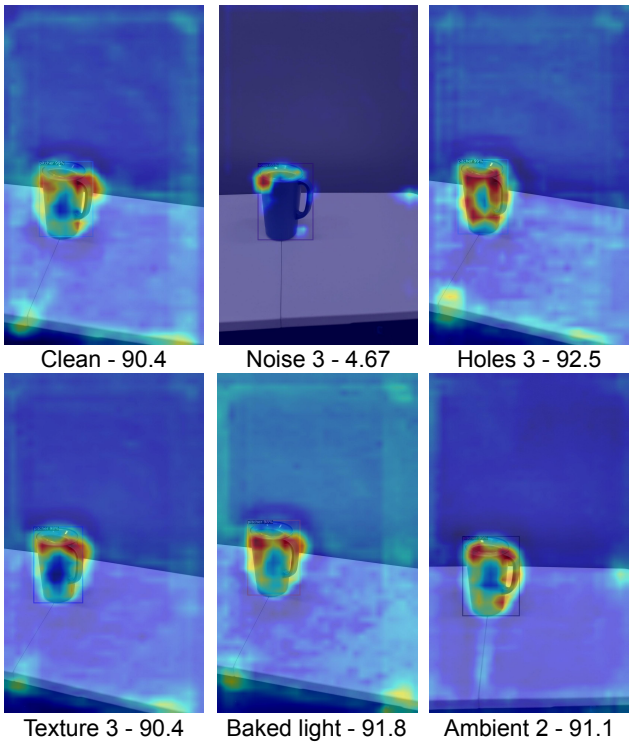
# Supplementary Materials for "Exploring the Sim2Real Gap using Digital Twins"

In these supplementary material, we show extra visualizations of our dataset and results. In addition, we include images that show the virtual and physical capture setup.

## Grad-Cam Visualization

First, we aim to understand why the model makes mistakes when trained with certain corruptions. To that extent, we produce gradcam makes which generate visual explanations for the model by visualizing the regions of input that are "important" for predictions from these models [1]. We create gradcam maps for each model trained on different corruptions in `YCB-Synthetic` and testing on pitcher in `YCB-Real`. When training on `YCB-Synthetic` noise level 3, the model receives a mAP of 4.67 on the pitcher classes and as seen in Figure 1, it only focuses on the top of the pitcher to make its decision. Whereas in clean, holes 3, texture 3, and ambient 2, it focuses on the pitcher's handle and receives a mAP of $\geq 90$. This indicates that due to the noise added when training a model on `YCB-Synthetic` noise 3, it learns to focus on the wrong part of the object for its decision making which hurts its performance greatly.

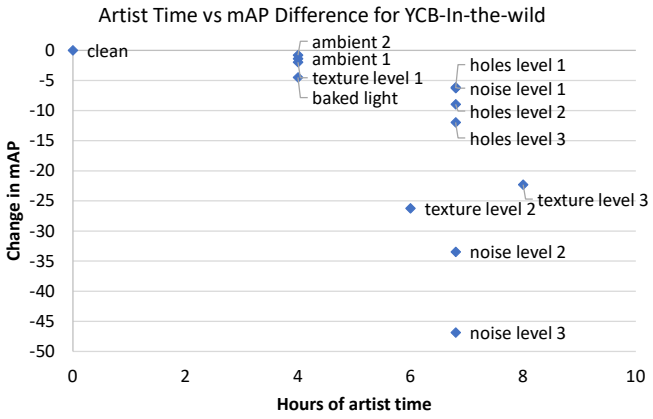
Figure 1: We visualize the gradcam maps when training models on different corruptions in `YCB-Synthetic` and testing on pitcher in `YCB-Real`. The noise in mesh trained models receives an mAP of 4.67 on the pitcher classes and it only focuses on the top of the pitcher to make its decision. Whereas in other training settings, it focuses on the handle of the pitcher and receives mAP $\geq 90$. This indicates that due to the noise added when training a model on noise in mesh, it learns to focus on the wrong part of the object for its decision making which hurts its performance greatly.

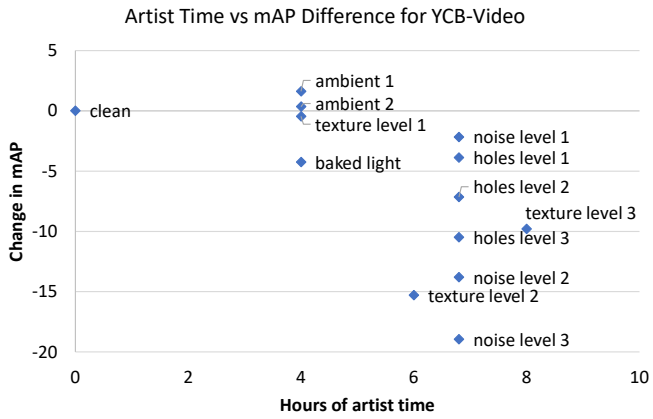


## Artist Time vs mAP Trade-offs

In the main paper, we provided a scatter plot highlighting the trade-off between how long it takes to fix each artifact and the accuracy benefit provided by that fix for object detection in `YCB-Real`. Here we provide the results for object detection on `YCB-In-the-wild`, `YCB-Video`, and segmentation on `YCB-Video` (see Fig 2).

Figure 2: For each artifact that can arise in synthetic data creation, we show the time it takes to fix it vs the drop in mAP of the model trained with that data. This provides actionable insights as to how to balance the time and cost of synthetic data generation.



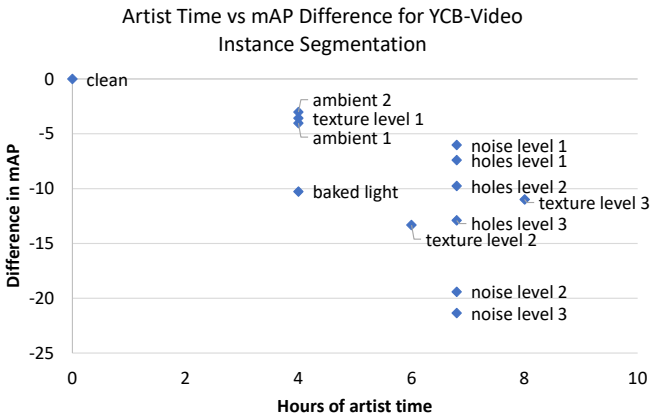
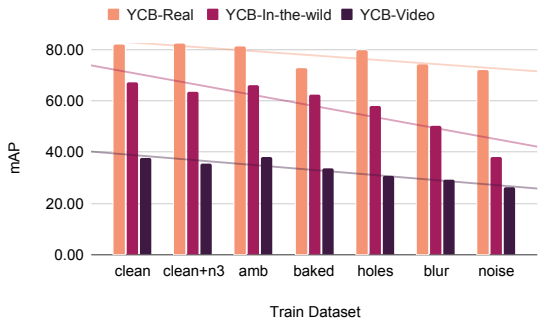



Figure 3: Results after averaging across the levels of severity of the different artifacts. Trends on performance drops due to artifacts hold quite consistently apart from baked lighting. These comprehensive results provide high-level intuition on how different artifacts impact model performance.




## Analyzing trends across datasets

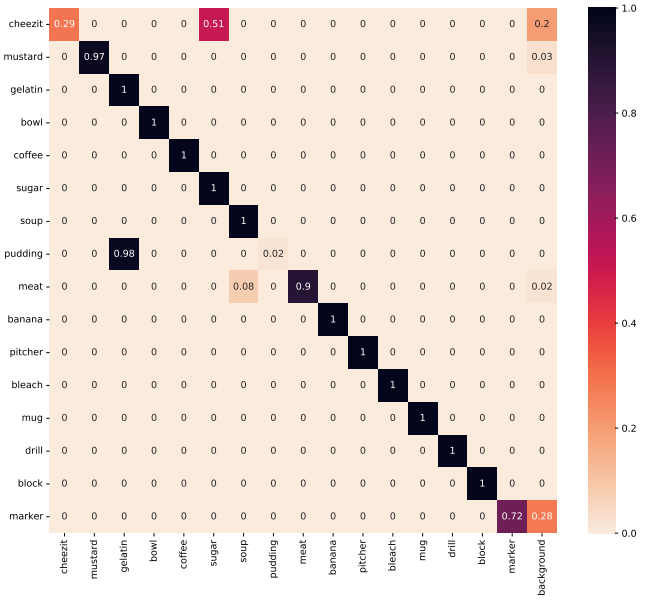
We discuss trends from `YCB-Real` results that held true in `YCB-In-the-wild` and `YCB-Video`. We average mAP across the levels of the different artifacts to get a high-level intuition on the results and present this in Figure 3. We noticed that like `YCB-Real`, changing the level of ambient lighting had the least drop in model performance from clean. Furthermore, baked lighting had a similar impact level as `YCB-Real` as seen by similar mAP values. Finally, we notice that across all 3 datasets, noise in mesh results in the greatest performance drops.

## In-Depth Per-object Analysis

We generate confusion matrices which help us understand the types of mistakes that the model makes. As highlighted in the results of the main paper, the model trained on `YCB-Synthetic` noise in mesh makes several classification errors. In Figure 4 we show that in contrast to noise in mesh, the model trained on `YCB-Synthetic` texture blurs makes several localization errors. Based on the application, certain corruptions can cause more critical mistakes than others. Further, we include standard deviations in Table 2 for all the objects in Table 1 of the main paper.

Figure 4: Confusion Matrix when training on `YCB-Synthetic` with different artifacts and testing on `YCB-Real`.

(a) Trained on `YCB-Synthetic` baked lighting




(b) Trained on `YCB-Synthetic` texture blur (averaged across level 1/2/3)

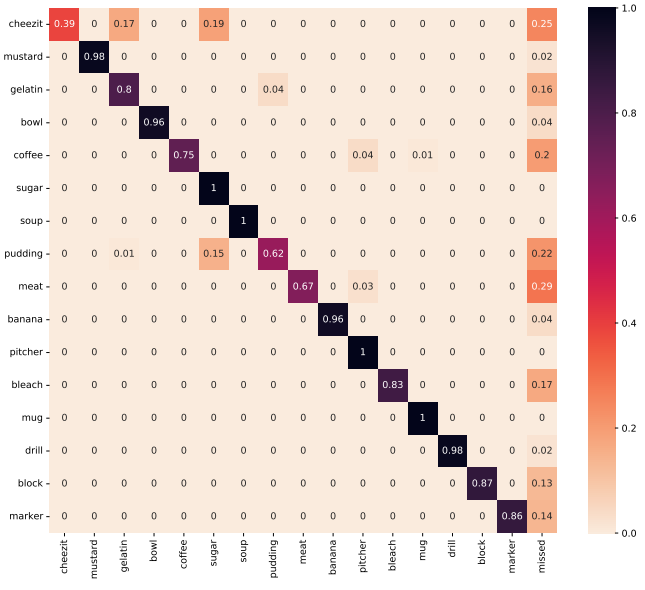


Table 1: First 8 objects - Object Detection results (mAP and standard deviation) from training on each type of artifact in YCB-Synthetic and testing on YCB-Real.

| train set | all | sugar | mug | gelatin | banana | bowl | drill | bleach | block |
|---|---|---|---|---|---|---|---|---|---|
| clean | 81.93 ± 0.98 | 86.69 ± 3.14 | 84.55 ± 1.88 | 82.14 ± 1.65 | 77.03 ± 2.21 | 86.71 ± 1.92 | 86.82 ± 1.80 | 67.84 ± 3.94 | 90.20 ± 1.89 |
| ambient 1 | 81.53 ± 0.75 | 86.51 ± 2.11 | 83.20 ± 2.97 | 80.70 ± 2.19 | 73.34 ± 3.21 | 85.50 ± 2.67 | 85.75 ± 1.81 | 74.49 ± 3.32 | 91.52 ± 2.66 |
| ambient 2 | 81.10 ± 1.08 | 85.10 ± 1.33 | 84.96 ± 1.43 | 80.01 ± 1.90 | 75.17 ± 2.95 | 85.61 ± 2.56 | 85.17 ± 1.81 | 73.03 ± 4.37 | 89.06 ± 3.65 |
| holes 1 | 78.85 ± 2.14 | 81.79 ± 6.50 | 83.14 ± 2.20 | 80.18 ± 2.38 | 75.94 ± 3.55 | 84.43 ± 2.67 | 86.76 ± 2.55 | 65.38 ± 3.20 | 91.17 ± 1.98 |
| holes 2 | 80.05 ± 1.30 | 82.55 ± 4.63 | 84.34 ± 1.87 | 81.74 ± 2.14 | 76.09 ± 3.12 | 83.65 ± 2.50 | 85.56 ± 1.79 | 63.47 ± 3.49 | 90.65 ± 3.38 |
| holes 3 | 79.99 ± 1.35 | 86.85 ± 3.54 | 85.01 ± 2.07 | 80.48 ± 1.80 | 80.03 ± 3.90 | 81.78 ± 2.58 | 86.61 ± 2.35 | 58.02 ± 3.21 | 89.98 ± 2.90 |
| texture 1 | 82.29 ± 1.11 | 86.31 ± 2.97 | 86.38 ± 1.16 | 81.81 ± 3.09 | 77.08 ± 6.90 | 87.21 ± 1.91 | 86.68 ± 1.78 | 69.72 ± 3.48 | 92.53 ± 2.57 |
| texture 2 | 69.82 ± 4.47 | 85.65 ± 2.45 | 84.78 ± 2.73 | 13.72 ± 15.77 | 78.23 ± 3.58 | 83.59 ± 2.26 | 86.20 ± 1.50 | 56.98 ± 18.26 | 87.03 ± 6.76 |
| texture 3 | 71.04 ± 4.74 | 83.47 ± 3.21 | 85.24 ± 2.36 | 74.49 ± 3.93 | 82.81 ± 2.61 | 82.56 ± 3.57 | 85.70 ± 2.60 | 63.20 ± 2.59 | 93.19 ± 2.03 |
| baked light | 72.87 ± 1.23 | 83.03 ± 2.61 | 84.47 ± 2.22 | 79.60 ± 3.65 | 70.79 ± 2.36 | 87.11 ± 2.37 | 83.60 ± 1.96 | 60.99 ± 3.27 | 90.02 ± 3.27 |
| noise 1 | 81.17 ± 1.17 | 84.81 ± 3.64 | 85.08 ± 2.89 | 80.59 ± 1.44 | 78.55 ± 2.97 | 85.29 ± 2.34 | 87.08 ± 1.20 | 65.69 ± 4.00 | 89.58 ± 6.18 |
| noise 2 | 69.20 ± 4.92 | 84.71 ± 3.28 | 36.10 ± 31.63 | 71.39 ± 9.65 | 82.41 ± 2.76 | 80.44 ± 5.89 | 81.96 ± 5.73 | 63.60 ± 2.51 | 89.52 ± 5.68 |
| noise 3 | 65.65 ± 4.84 | 83.38 ± 2.86 | 63.59 ± 27.23 | 78.77 ± 3.14 | 81.76 ± 4.19 | 68.18 ± 31.30 | 80.64 ± 12.23 | 70.31 ± 3.34 | 87.45 ± 7.38 |

Table 2: Object Detection results (mAP and standard deviation) from training on each type of artifact in YCB-Synthetic and testing on YCB-Real.

| train set | all | meat | marker | cheezit | pitcher | mustard | pudding | coffee | soup |
|---|---|---|---|---|---|---|---|---|---|
| clean | 81.93 ± 0.98 | 86.90 ± 2.30 | 38.17 ± 3.54 | 92.16 ± 1.65 | 90.44 ± 1.02 | 82.65 ± 2.53 | 83.75 ± 5.02 | 90.18 ± 3.16 | 84.64 ± 2.05 |
| ambient 1 | 81.53 ± 0.75 | 87.02 ± 2.57 | 38.34 ± 2.36 | 90.81 ± 1.48 | 90.48 ± 1.92 | 81.46 ± 3.09 | 80.26 ± 6.06 | 91.95 ± 1.63 | 83.20 ± 3.02 |
| ambient 2 | 81.10 ± 1.08 | 80.44 ± 7.32 | 36.60 ± 3.37 | 90.64 ± 2.12 | 91.14 ± 1.14 | 81.97 ± 2.89 | 84.77 ± 3.93 | 91.78 ± 1.70 | 82.11 ± 2.89 |
| holes 1 | 78.85 ± 2.14 | 85.91 ± 4.22 | 37.64 ± 3.63 | 90.67 ± 2.81 | 90.84 ± 2.23 | 82.58 ± 3.17 | 53.33 ± 25.31 | 86.11 ± 6.69 | 85.69 ± 2.58 |
| holes 2 | 80.05 ± 1.30 | 82.75 ± 2.98 | 43.62 ± 3.14 | 91.04 ± 2.40 | 90.46 ± 2.53 | 82.19 ± 2.35 | 71.57 ± 12.49 | 86.65 ± 11.88 | 84.48 ± 2.60 |
| holes 3 | 79.99 ± 1.35 | 83.43 ± 3.38 | 35.32 ± 3.49 | 91.10 ± 2.73 | 91.50 ± 1.38 | 81.90 ± 0.78 | 78.55 ± 14.92 | 85.50 ± 10.01 | 83.84 ± 1.21 |
| texture 1 | 82.29 ± 1.11 | 87.20 ± 2.49 | 35.96 ± 3.41 | 92.41 ± 2.00 | 91.26 ± 0.89 | 82.80 ± 2.35 | 85.93 ± 3.39 | 89.11 ± 4.82 | 84.29 ± 1.09 |
| texture 2 | 69.82 ± 4.47 | 34.55 ± 27.94 | 31.74 ± 3.57 | 61.11 ± 34.59 | 92.10 ± 1.61 | 83.29 ± 2.00 | 75.22 ± 14.65 | 77.25 ± 16.17 | 81.39 ± 1.47 |
| texture 3 | 71.04 ± 4.74 | 67.63 ± 10.77 | 33.16 ± 1.95 | 21.37 ± 30.78 | 90.39 ± 2.75 | 82.18 ± 1.76 | 43.62 ± 35.73 | 66.25 ± 20.31 | 81.41 ± 2.40 |
| baked light | 72.87 ± 1.23 | 80.01 ± 6.80 | 21.68 ± 6.87 | 74.93 ± 16.10 | 91.77 ± 1.80 | 83.83 ± 2.58 | 4.82 ± 12.06 | 87.30 ± 4.15 | 81.88 ± 4.36 |
| noise 1 | 81.17 ± 1.17 | 79.64 ± 10.21 | 37.25 ± 2.77 | 91.23 ± 1.77 | 90.44 ± 1.78 | 84.65 ± 2.05 | 84.13 ± 11.30 | 91.14 ± 3.22 | 83.51 ± 1.48 |
| noise 2 | 69.20 ± 4.92 | 51.01 ± 26.42 | 37.38 ± 2.71 | 87.19 ± 5.89 | 9.26 ± 28.24 | 84.15 ± 2.31 | 80.13 ± 7.02 | 84.27 ± 23.36 | 83.63 ± 1.52 |
| noise 3 | 65.65 ± 4.84 | 31.09 ± 21.30 | 38.07 ± 4.36 | 71.28 ± 8.53 | 4.67 ± 14.78 | 85.04 ± 1.33 | 85.76 ± 2.60 | 36.52 ± 41.50 | 83.94 ± 2.46 |

Table 3: Object Detection results: Train on combined YCB-Synthetic clean and noise l2/l3 artifact. Test YCB-Real, YCB-In-the-wild, and YCB-Video.

| train set | YCB-Real | YCB-In-the-wild | YCB-Video |
|---|---|---|---|
| clean | 81.93 $\pm$ 0.98 | 67.15 $\pm$ 2.56 | 37.83 $\pm$ 2.28 |
| 1/2(c+n2) | 82.48 $\pm$ 1.23 | 64.70 $\pm$ 3.02 | 35.80 $\pm$ 1.58 |
| 1/2(c+n3) | 82.24 $\pm$ 1.13 | 63.53 $\pm$ 2.68 | 35.51 $\pm$ 1.99 |

## Training on Combination of Clean+Corrupted Data

Obtaining clean labeled data is often challenging or impossible in real-world scenarios (ex: privacy issues with medical records data), so by excluding these factors, we remove another confounding factor from our analysis. However, it is still interesting to conduct experiments with different ratios of corrupted and clean YCB-Synthetic data. We have conducted experiments when training with 50% corrupted noise Level 2/3 and 50% clean data (See Table 3). Notice that due to the noisy data, the model performs better than when training on just clean data, indicating that Synthetic data with corruptions can actually improve model performance in certain settings! However, there are many more ratios and setting yet to test (due to time constraints in this rebuttal) and a per-object category analysis could be conducted. We encourage the community to use this dataset and continue investigating this question in further detail.

Figure 5: After creating the 3D models, we rendered these objects from several different camera viewpoints, as seen in this figure.

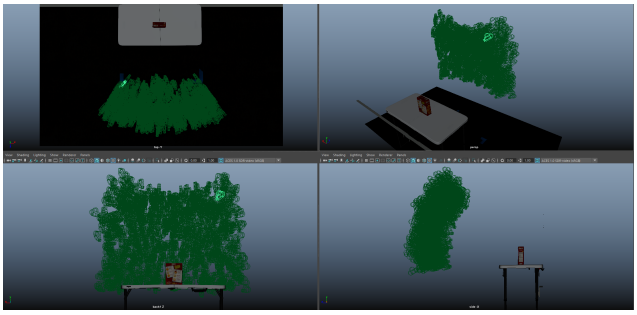
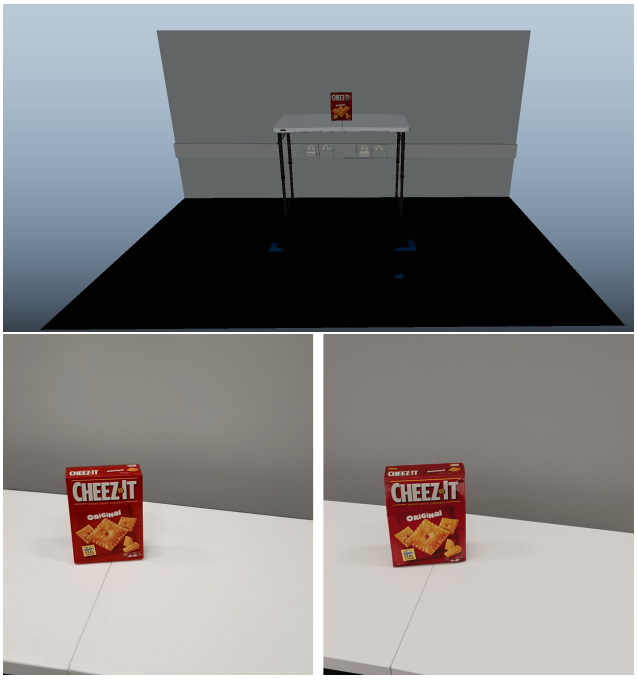


Figure 6: Left image shows the synthetic environment used in our synthetic data generation pipeline. Two images rendered using the environment are shown on right.




## Dataset Creation Diagrams

We visualize the rendering systems used to generate synthetic data in Figure 5 and the environment in 6. We show some more examples of synthetic training data with different corruptions in Figure 7. Finally, we show the real test datasets in Figure 8 and 9.

Figure 7: More examples of `YCB-Synthetic` data.

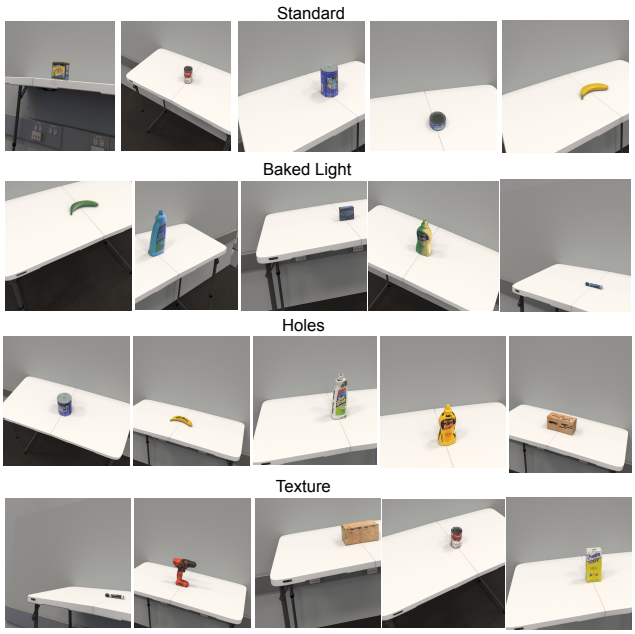



Figure 8: Examples of `YCB-Real`, `YCB-In-the-wild`, and `YCB-Video` data.

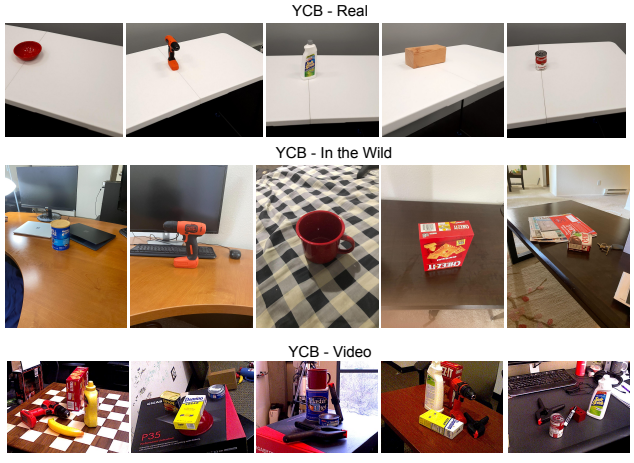


Figure 9: Real world environment used in the data capture. We show YCB objects and distractor objects used to create occlusions are shown in the left and right images respectively.

## References

[1] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016. 1