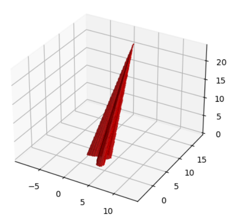
Problem Set 3, Due: Friday February 27th 2023

Bryan Olisaemeka Mbanefo
Computer Science
bmbanefo@stanford.edu

1 Space Carving [45 points]

a) Implement the form initial voxels function. See Code submission for implementation.
b) Implement the carve function.

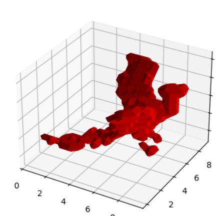
- First step is to project each voxel into the camera image plane.
- Check if it lies within the silhouette captured by the camera.
- If visible then add the voxel to the list of carved voxels.



(a) Output of Carving 4000

Figure 1: Problem 1 initial output better bounds false

c) Using number of voxels = 6000000. Submit the final output after all carvings have been completed, What are some flaws you notice in the reconstruction?



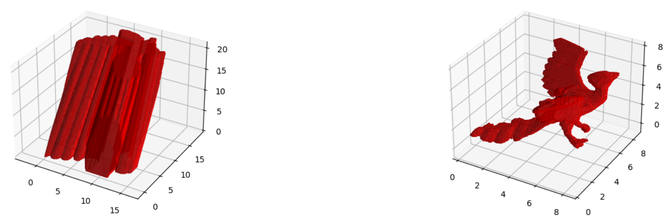
(a) Output of Carving 6000000

Figure 2: Problem 1 Final output better bounds false

Flaws include the following:

- The image is still a bit blocky, which suggeests that better voxel bounds can still be estimated
- This means that there are still voxels that do not pertain to the object being included in the final carve.

d) Estimate better bounds. Include your new carving in the report, which should be more detailed. Are there any remaining issues with the reconstruction? Why do you think they are still there



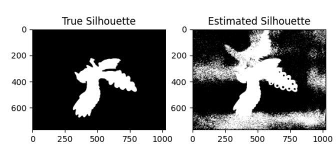
(a) Carving output Better bounds (b) Final output better bounds

Figure 3: Problem 1 Results Better Bounds set to True

Still some issues:

- For example the tail is not perfect and also the right foot of the bird. This could be due to the limitation of space carving where it cannot detect concavities. Therefore depending on the orientation of the bird in the image, there is bound to be some spaces missed.

e) Finally, let's have a fun experiment. Notice that in the first three steps, we used perfect silhouettes to carve our object. Look at the estimate silhouette() function implemented and its output. Notice that this simple method does not produce a really accurate silhouette. However, when we run space carving on it, the result still looks decent!



(a) Silhouettes

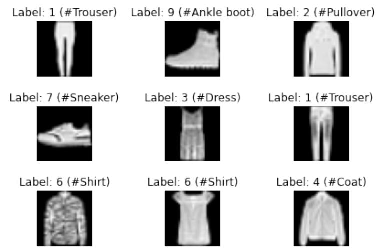
Figure 4: Problem 1 Estimated Silhoutte

- **Why is this the case?.**
  This could be because space carving is good for noisy or incomplete data. Simple techniques may fail in the presence artifacts, occlusions or other factors whereas space carving is more robust in this sense though it has its own limitations.
- **What happens if you reduce the number of views?.**
  When the number of views reduces, we end up with a more conservative set of voxels and our resulting model has voxels that do not belong to actual object
- **What if the estimated silhouettes weren't conservative, meaning that one or a few views had parts of the object missing?**

If the estimations were not conservative and a few views had parts of the object missing, then those parts will be carved out of of the final model.

2 Representation Learning [15 points]

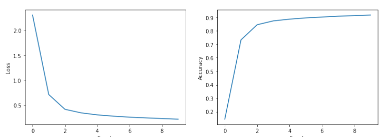
a) Include the 3 by 3 grid visualization of the Fashion MNIST dataset.



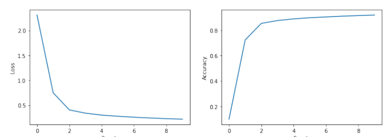
(a) Fashion MNIST data

Figure 5: 3 by 3 Grid visualization

b) Once you finish the section ”Training for Fashion MNIST Class Prediction”, include the two graphs of training progress over 10 epochs, as well as the test errors. [5 points]



(a) Training over 10 epochs

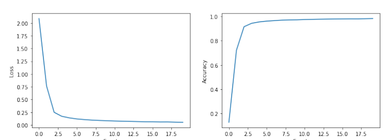


(b) Training smaller dataset

Figure 6: Training Errors

- Test Accuracy: 0.9
- Test Accuracy smaller dataset: 0.898.

c) Once you finish the section ”Representation Learning via Rotation Classification”, include the 3 by 3 grid visualization and training plot, as well as the test error [5 points].



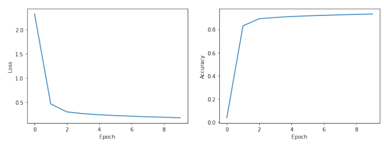
(a) Rotation Classification Dataset

Figure 7: Rotation Classification

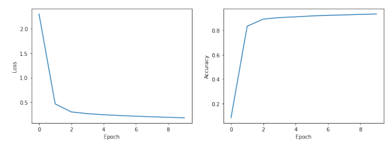
- Test Accuracy: 0.983

d) Once you finish the section "Fine-Tuning for Fashion MNIST classification", include all 3 sets of graphs from this section, as well as the test errors. Why do you think learning to un-rotate images works as a method of representation learning?.

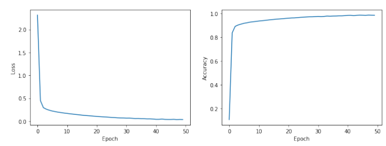
By un-rotation an image, the model learns robust features of the image that are important for manipulating data. Such features may include spatial awareness, geometry, relationships between objects, shape and color etc.



(a) Training large dataset



(b) Training smaller dataset



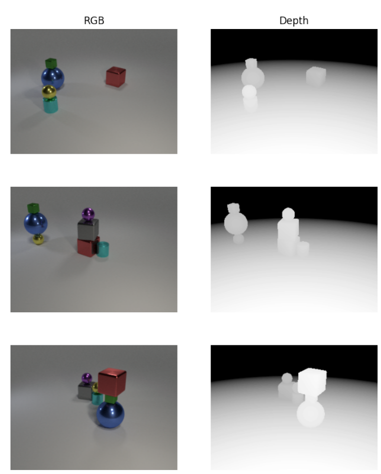
(c) Training smaller dataset longer

Figure 8: Training Errors

- Test Accuracy: 0.910
- Test Accuracy smaller dataset: 0.907.
- Test Accuracy smaller dataset longer: 0.906.

3 Supervised Monocular Depth Estimation [15 points]

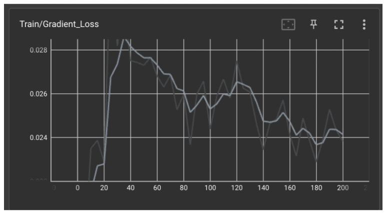
a) Once you finish the section "Checking out the data", include the grid visualization of the CLEVR-D data [5 points)].



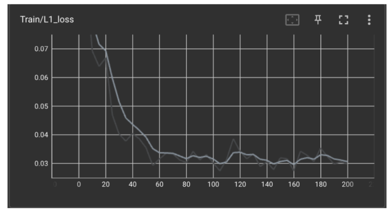
(a) Training large dataset

Figure 9: Checking out Data

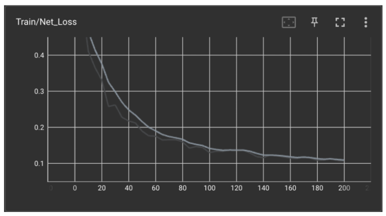
b) Once you finish the section "Training the model", include a screenshot of your train and test losses from Tensorboard as well as the final outputs of the network [10 points].



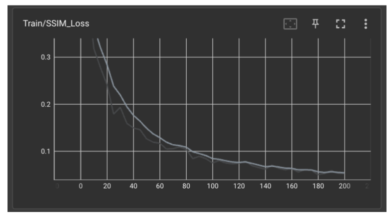
(a) Gradient Loss



(b) L1 Loss

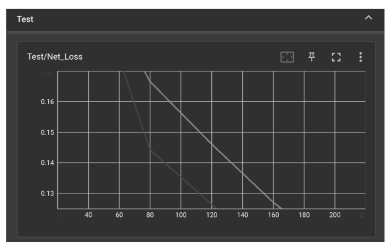


(c) Net Loss



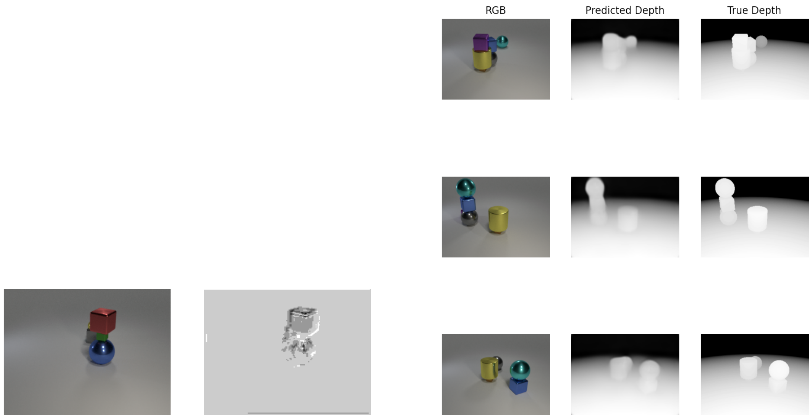
(d) SSM Loss

Figure 10: Training Losses



(a) Test Loss

Figure 11: Test Losses

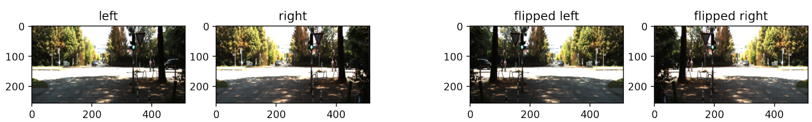


(a) RGB to Depth (b) Final Out

Figure 12: Final Output

4 Unsupervised Monocular Depth Estimation [25 points]

a) In your report include the images generated by this part of the code (no need to include the input images). [5 points]



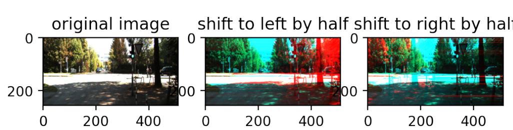
(a) Original Images (b) Augumented Images

Figure 13: Data Augumentation

b) Can you think of any other techniques we can use to apply data augmentation for this task? [5 points]

- We can also vary the brightness and intensity of the images

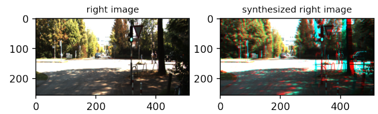
c) Implement a function bilinear sampler which shifts the given horizontally given the disparity. The core idea of unsupervised monocular depth estimation is that we can generate left image from right and vice versa by sampling rectified images horizontally using the disparity. We will ask you to implement a function that simply samples image with horizontal displacement as given by the input disparity. In your report include the images generated by this part of the code (no need to include the input images). [5 points]



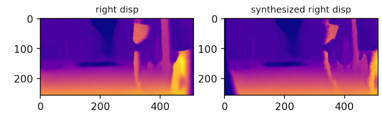
(a) output bilinear sampler

Figure 14: Bi-linear sampler

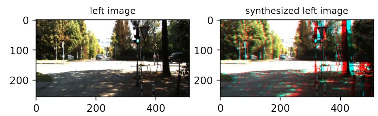
d) Implement functions generate image right and generate image left which generates right view of the image from left image using the disparity and vice versa. This will be a simple one-liner that applies bilinear sampler. In your report include the images generated by this part of the code (no need to include the input images). [5 points)]



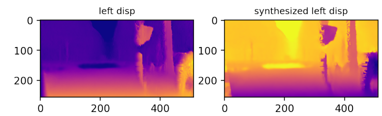
(a) Right Synthesized



(b) Right Synthesized Disparity



(c) Left Synthesized



(d) Left Synthesized Disparity

Figure 15: Synthesized Images

e) In Figure 5, we visualize output of the networks trained with the losses you have implemented. You may notice that there are some boundary artifacts on the left side of the left disparity and right side of the right disparity. Briefly explain why these artifacts may exist. [5 points]

This is due to the limitation of stereo vision. Stereo vision works by trying to identify corresponding points in the 2 images but due to things such as occlusion or light intensity variations we may not get accurate correspondence near the boundary which can lead to artifacts and missing depth information.