# Dirty Room Notifier

## Final Report

Gonzalo Martinez Santos - 1004948300
Mobin Malmiran - 1005023608
Peter Sorbo - 1004109975

Word Count: **1987**

## Introduction

Goal:
Develop a classifier that will determine whether a room is clean or dirty. Use image segmentation to locate where the dirt (or garbage) is in the input image.

Why:
The product will help Airbnb hosts and people managing properties ensure that their cleaning staff is doing a good job.

Motivation:
Peter is working on a Startup associated with Airbnb where he develops a chatbot for clients to ask questions about certain houses that they are interested in renting. From what he has learned, there is a surprisingly high number of cases where guests complain about arriving to garbage in certain places of the property. Therefore, this project serves as a complementary addition to Peter's startup and will allow the hosts to remotely monitor whether a room has been properly cleaned by the cleaning staff, and get notified when a room is left dirty.

Future vs Project Objective:
Our classifier will simply recognize garbage items in a room and decide whether or not the room is dirty using the input from a stationary camera. However, the long term goal for this product is for a drone or robot to capture the photos and to check smaller details around the house (such as dust). This is what motivates the objective of garbage location using image segmentation since we hope to build on this project for real uses in the Airbnb industry.

Why a Neural Network:
Will allow the property owner to be notified remotely (once the cleaning staff has left) if there is still garbage in a room. The NN can use image segmentation to identify garbage and notify the host only if needed.

## Illustration

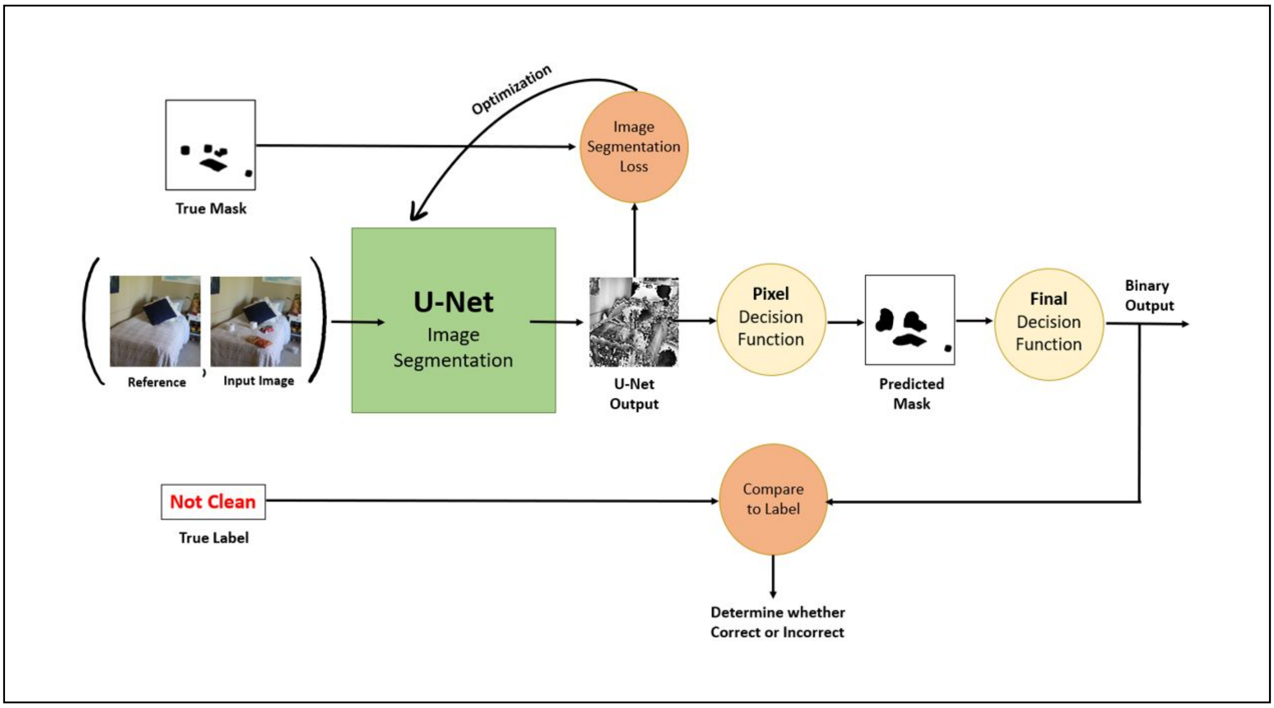


*Figure 1: Overall process of proposed model*

NOTE: The pixel decision function used to develop the predicted mask is a simple step function, and its output is passed through a final decision function whose output determines whether there is garbage and the owner should be notified or not. If any pixel is classified as garbage, the image will automatically be labelled as "dirty".

## Background and Related Work

During our initial research stage, we came across the article "U-Net: Convolutional Networks for Biomedical Image Segmentation" [1]. This group of scientists changed the traditional CNN to develop an architecture called the U-Net which achieves higher-precision image segmentation on a much smaller number of training samples. Although they used the U-Net to classify microscopic cell images, we decided to borrow this architecture since our goal is also to perform pixel classification on a limited set of manually gathered image data.

Additionally, we were very inspired by the article "Spot the difference by Object Detection", where the difference in book covers are found by using a CNN, and the input technique of aligning the two images being compared and passing this new 6-channel image through a series of convolutions which are able to find differences between the two initial pictures [2]. We considered this input method appropriate since we will always have a reference picture of how the owner of an Airbnb property wants their rooms to look like, and thus we decided to incorporate it into our model as a way to help the network in detecting unwanted garbage.

## Data Processing

We took pictures of our own rooms (6 rooms in total, each from 1 or 2 different angles), shown below. We took a clean picture of each angle and many dirty images by adding pieces of garbage in different locations of the room. We used common pieces of garbage including toilet papers, cans, food wrappers, and containers. About 250 images were gathered in total.

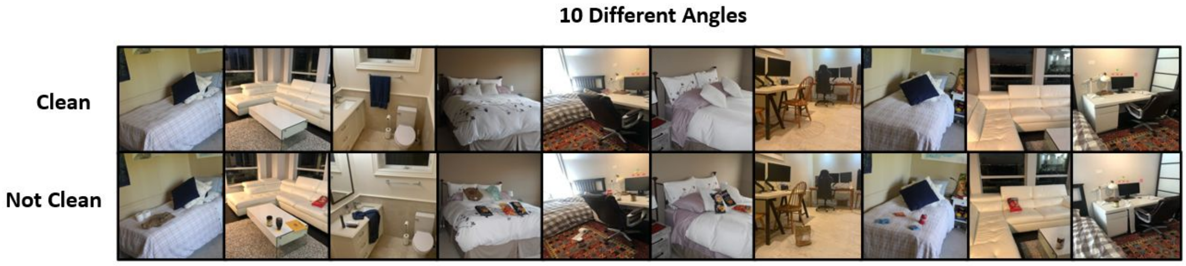


*Figure 2: 10 Different angles with the labels clean/not clean*

We used data augmentation by changing the brightness of the clean images, where 1 picture produced 18 new pictures. This helped us to increase our sample to about 450 images and balance the clean and dirty classes.

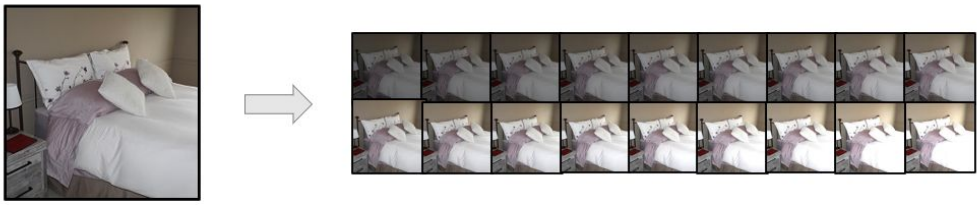


*Figure 3: 18 different inputs that we generated from on image through data augmentation by changing the brightness.*

All images were cropped and resized to 200 x 200 pixels. We then manually created ground truth segmentation masks for the "not clean" rooms by drawing over the garbage on each picture (sample masks and labels shown in Figure 4).

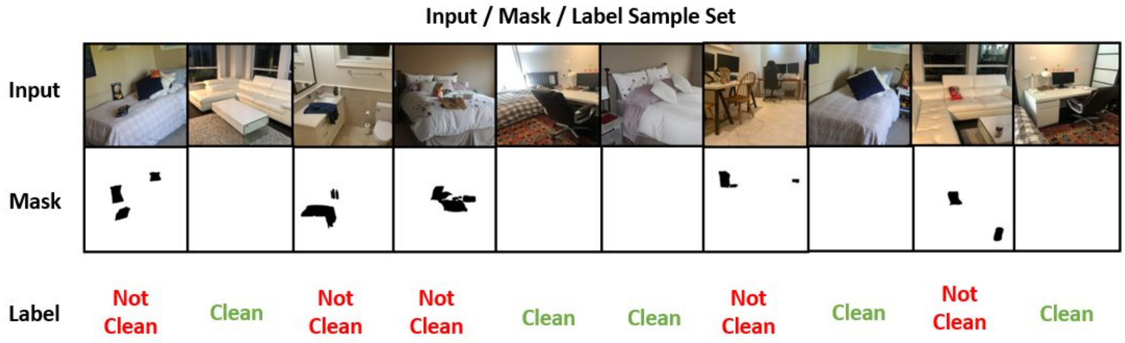


*Figure 4: The ground truth masks and labels for the inputs*

Inspired by our research, we concatenated each "not clean" picture with a reference clean image of its same angle and produced a 6-channel image that would be the input to our main model. Lastly, we split the data into 80% training, 10% validation, and 10% testing.

## Architecture

The proposed model consists of a U-Net architecture adapted to receive 6-channel image inputs and produce 1-channel segmentation map outputs as shown below.

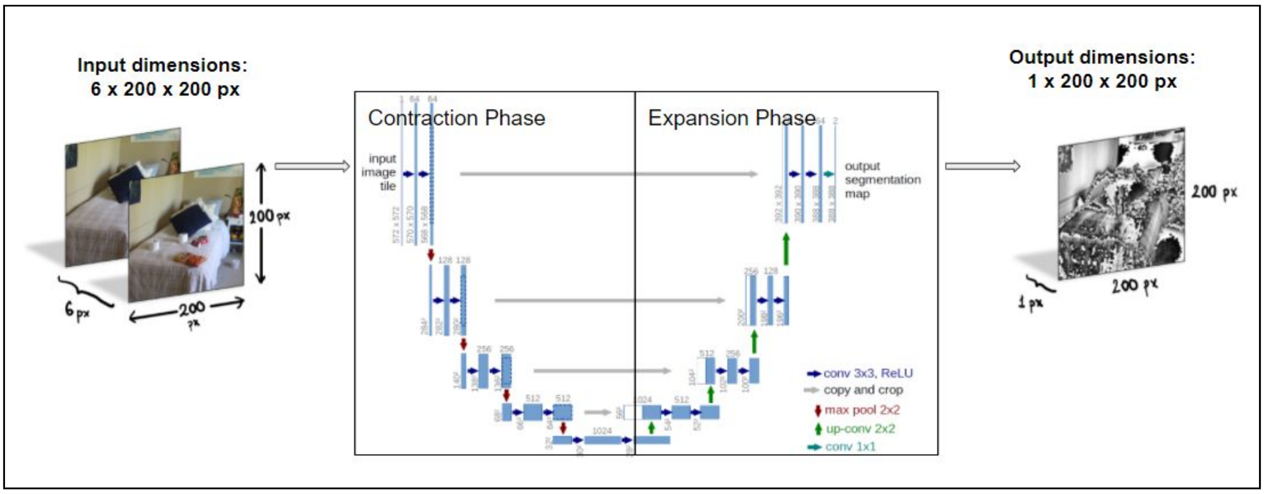


*Figure 5: Proposed model - U-Net architecture*

As described in the figure, the U-Net architecture achieves image segmentation through a series of "down convolutions" in an image contraction phase, and "up convolutions" in an image expansion phase. Specifically, this model consists of the following convolutional steps in order:

- ➔ 1 **Double Convolution** = 2 x (Convolution + BatchNorm + ReLu)
- ➔ 4 **Down convolution** layers (each containing a Double Convolution + maxpool2d)
- ➔ 4 **Up Convolutions** (each containing an Upsample, Concatenation & convolution)
- ➔ 1 **Double Convolution** = 2 x (Convolution + BatchNorm + ReLu)

The proposed hyperparameters that achieved the best performance are the following:

| Image Size | Conv. Dimensions | Batch Size | Learning rate | Epochs |
|---|---|---|---|---|
| 200 x 200 px | 64, 128, 256, 512 | 5 | 0.0001 | 90 |

## Baseline

Various baseline models were explored throughout the project. The one we chose to work with is a CNN followed by a Multi-Layer Perceptron that performs binary classification to decide whether an input image is clean or dirty. We understand that this model is not designed to perform image segmentation like our proposed model does (so the model's comparison is not exactly "apples to apples"), but it does achieve the overall and final goal of determining whether a room is clean or dirty, and thus allowing us to notify the owner of the property accordingly, which is the reason why we think this model is an appropriate baseline to contrast against the more complex proposed model.

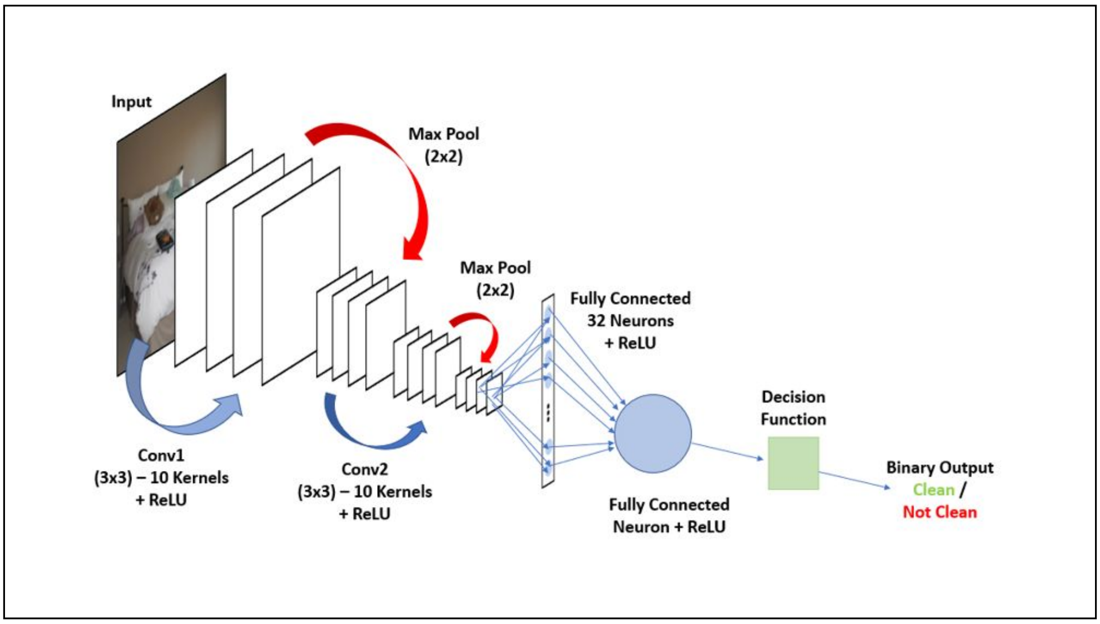


*Figure 6: CNN + MLP Baseline Model Architecture*

Recall that the image segmentation functionality was motivated by the long-term goal of using a drone/robot to find precisely where the garbage is located. Therefore, although we added pixel classification to the proposed model, the baseline simply performs the final binary classification task that we were interested in optimizing throughout this project.

Other baseline models were considered, such as Encoder/Decoder networks that are trained to receive a room image, encode it, and decode it producing a final mask with the location of the garbage in the input image. We explored the MLP Autoencoder from ECE324 Lecture 27 [3]. However, the results for this model were too poor and inaccurate to include in this paper.

## Quantitative Results

I. <u>Baseline Performance:</u>

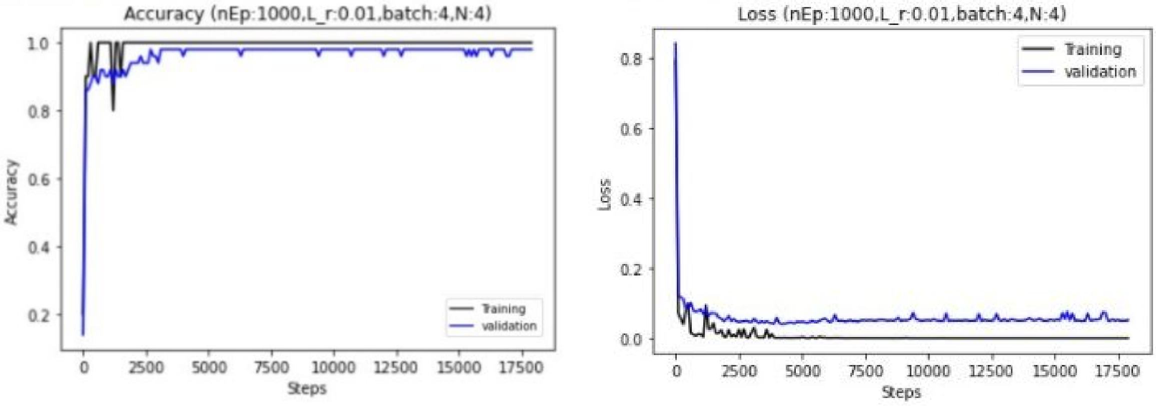


*Figure 7: Accuracy and Loss Plots for the Training and Validation Datasets using Baseline Model*

Surprisingly, our baseline model was able to achieve high accuracy in both the training and validation datasets, as shown above, as well as the following final test results:

| Baseline Test Accuracy | Baseline Test Loss |
|---|---|
| 0.967 | 0.0648 |

II. Proposed Model Final Classification Performance:

A confusion matrix was used as a metric to understand both how well our proposed model is doing incorrectly predicting whether a room is clean or dirty, and to understand where it is failing, as shown below:

| | | True | |
|---|---|---|---|
| | | Clean | Dirty |
| Predicted | Clean | 8 | 4 |
| | Dirty | 0 | 8 |

As we can see, we have a significant number of dirty images that were classified as being clean. This is not ideal at all. In fact, we are mainly concerned about minimizing these false negatives because if an owner's house is dirty and they are not notified, the future guests will file complaints against the owner, which is what we want to avoid. The root of these false negatives comes from the image segmentation process which is not being able to find certain garbage items. This causes the model to achieve **80% test accuracy** in the final classification stage, which is worse than the baseline. Thus, more specific segmentation metrics are necessary to look at where the model is failing to detect garbage.

III. Proposed Model Segmentation Performance:

The metric we used to measure the image segmentation performance of the U-Net was the Dice Coefficient [4], which uses the following formula in a "pixel-wise" manner:

$$Dice = \frac{2 \times Intersection}{Union + Intersection}$$

$$= \frac{2TP}{2TP + FN + FP}$$

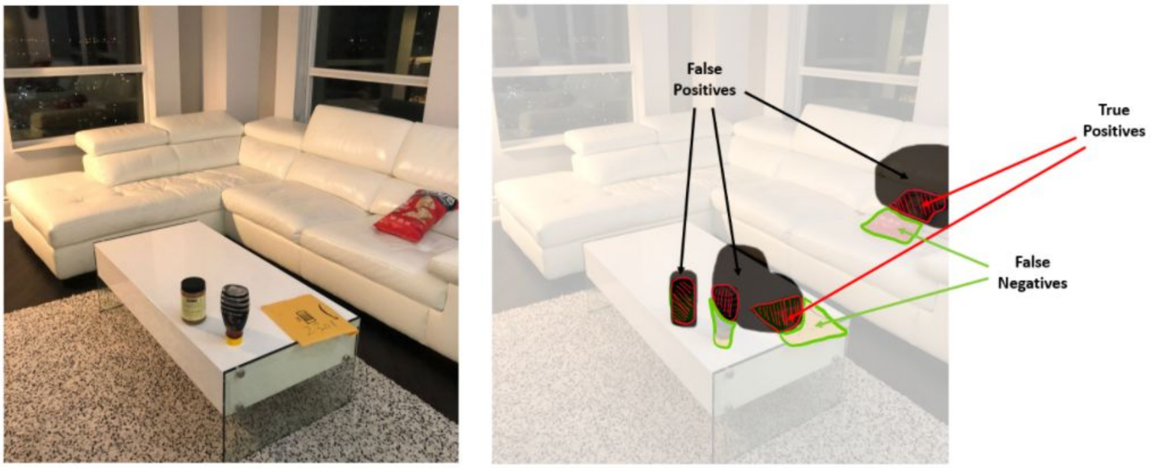
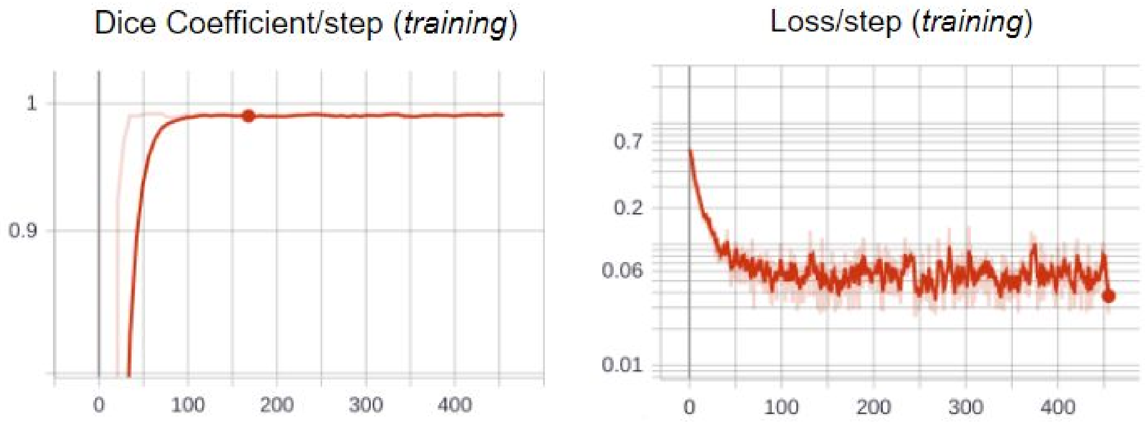


*Figure 8: Depiction of the True Positive, False Positive, and False Negative regions in an image used to find the Dice Coefficient*

The following graphs illustrate the learning process of the U-Net over each step:


Dice Coefficient/step (*training*)

Loss/step (*training*)

*Figure #: Dice Coefficient and Loss plots of U-Net over each iteration*

More specific numerical results are given in the table below:

| | |
|---|---|
| Training Loss | 0.0722 |
| Training Dice | 0.9871 |
| Validation Loss | 0.0934 |
| Validation Dice | 0.9621 |
| Test Dice (images from same distribution) | 0.8531 |
| Test Dice (images from new distribution) | 0.1456 |

As we can see, our segmentation model performed almost perfectly on the training data and decently well on the validation data, where the dice coefficients are very close to 1.0. However, the performance significantly worsens during testing on images from the same distribution. Additionally, when testing on images from a different distribution, the model completely fails to generalize.

Lastly, hyperparameter tuning was performed in order to converge on a good set of hyperparameters.

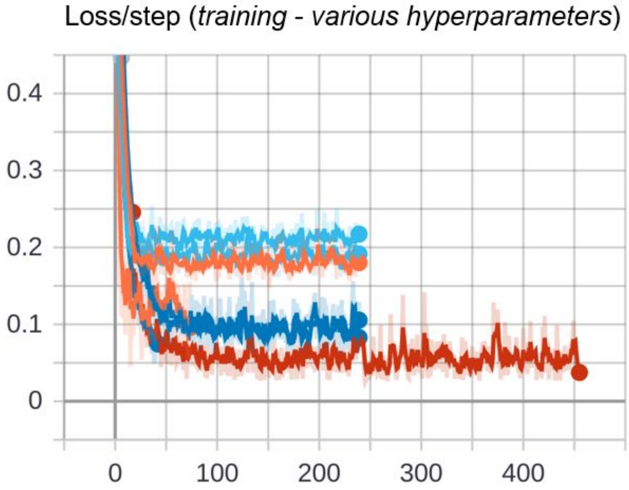


*Figure 9: Loss vs Step graph for various hyperparameters*

## Qualitative Results

The figure below demonstrates how the U-Net model overfits a small dataset very well and achieves high accuracy on the training dataset. In the first two images (dirty), the garbage is spotted almost perfectly, and the masks for the clean images are left blank as desired.

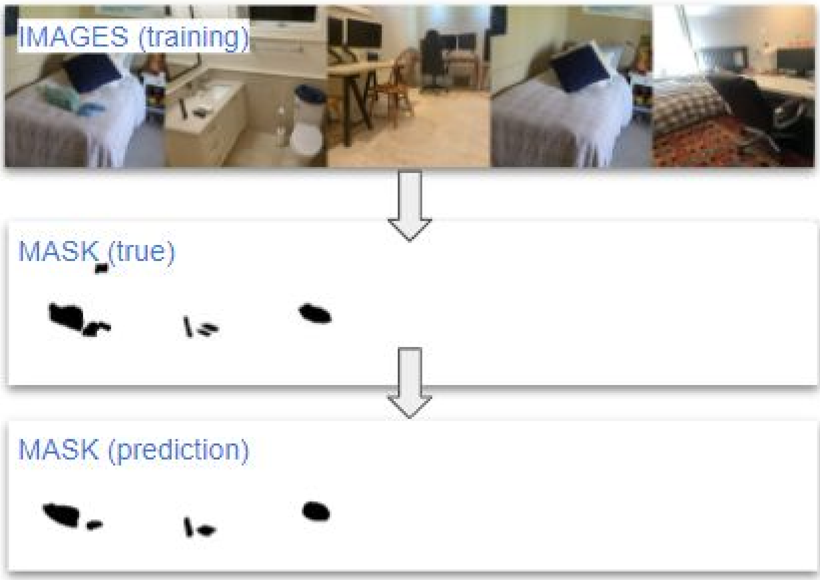


*Figure 10: Input training images alongside their true masks and their predicted masks*

Validation results were worse than training, although not substantially. Below we can see an example where one large garbage item was not recognized altogether. This also shows that even though the final prediction will be correct (i.e. classified as dirty), the U-Net is not performing as it should since it is missing full items. We noticed that low contrast objects are usually incorrectly predicted.

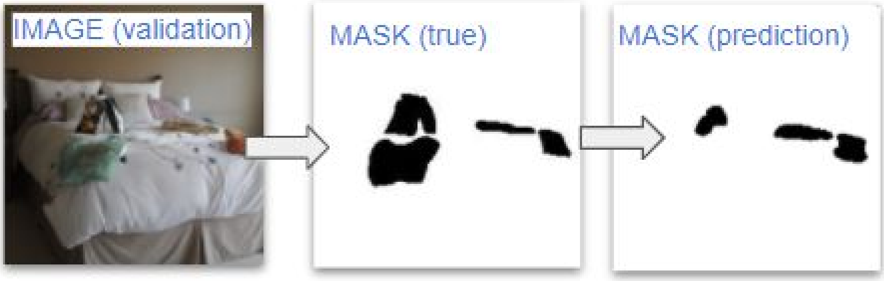


*Figure 11: Sample validation image alongside its true and predicted mask*

Lastly, we can see below that the testing masks generated were not very accurate, where many garbage items fail to be recognized.

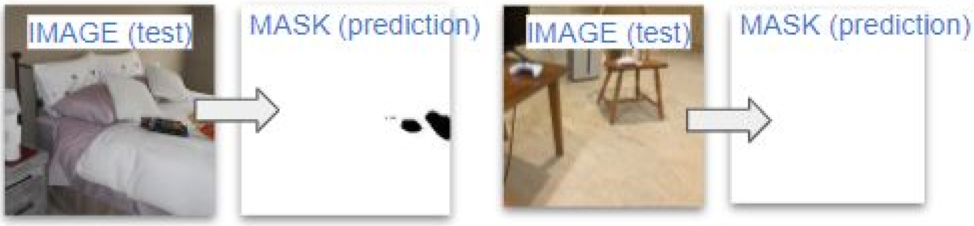


*Figure 12: Sample testing images alongside their corresponding mask predictions*

## Discussion and Learnings

Ultimately, our model was able to successfully learn to distinguish and segment garbage on the scenes it was trained on. Nevertheless, our model did not turn out to be as robust as we had hoped and has several important caveats:

- Novel camera angles performed very poorly
- Smaller, low-contrast items performed poorly
- Small shifts in furniture, doors introduced noise into the mask.

Therefore, in order for our model to perform effectively in production, a user would have to label several images from a fixed camera angle and make sure no movable furniture or doors are in the frame. While we could solve the door/furniture problem by having the user (or another network) draw bounding boxes over the image, the problem of generalizing to any scene given only a single clean reference image remains our biggest challenge.

Although we augmented images by manipulating brightness & contrast, we believe that the failure to generalize was fundamentally a problem with our training data, or rather a lack thereof at only 10 camera angles. In other words, even at the full dataset size, we run the risk of overfitting to those particular angles. Therefore, if we were to redo this again, instead of starting by overfitting our model to a single angle, we would train our model on a synthetic dataset with no unique camera angles. The simplest approach would be to overlay pictures of garbage over room images, although given more resources we could generate cohesive 3D scenes using Nvidia's Deep Learning Dataset Synthesizer. Either way, taking this approach would ensure that our model is not fitting to any one angle but actually extracting the difference between images.

## Ethical Framework

Our Stakeholders:

- Host (could be a property management team or just one person)
- Airbnb guests
- Airbnb team
- Cleaning staff
- Similar properties' hosts
- Town's tourism industry
- Local restaurants and shops

In terms of Reflective Principlism, our product would **benefit** the host when the guest gives good feedback about their pleasant experience. Also, by making sure the room is clean the host would **avoid the harm** of getting negative feedback, which could hurt their business. This solution is **just** because the host is paying for a cleaning service, and it is fair for him to know about the state of his property before renting it. Lastly, it gives hosts **autonomy** to manage their houses remotely.

The guest **benefits** from having a clean room and will **avoid the harm** of being given a poor experience during their stay. We provide a **just** solution because all guests will enjoy the clean house that they have rightfully paid to spend time in. Their **autonomy** is respected because all pictures would be taken prior to their arrival, after the cleaning staff have left, thus avoiding any privacy issues.

## References

[1] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” May 2015. [Online serial]. Available: https://arxiv.org/abs/1505.04597.
[Accessed Oct. 26, 2020]

[2] J. Wu, Y. Ye, Y. Chen, and Z. Weng, “Spot the Difference by Object Detection,” Jan. 2018. [Online serial]. Available: https://arxiv.org/abs/1801.01051. [Accessed Oct. 26, 2020]

[3] J. Rose. ECE324. Class Lecture, Topic: “Autoencoders.” Engineering Science, University of Toronto, Toronto, ON, Nov. 3, 2020

[4] E. Tiu, “Metrics to Evaluate your Semantic Segmentation Model,” *Medium*, 03-Oct-2020. [Online]. Available: https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2. [Accessed: 03-Nov-2020].