupervised manner to do instance segmentation on a subset of the **CLEVR Dataset [2]** by using K-Means Clusterization method on the extracted feature maps of the model after it processes an image. Then, we also try to do a semantic segmentation on the dataset using a a similar model by training it in supervised manner, and we compare the obtained results.


Initial Test Image


Ground Truth Mask

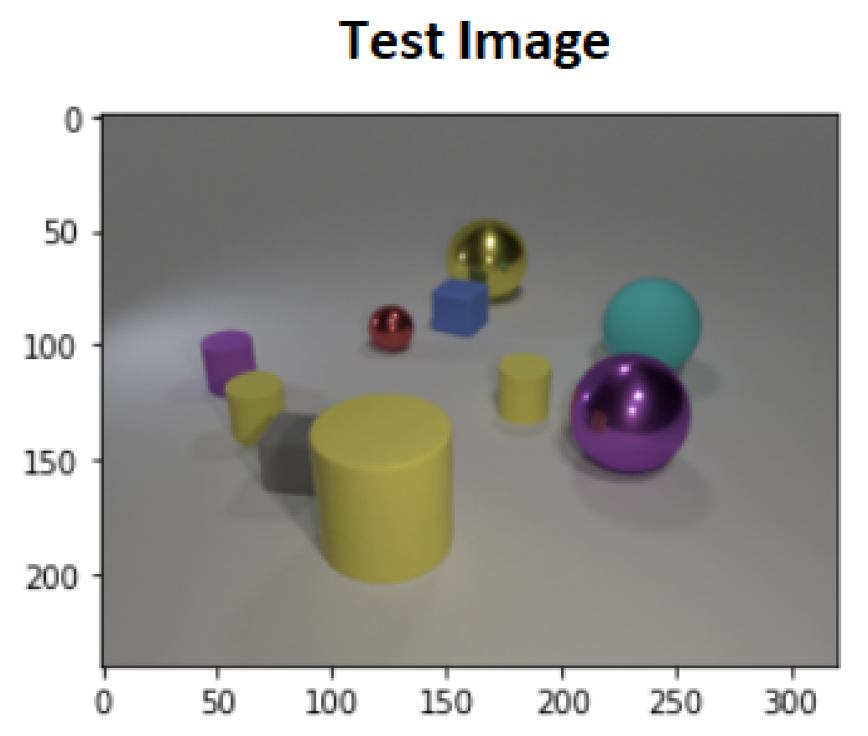

Ground Truth Mask

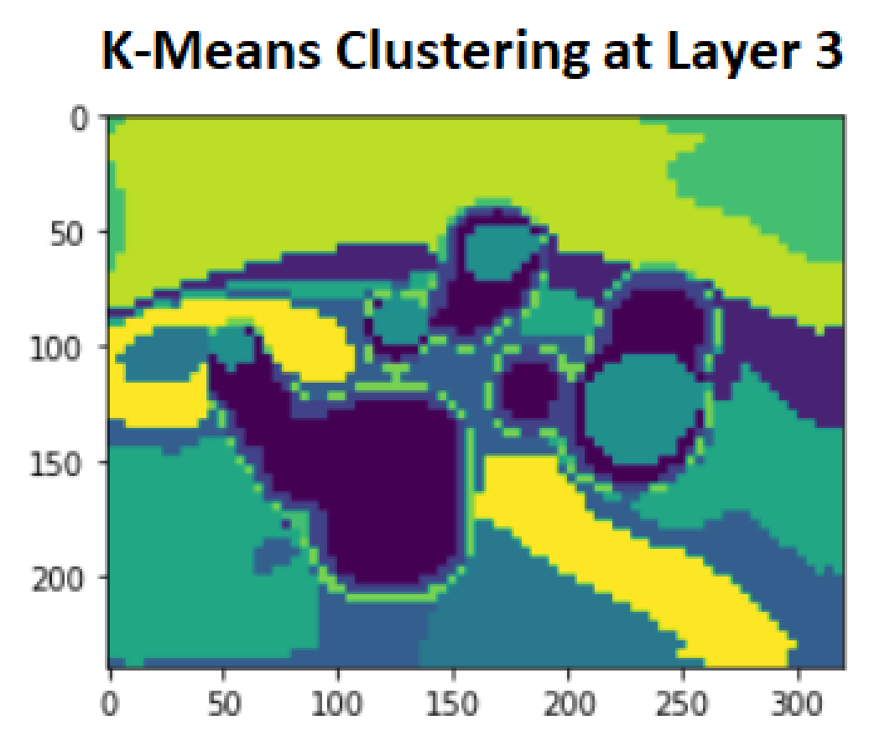
Unsupervised Procedure

We start by training a **U-Net** and an **Auto-Encoder** to reconstruct the input images and we minimize its loss per pixel using Mean Squared Error Loss Function.

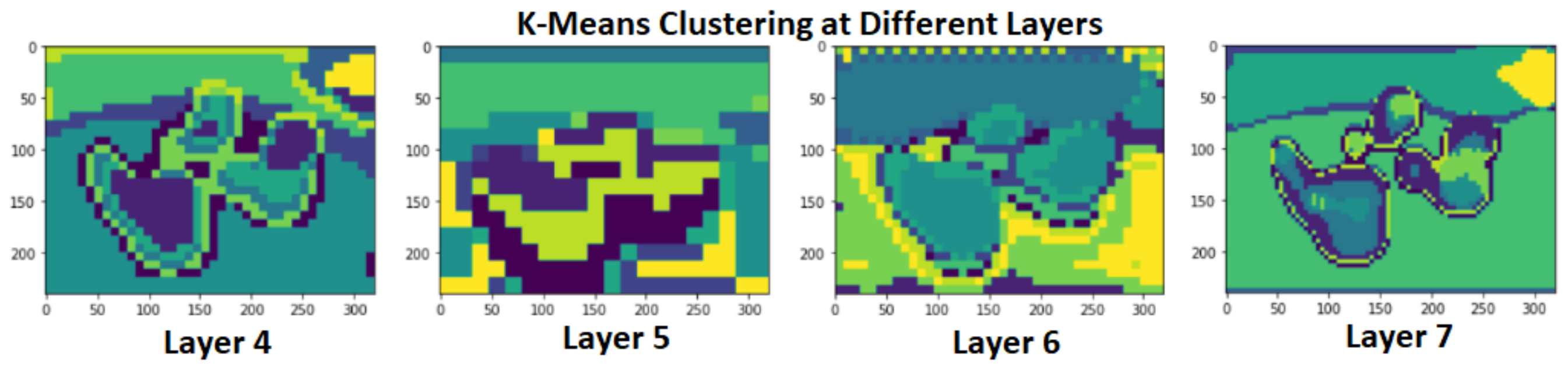
$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \qquad (1)$$

After the previous step, they should learn various representations of the images at each layer, differentiating them by color, shape, material and so on. Using the extracted feature maps, present in the model layers, we use the **K-Means Clusterization Algorithm** to segregate the objects present in the image from one another.


Test Image


K-Means Clustering at Layer 3

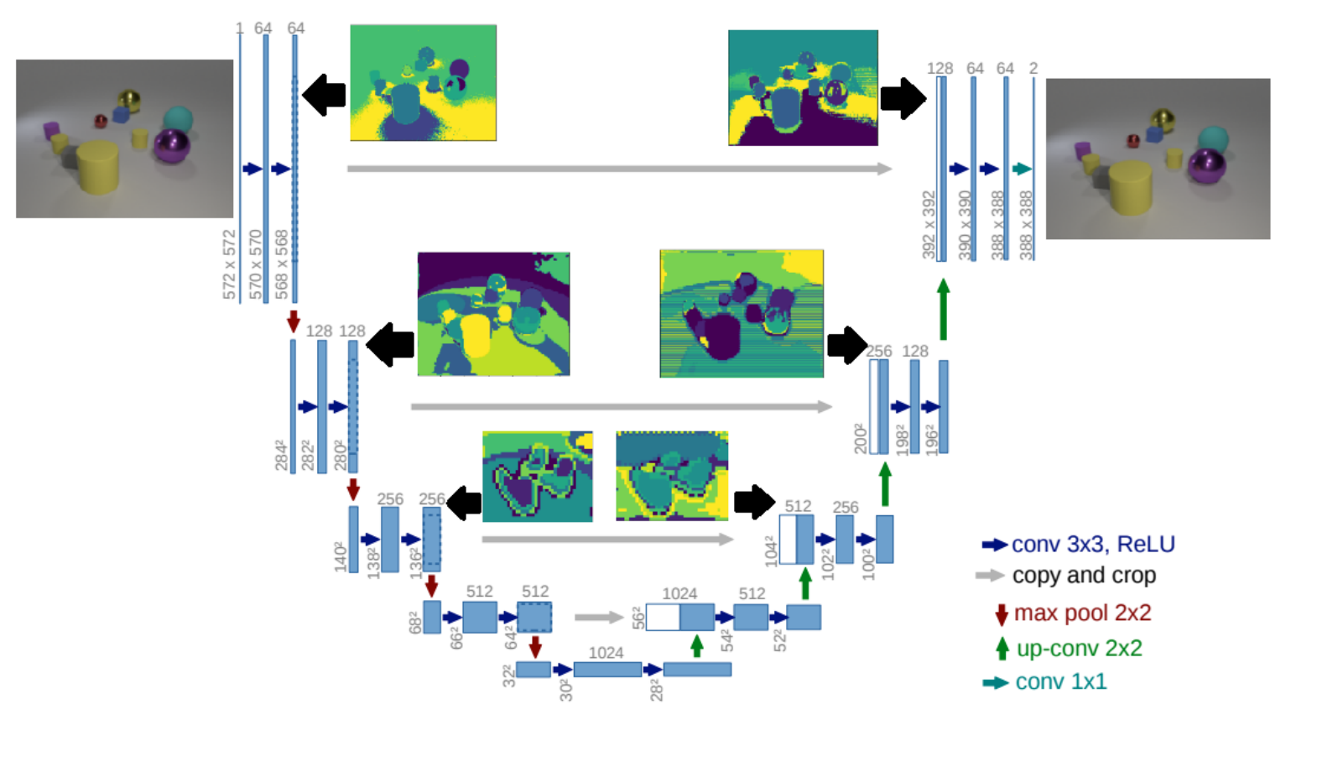
Because the background occupies the largest area in the images, it skews the model's learning towards gray values. Considering this fact, we approach the problem from two angles using the **RGB** and the **Grayscale** versions of the images (with a weighted loss using the inverse number of samples).


K-Means Clustering at Different Layers

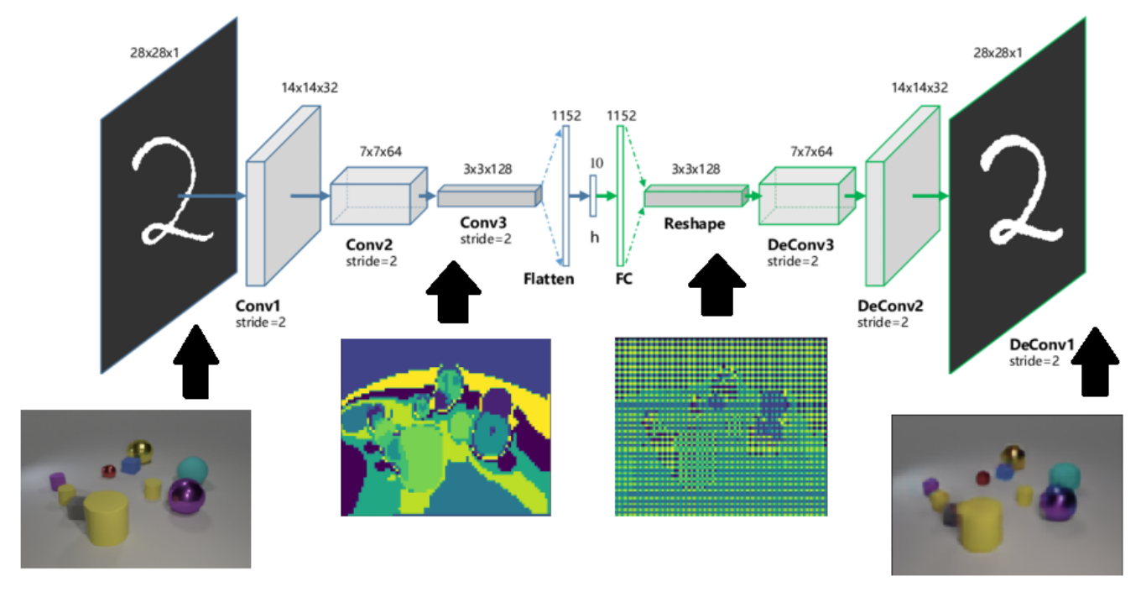
Architectures Used

We used a similar U-Net architecture as in the original paper [3] and an Auto-Encoder with the same internal architecture without the skip connections. The input images are of size 240×320 and we normalized them in $[0, 1]$. We used batch normalization before the convolutional layers. We also tried to use dropout but it had a negative impact on the results.

The architectures as well as some intermediary outputs can be seen in the following figure:

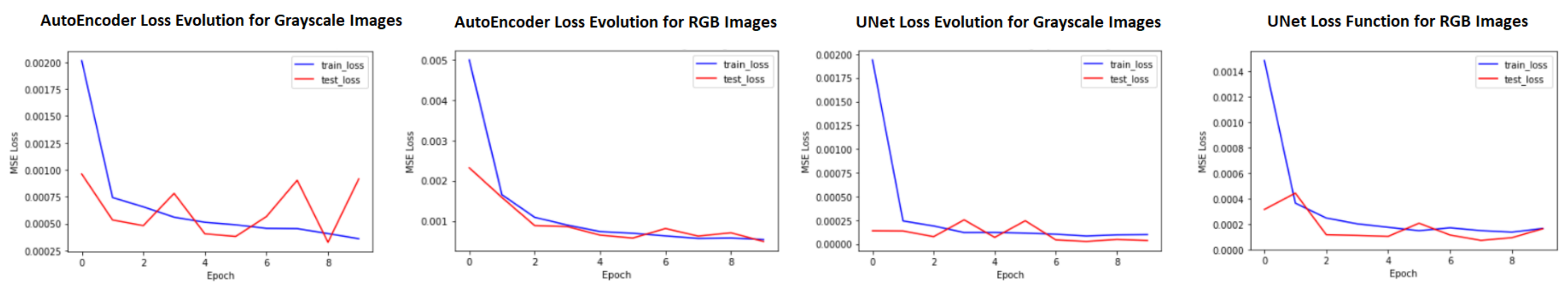


U-Net



Auto-Encoder

We trained the models on 7000 images and tested them on another set of 1000 images for 10 epochs using AdamW as the optimizer, with a learning rate and weight decay of $1e-2$:



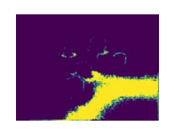
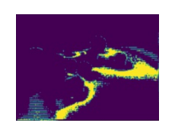
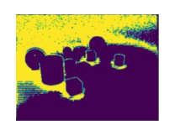
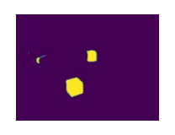
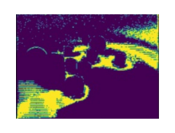
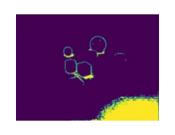
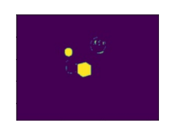
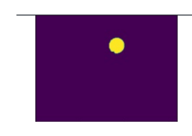
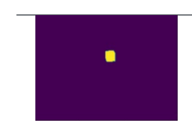
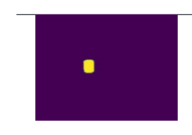
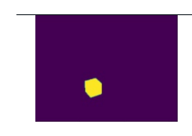
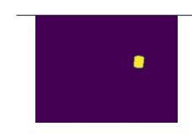
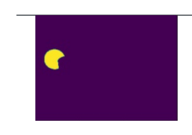
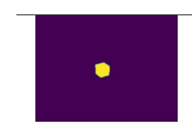
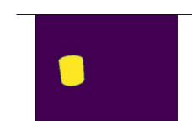
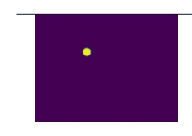
The models were trained in unsupervised manner to solve an auxiliary task, namely image reconstruction. In the case of the U-Net, it cheated by using the skip-connections to copy the input from earlier layers and it did not learn complex representations of the data.

Results

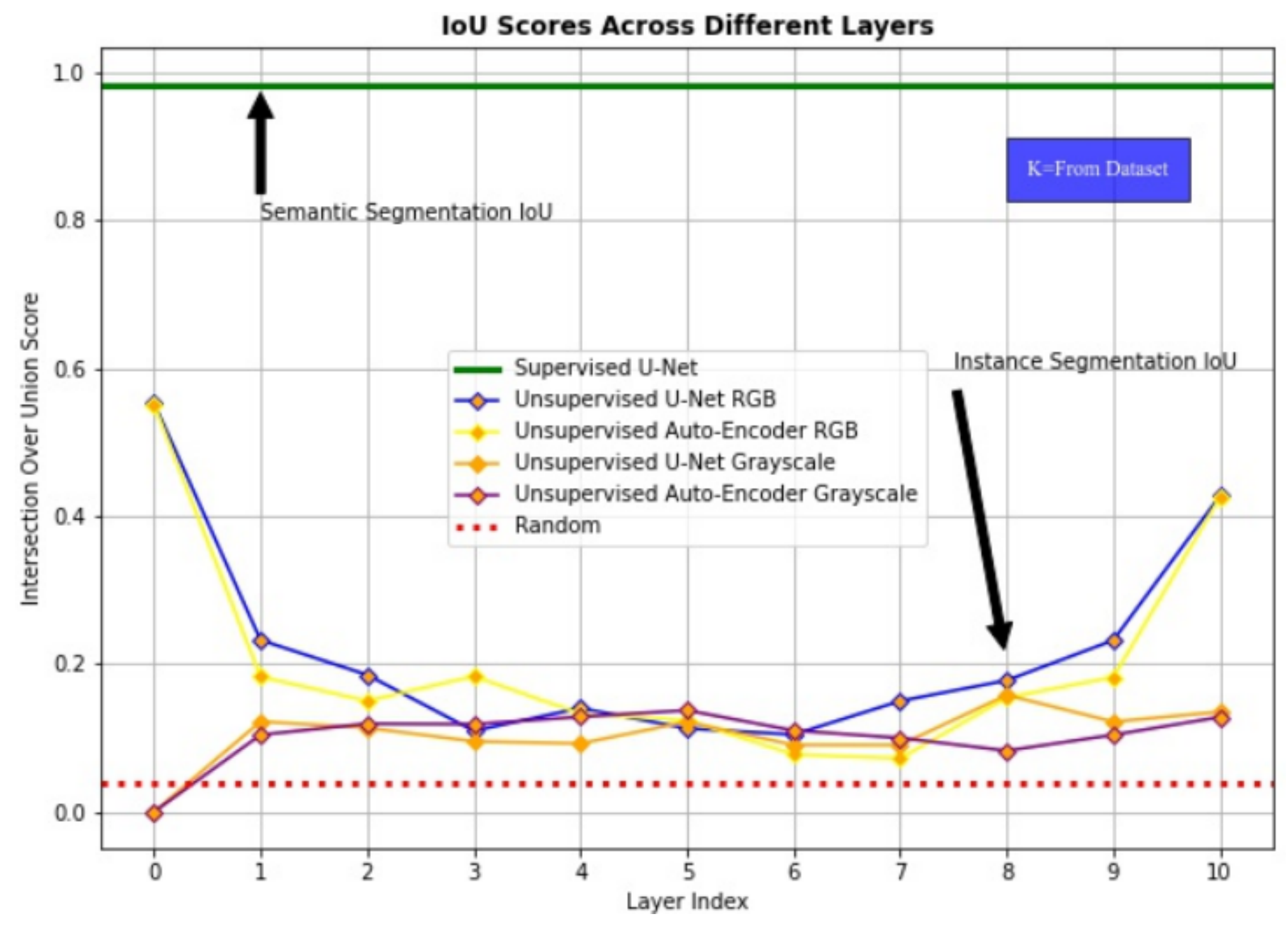
- Send one test image to extract intermediary features.
- Perform K-Means Clustering on upsampled maps.
- Label every pixel from an image.
- Segregate images into multiple masks.
- Compare predicted masks with ground truth using **Intersection over Union**.

$$IoU(TrueMask, PredictedMask) = \frac{|TrueMask \cap PredictedMask|}{|TrueMask \cup PredictedMask|} \qquad (2)$$

- **Problem** ! No association between predicted and ground truth masks.
- **Solution** ! Apply Hungarian Matching Algorithm [1].
- Penalize model for less masks by padding zeroes.
- Choose the top $N_{TrueMasks}$ most relevant pairs.
- Compute mean of the relevant IoU scores $(3.9e-1)$ as shown below.

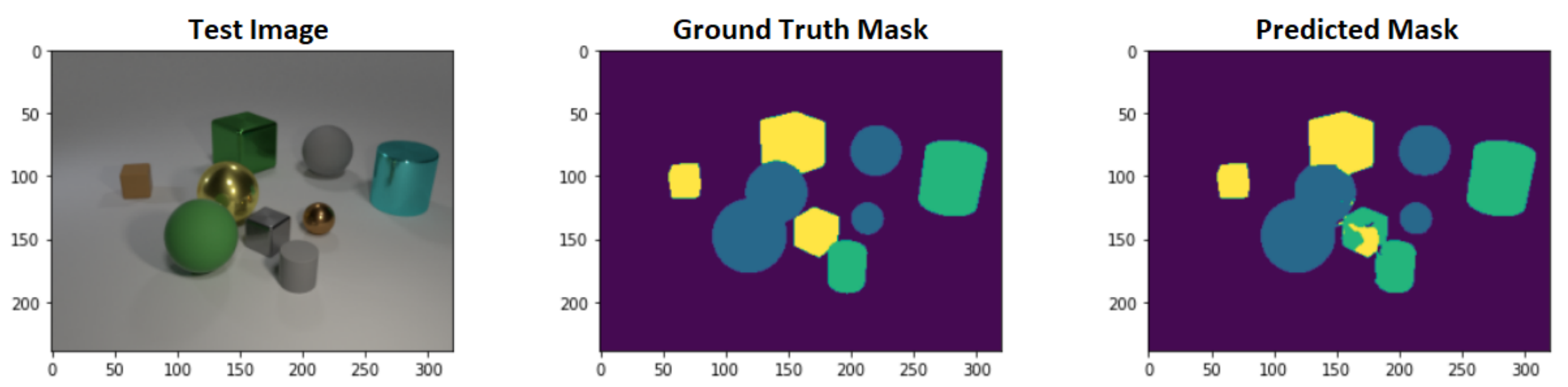
TM\PM										
	1.06e-3	1.58e-1	1.03e-1	**4.09e-1**	1.16e-4	2.26e-1	7.2e-5	9.90e-2	1.13e-4	6.74e-4
	0.00e+0	0.00e+0	0.00e+0	3.49e-5	0.00e+0	6.21e-5	**.55e-1**	3.40e-3	0.00e+0	2.65e-2
	0.00e+0	0.00e+0	**0.00e+0**	1.09e-3	2.56e-1	0.00e+0	.00e+0	0.00e+0	5.28e-4	0.00e+0
	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	.02e-1	**1.90e-3**	0.00e+0	7.76e-3
	0.00e+0	0.00e+0	0.00e+0	8.97e-4	**6.80e-1**	0.00e+0	.00e+0	0.00e+0	0.00e+0	0.00e+0
	2.80e-1	**0.00e+0**	0.00e+0	5.66e-4	0.00e+0	0.00e+0	.00e+0	0.00e+0	0.00e+0	0.00e+0
	**6.43e-1**	0.00e+0	0.00e+0	5.85e-4	2.47e-2	0.00e+0	.00e+0	0.00e+0	2.16e-4	0.00e+0
	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	.00e+0	3.84e-3	0.00e+0	**6.72e-1**
	1.92e-4	0.00e+0	0.00e+0	2.80e-3	9.63e-4	0.00e+0	.00e+0	0.00e+0	**9.69e-1**	0.00e+0
	0.00e+0	0.00e+0	0.00e+0	0.00e+0	0.00e+0	**0.00e+0**	.00e+0	7.16e-4	0.00e+0	2.13e-1

- The models seems to take in consideration only the color (U-Net and Auto-Encoder).
- Grayscaling images to handle the bias towards colors brings no success.
- The auxiliary task may be too simple to learn more complex features.
- The final results can be seen in the following image.

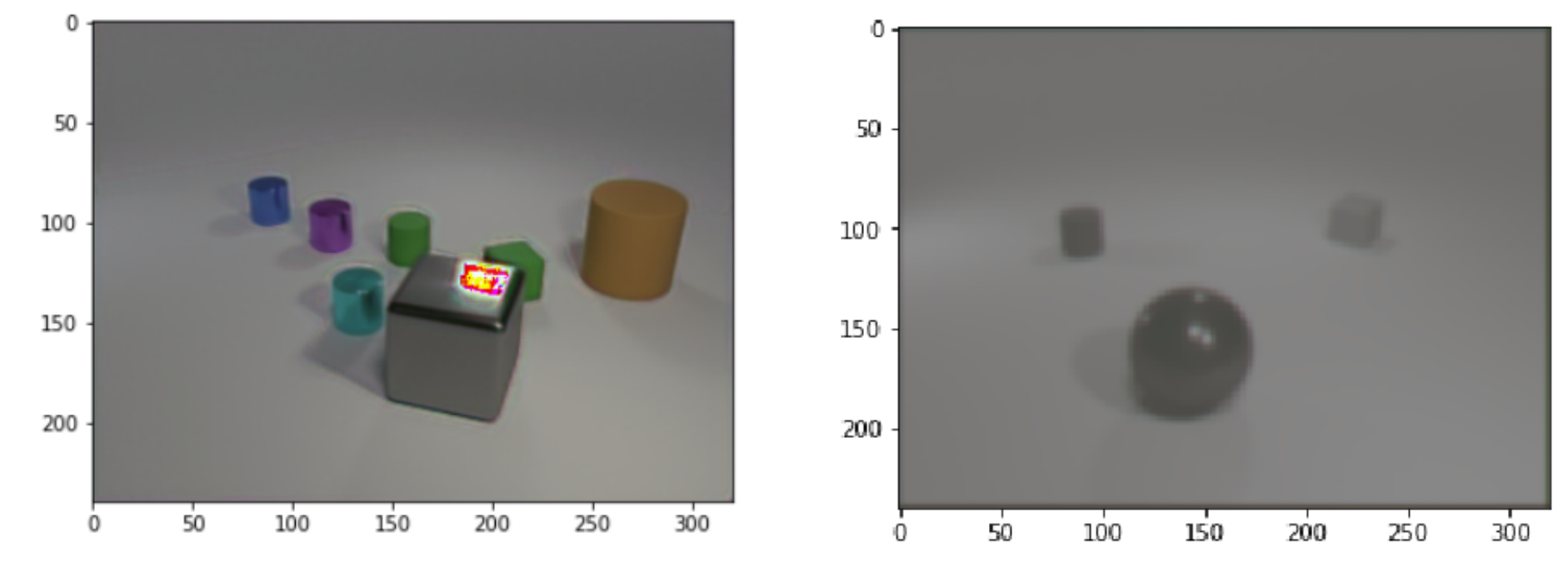


Supervised Procedure

- Train a U-Net for semantic segmentation in a **supervised** way.
- Aggregate the masks according to the shape to create masks with 4 possible classes: cube, sphere, cylinder and background.
- Similar architecture with the reconstruction task (change the output to $4 \times 240 \times 320$).
- Use Cross Entropy Loss at pixel level.
- **Accuracy:** 0.9828**!**



Encountered Problems



References

[1] *Hungarian Algorithm*. URL: `https://en.wikipedia.org/wiki/Assignment_problem`.

[2] Rishabh Kabra et al. *Multi-Object Datasets*. https://github.com/deepmind/multi-object-datasets/. 2019.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: `1505.04597`. URL: `http://arxiv.org/abs/1505.04597`.