Playing The Strengths of Synthetic Data

Carson Trego

September 2024

1 Introduction

In recent years, we have seen neural networks do a wealth of varied tasks from object detection, code generation, and sentiment analysis [3][1][2]. Neural networks, for all of the flexibility they can provide, often require significantly more training data compared to other machine learning methods [6]. Many large datasets are available for those seeking to train their own neural networks on established tasks, but if a large dataset cannot be used for a user's task, they will have to create a new dataset for said task [3][6]. Creating a high quality dataset is challenging: dataset creators will have to gather a large number of samples, annotate those samples mostly by hand, and those samples must have a fair distribution of characteristics, or the model could be biased. For example, one model gave the appearance of being able to classify huskies and wolfs, but because the backgrounds for wolfs consistently contained snow, the model would classify the subject as a wolf instead of a husky [2]. Gathering and annotating data for a neural network dataset is not only time and labor intensive, but ensuring fairness of the samples and preventing the wrong features from being used in prediction is a difficult task, so it is easy to put a large amount of work into a dataset that has a fatal flaw of some kind [6][2]. One proposed method to make datasets faster to create and more fair is using synthetic data to train a model [6][3][1][7][11]. Synthetic data is broad term to describe data that is simulated instead of being gathered directly from reality [1][11]. For example a classifier model may need to differentiate between two types of cups. Rather than obtaining a wealth of images of the cups in a variety of environments, a set of images could be rendered from the 3D models of the cups and then used to train the model. Ideally, a 3D rendered image should be so realistic that the model learns as effectively as it does with real images. In practice, there is a well-known issue called the "domain gap" that often occurs when synthetic data is used, in which the model performs well on the virtual training, validation, and testing data, but performs relatively worse when given real data [4][3]. The domain gap does not mean that the model does not perform well, just worse than the real data model [4]. Advances in rendering software could bring the domain gap closer, but there is still advantages to using synthetic data for a model, particularly when real data is either limited or completely unavailable otherwise [3][4][1]. Advantages to synthetic data methods are not just in cost

and effort, as that extra information, such as a 3D model of the object that is intended to be classified, can be used to do more with data that is available, as a 3D model has some information that is often not available from photos alone, and synthetic methods in general could provide more flexibility in classification, such as allowing for different classes that are slight modifications of the previous ones.

2 Theory

The method intended for this project begins with a 3D model of the objects that are intended to be classified. While classifying objects that vary significantly could be a topic that could be an aspect of this project, we will begin with assuming the simple case where each object is consistent in form, is fairly rigid, and opaque.

The first advantage in using synthetic data is that it eliminates a lot of redundancy in gathering data. Say a model needs to classify 8 different cups, which are manufactured to be nearly identical, so a cup that is in class 1 would look almost the exact same as a cup that is also in class one (As is true with disposable cups, which are often sold in large packs of identical cups). With the traditional method, several photos, at several angles, distances, lighting, and environments would have to be obtained, which would be a large effort to describe just one object. Using synthetic data and rendering system, a single model for each cup would be sufficient.

3D models contain information on how the object would look from all angles, so one 3D model can be used to simulate thousands of photos capturing different angles and distances. Assuming that a 3D render of an object is sufficiently realistic, the object could be observed in a variety of different chosen simulated situations with only one 3D model. The advantage with choosing the environments is that there is no background bias issue. Images of coconuts in tropical areas might be more common than images coconuts of in factories, but with synthetic data, a classifier could be trained in rare or nonsensical situations such as an image with a coconut in a steel mill. This ensures that, provided the rendering system has a wealth of virtual spaces, the model does not have a classification bias based on background features. The images in a synthetic data system can be easily balanced to represent each class equally and in varied environments.

The size of the object is another piece of information provided in 3D models that can be used. While normal photos often are not paired with specific information of the object, such as the angle the object is facing relative to some central point, the distance from the camera, or the actual size of the object, a 3D render has all of this information by design. If the model is then used with some sort of depth seeing method, such as a range finder, LIDAR, or stereo camera, the model could know how far away the object is, and combined with the known size of the class, "know" when a larger version of one class is being shown. This could also be achieved without depth-seeing hardware, provided the camera is

placed at a consistent distance from the objects, and is given information about how far the platform is. This aspect of synthetic data directly leads to 6 degree of freedom pose estimation, which is very useful for robotics applications. This will be explored in greater depth in a later part of the document.

Black box machine learning algorithms do not allow the operators to predict how the model will respond to certain conditions, and the method of finding out how a model will respond to specific environments is by testing the model directly. This adds another step to the model making process, where a model must first be completed in entirety and then tested under a variety of conditions. Using synthetic data, this aspect of creating models is not fully removed, but the method of producing new data to test the model on can be used to predict how the model will fail, and potentially adjust for it. For example, a model may struggle to classify two objects that look identical from certain angles. Using synthetic data, the model could automatically test itself on all of the classes to check for weak points, and "know" before hand that it will struggle with the class at certain angles, and when the model is presented with a said object at said angle, it can instruct itself to obtain more information on the object, rather than using the current image to make a definitive classification.

Theoretically, synthetic data in the form of 3D models allows for single models of objects to provide images from all angles, a wealth of environments, information in 6 degree of freedom pose estimation, the size of the object, and self diagnostics for when the model has shortcomings. This all assumes a strong 6 degree of freedom pose estimation and classification model to begin with, alongside a high quality rendering system with a wealth of environmental conditions prepared in advance. With these assumptions, there should be no domain gap, yet there is, so while the information needed to make these conclusions is present in the system, and while a human operator given a 3d model could theoretically perform this task in entirety, how the model responds to the actual synthetic data, and the quality of the synthetic data, is to be seen.

3 Defining The Challenge

The goal of this challenge is to exploit the advantages of synthetic data, in which all objects are placed on a platform which the model can make observations of through a camera. The model should "know" how far away the platform is, and be able to rotate the platform and gather additional pictures if needed. To quantify the model's performance, the model will have to correctly classify the object and its 6 degree of freedom parameters in its final conclusion, as well as do so with as few photos and mechanical actions as possible.

To demonstrate this we will give two examples in a model with 3 classes: Flathead screw, Phillips head screw, and plastic banana. In the first case, the model identifies that the object is a plastic banana and recognizes that the probability of misclassifying a banana from this angle is rare, so it automatically logs the class and 6 degree of freedom parameters. In the second example, the model sees a screw with the head facing the opposite direction. Having

prior tests show that a screw of either type at this angle is likely to result in a misclassification, the model calculates what angles of the objects are most likely to provide a definitive result between the two, by finding angles where the correct classification rate is high for both objects. Using this information in the form of a spherical space, the model runs a second calculation to find the area that would function as a tie breaker that is the closest to the current angle (thus reducing the amount of mechanical effort and time needed), and snaps an additional photo of the object. This next photo (and potentially prior information) is used to make a final classification of the object and its 6 degree of freedom parameters.

4 Other Efforts

Many studies and model creators have used synthetic data as a way to augment their dataset or create a dataset from scratch [3][4][6][11][7]. 3D models have been used as synthetic data for 6 degree of freedom pose estimation, as in the case of the LINEMOD template matching method [9][10]. There is also 6D models such as EfficientPose and YOLO6D, which both perform 6 degree of freedom pose estimation using 3D models, but make use of a convolutional neural network. [13][5]. In a prior project, I had used a method similar to the copy paste method, but rather than using segmented real images, I had taken 3D models and segmented them onto a different background, this method was later used in plain classification and with other models [8][12][14]. While the object detection models were able to perform quantifiable well on the virtual data, the real life tests were limited, but showed fair results [12][14]. Much of the real world testing has been fairly qualitative, however the real time capability of the model allow for a better intuition of its performance. For this project, a classifier and pose estimator could be used independently to improve accuracy, and potentially, the amount of operations required for processing an object.

These model articles tend to not discuss a method for scaling its use, and the goal of this project is to bring these models closer to being part of a system that could be efficiently used in a variety of devices. One aspect to make the models more usable will be the efficient tie breaking method, in which the model prepares angles and objects that tend to produce confusion, and how to most efficiently determine which class is being shown. The other aspect of this model is physical, as the platform that is used for the objects and the method of scanning new objects and automatically adjusting the model to accommodate them will be part of the model considerations.

5 Method

Several objects will be prepared as 3D model and physical object pairs obtained by either scanning a physical object or 3D printing a 3D model. The 3D models will be used with a synthetic data generator to produce training data and tem-

plates, which will then be used in the 6 degree of freedom pose estimator and the classifier. Once the models are prepared, the synthetic data generator will be used to access potential weak points of the model by attempting to classify the object at a multitude of angles and conditions. These results will be mapped to a spherical plane and saved for future reference. When performing the physical test, these saved tests and the pose estimator will be used to estimate the risk of misclassification caused by having an angle that does not offer enough information to adequately classify the object, and the classification test spheres will be used to find positions that would provide more information on the object, while comparing the results to other objects that are at risk, with a set of points on the spheres chosen to maximize the chance of a decisive classification of the object. In addition to searching for regions that would provide the most information, the model will be attempting to find the shortest path and least number of photos required to come to a decisive conclusion. This test will be ran several times on multiple different objects, and the amount of operations, camera movement, and correct classifications will all be incorporated into how well the overall system performs. This should be more accurate and faster than taking a large amount of photos and processing all of them, or taking photos of the object uniformly at random, but this remains unknown. The goal of this challenge is to see if applying this method provides more accuracy and speed than simply using an object detector on one photo alone, grid searching the object processing a large number of photos, or randomly taking photos of the object and processing those.

6 Hardware

The camera angle should be restricted to only rotate around the object with angles theta and phi. For the duration of this experiment, this does not have to be motorized, and thus can be simply clamped in place. The camera itself needs to be able to create depth maps, and thus a stereoscopic camera will be used. (I can get access to an Intel RealSense Depth Camera)

[ADD INFO]

7 Training Data

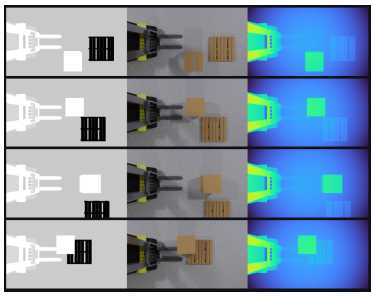


Figure 1: .

Training data will consist of 4 primary sources [ADD INFO]

7.1 Color-RGB (nxmx3)

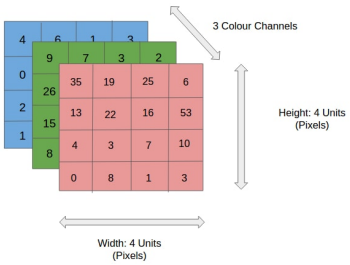


Figure 2: .

To represent color, we will split standard Red, Green, Blue (RGB) images into 3 single-color matrices, resulting in a single nxmx3 tensor, with n and m being the resolution of the input layer. This resolution is often low, and often necessitates a framework for processing higher resolution images effectively, such as *Slicing Aided Hyper Inference*.

Contrast to using luminance, which helps determine the form visual size of objects, using the full visual spectrum will help the detector differentiate objects that have identical form and size, but differ in color scheme.

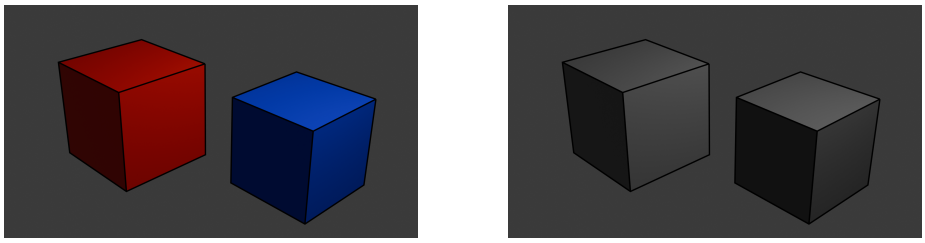


Figure 3: Two objects with nearly identical luminance displays (right) can be differentiated using color (left). The objects above can only adequately be differentiated by the model if color is an input in the model

7.2 Depth-D (nxmx1)

Size and distance both occupy the same visual channel in human vision which are combined into visual size: the amount of space an object occupies in an image. Given that the true size and true distance of an object is combined into a single channel, neither the true size nor true distance of an object can be inferred from the amount of space an object occupies in an image, unless either the true size or true distance is known. This means that images alone cannot consistently differentiate objects with the same appearance but are different sizes. This issue can be solved by gathering more information about the scene, labeling the objects with the true dimensions, constraining how the images can be taken, or by including hardware that can directly infer distance.

Having a physical label of dimensions on an object is a simple way of determining the true size of an object, and thus the distance the object is from the camera. For example, if two people are wearing name tags that show that they are both 1.8 meters tall, and both people appear to be the same height in the image, then it can be inferred that they are both roughly the same distance from the camera. This is often impractical, as it requires every object to have physical label. Another option is to restrict how the photos are taken. If two people are positioned against a wall of a known distance, and occupy the same length against that wall, it can inferred that their height's are the same. While this works in many settings, such as "mugshots", it requires some information about the environment, and assumes that the effect of the thickness of the object is negligible.

Alternatively, if two people seem to be the same height in an image, one can check if this is an optical illusion by looking at the two people at a different angle. If two people appear to be the same height in an image, but a 45 degree rotation shows that one person is standing several meters behind the other person, then it can be determined that they are not the same height. Rather than viewing the scene at two separate angles and making complex inference from that, two images taken from different angles can be combined into a single depth map, providing estimates of the distance of every surface within close view of both cameras. Similar information can also be obtained by measuring the distance of

the camera from each surface by the time it takes for a phenomena to travel to and from the object. This can be done with light using LiDAR, or sound using SoNAR.

For simplicity, this project will use depth maps, as they can deliver information about the distance of the object using only a nxmx1 tensor bitmap, which models can easily as a consistent source of depth information, and this information can be gathered without any mechanical actions.

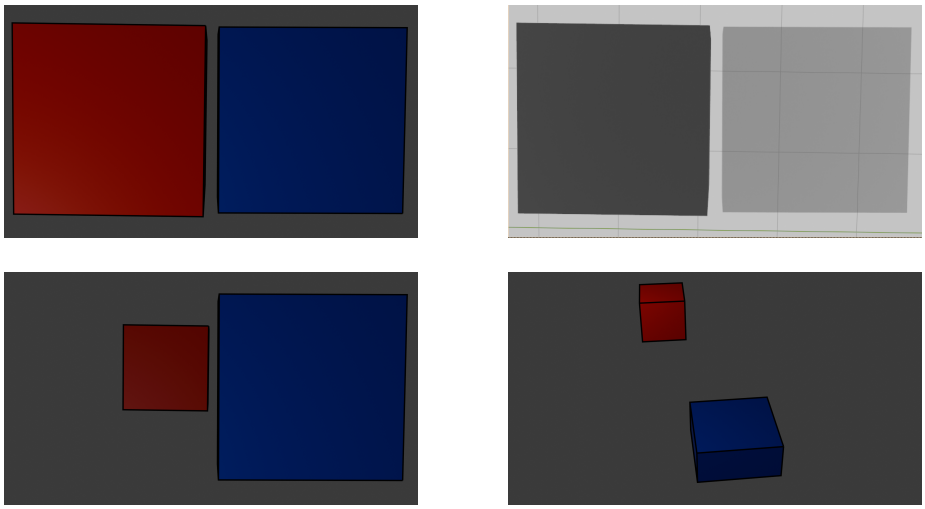


Figure 4: A small object appears to be the same size as a larger object at a greater distance (top left). These objects can be differentiated by setting the objects on an equally distanced platform (bottom left), inspecting the scene at a different angle (bottom right), or by using a depth map (top right)

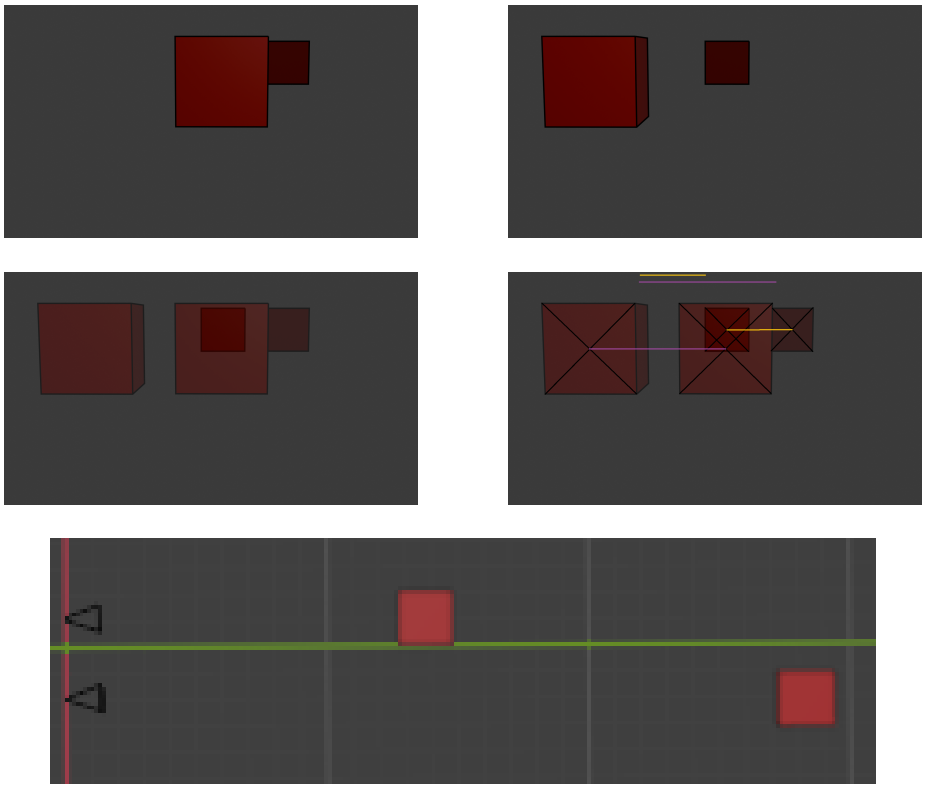


Figure 5: Stereoscopic image of both image taken 1 unit to the left (top left) and 1 unit to the right (top right) superimposed into one image (middle left) with the each of the objects' face centroids connected to display the discrepancy between the left and right stereoscopic images, which shows that the centroid discrepancy of the closer object is slightly less than half of the further object (middle right). The overhead view shows the true distance of the closer object to the camera to be slightly less than half the true distance of the further object to the camera

The perpendicular distance D a point is from two cameras with focal length f placed B distance apart is given by the following formula:

$$Disparity \propto \frac{B * f}{D} \tag{1}$$

Where *Disparity* is the distance from the two points formed in a superimposed stereoscopic image. This means that if the distance is doubled, the disparity halves, which is demonstrated in figure 5.

7.3 Six Degree of Freedom Pose (6x1)

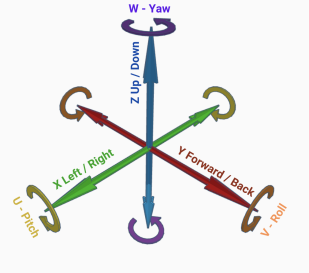


Figure 6:

The Six Degree of Freedom (6DoF) pose information will begin by arbitrarily specifying the default rotation of the object.

8 Training Loss

Given that all of the training data will be generated synthetically by 3D rendering, all of the depth, pose, segmentation mask, and object class information can be deduced directly from the 3D rendering software. This means that all synthetic images will have an established ground truth value for depths, poses, segmentation masks, and object classes.

8.1 Class Loss

The loss function for the object classes will be provisionally given by categorical cross entropy, in which all classes are treated equally, even when objects have similar characteristics.

$$-log(\frac{e^{s_p}}{\Sigma_j^C e^{s_j}}) \tag{2}$$

8.2 Pose Loss

Where C is all classes, C_P is the true class, s_x is the score for the class C_x.

The pose loss will be given by the following formula, based on the PoseNet Deep Convolutional network.

$$loss(pose) = ||\hat{x} - x||_2 + \beta||\hat{q} - \frac{q}{||q||}||_2 \tag{3}$$

Where:

$$||\hat{x} - x||_2 \tag{4}$$

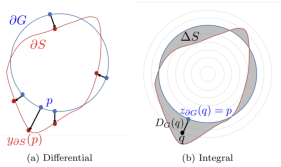
Represents the loss for the predicted position of the object.

$$||\hat{q} - \frac{q}{||q||}||_2 \tag{5}$$

Represents the loss for the predicted orientation of the object, and Beta represents the scaling factor for position and orientation.

8.3 Segmentation Loss

To have a differentiable loss function that weights completely disjoint segments, boundary loss will be used.



(a) Differential (b) Integral

Figure 7:

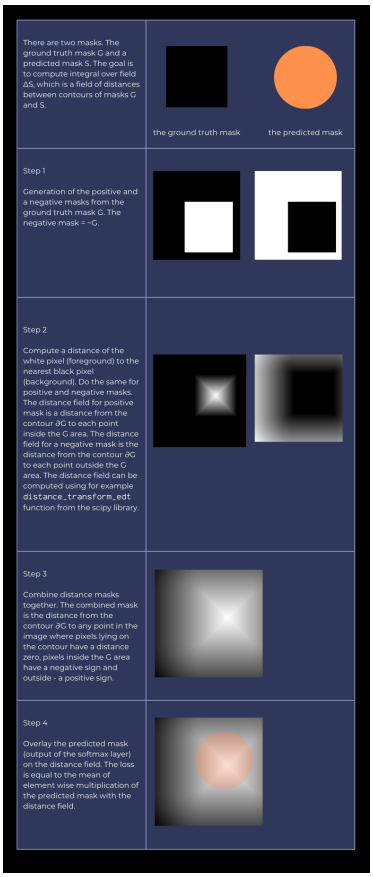


Figure 8:

9 Input Data (features)

The Classes, 6DoF metrics, and boundary loss are all related to the targets of training, but the model will not have access to this information in operation. However, it will have access to:

- An RGB image.
- A depth map.
- The results of previous inference on the same object.

10 Three Strategies

10.1 Single Image

Model will encounter an object in a random orientation, it will have on attempt to classify the object by selecting the highest confidence score.

10.2 Monte Carlo Search

Model will encounter an object in a random orientation, it will select N orientations at random angles theta and phi. After aggregating all of the confidence scores, it will select the highest one to classify.

10.3 Attempted Optimal

Prior to running any inference, the model will use the 3D models to test where it scores low confidence levels. Since the 3D model represents the ground truth for the class, any low confidence scores will indicate that the model performs poorly at that orientation. If the model has a top one confidence score below a selected threshold, then the model estimate the pose of all object classes with confidence score above some minimum threshold. The model will then take the spherical synthetics tests, turn them into a Voronoi diagram, bin the results, and then find all orientations that provide the maximum binned predicted confidence scores, and then attempt to find the union of all maximum regions for all selected classes

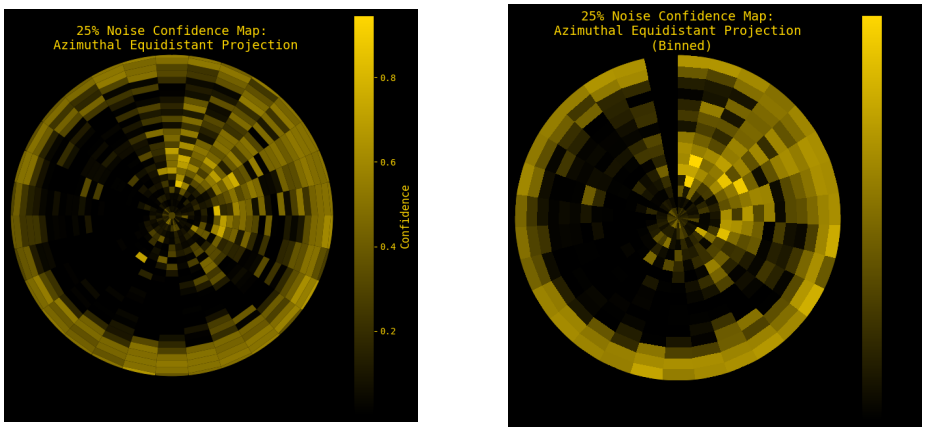


Figure 9: Default Voronoi diagram of confidence map (left) and further binned version (right). Note, there was an issue with completing polar coordinate maps.

11 Scoring Criteria

Each method will be first simulated, and then performed on an assortment of real-world objects in random orientations. The best method should reduce accurately classify each object, use as few inference steps as possible to reduce computational needs, and require the camera to move as little as possible to make accurate inference. To measure the accuracy, this will be based on the number of correct classifications vs the total rounds, running inference on the same object multiple times before classifying will not harm the accuracy score. The number of inference steps is based on how many times the model needs to scan photos of the object to make a conclusion. With the *Single Image* and *Monte Carlo Search methods*, this number should be flat, with 1 and n inferences respectively. The only strategy that could vary in this result is the Attempted Optimal, which has an unknown amount of inferences required for each object. The third criteria will be the amount of spherical distance the camera needed to travel in total to complete each classification. For the *Single Image*, this should be 0, but for the other two strategies, this number is not known.

References

[1] What is synthetic data? URL: https://www.ibm.com/topics/synthetic-data.

[2] Husky or wolf? using a black box learning model to avoid adoption errors, 2017. URL: https://innovation.uci.edu/2017/08/husky-or-wolf-using-a-black-box-learning-model-to-avoid-adoption-errors/.

[3] Bridging the domain gap for neural models, 2019. URL: https://machinelearning.apple.com/research/bridging-the-domain-gap-for-neural-models.

[4] Xiangyu Bai, Yedi Luo, Le Jiang, Aniket Gupta, Pushyami Kaveti, Hanumant Singh, and Sarah Ostadabbas. Bridging the domain gap between synthetic and real-world data for autonomous driving, 2023. URL: https://arxiv.org/abs/2306.02631, https://arxiv.org/abs/2306.02631 arXiv:2306.02631.

[5] Yannick Bukschat and Marcus Vetter. Efficientpose: An efficient, accurate and scalable end-to-end 6d multi object pose estimation approach, 2020. URL: https://arxiv.org/abs/2011.04307, https://arxiv.org/abs/2011.04307 arXiv:2011.04307.

[6] Terrance DeVries and Graham W. Taylor. Dataset augmentation in feature space, 2017. URL: https://arxiv.org/abs/1702.05538, https://arxiv.org/abs/1702.05538 arXiv:1702.05538.

[7] Killeen B. D. Hu Y. Grupp R. B. Taylor R. H. Armand M. Unberath M. Gao, C. Synthetic data accelerates the development of generalizable

learning-based algorithms for x-ray image analysis. *Nature machine intelligence*, 2023. URL: https://doi.org/10.1038/s42256-023-00629-1.

[8] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation, 2021. URL: https://arxiv.org/abs/2012.07177, https://arxiv.org/abs/2012.07177 arXiv:2012.07177.

[9] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Linemod dataset, 2013.

[10] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In Kyoung Mu Lee, Yasuyuki Matsushita, James M. Rehg, and Zhanyi Hu, editors, *Computer Vision – ACCV 2012*, pages 548–562, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[11] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016. https://doi.org/10.1109/CVPR.2016.352 doi:10.1109/CVPR.2016.352.

[12] Ben Laube Carson Trego Maxwell Kline Tyson Shields. Yosco, you only scan once. using a single 3d scan to automatically generate a labeled object detection training dataset, 2024.

[13] Bugra Tekin, Sudipta N. Sinha, and Pascal Fua. Real-Time Seamless Single Shot 6D Object Pose Prediction. In *CVPR*, 2018.

[14] Carson Trego. Yosco cls, 2024. temp, currently a program on its own with no written document, as this is part of that.