# Sensitivity of Object Detection Models to Synthetic Training Data

Afreen Alam, Vishal Bhatnagar, Rishitha Dubbaka, Nem Mehta, Kate Plas, Sonal Shah, Nikhil Thimmadasaiah

## I. Introduction

Collecting substantial quantities of training data for machine learning models is often challenging and costly. For example, if a recognition model was being developed to identify an extremely rare form of cancer, it is unlikely that enough training images could be obtained [8]. Because of these issues, training sets can be supplemented with synthetic data created using generative models. By simulating real samples, synthetic data can be added to a limited training dataset, reducing the difficulties associated with collecting and labeling sufficient real data.

A hybrid synthetic and real dataset allows for effective model training even when there are limited real-world samples. In the RarePlanes dataset, Shermeyer et al. found that combinations of $90\%$ synthetic to $10\%$ real can result in nearly indistinguishable performance as $100\%$ real data in aircraft identification tasks [6]. This displays the efficacy in using generated images for training object detection models. However, some models are more prone to overfitting synthetic data which results in degraded performance when it comes to testing on the entirely real test dataset [6]. Thus, each model has an optimal ratio of synthetic to real data. The primary problem addressed in this study is to examine the impact of incremental increases in synthetic training data on a variety of object detection models' performances.

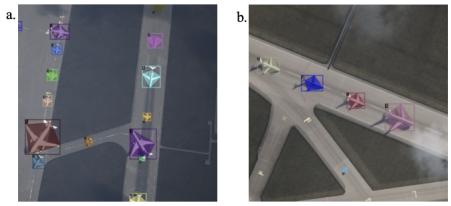


Fig. 1: Example of a real (a) and synthetic image (b) with mask and bounding box annotations from RarePlanes [6].

The motivation for identifying how different models react to synthetic training data is to be able to optimize model performance by balancing synthetic and real training data. Additionally, the study could help understand the domain gap, or the subtle differences in synthetic and real data that can cause performance degradation, even when they appear identical [7]. This could be done by identifying which model architectures minimize performance impacts induced by the domain gap. Our study could allow for training sets to be optimally calibrated by tailoring data composition to model structure.

Motivated by the need to maximize utility of synthetic data, this study is relevant because it can inform which models are most robust against the domain gap brought on by differences in the real and synthetic data. That way, tasks that were before limited by quantity of training data can effectively supplement using synthetic data in the most suitable amount for a specific model.

## II. Related Studies

We review both two-stage and one-stage object detection models and evaluate their performance based on their average precision. DiffusionDet, Faster R-CNN, Mask R-CNN, Cascade R-CNN, and YOLOv8 are examined using the RarePlanes dataset [6].

### A. DiffusionDet

DiffusionDet is a state-of-the-art model developed by Shoufa Chen *et al.*, which applies diffusion to the task of object detection by iteratively denoising bounding boxes into the ultimate prediction [2]. This approach addresses the problem of object priors associated with anchor boxes and region proposals because the model begins from purely random boxes, devoid of learnable parameters.

*1) Methodology:* The key technique in DiffusionDet is the use of the *noise-to-box* approach [2]. This method initially diffuses object boxes from the ground truth by incrementally adding noise from a random distribution. In the training stage, the model gradually denoises the bounding boxes into the prediction, where the goal is to minimize the difference between the predicted denoised boxes and the ground truth, characterized using L2 loss. Because the domain gap can be considered noisy data, DiffusionDet could be well suited for the problem at hand. In particular, the lack of object priors could support its robustness against synthetic training data.

*2) Performance:* Using the standard benchmark, Average Precision, on the COCO, LVIS, and CrowdHuman datasets, DiffusionDet outperforms RestinaNet, Faster R-CNN, Cascade R-CNN, DETR, and Sparse R-CNN [2]. For example, the next closest AP is Sparse R-CNN with an AP of 45.0 while DiffusionDet attains an AP of 46.8. Its primary weakness is in decreased speed performance but it's strength is enhanced AP.

### B. Faster R-CNN

Faster R-CNN is a two-stage model proposed by Shaoqing Ren *et al.*, which extended Fast R-CNN through the introduction of the Region Proposal Network (RPN), thereby

addressing the computational bottleneck associated with region proposals [5]. The integration of RPN for region proposal generation and Fast R-CNN for object detection into a unified network by sharing convolutional features allowed for near real-time processing as well as improved accuracy.

*1) Methodology:* In comparison to selective search methodologies, Faster R-CNN's RPN is a fully convolutional network that takes the input image and outputs object proposals with their objectness scores [5]. The region proposals are generated by sliding a small network over the extracted feature maps from the last shared convolutional layer. For each sliding window, multiple region proposals are generated using anchor boxes which are references at multiple aspect ratios and scales. These developments in Faster R-CNN enhance its resilience to noisy data by the implementation of RPN to focus on high-objectness regions, which reduces the impact of background noise.

*2) Performance:* Faster R-CNN was evaluated on PASCAL VOC 2007, 2012 and MS COCO datasets using mean Average Precision (mAP) [5]. On the PASCAL VOC 2007 dataset, RPN and Fast R-CNN with ZF achieved an mAP of $59.9\%$ while RPN and Fast R-CNN with VGG-16 achieved an mAP of $69.9\%$. Compared to traditional region proposal methods like Selective Search, Faster R-CNN obtained better accuracy using only 300 proposals per image.

### C. Mask R-CNN

Mask R-CNN which was introduced by He *et al.*, provides a framework for instance segmentation, extending Faster R-CNN by adding a section for predicting masks, bounding boxes, classification labels [3]. These advancements allow Mask R-CNN to effectively perform both object detection and segmentation. A key advancement developed in Mask R-CNN is the ROIAlign layer which improves the pixel-level accuracy. The model enhances mask accuracy by sectioning off the mask prediction from the class selection, so that each of the object's mask is created without being influenced by competition for the classification decision. These advancements help to lower errors and results in more accurate segmentation and detection for each object.

*1) Methodology:* Mask R-CNN addresses the challenges for both segmentation and objection detection when noisy synthetic training data [3]. The model uses an RPN to propose the areas where objects of interest could be located. These proposals are then fine-tuned to maintain only the best possible bounding box options so that the model can concentrate on the most likely object regions and reduce the chances of false positives. RolAlign is integrated in Mask R-CNN to address the issue of spatial quantization errors that occur during the process of feature extraction. By maintaining the input and output in an aligned order, the model can be more resilient to any noise.

*2) Performance:* This model's overall performance was compared to other models on AP metrics using IoU thresholds for the COCO dataset [3]. Mask R-CNN outperforms previous models including Faster R-CNN, FCIS, and MNC on AP metrics, especially for higher IoU thresholds (AP75). The most significant strengths of Mask R-CNN were in its capacity to achieve high accuracy because of its RoIAlign layer and separate mask and class predictions. These advancements made it more effective in challenging scenarios than its predecessors.

### D. Cascade R-CNN

Cascade R-CNN is an object detection model developed by Cai *et al.* that consists of sequential detectors trained with increasing IoU thresholds so that each detectors are increasingly filter out close false positives [1].

*1) Methodology:* The key technique in this algorithm is the use of progressive detectors which are trained stage by stage in order to take advantage of the observation that a detector's output is a good distribution to train the following higher quality detector [1]. This resampling of sequentially elevated IoU thresholds ensures that all detectors have equally sized sets of objects meeting that threshold, which reduces overfitting. An equivalent cascading method is applied at the inference stage, to enable a close match between the detector quality and the hypothese for each stage. This model approaches the problem of handling noisy training data by training each detector stage with higher IoU thresholds, so that more background noise is removed with each detector.

*2) Performance:* Cascade R-CNN was evaluated on COCO-2017 using average precision across IoU thresholds of 0.5 to 0.95, incremented by 0.05 [1]. It was compared to YOLOv2, SSD513, RetinaNet, Faster R-CNN, AttractioNet, and Mask R-CNN using various backbones, but primarily ResNet-101. With an AP of 42.8, Cascade R-CNN outperformed the next closest model, Mask R-CNN, with an AP of 38.2.

### E. YOLO

You Only Look Once, by Redmon *et al.*, describes an object detection model that works at very fast speeds by condensing the object identifying and classification process to a single regression problem [4].

*1) Methodology:* YOLO works by creating square-sized grid cells of length *S*, and estimating *B* bounding boxes for each grid cell, where each box predicts for *C* classes, with *S* and *B* being preset values, and *C* depending on the dataset being used [4]. The detection model of YOLO uses 24 convolutional layers and 2 fully connected layers. Also described is a model referred to as Fast YOLO, which is functionally similar, but uses 9 convolutional layers instead of 24.

*2) Performance:* YOLO's advantages lie in the speed with which it can process and its ability to adapt to artwork, when compared to SOTA models [4]. In a comparison between YOLO, Fast YOLO, and other fast speed detectors on Pascal VOC 2007, YOLO had the highest mAP at 63.4 and an FPS of 45, while Fast YOLO had the highest FPS at 155 and second highest mAP at 52.7. Even compared to less than real-time detectors, the highest performing model that was compared, Faster R-CNN VGG-16, was only 10 mAP higher than YOLO,

but was 6 times slower [4]. The key to YOLO's performance, apart from its structure, was that the first 20 layers were pretrained with an average-pooling and fully connected layer on the ImageNet 1000-class competition dataset for about a week.

### F. Methods

*1) Data Preparation and Analysis:* We have performed our experiments on the RarePlanes dataset which contains both real and synthetic satellite images [6]. For training our models we used two different percentages of synthetic training data. The $98\%$ synthetic training set consists of $98\%$ synthetic training data and $2\%$ real training data while the $100\%$ synthetic training set consists of $100\%$ synthetic training data and $0\%$ real training data. The total number of images for both the training datasets is 29,075. The test dataset remains $100\%$ real dataset consisting of 2710 images.

We then analyze the difference in distributions between the training and test sets using KL Divergence and EMD. As shown in Table I, the KL Divergence for 100% synthetic vs 100% real dataset is lower in comparison to 98% synthetic vs 100% real dataset. This shows that the 100% synthetic dataset has a closer data distribution to the 100% real dataset. Whereas, the higher KL Divergence score for the 98% synthetic vs 100% real dataset indicates a greater difference in the two data distributions. However, the low EMD scores for both 98% synthetic vs 100% real dataset and 100% synthetic vs 100% real dataset suggest a greater similarity between the distributions for both percentages of the synthetic training datasets.

| | 98% Synth vs 100% Real | 100% Synth vs 100% Real |
|---|---|---|
| KL Divergence | 1.00521 | 0.40243 |
| EMD | 0.09061 | 0.09269 |

Table I: Results of KL Divergence and EMD for different percentages of synthetic data

*2) Training and Testing the Models:*

We trained each of the models for 45,000 iterations on the 98% synthetic training dataset and 100% synthetic training dataset. The models were then tested against the 100% real test dataset. The architecture, activation function, loss function, optimizer and learning rate for each of the models remained unchanged from the original architectures.

**Inputs:**

$$\mathcal{M} = \left\{ \begin{matrix} \text{Faster R-CNN, Mask R-CNN,} \\ \text{Cascade R-CNN, DiffusionDet} \\ \text{YOLOv11} \end{matrix} \right\} \quad \text{(Set of models of interest)}$$

$$\mathcal{D}_{\text{train}} = \left\{ \begin{matrix} \text{98\% Synthetic,} \\ \text{100\% Synthetic} \end{matrix} \right\} \quad \text{(Training datasets)}$$

$$\mathcal{D}_{\text{test}} = \{\text{100\% Real}\} \quad \text{(Test dataset)}$$

**Procedure:**

```
for m ∈ M do
    for d_train ∈ D_train do
        Train model m on dataset d_train
        Test model m on dataset D_test
    end for
end for
```

**Outputs:**

bbox AP (performance metric) for each model $m \in \mathcal{M}$ on $\mathcal{D}_{\text{test}}$

### G. Results

To evaluate the performance of our models, we used bounding box Average Precision (bbox AP) as the metric. Bounding box AP measures the capability of a model to accurately detect, localize and classify objects from an image. It is also a standardized object detection metric used in all of the literature we reviewed.

From the results shown in the Table II we can see that the bbox APs for the 98% synthetic data tends to be higher than that of 100% synthetic data for all of the models. This demonstrates that models trained on the 98% synthetic dataset have a better object detection performance. Although, DiffusionDet has the highest AP score for the 98% synthetic training dataset, it's performance significantly degrades when the model is trained on 100% synthetic dataset.

From Figure 2, we determine that YOLOv11 was the least sensitive to changes in the percentage of synthetic training data as it has the smallest change in bbox AP value from 98% synthetic training data to 100% synthetic training data. DiffusionDet has the largest change in bbox AP value from 98% synthetic training data to 100% synthetic training data which makes it the most sensitive to changes in the percentage of synthetic training data.

Unfortunately, we were unable to get our Faster R-CNN model to work which is why we have not included the Faster R-CNN model in our results.

| Models | 98% Synth | 100% Synth |
|---|---|---|
| Mask R-CNN *(Trainable params: 110,319,318)* | 66.5275 | 41.3719 |
| Cascade R-CNN *(Trainable params: 140,746,466)* | 62.480 | 35.48 |
| YOLOv11 *(Trainable params: 20,114,688)* | 54.3933 | 30.4704 |
| DiffusionDet (swin backbone) *(Trainable params: 173,350,102)* | 70.68 | 31.188 |

Table II: Overall bbox AP Results

### H. Discussion

The reviewed object detection models demonstrate varying capacities to handle noisy training data, which could be

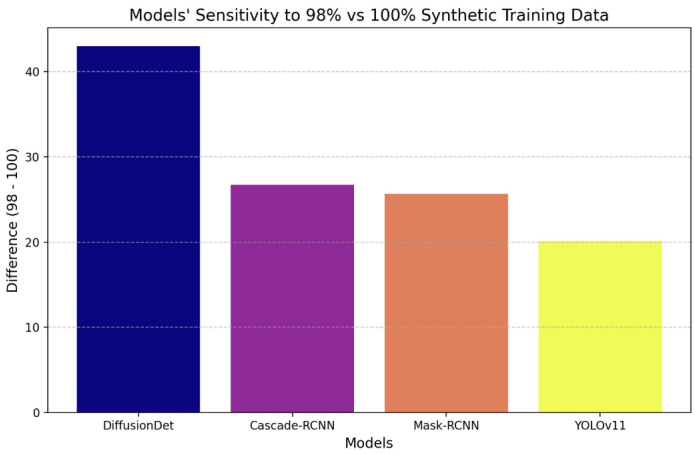


Figure 2: Models' sensitivity to 98% vs 100% synthetic training data

beneficial when trained with higher levels of synthetic data. YOLOv11 stands out for its potential robustness against the domain gap while DiffusionDet has the best object detection performance amongst all the models when trained on 98% synthetic data.

This study is the first known to examine an object detection model's ability to withstand high levels of synthetic data and to compare the resulting APs of DiffusionDet, YOLOv11, Faster R-CNN, Cascade R-CNN, and Mask R-CNN when trained on the RarePlanes dataset ratios described above. Our approach builds on previous research by studying AP trends over incremental changes in synthetic data proportions rather than traditional benchmarks. This analysis reveals models' sensitivities to the domain gap and could provide guidance in determining optimal ratios of synthetic-to-real data.

Contrary to our initial hypothesis, DiffusionDet was not the most robust model against the domain gap. This is an unexpected behavior which we would like to investigate further in the future. We would also like to perform runs on the RarePlanes dataset with different seeds when partitioning the dataset as well as experiment with different percentages of synthetic training data – 92%, 94% and 96%.

## REFERENCES

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[2] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19830–19843, 2023.

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[5] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[6] Jacob Shermeyer, Thomas Hossler, Adam Van Etten, Daniel Hogan, Ryan Lewis, and Daeil Kim. Rareplanes: Synthetic data takes flight. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 207–217, 2021.

[7] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[8] Christos G Xanthis, Dimitrios Filos, Kostas Haris, and Anthony H Aletras. Simulator-generated training datasets as an alternative to using patient data for machine learning: an example in myocardial segmentation with mri. *Computer Methods and Programs in Biomedicine*, 198:105817, 2021.