Towards Neural Scene De-Rendering: Investigating Three-Dimensional Object Detection and Fast Scene Understanding with Generative Models

Ryan Chen
Department of Computer Science
Stanford University
rjc45@stanford.edu

Kavita Selva
Department of Computer Science
Stanford University
kselva@stanford.edu

Abstract

Holistic scene understanding given images has been a popular problem in computer vision, especially given its plethora of useful and intriguing applications like image editing, captioning, and inpainting. Previous research has used neural network architectures to convert images into compact representations that can be used to encode information about images. However, in order to move towards better understanding of scenes, which is more difficult with the more abstract representations created by typical neural-network-based encoding approaches, work has been done -- especially in the two-dimensional case — to use neural networks to convert images into representations that are more easily interpretable for rendering engines. However, making these interpretable representations in the three-dimensional case is much more complex due to the difficulty of recovering fine details about three-dimensional scenes. This would require us to reason about the geometric relationship between different objects in the scene, and to acquire three-dimensional information from the original two-dimensional image. Thus, for this final project, we decided to reduce the scoping of this ambitious task to better understand the problem space and relationships between two-dimensional and three-dimensional information. First, we looked into some of the related work that had been referenced in the original Neural Scene De-Rendering work posed by Wu et al., in particular Attend, Infer, Repeat — sequential generative models for image recognition and synthesis. We investigated both a Multi MNIST dataset, and started to investigte the Fashion MNIST dataset as well. We also wanted to better understand the three-dimensional datasets through the lens of object detection, since particular object detection models would output information that is similar to those used in the Scene XML encodings presented by Wu et al. We used the fridge dataset and Fast-RCNN models to perform object detection, and found that these were highly effective for multiple objects and obstructions.

1. Introduction

In this research project, we decided to investigate Neural Scene De-Rendering. In this problem, we wanted to look into a method for "holistic understanding" of scenes, namely by transforming an input image into a structured scene represented in XML, which has high interpretability (in contrast to image representations learned by neural networks). This problem is interesting because it presents a method for visual understanding that can be applied to a variety of use cases, like image editing, image captioning, and inpainting. This is particularly useful in that with a compact, expressive, and interpretable representation that can be automatically generated for images, we can more easily reason about and potentially reconstruct elements of an image.

Traditional image representations learned by neural networks are indeed compact, but they are frequently hard to interpret in ways that are meaningful for human understanding and manipulation. Current work on creating interpretable representations for images by using neural networks mainly focus on the case where there is an unobstructed object in front of a clear and clean background, but these methods are not nearly as effective on cases that involve multiple objects and more complex backgrounds. Moving towards better understanding these more complex image situations will help guide us to future neural scene de-rendering for these difficult, three-dimensional cases.

However, through the course of our project, we found that this task was incredibly complex, especially for the three-dimensional case. This complexity follows directly directly from the difficulty of recovering the fine details in three dimensional scenes. We would need this three-dimensional information in order to properly train a neural network, in particular, the generalized encoding-decoding structure that was used by the authors of the original Neu-

ral Scene De-Rendering paper. Without this information, this task was too hefty for the scope of this project, and would require reasoning about the three dimensional scene given the two dimensional cues that we have access to. So, in order to tackle this problem, we decided to pinpoint our project on previous work and investigate some of the related research that has been done, as well as working with datasets that we found interesting for these applications.

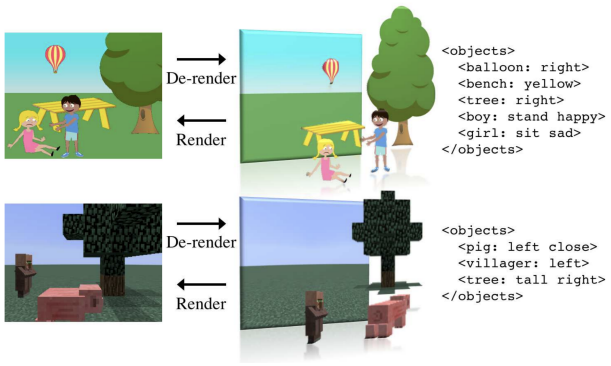

Figure 1. The Neural Scene De-Rendering paper presented by MIT CSAIL and Microsoft Research can derender images into a Scene XML representation, which can be used by a graphics engine to reconstruct or rerender the image [5].

In particular, we decided to look further into one of the related works referenced by the Neural Scene De-Rendering research by Wu et al. [5] – Attend, Infer, Repeat: Fast Scene Understanding with Generative Models presented by Ali Eslami et al. [2]. Wu et al. mention this work as an example of closely related research on sequential generative models for image recognition or synthesis that typically involve recurrent networks, training a renderer simultaneously. Wu puts their work in contrast, in that their rendered images are created not by a trained approximate renderer, but instead directly by a real renderer that uses the generated interpretable representation of the image. In order to better understand the other options when rendering images, we decided to look further into Ali Eslami et al.'s work and use the Attend, Infer, Repeat on datasets that we were interested in. As we will discuss in later sections, we mainly investigated strictly 2-dimensional data, such as multiple MNIST digits and the Fashion-MNIST dataset.

However, we were also interested in three-dimensional data to gain a better understanding of work that has been done with three-dimensional object positions, transformations, and obstructions, since the MNIST datasets didn't incorporate three-dimensional objects. So, we also researched into object detection methods for three-dimensional datasets that we were interested in. We did this because similarly to the Scene XML that was encoded by the results of the Neural Scene De-Rendering work, certain object detection models also encode information about category, size, and positioning with the bounding boxes given by the outputs. There is clearly a long way to go for our initial task of interest in neural scene de-rendering for the three-dimensional setting, but we think that this investigation into both previous work with image recognition and three-dimensional multiple object detection definitely strengthened our understanding of the problem setting and the reasons for why this particular task is so difficult.

2. Background/Related Work

2.1. Natural Scene De-Rendering

As previously mentioned, most prior works on Natural Scene De-Rendering focus on 2D vision settings due to the difficulty of recovering the fine details of 3D scenes. In theory, the beginning steps of applying this process to 3D settings are when derendering each object in the scene, one could reason about their geometric relationship from the 3D information acquired from the 2D image. The original work presented by MIT CSAIL and Microsoft Research, uses the two-dimensional the Abstract Scene dataset and their created Minecraft images dataset. They present a generalized encoding-decoding structure, in which a neural network encodes an image input into a compact and informative representation.

However, in contrast to a neural decoder, Wu et al. present the usage of the graphics rendering engine as a generalized decoder: in contrast to a neural decoder, the generalized decoder in this format requires the interpretable image representation as input in order to reconstruct/render the image. As previously mentioned, Wu et al.'s work on neural scene de-rendering focus on the two-dimensional case, and work with the Abstract Scene dataset as well as their handcrafted Minecraft scene dataset – which does include a bit more depth in the scene, but still involves a 2-dimensional image. In order to learn more about how this process works, we look towards some of the related works that can be reproduced within the scope of this class.

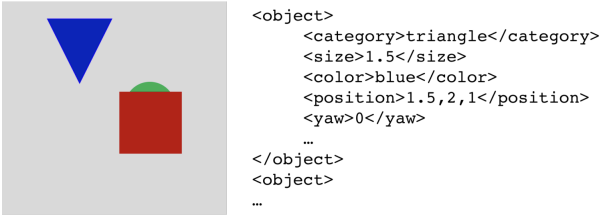

Figure 2. An example of an image and its corresponding Scene XML, which encodes information about the object, including pose, position, yaw, color, size, and category [5].

In addition, Wu et al. introduce Scene XML, which they

describe as a "cross-platform structured image representation" that is generalizable to various graphics engines for image rendering. Scene XML includes information on the object including category, appearance, size, color, position in 3D space, and pose. The Scene XML is the output of the decoder, and is subsequently translated into a format that is understandable by graphics engines for re-rendering.

2.2. Attend, Infer, Repeat

The research presented by Wu et al. builds off of related work on sequential generative models for image recognition and synthesis, many of which use recurrent networks such as LSTM. As previously mentioned, the difference between this previous work and Wu et al.'s approach is the decoder; a neural network was trained in earlier approaches to approximate the graphics renderer, while in Wu et al.'s work the output of the encoder in Scene XML was to be used in a real graphics renderer itself. We wanted to better understand this previous work and thus looked further into one of these generative model approaches for fast scene understanding — in particular, Attend, Infer, Repeat, as presented by Ali Eslami et al.

In this work, Ali Eslami et al. present a framework for efficient inference in structured image models by conducting probabilistic inference using a recurrent neural network that is able to process individual elements of a scene one at at time [2]. The model is able to find the correct amount of inference steps to take, and is able to learn how to identify, count, and locate, multiple objects in a three-dimensional scene using this neural network-based architecture.

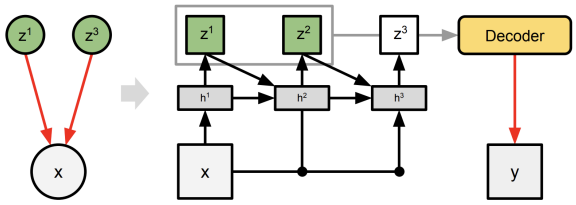

Figure 3. The general framework proposed by the Attend, Inference, Repeat methodology; for some image of interest, the inference network as represented by black arrows focus on one object at a time, to train the inference network jointly with the Fast-RNN model[2].

Attend, Inference, Repeat, or AIR, can learn to reconstruct images. It is a Variational Autoencoder (VAE) construction, and focuses on interesting parts of the image one-by-one. It looks at the image, figures out parts of the image it wants to focus on, and then reconstructs the image by putting back each part onto an initially blank canvas one at a time. AIR attends to a part of an image, effectively cropping it, infers the variables that best describe this crop, and then repeats this procedure for each part of the rest of the image [2]. It is an autoencoder, that uses an encoder to transform the image into some representation, and uses the Recurrent Neural Network architecture (RNN) to keep track of the hidden state since the image is "looked" at multiple times. In this hidden state, information such as the appearance, location, and presence of the object is latently encoded.

Overall, recreating and analyzing this framework will hopefully give better insights into the benefits and drawbacks of using this type of framework for reconstruction of images, as compared to the neural scene de-rendering approach of rendering based on the encoded Scene XML.

2.3. MNIST Datasets

The approach presented by Ali Eslami et al. as described above was tested on the MNIST dataset, particularly multiple MNIST digits. The MNIST dataset contains binary images of handwritten digits, commonly used for training image processing systems [1]. In the multiple MNIST dataset used by Ali Eslami et al, each image contained zero, one, or two non-overlapping random MNIST digits with equal probability.

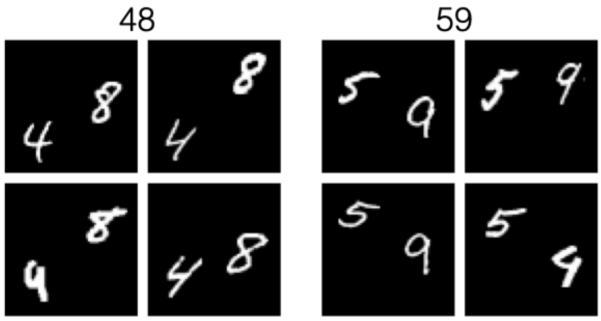

Figure 4. Example of data in the MNIST datasets, the two number versions include two digits selected at random, specific data/handwriting selected from the original MNIST dataset [1]

Figure 5. Example of data used in the Attend, Inference, Repeat paper: their custom Multiple MNIST dataset samples unobstructed MNIST digits – either 0, 1, or 2 – from the original dataset, with equal probability of number of digits present in each data image.

In our research, we planned to use this dataset as a base-

line to make sure our implementation of the model was consistent with the original results. However, this particular dataset is not quite as interesting for the three-dimensional case that defines our broader problem space. Therefore, we supplement our work by doing additional testing with other datasets that involve more complex objects as well, such as the Fashion NMIST dataset. In the future, we'd also like to experiment with more three-dimensional based datasets as well.

2.4. Object Detection

The last part of our project that we wanted to experiment with was work that had been done in the Object Detection space. Inspired by the Scene XML format that included information on the position and size for encoding objects in the image, we thought it would be interesting to additionally explore methods for object detection on datasets that we were interested in.

Work on Object Detection, in contrast to Object Classification, particularly involves the drawing of bounding boxes around objects of interest. In particular, Ross Girshick et al. proposed R-CNN, or Regions with CNN Features, in which about 2000 region proposals are extracted from an input image, warped into a square, and fed into a CNN to produce a feature vector as output [3]. However, this method still takes a long time to train the network, since 2000 region proposals per image need to be classified in training.

Girshick solved this issue in a follow up methodology called Fast R-CNN, in which instead of using the region proposals as inputs to the CNN, the input image is fed itself into the CNN to create a convolutional feature map, from which we identify proposals to warp into squares. Now, the convolution operation is only done once per image, not 2000 times per region in the image.

An even faster method called Faster R-CNN proposed by Shaoqing et al. performs object detetion that instead of using selective search to find the region proposals, has a separate network learn which regions to propose for the convolution [4].

As we will later elaborate, in our experiments, we decided to investigate Faster R-CNN in particular for object detection, and apply it to datasets in three dimensional space – both those with individual objects (one in each image) and with multiple objects in an image in different positions.

3. Approach

Our initial plan for our project was to investigate the existing implementation proposed by MIT CSAIL and Microsoft Research, in which an autoencoder encodes an image input into a compact representation and a graphics engine is used as a decoder, which allows the generalized autoencoder to naturally learn to encode the image in an "interpretable image representation" (as opposed to a hard-to-interpret representation as learned by a neural decoder). Initially, we planned to modify this implementation by trying to apply it to a three-dimensional case.

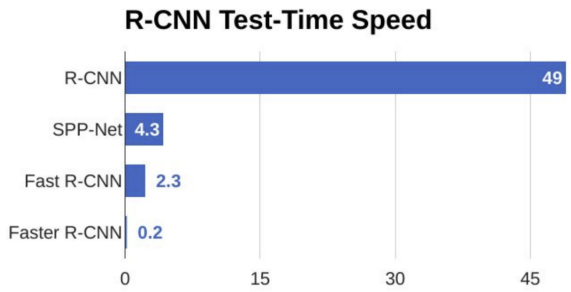

Figure 6. Comparison of different object detection algorithms with respect to training time; Faster R-CNN is much faster than the other predecessors.

However, as mentioned in previous sections of our report, we realized that the scope of this project to be used in the three-dimensional case was a task that was too far out of our scope of understanding for what we learned in this class; it is a difficult task that is an active research topic as of current. As such, we pivoted from our initial goal of performing neural scene de-rendering in the three-dimensional space, and instead decided to investigate a relevant related work – in particular, the Attend, Infer, Repeat method for image reconstruction.

3.1. Attend, Infer, Repeat

We investigate the AIR methodology with both the original dataset in the Ali Eslami et al. paper (the Multiple MNIST dataset) as well as applying the procedure to a different dataset that we were interested in, in particular, the Fashion-MNIST dataset that we looked at in class.

As described by Ali Eslami et al., the AIR method takes a Bayesian approach to scene interpretation: the task of scene interpretation is treated as inference in a generative model. Concretely, given some image x and a model $p^x_\theta(x|z)p^z_\theta(z)$ parameterized by θ, the goal is to recover a scene descriptor z by computing the posterior $p(z|x) = p^x_\theta(x|z)p^z_\theta(z)/p(x)$ [2]. Overall, the scene can be decomposed into various objects, which are called groups of variables z^i, in which each group describes some attribute of an object, such as pose or type. Broadly, this approach can model how a scene description can be rendered to an image, using our assumptions about the underlying scene captured in the prior $p^z_\theta(z)$, and the likelihood $p^x_\theta(x|z)$ of how the scene description can be rendered to create an image [2].

From here, the parameter θ from the model and ϕ from the inference network (approximation to the true posterior) can be optimized using the likelihood of the image under the model. Further details can be found in the Ali Eslami et

al. paper [2].

3.2. 3-Dimensional Object Detection

The next part of our research for this project was to investigate object detection, since we hoped that this would help give us more insight into working with the three-dimensional space and positionings for singular and multiple objects. In particular, we experimented with Faster R-CNN on two datasets that we were interested in, namely an American Sign Language (ASL) dataset that provided pictures of hands in particular letter gestures, as well as a Fridge dataset that provided different bottles and cartons in various configurations, positions, and transformations for the multiple object case.

Faster R-CNN, as proposed by Ren et al., consists of two modules: first, a deep convolutional network to process regions, and a Fast R-CNN detector that uses these proposed regions, coming together to create a fully unified object detection network [4].

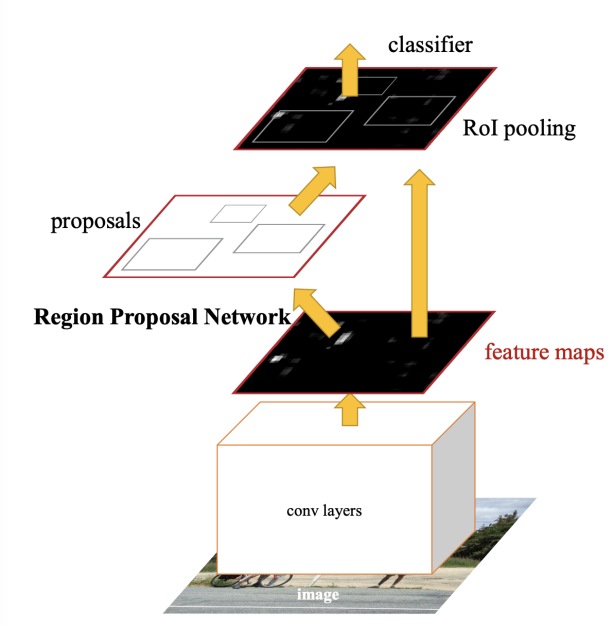

Figure 7. Visual representation of Faster R-CNN, which effectively functions as a single network with both the RPN module and the Fast R-CNN module [4].

The Regional Proposal Network (RPN) takes an image as an input and outputs rectangular "object proposals" that are assigned "objectness scores", from which region proposals are created using a network over the generated convolutional feature map outputted by the last layer of the shared convolutional network between the RPN and the Fast R-CNN objection detection network; more extensive details can be found in the Ren et al. paper on Faster R-CNN [4].

4. Experiment

With both of these parts of our project implemented, we aimed to get a better understanding of both image reconstruction tasks and the three-dimensional positioning and identification tasks that overall would make us more comfortable with working with the three-dimensional case in future applications to work towards Neural Scene De-Rendering in the 3D space. We will go into the experimental details of each of these tasks.

4.1. Attend, Infer, Repeat Experimentation

In the Attend, Infer, and Repeat experiments, we first use the Multiple MNIST dataset, which consists of multiple MNIST numbers in a single image that are not obstructing one another, with either 0, 1, or 2 numbers in the image, chosen at random. We were then interested in applying the model on a different dataset that we were interested in, the Fashion-MNIST dataset, which consists of a training size of 60K images and a test size of 10K images. In order to measure the quality of our results, we use count accuracy. On the mutliple MNIST dataset, we were able to achieve a count accuracy of around **98%**.

We found that using the Attend, Infer, Repeat method that image reconstruction can be effective in simple cases, in this case for digits. We are still working on applying these methods to a different, more complex dataset, namely continuing our work with the Fashion-NMIST dataset, which we began quite late into the project process. We are currently in the process of applying AIR to this dataset, but the inference process takes many hours to run which set us a little behind track since we made this pivot so late into our project process. We hope to achieve additional results on this dataset in the future.

4.2. Object Detection Experimentation

For the Object Detection experiments, we initially worked with an American Sign Language dataset, which consists of about 1600 labeled images. For this task, we used a pretrained Faster R-CNN model that was trained on COCO, which contains over 200K labeled images and 80 categories for the labels. We finetuned this Faster R-CNN model on our ASL dataset, with 10 epochs and a learning rate of 0.005, and got an average precision of **0.863** and average recall of **0.890**.

Overall, the model does a very good job of correctly labeling hand gestures as their corresponding ASL letter, clear from the qualitative and quantitative results. However, we were also interested in the case where there were multiple objects in a three-dimensional environment. So, we worked with the fridge objects dataset, which involved multiple fridge-related objects like bottles and cartons in different transformations, sometimes partially occluding one another from the view of the camera. Again, we fine-tuned the

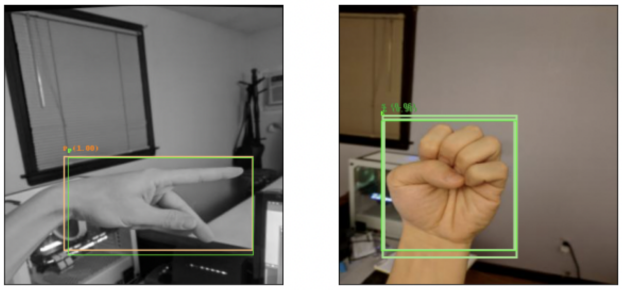

Figure 8. Accurate identification and labeling of American sign language hand gesture to the corresponding letter by the model, bounding boxes for the object detection also appear visually accurate.

Faster R-CNN model with a variety of parameters, and the results were once again quite successful, as we can see in the information below.

Parameters	Average Precision	Average Recall
LR 0.01, 10 Epochs	0.903	0.920
LR 0.005, 10 Epochs	0.901	0.929
LR 0.005, 5 Epochs	0.905	0.920
LR 0.001, 10 Epochs	0.914	0.930

Table 1. Results for a variety of parameters for epochs and learning rate for fine-tuning the Faster R-CNN model on the fridge objects dataset.

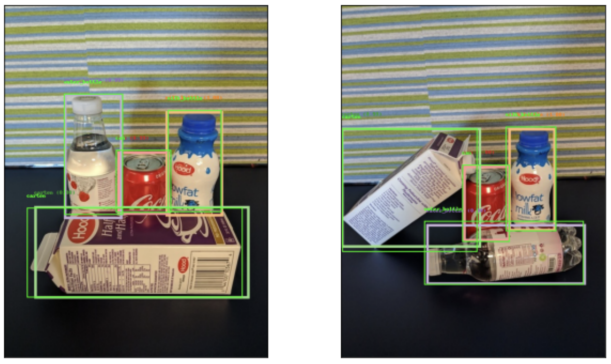

Figure 9. Accurate identification and labeling of the fridge dataset objects by the model, bounding boxes for the object detection also appear visually accurate despite obstructions of objects in front of or behind one another.

With these results, we find that it is possible to accurately output bounding boxes for objects in three-dimensional space, even when objects are in front of or on top of one another, which is interesting to confirm geometrically. What becomes difficult is that we are unable to find from these raw results the three-dimensional information about

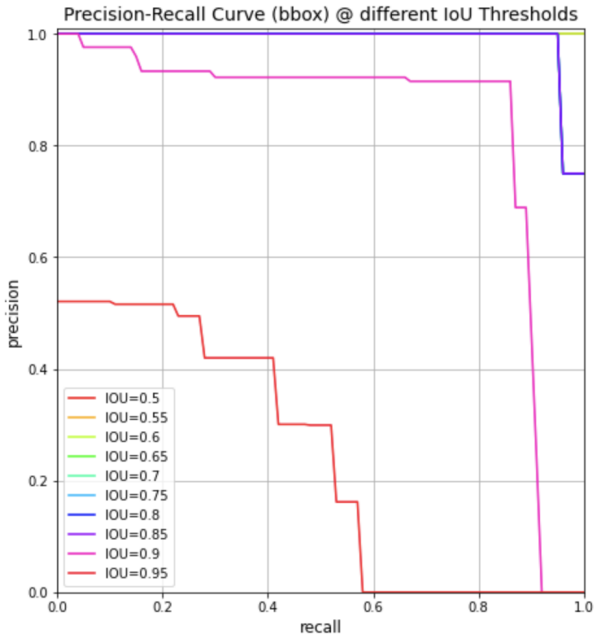

Figure 10. Precision-recall curves as a function of the IoU (Intersection over Union) threshold that describes the accuracy of the bounding box for object detection; lower IoU (higher leniency) corresponds to higher Precision-Recall curve, as expected.

depth of positioning, which is the main goal of the three-dimensional Neural Scene De-Recognition case. Future work could involve using the two-dimensional information that we were able to extract from this detection, and potentially information about the known size of these objects, to help guide the three-dimensional case.

5. Conclusions

Through this project, we were able to gain a lot of exposure to various initiatives towards understanding objects in three-dimensional space. First, by applying Attend, Infer, Repeat to both the Multiple MNIST and Fashion-MNIST datasets, we were able to gain exposure to relevant work in image reconstruction that can be directly compared to and give us insight into the work being done in Neural Scene De-Rendering into more meaningful representations.

In the future, we'd like to explore even more datasets for both applications, in particular three-dimensional datasets with multiple objects that could be occluding one another for the Attend, Infer, Repeat case. We would also like to use what we learned to replicate the code for the Neural Scene De-Rendering experiments, potentially with our own two-dimensional datasets, like the websites idea that was proposed by the course staff. We would also like to continue to explore Faster R-CNN with more datasets that we're interested in, in particular more robust/larger datasets with a

greater variety of objects. Finding such a dataset was difficult, and in the meantime we would like to continue working on tuning the Faster R-CNN model to other applications that we find interesting.

The relevant code for our project can be found here, for both tasks that we were interested in for our final report: `https://github.com/kaselva/cs231a_final_project`.

6. Acknowledgements

We would like to thank the CS 231A teaching staff for guidance on our project, especially with regards to pivoting our goals late into the project process! We hope to continue working on this project to achieve some more results that are interesting.

References

[1] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. High-performance neural networks for visual object classification. *CoRR*, abs/1102.0183, 2011.

[2] S. M. A. Eslami, N. Heess, T. Weber, Y. Tassa, K. Kavukcuoglu, and G. E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. *CoRR*, abs/1603.08575, 2016.

[3] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[4] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[5] J. Wu, J. B. Tenenbaum, and P. Kohli. Neural Scene De-rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.