Self-Supervised 3D Shape Reconstruction from Single-View 2D Images

Honglin Zhu, Kangyun Chen, Wenyu Jiao, Zhengyuan Zhang

Johns Hopkins University | Whiting School of Engineering | Baltimore, MD

Introduction

Nowadays, three-dimensional (3D) Reconstruction from two-dimensional (2D) images are widely used in a large variety of fields, including computer graphic, medical imaging, and so on. Besides, plenty of 2D images as well as its 3D ground truth are required to get a satisfactory 3D reconstruction result. However, in order to release the stress of both collecting images from different viewpoints and 3D supervising annotations, here comes the importance of single-view input source and self-supervision approach.

Objectives

Given single-view 2D inputs, our project aims to reconstruct 3D shapes in mesh representation under self-supervision, which consists of polygons with vertices and faces.

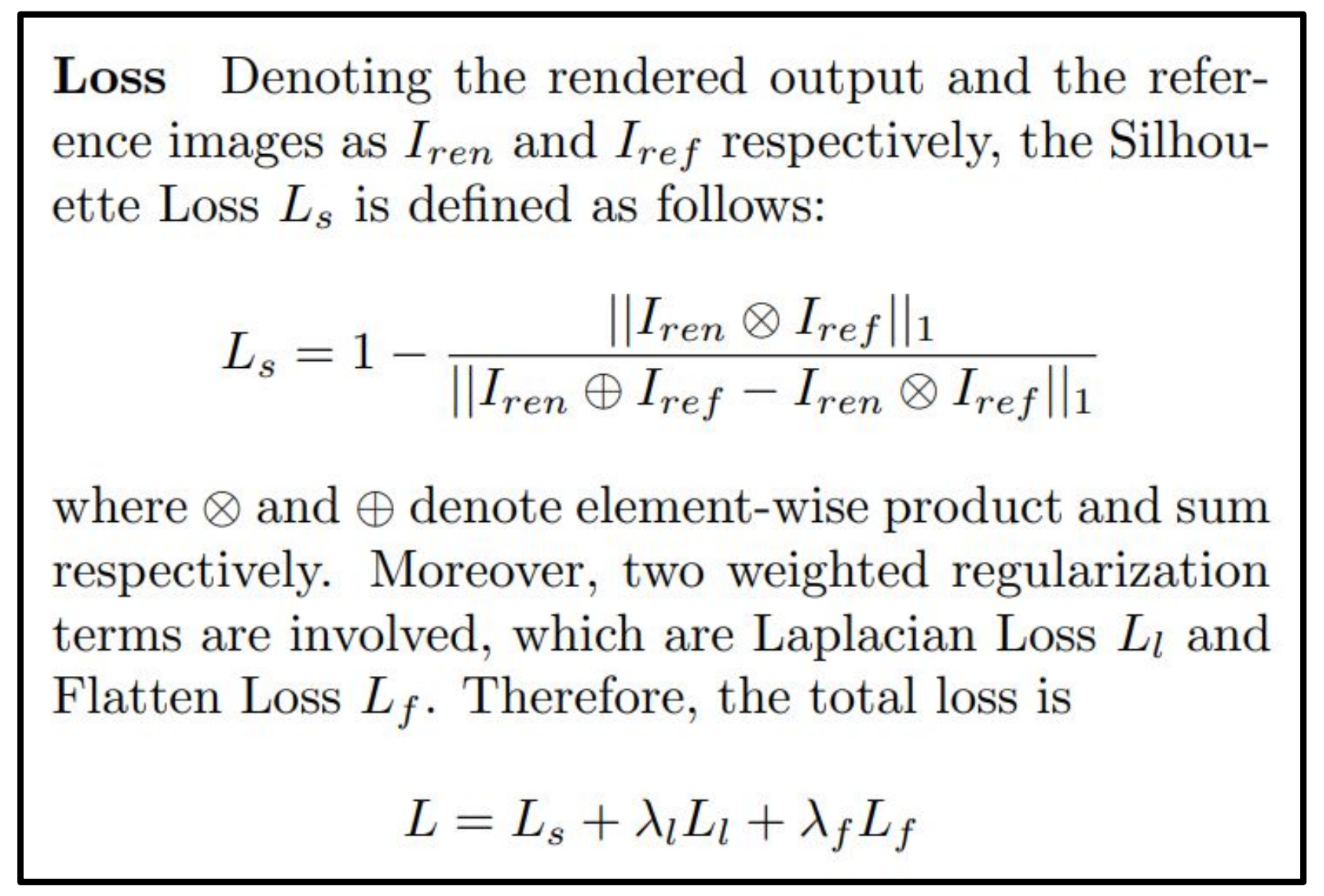
Methods

We choose Soft Rasterizer {arXiv:1904.01786} based reconstruction network as our baseline model, consisting of an encoder-decoder mesh generator and a differentiable renderer.

According the provided codebase, for each object to be reconstructed, two 2D images from two viewpoints are taken, one is for input and the other is for reference.

Given the input 2D image, a 3D mesh representation is generated first, which is fed into a differentiable renderer subsequently to get a rendered 2D output.

In the end, we will calculate the loss based on the difference between the rendered 2D image and the ground truth reference 2D image, where the differentiable rendering procedure ensures the end-to-end backpropagation, where the loss is defined as follows:



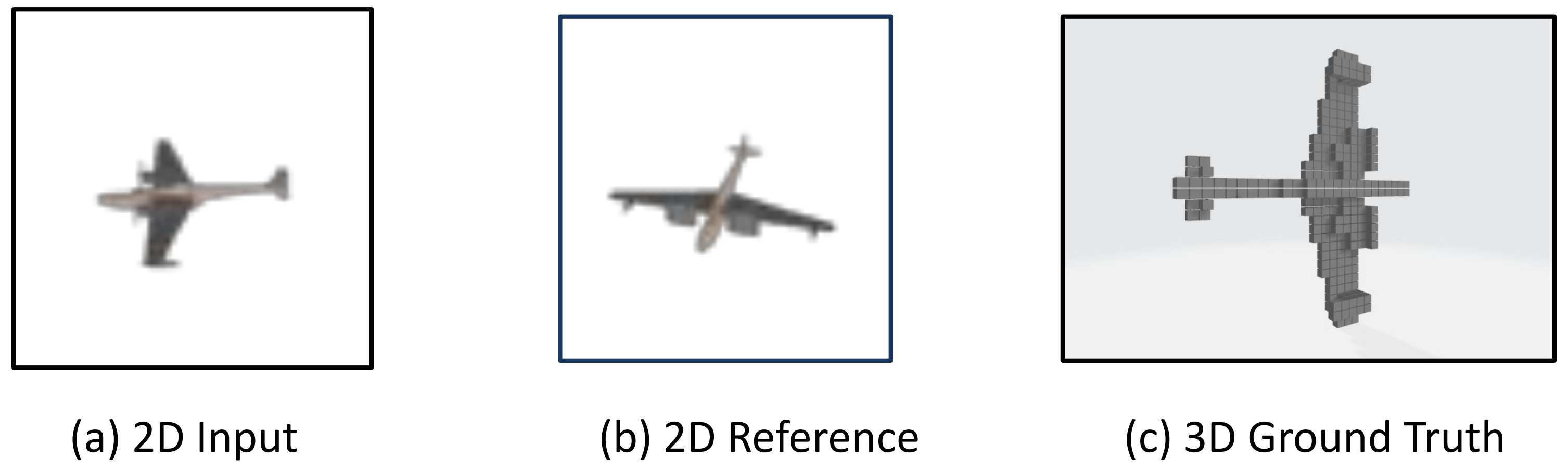
Loss Denoting the rendered output and the reference images as I_{ren} and I_{ref} respectively, the Silhouette Loss L_s is defined as follows:

$$L_s = 1 - \frac{||I_{ren} \otimes I_{ref}||_1}{||I_{ren} \oplus I_{ref} - I_{ren} \otimes I_{ref}||_1}$$

where $\otimes$ and $\oplus$ denote element-wise product and sum respectively. Moreover, two weighted regularization terms are involved, which are Laplacian Loss L_l and Flatten Loss L_f. Therefore, the total loss is

$$L = L_s + \lambda_l L_l + \lambda_f L_f$$

Result



(a) 2D Input

(b) 2D Reference

(c) 3D Ground Truth

Figure 1: A Graphical Depiction of the Dataset

Each object is rendered from 24 azimuth angles with image resolution 64 x 64 x 3, with a fixed elevation angle under the same camera and lighting setup.

Each object is also accompanied by a 3D ground truth voxel representation.

Only 2 viewpoints were applied in our task and the ground truth voxel representation was only involved in the validation and test process.

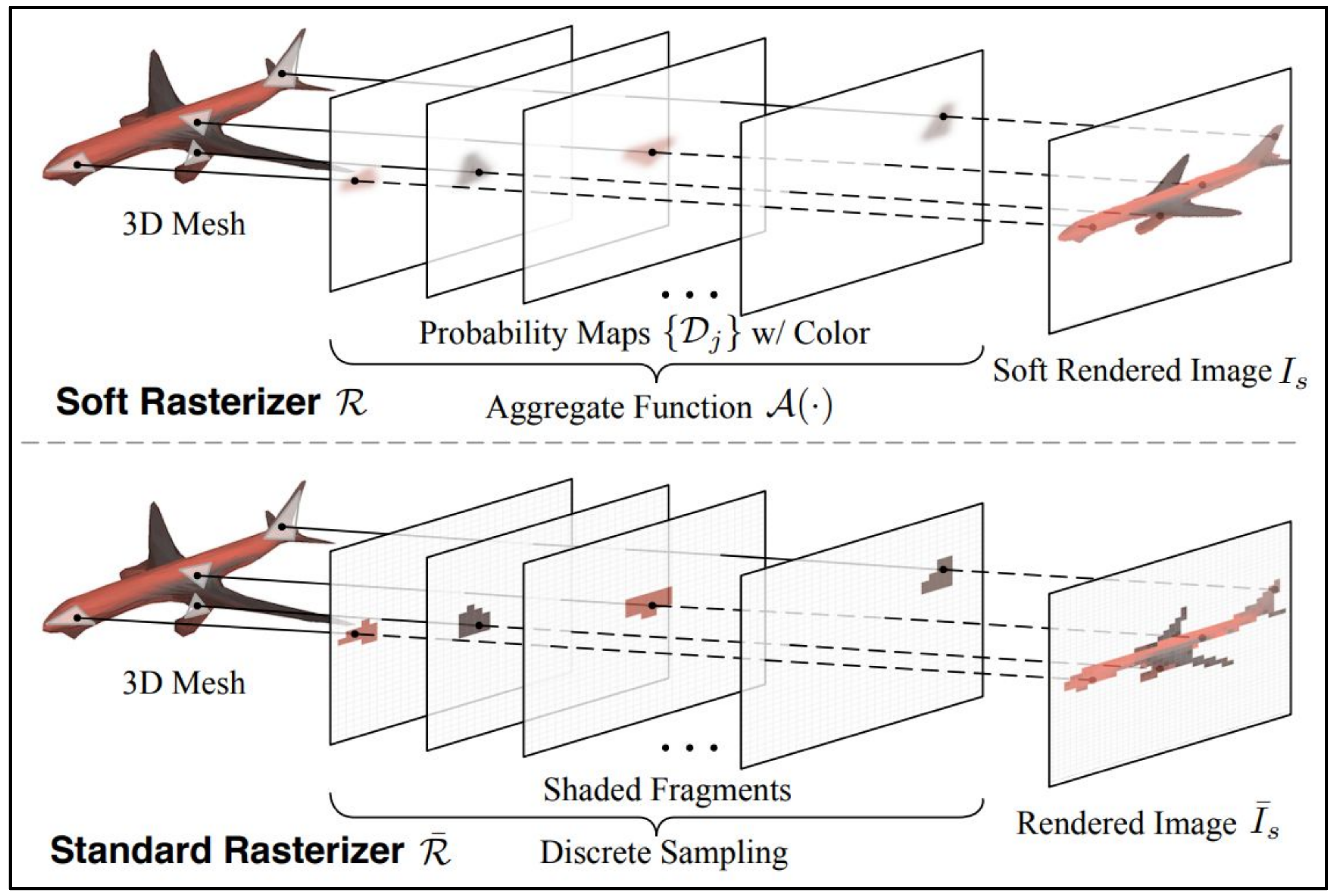
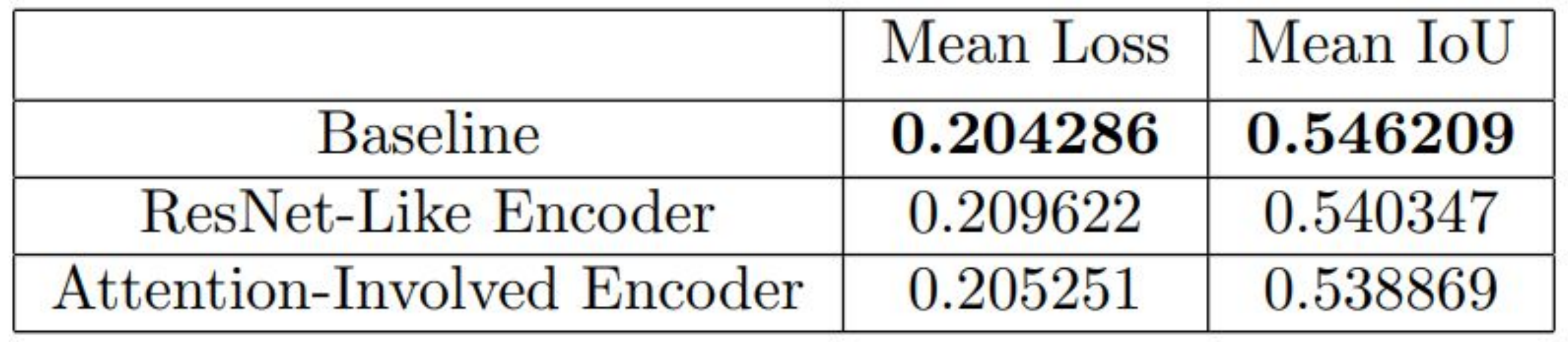

3D Mesh
Probability Maps $\{\mathcal{D}_j\}$ w/ Color
Soft Rendered Image I_s
Soft Rasterizer $\mathcal{R}$
Aggregate Function $\mathcal{A}(\cdot)$
3D Mesh
Shaded Fragments
Rendered Image $\bar{I}_s$
Standard Rasterizer $\bar{\mathcal{R}}$
Discrete Sampling

Figure 2: Soft Rasterizer (Differentiable Renderer)

In traditional rendering procedure, both rasterization and z-buffering are not differentiable and cannot flow gradient information.

However, Soft Rasterizer ensures differentiable rasterization and z-buffering by replacing them with probability maps computation and aggregations respectively.



	Mean Loss	Mean IoU
Baseline	**0.204286**	**0.546209**
ResNet-Like Encoder	0.209622	0.540347
Attention-Involved Encoder	0.205251	0.538869

Figure 5 – Testing Results

The result shows the baseline model has the best performance. We think one of the reasons is that the dataset used for training and testing might not be complex enough to benefit from the advanced architectural features in the ResNet-like and Attention-Involved encoder models. The baseline model might perform better in this case due to its simpler architecture and fewer parameters.

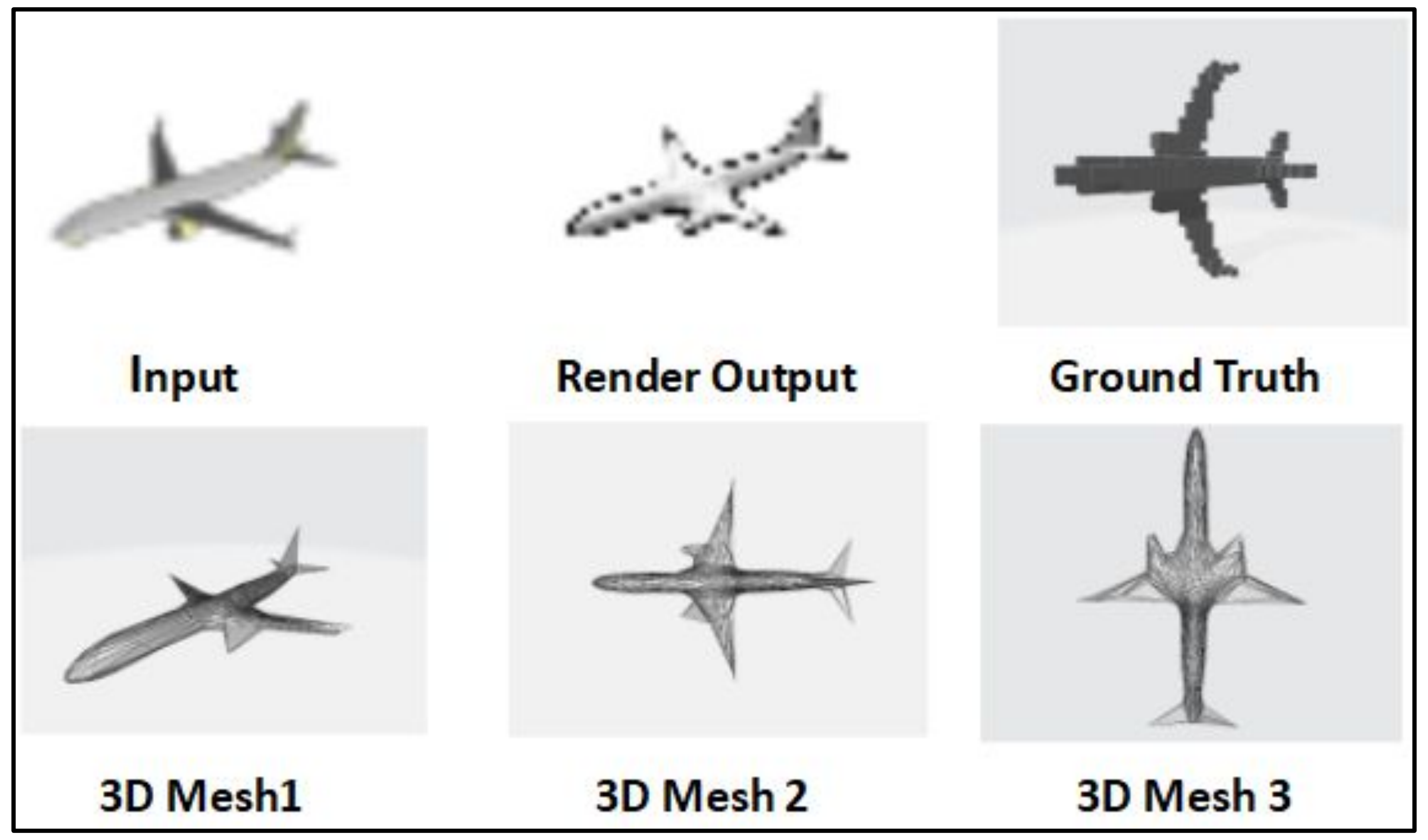


Figure 6 – Output Visualization

The input is a 2D image. The render output is for calculating loss with reference image. The three 3D mesh images are presented from different viewpoints for our generated object, which can be evaluated by being compared with the 3D ground truth.

Future Work

1. Tune Hyperparameters
2. Modify Both of Encoder and Decoder
3. Apply Networks for Reconstructing Colors and Textures