Learning Single-View & Multiple-View 3D Object Reconstruction

Presented by,

Ronak Dedhiya

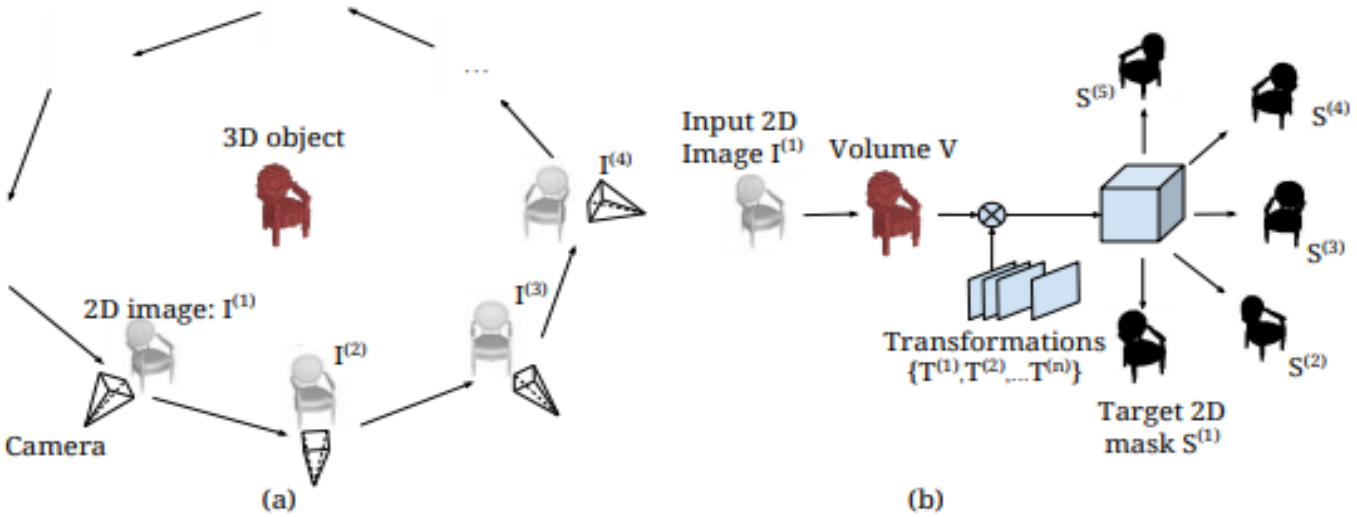
11/12/2022

Goal

- Given a single view or multiple views of the images to construct the 3d object.
- Inferring the shape and layout of 3D scenes from 2D images has long been a fundamental problem
- Has wide applications in robotics, autonomous vehicles, graphics, and AR/VR.

Perspective Transformer Nets*:

Learning 3D shape recovery from 2D images with object silhouettes.



$$L_{proj}(I^{(k)}) = \frac{1}{n}\sum_{j=1}^{n} ||P\left(f\left(I^{(j)}\right); \propto^{(j)}\right) - S^{(j)}||^2$$

- Input is the object Image, and output is its volumetric 3D shape such that the perspective transformations of the predicted shape match well with corresponding 2D observations.

* Yan, Xinchen, et al. "Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision." *Advances in neural information processing systems* 29 (2016).

Dataset Description

ShapeNet Dataset:

A subset of the ShapeNet dataset consists of 50,000 models and 13 major categories.

The 13 categories are airplane, bench, cabinet, car, chair, display, lamp, speaker, rifle, sofa, table, telephone, and watercraft.

- ShapeNet rendered images ftp://cs.stanford.edu/cs/cvgl/ShapeNetRendering.tgz (12.8 GB)
- ShapeNet voxelized models ftp://cs.stanford.edu/cs/cvgl/ShapeNetVox32.tgz (100mb)

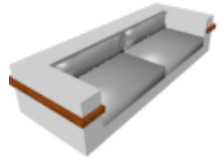
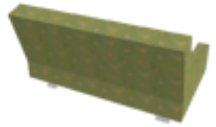
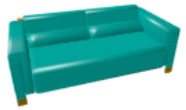
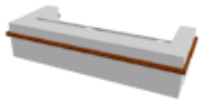
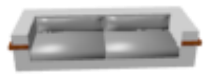
Training and testing split were predefined. Almost 4/5 models from each category was used for training and 1/5th for testing.

Dataset Examples

Airplane

	Model 1	Model 2	-----	Model 5000th
View 1			-----	
View 2			-----	
View 24th			-----	

Sofa

	Model 1	Model 2	-----	Model 5000th
View 1			-----	
View 2			-----	
View 24th			-----	

Baseline run

Average Prediction voxel IOU with the ground truth 3d object

Test Category	Airplane	Bench	Car	Chair	Display	lamp
Reported in paper	0.5556	0.4924	0.7123	0.4494	0.5395	0.4223
Baseline result obtained by local running for 10 epochs	0.234	0.217	0.343	0.323	0.345	0.289

- The local run was done on the CPU local machine and was trained for fewer epochs.

Difficulties:

1. Implementation is available only in theono or TensorFlow1. Unable to get it run on google colab.

2. Perspective transformation layer transforms the 3d voxels into 2d projections using the camera's intrinsic & extrinsic parameters. However, It is very complex logic and difficult to convert code to pytorch or tensorflow.

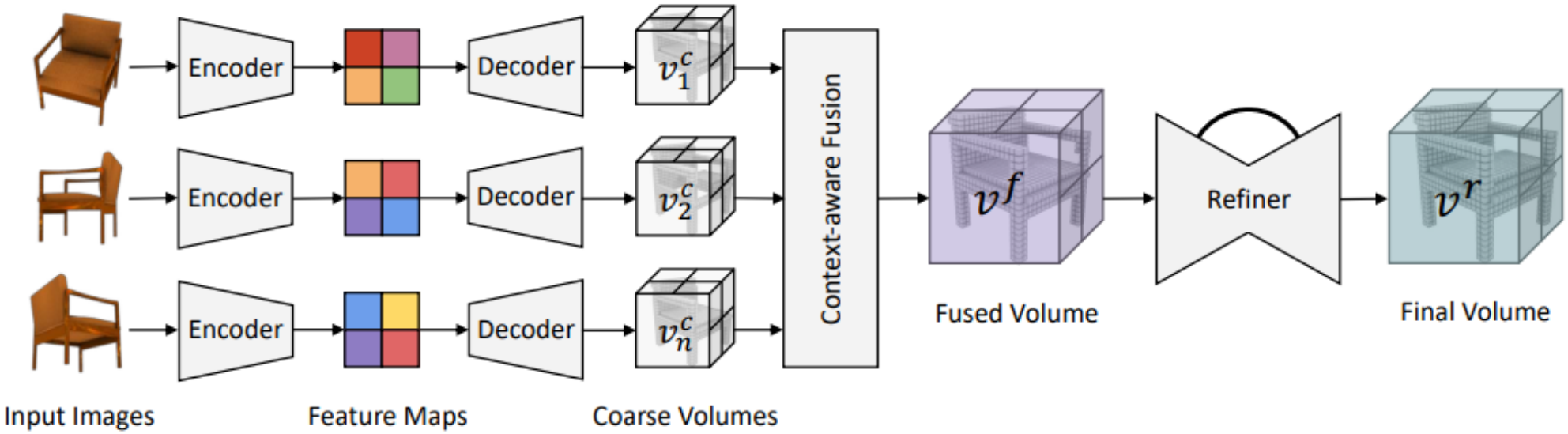
3. Alternative Pytorch 3d rendering methods are available but works with only specific Cuda version. Again unable to run it on google colab.

4. Difficult to do unsupervised learning without renderer or perspective transformation.

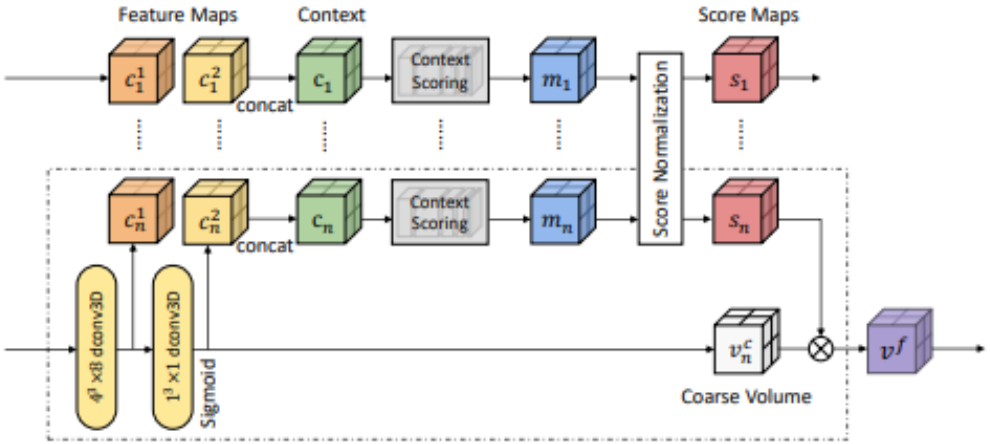
Supervised 3d reconstruction

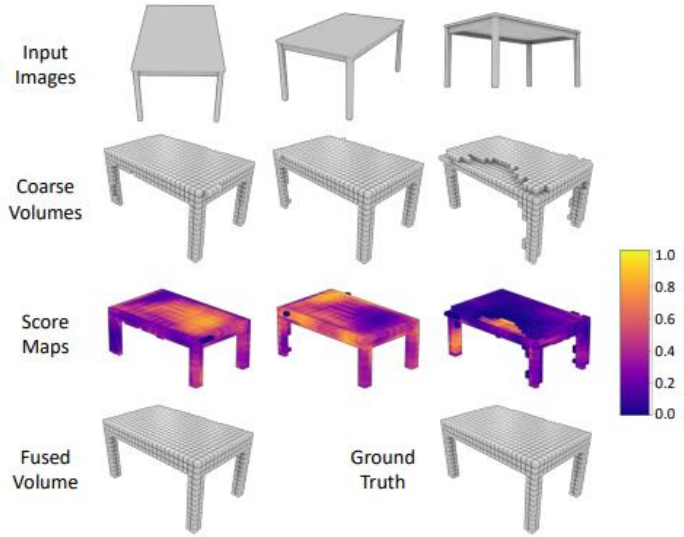
Pix2Vox:

1. Encoder produces feature maps from input images.
2. Decoder takes each feature map as input and generates a coarse 3D volume correspondingly
3. Single or multiple 3D volumes are forwarded to the context-aware fusion module, which adaptively selects high-quality reconstructions for each part from coarse 3D volumes to obtain a fused 3D volume
4. Refiner with skip connections further refines the fused 3D volume to generate the final reconstruction result

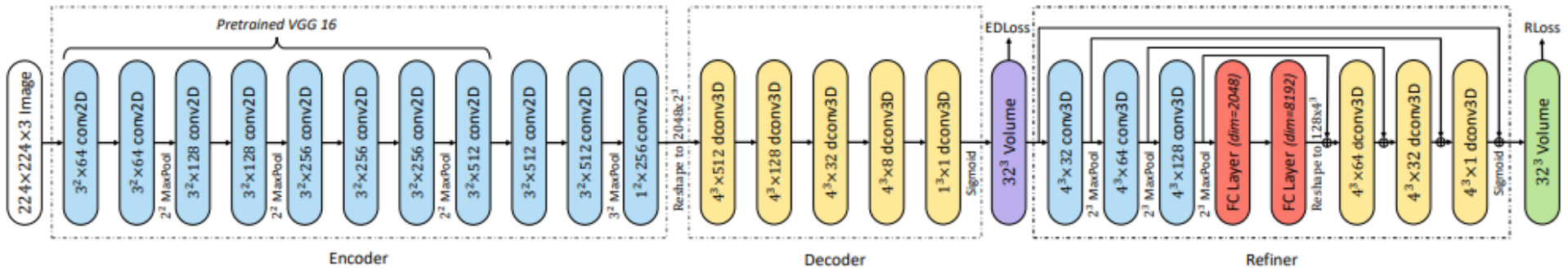


Context – Aware fusion





ED Loss = Rloss = Binary cross entropy loss between predicted voxel and groundtruth voxel

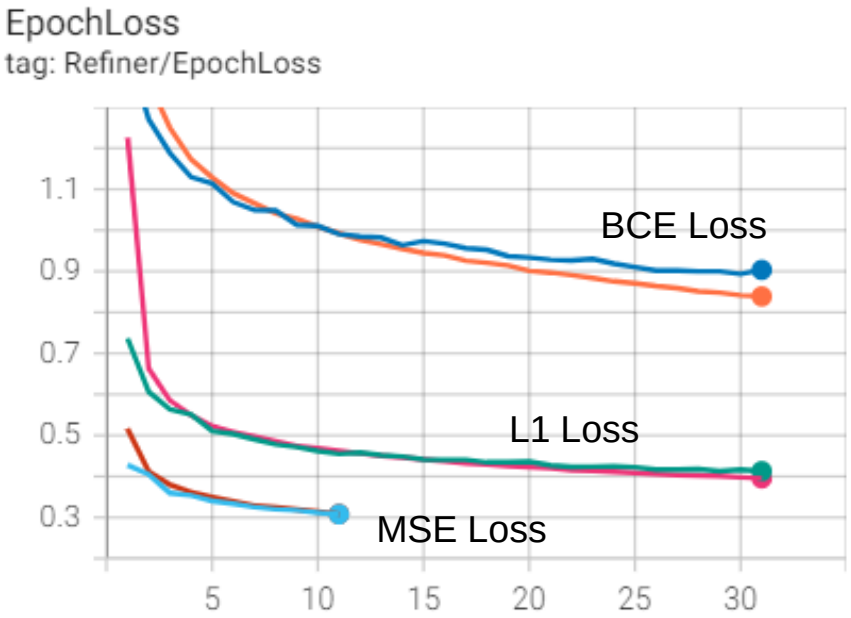


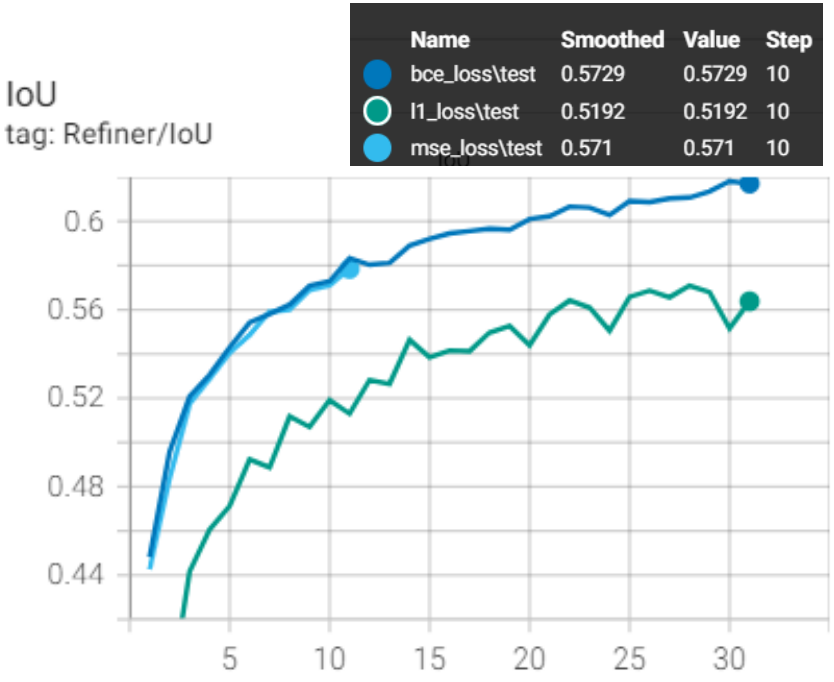
Results

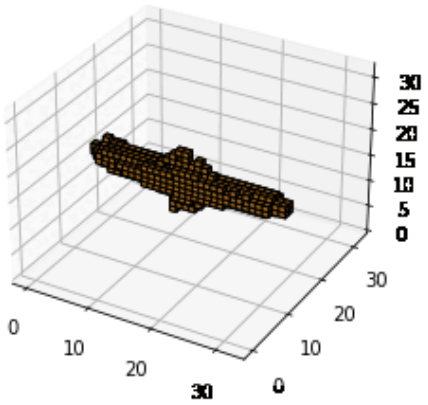
Average Prediction voxel IOU with the ground truth 3d object

Test Category	Airplane	Bench	Car	Chair	Display	lamp
Reported in paper (250 epochs)	0.684	0.616	0.854	0.567	0.537	0.443
Baseline result (with 30 epochs)	0.5958	0.4941	0.8216	0.5175	0.4953	0.4432
Result with L1 Loss (with 30 epochs)	0.5273	0.4109	0.8178	0.4405	0.3926	0.3561
Result with MSE Loss (with 10 epochs)	0.5421	0.4455	0.8114	0.4802	0.4304	0.3978

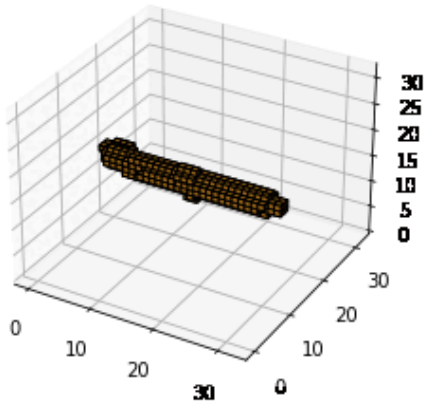
Plots



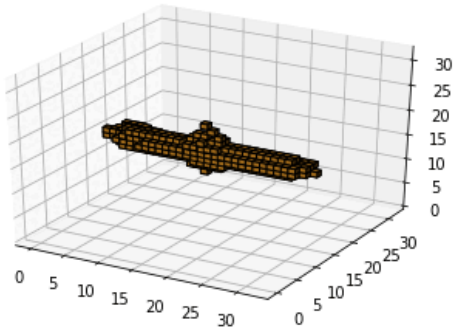




BCE Loss



L1 Loss



MSE Loss

Future scope

- Combination of MSE loss and binary cross entropy loss may improve performance.
- Using ResNet architecture instead of vgg-net
- Using a large set of data to train the network

Code:

https://www.kaggle.com/code/ronak555/pix2-vox

Questions?

Thank you!!