Blocks World Revisited: The Effect of Self-Occlusion on Classification by Convolutional Neural Networks

Markus D. Solbach
York University
solbach@eecs.yorku.ca

John K. Tsotsos
York University
tsotsos@eecs.yorku.ca

Abstract

Despite the recent successes in computer vision, there remain new avenues to explore. In this work, we propose a new dataset to investigate the effect of self-occlusion on deep neural networks. With TEOS (The Effect of Self-Occlusion), we propose a 3D blocks world dataset that focuses on the geometric shape of 3D objects and their omnipresent self-occlusion. We designed TEOS to investigate the role of self-occlusion in the context of object classification. In the real-world, self-occlusion of 3D objects still presents significant challenges for deep learning approaches. However, humans deal with this by deploying complex strategies, for instance, by changing the viewpoint or manipulating the scene to gather necessary information. With TEOS, we present a dataset with two subsets (L_1 and L_2), containing 36 and 12 objects, respectively. We provide 768 uniformly sampled views of each object, their mask, object and camera position, orientation, amount of self-occlusion, as well as the CAD model of each object. We present baseline evaluations with five well-known classification deep neural networks and show that TEOS poses a significant challenge for all of them. The dataset, as well as the pre-trained models, are made publicly available for the scientific community under https://data.nvision.eecs.yorku.ca/TEOS.

1. Introduction

Over most of the last decade, computer vision was pushed by efforts put into deep learning. The exact advent of this deep learning dominated era is often dated to the ImageNet challenge ([39]) in 2012. Since then, the performance of models on various tasks has been improving at unparalleled speed; for instance, image classification on the ImageNet dataset surpassed the reported human-level performance in 2015 ([13]). Two of the enablers for the recent successes are faster computers, specifically graphic processors, and the availability of large scale and often well-curated data sets to learn from.

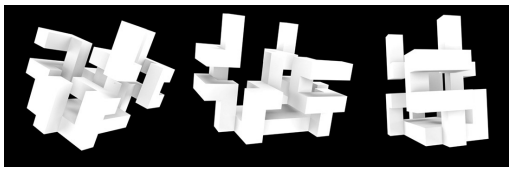

Figure 1. Example of the proposed objects from three different viewpoints.

The deep learning paradigm is omnipresent, and, with it, the need for data with specific statistics to work in certain domains. [26] goes as far as saying that "Data is playing an especially critical role in enabling computers to interpret images as compositions of objects, an achievement that humans can do effortlessly while it has been elusive for machines so far."

Many domains exist in which one would like machines to perform visual tasks ([5]). One of these is object classification, which is defined as whether a particular item is present in the stimulus ([51]).

Object classification is an essential capability of humans, as well as for any robotic system whose goal is to be a real-world assistant; in a factory, hospital, or at home, just to name a few. Even though very successful in many domains, deep learning methods are challenged with occlusion ([24]), which is inevitable in real-world scenarios. Here, we go a step further and show that deep learning methods are also challenged by the self-occlusion of objects, hence not generalizing to objects' 3D structure.

The problem of understanding the 3D structure from a 2D description, for instance, a line drawing, was first put forward independently by [18] and [6], and they both showed that the necessary critical condition for a line drawing to represent an actual arrangement of polyhedral objects was labelability.

As the human brain is very efficient at reconstructing a scene's 3D structure from a single image with no texture, colour or shading, efforts have been concentrated on com-

putational complexity issues; one might think an efficient solution exists (e.g. polynomial-time). [23], however, proved that this problem is NP-Complete, also for simple cases like trihedral, solid scenes. To further research in this field, [34] proposed a method to generate random instances of line drawings with useful distribution to investigate questions related to complexity of understanding images of polyhedral scenes. More recently, [47] provided a 3D extension with controllable camera parameters and two different light settings. It is designed to enable research on how a program could parse a scene if it had multiple and definable viewpoints to consider. An example of a polyhedral scene from [47] is shown from three different viewpoints in Fig. 2.

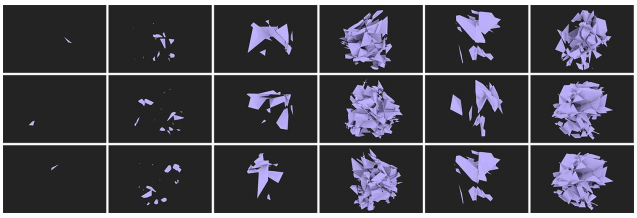

Figure 2. Six polyhedral scenes from three different viewpoints with increasing complexity ([47]).

With the increasing successes, contemporary computer vision approaches show a trend away from artificial problems and provide solutions to real-world problems, already deployed in many domains ([1]), for example, optical character recognition, industrial inspection systems, medical imaging, and biometrics. While toy-domains are essential demonstration vehicles even in the deep learning era ([8, 19, 20, 21, 31]), a disparagement of artificial domains can be seen ([46]). At the very least, these domains can support meaningful systematic experiments. Here we revisit one such artificial domain; the Blocks World. In visual perception, the basic physical and geometric constraints of our world play a crucial role.

Larry Roberts argued that "the perception of solid objects is a process which can be based on the properties of three-dimensional transformations and the laws of nature" ([38]). Roberts' popular Blocks World was an early attempt to build a system for complete scene understanding for a closed artificial world of textureless polyhedral shapes by using a generic library of polyhedral block shapes. This toy domain that has remained as a staple of the AI literature for over 50 years.

The polyhedral scenes shown in Fig. 2, showing a kind of "extreme" blocks world setting, feature significant self-occlusion. However, the space of possible objects and their characteristics are far too large to conveniently use in a learning scenario. Motivated by this set, we present a new, more tightly controlled dataset, *TEOS*: The Effect of Self-Occlusion.

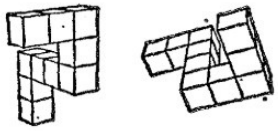

Figure 3. Example of the objects by [42] which are used as an inspiration.

TEOS is a Blocks World based set of objects with known complexity, controlled viewpoints, with a known level of self-occlusion and 3D models. *TEOS* shares similarities in appearance with the so-called Shepard and Metzler Objects ([42]), which are widely used in the literature for mental rotation tasks. See Fig. 3 for an illustration of two such objects.

However, with *TEOS*, we present a set of objects that go beyond the Shepard and Metzler objects. Specifically, our objects have known, incrementally increasing complexity, they are designed to require self-occlusion to be solved, they share a common coordinate system, and we will show that they are challenging for visual tasks using modern classification algorithms.

Our contributions are an investigation of the effect of self-occlusion for object classification. To accomplish this, we provide a novel set of objects, a carefully created dataset, including an in-depth explanation of the objects and generated data with a focus on self-occlusion and a baseline evaluation with modern classification algorithms.

The remainder of the paper is structured as follows. First, we will explain in detail the objects we have created for *TEOS*. We then continue by giving an overview of related work, describing the data acquisition, presenting our self-occlusion measure, evaluating the dataset against modern classification algorithms, and finally finishing with our conclusions and future directions.

2. Related Work

To the best of our knowledge, self-occlusion has not attracted much attention in the literature. However, occlusion caused by other objects has. In addition to several datasets, a number of approaches were introduced to deal with occlusion.

2.1. Occlusion Datasets

A burden of deep learning is its need for vast mounts of training data. Even though occlusion and its effect on vision tasks has been addressed for some time ([16, 33, 4, 17]), occlusion datasets created are usually too small to be used to train successful deep learning models. Furthermore, to our knowledge, datasets, if considering occlusion, mostly intro-

duce various levels of clutter but fail to define occlusion in a generic way. For instance, the CMU Kitchen Occlusion dataset (CMU_KO8) by [17] consists of 1,600 images of eight kitchen objects, which only yields 200 examples per class. The dataset has explicitly been designed to challenge object recognition algorithms with strong viewpoint and illumination changes, occlusions and clutter. Besides this, an occlusion reasoning module is also proposed (Section 2.2). With the *ICCV 2015 Occluded Object Challenge* ([15, 4]), a dataset with eight objects positioned in a realistic setting of heavy occlusion is presented. The objects can be described as being of different domains (animals, office supplies, kitchenware, ...). However, neither a definition of occlusion nor a metric is given. Fig. 4 shows an example image of the dataset.

Figure 4. A scene with different objects under occlusion from the *ICCV 2015 Occluded Object Challenge*.

The majority of occlusion datasets, however, deal with the occlusion of pedestrians. Specifically, in the context of autonomous driving, detecting pedestrians, even if occluded, is crucial to detect potential collisions. It is argued that most existing datasets are not designed for evaluating occlusion. For instance, the Caltech dataset ([7]) only contains 105 out of 4250 images with occluded pedestrians. The CUHK Occlusion Dataset ([33]) is specifically designed as a pedestrian dataset with occlusion. The authors selected images from popular pedestrian datasets and recorded images from surveillance cameras and filtered them for occluded pedestrians. The dataset contains 1,063 images with binary classification to indicate occlusion.

2.2. Occlusion Reasoning

Reasoning about occlusion has been used in many areas, from object recognition to tracking and segmentation. Reported in [17], the literature is extensive, but there has been comparatively little work on modelling occlusion from different viewpoints and using 3D information until recently. Further, occlusion reasoning is broadly classified into five categories; inconsistent object statistics, multiple images, part-based models, 3D reasoning, and convolutional neural networks.

The first category uses inconsistent object statistics to reason about potential occlusion. For instance, [32] use inconsistencies in 3D sensor data to classify occlusions. [12] introduce an occluder part in their grammar model when all parts cannot be placed. [54] use a scoring metric based on individual HOG filter cells. [17] incorporate occlusion reasoning in object detection in a two-stage manner. First, in a bottom-up stage, occluded regions are hypothesized from image data. Second, a top-down stage is used that relies on prior knowledge to score the candidates' occlusion plausibility. Extensive evaluation on single and multiple views shows that incorporating occlusion reasoning yields significant improvement in recognizing texture-less objects under severe occlusions.

The use of multiple images characterizes the second category. For these approaches, consecutive images are necessary to disambiguate the object from occluders. For instance, [9] detects the objects and extrapolates the state of occluded objects using an Extended Kalman Filter. Reliable tracklets that are used in a temporal sliding window fashion are generated to disambiguate occluded objects in [56].

One of the largest categories is part-based model approaches. A challenge of global object templates is occlusion as their performance degrades with its presence significantly. A popular solution to this problem is to separate the object into a set of parts and detect parts individually. This approach yields more robust detections towards occlusion. For example, [44] analyze the contribution of each part using a linear SVM and train the classifier to use unoccluded parts to maximize the probability of detection. [55] go a step further and use multiple part detectors to maximize the joint likelihood. Binary classification of parts is introduced by [52]. They decompose the HOG descriptor into small blocks that selectively switch between an object and an occlusion descriptor.

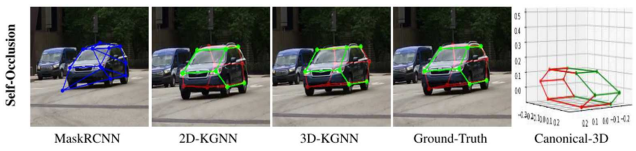

Figure 5. The effect of occlusion reasoning used in a CNN. Left the original CNN (MaskRCNN) and different (2D and 3D) occlusion reasoning approaches improve the detection ([37]).

More recent work is using 3D information. [35] train multiple occlusion detectors on mined 3D annotated urban street scenes that contain distinctive, reoccurring occlusion patterns. [53] use RGB-D information and an extended Hough voting to include object location and its visibility pattern. [36] addresses precisely the problem of self-

occlusion in the context of human pose estimation and adds an inference step to handle self-occlusion to an off-the-shelf body pose detector to increase its performance under self-occlusion. [3] propose an object recognition system that also works in the presence of occlusion and clutter. They use a soft label *Random Forest* to learn the shape features of an object. Using occlusion information, taken from the depth data, the forest emphasizes the shape, thus making it robust to occlusion. More recently, [40] propose a part-based architecture to recover the 6D object pose in-depth images that is also able to deal with occlusion. Their *Intrinsic Structure Adaptor* adapts the distribution shifts arising from shape discrepancies and removes the variations of texture, illumination, pose, etc.

Convolutional neural networks form the last group of approaches. [37] introduce a framework to predict 2D and 3D locations of occluded key points for objects to mitigate the effect of occlusion on the performance. Evaluated on CAD data and a large image set of vehicles at busy city intersections, the approach increases the localization accuracy of MaskRCNN by about 10%. A self-occlusion example can be seen in Fig. 5. [27] uses deep supervision to fine-grain image classification. In their approach, they simulate challenging occlusion configurations between objects to enable reliable data-driven occlusion reasoning. Occlusion is modelled by rendering multiple object configurations and extracting the visibility level of the object of interest. [25] introduce CompositionalNets, which is combined with part-based models. The fully-connected classification layer is replaced with a differentiable compositional model. The idea is to decompose images into objects and context, and then decompose objects into parts and objects' pose. The approach can learn occlusion invariant features and discard occluders during classification, hence increasing performance under occlusion. However, a trade-off is that a good occluder localization lowers classification performance because classification benefits from features that are invariant to occlusion, where occluder localization requires a different type of features. Namely, ones that are sensitive to occlusion. It is pointed out that it is essential to resolve this trade-off with new types of models.

3. Object Definitions

With *TEOS*, we present in total 48 objects, split into two sets; L_1 and L_2. L_1 consists of 36 objects in 18 complexity classes, hence tailored towards research exploring the effect of finely grained complexity changes.

All objects consist of the following two elements: One 20mm x 60mm x 120mm base (Fig. 6 right) and n 20mm x 20mm x 60mm cuboids (Fig. 6 left). The complexity of an object is simply calculated as

$$compl = n + 1 \tag{1}$$

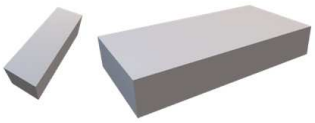

Figure 6. The building blocks used to create the objects of *TEOS*; cuboid (left) and base (right).

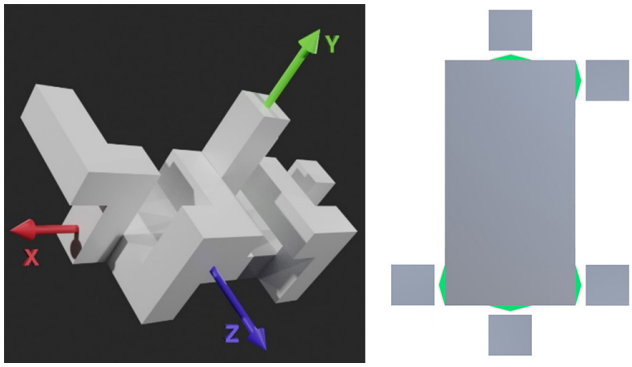

Figure 7. Left: Illustration of the common coordinate system of the objects. Right: Possible cuboid connection points on the base.

Where n is the number of cuboids used. Further, *inter-class object complexity* refers to objects that are not of the same object complexity class, while *intra-class object complexity* refers to objects that are of the same object complexity class but differ in their configurations.

Building an object, the base has five connection points for cuboids. All cuboids are only attached upright, sitting flush with the bottom of the base. This also makes it simple to define a coordinate system.

All objects share the same coordinate system, which is crucial for any research that looks at the effect of the orientational difference of 3D objects. The coordinate system is defined as depicted in Fig. 7 (left); the Y-Axis is running orthogonal out of the base, the X-Axis running through the base from its center of gravity towards the end with three cuboids-connectors, and the Z-Axis runs orthogonal to the Y- and X-Axis with the positive direction through the side of the base with two cuboid connections.

A cuboid has eight connectors at which another cuboid can be attached (Fig. 7 (right)). Consecutive cuboids are always orthogonally and never aligned in their direction, which is one of the differences to the Sheppard and Metzler objects. Furthermore, cuboids never intersect or touch neighbouring cuboids, hence avoiding geometrical loops. Creating the objects for L_1, we focused on making the complexity comparable by consecutively adding one cuboid per complexity class to the object of the previous complexity class.

In several empirical studies with human subjects, we have studied the relationship between the number of elements per object and classification accuracy. The performance to clas-

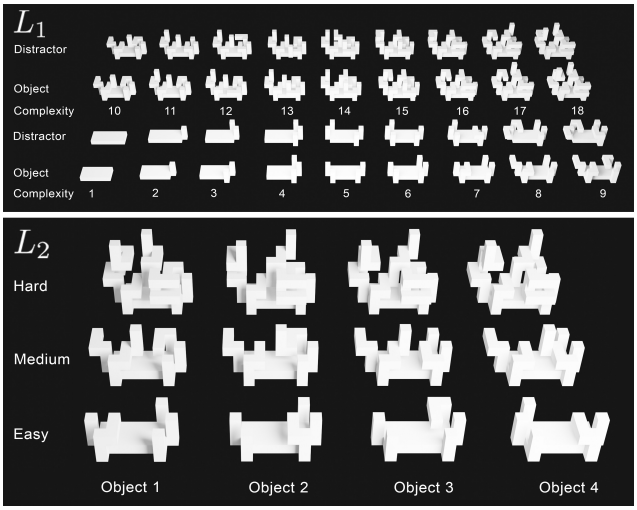

Figure 8. Top: Illustration of L_1 with all 36 objects. Bottom: Illustration of L_2 with all 12 objects, split into three different complexity classes.

sify L_1 objects is reliable (accuracy of $> 98\%$) for objects of $compl = 7$ (Eq. 1). The classification is less accurate (89%) with objects of $compl = 10$. Finally, the classification gets challenging (57%) with objects of $compl = 18$. Based on these findings, we have created the L_2 set. It is designed with less variation across complexity class but more variation within a complexity class. Twelve objects are evenly split into three complexity classes; easy with seven elements, medium with ten elements, and hard with 18 elements. Within a complexity class, the objects only differ in one small detail by changing one of the elements' orientation. This said, the L_1 and L_2 subsets enable two classes of self-occlusion analysis; one with high inter-class object complexity variability and another with high intra-class object complexity variability in appearance, respectively. Furthermore, as will be discussed later, this set also provides self-occlusion distributions with unique means for each of the three complexity classes (see Fig. 14).

Lastly, we present the L_1 and L_2 object sets. The L_1 objects can be seen in Fig. 8 (top), and consist of 36 objects split into 18 complexity classes. There is a distractor object of the same complexity for each object that differs only in one small detail; one of the items is oriented differently. The introduction of the distractor objects is intended to support research in visual recognition, where merely counting the number of elements would reveal the object class. The L_2 objects can be seen in Fig. 8 (bottom). Fig. 9 shows how an increase of complexity of the L_1 dataset also increases the average amount of self-occlusion among all viewpoints. Each point shows the self-occlusion of the respective object from a specific viewpoint. The viewpoints are evenly distributed on a sphere around an object, resulting in 768

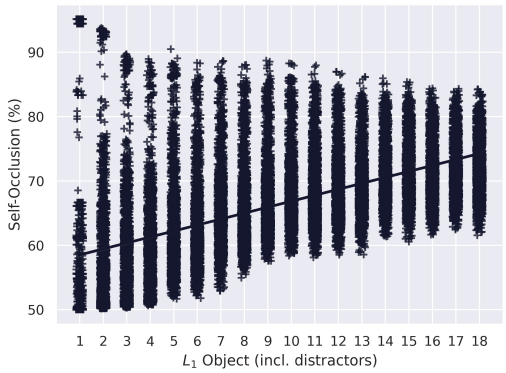

Figure 9. Illustration of the amount of average self-occlusion per object of L_1. Each point shows the self-occlusion of the respective object from a specific viewpoint. The straight line illustrates the increase in average self-occlusion as the complexity increases.

unique views. The straight line illustrates the increase in average self-occlusion as the complexity increases. However, worth noting, with an increasing amount of complexity, the self-occlusion distribution per class decreases. Further information about our self-occlusion measure will be explained in Section 5.

4. Dataset Acquisition

TEOS is a dataset that is designed to be used in the virtual as well as the real world. For the former, one can use the rendered images and provided 3D Models (.STL file). For the latter, the objects are designed to be printable with a 3D printer. However, in this Section we want to focus on the generation of the rendered dataset images for which we have used Blender ([2]), a free and open-source 3D computer graphics software toolset. For *TEOS*, each object was rendered from 768 views – Totalling 36,864 images. To achieve realistic renderings of the objects, we used the Cycles Path Tracing rendering engine, created a white, smooth, plastic imitating material, set six light sources in the rendering scene and used 4,096 paths to trace each pixel.

Each object is rendered from the same set of views. To determine the views, we used the Fibonacci lattice ([48]) approach. This approach allows distributing points on a sphere uniformly. Other approaches, for example, using radial distance, polar angle and azimuthal angle, will result in an unevenly sampled sphere; dense on the poles and sparse closer to the equator. Fig. 10 illustrates the chosen views to generate the dataset. Each blue-coloured point represents a location where the camera is placed and oriented to the center where the object (red) is. We chose a sphere radius of two such that the object is view-filling but not cropped. Further, as it is sometimes practiced in the machine learning community ([10, 29, 30, 22]), we also provide the object mask and renderings with a dark and bright background for

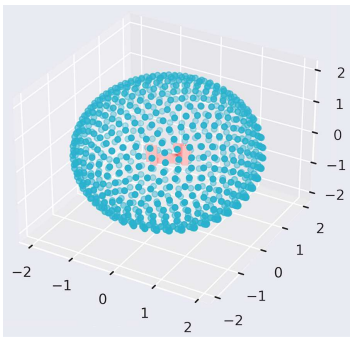

Figure 10. Illustration of viewpoints used to render each object of L_1 and L_2. Views are evenly distributed on a sphere around an object (blue points) and point towards the object (light red). In total 768 views are taken.

data augmentation purposes. The annotation file contains the object-type, view-id, bounding box information, object and camera positions and orientations, object dimensions, and self-occlusion value.

5. Self-Occlusion Measure

It seems evident that if we see less of an object, it is harder to classify it. Regions of the object that are occluded to us might hold distinct features to tell object X apart from object Y. In other words, occlusion for visual classification plays an important role. However, it is not only dependent on the view but also on the object. Let us take, for example, a sphere. No matter from which angle we look at it, we always observe 50% of it. On the contrary, for a complex polygonal shape, this cannot be answered as quickly as it is dependent on its geometry.
[11] distinguishes two kinds of occlusions; "external occlusion" and "self-occlusion." "External occlusion" is caused by an object entering the space between the camera and the object of interest and "self-occlusion "which describes the occlusion caused by the object of interest to itself. For *TEOS*, we are interested in the latter, as we always have one object in the scene. To our knowledge, no standard self-occlusion measure is used for computational approaches; therefore, we aim to specify our own intuitive measure as:

$$SO_{c_i} = \frac{A_\phi^{c_i}}{A_\sigma} \tag{2}$$

Where A_ϕ is defined as the occluded (not visible) surface area of the object and A_σ stands for the total surface area of the object. These values are computed as shown in Algorithm 1. Note that for this calculation, the object identity must be known. An object might have different views from which it causes the same amount of self-occlusion, resulting in perhaps a considerably different appearance. Fig. 11 shows an example of two objects from two different views

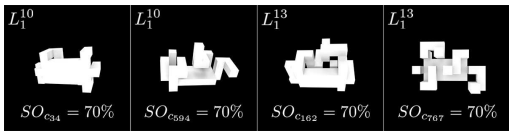

Figure 11. Examples of different objects (Object 10 and 13 of L_1) and poses causing the same amount of occlusion but different appearances.

with the same amount of occlusion.
Therefore, we also consider the camera's point of view with c_i as the camera pose. Here, c_i is defined as the camera position $c_i = (x_i, y_i, z_i)$ and computed based on the Fibonacci lattice approach (see Fig. 10). The camera orientation is automatically set such that the object is in the centre of the viewpoint.

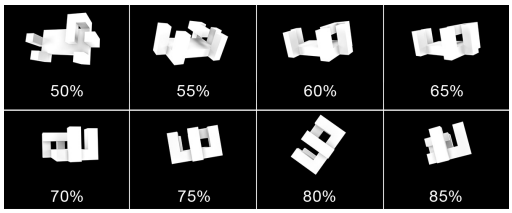

Figure 12. Some object viewpoints and their corresponding SO_{c_i}.

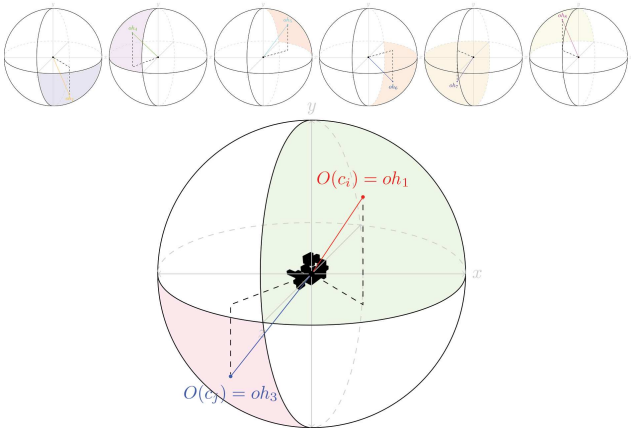

Figure 13. Visualization of the octahedron based projection used to map camera positions. Bottom: two example camera poses (c_i and c_j) mapped to oh_1 and oh_3.

For evaluation purposes, we also define a function that maps a camera position (c_i) onto one of the eight regions of the octahedral viewing-sphere placed at the centre of an object. Fig. 13 illustrates a mapping example for two camera-positions. We represented the viewing sphere around an object as a spherically tiled octahedron, resulting in eight uniformly distributed triangles. To map a viewpoint c_i to a tile, we perform a determinant check to see in which tile a

given camera pose c_i is located.
In our rendered data set, the self-occlusion was calculated by using the following steps:

Algorithm 1 Self-Occlusion

1: Iterates over all faces of the object with valid normals and calculate the (A_σ)
2: Subdivide the objects into many thousand elements
3: Position the camera at a given location and pointing it at the object (see Fig. 10)
4: Select vertices that are visible through view-port
5: Divide object into visible and not-visible part
6: Iterate over all faces of the not-visible object with valid normals and calculate (A_ϕ)
7: Lastly, calculate Self-Occlusion (Equation 2)

Fig. 12 shows eight examples of the same object (object-7) from different viewing angles and sorted based on their amount of self-occlusion. As can be seen in the illustration, a single object can cast many different appearances based on the viewing angle and a significant change in the amount of what is observable of it.

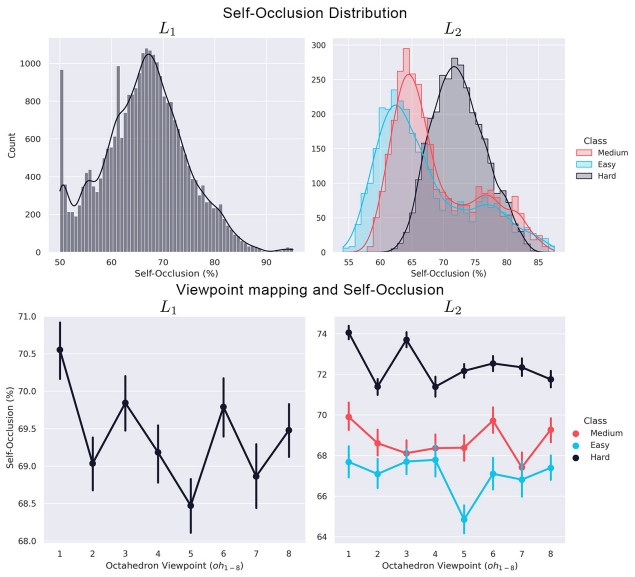

Figure 14. Illustration of the self-occlusion distribution for L_1 and L_2 (top), as well as the distributional relation between viewpoint mapping and self-occlusion for L_1 and L_2 (bottom).

Fig. 14 illustrates the self-occlusion distribution for L_1 and L_2 (top) and the distributional relation between viewpoint mapping and self-occlusion for L_1 and L_2 (bottom). Self-Occlusion for L_1 ranges from 49.99% to 95.16% with a mean at around 68% and L_2 from 54.08% to 87.5% with a mean at 61% (Easy), 63% (Medium), and 71% (Hard). The lower half of the Fig. shows that different octahedron viewpoints result in varying amounts of self-occlusion. For both L_1 and L_2, an overall sweet-spot with the least self-occlusion is at oh_5, presumably resulting in the best classification result. More specifically, for L_2 class "Hard", this spot is at $oh_{2/4}$ and for class "Medium" at oh_7.

6. Baseline Evaluation

In this Section, we discuss how well modern classification approaches perform on *TEOS*. We have chosen five deep learning models with different properties, carefully trained and evaluated them on *TEOS*.
We have chosen Inception-V3([49]), MobileNet-V2 ([41]), ResNet-V2 ([14]), VGG16 ([45]) and EfficientNet ([50]) as reference networks for *TEOS*. Their trained version of *TEOS* will be made publicly available. Table 1 shows more details about the networks in ascending order of their parameter count.

Table 1. High-Level CNN Characteristics

CNN	Layers	Parameters (mil.)
MobileNet-V2	53	3.4
Inception-V3	48	24
ResNet-V2	152	58.4
EfficientNet-B7	813	66
VGG16	152	138

Besides the architecture of CNNs, a crucial element is the choice of training-parameters and so-called hyperparameters. In our case, we have looked at the input size, input noise, dropout rate, learning rate, optimization algorithm and lastly, the difference between learning from scratch and fine-tuning the networks. Hyperparameters such as input noise, drop rate, learning rate were determined using the hyperparameter optimizer Hyperband by [28]. The remaining parameters were empirically determined. Table 2 presents the parameters used to establish the baseline of *TEOS*.

Table 2. Chosen Training Parameters

Parameter	Value
Input Size	224 x 224 – 800 x 800 (dependent on CNN)
Input Noise	Gaussian Noise of 0.1
Drop Rate	20%
Learning Rate	1e-5
Optimizer	Adam Optimization
Learning Method	Fine Tuning

To prepare the data for training, we chose a 20% validation split and augmented the remaining 80% with the following data augmentation techniques ([43]): rotation ($0 - 40°$), width/height shift (0-20%) and zoom (0-20%).
Our results show that MobileNet-V2 performed best across L_1 and L_2. Specifically, for L_1, it achieved a top-1 accu-

racy of 17.25% and 10.83% on the L_2 data set. See Fig. 15 for the classification accuracies of all networks. It seems that MobileNet-V2 is the only network that was able to learn some aspects of *TEOS*, performing with a large (L_1) or small (L_2) margin above chance, whereas all other networks perform at around chance. This, perhaps, has something to do with the relatively homogeneous appearance of *TEOS*, not allowing the more complex CNNs to learn from. However, this needs to be investigated further in the future.

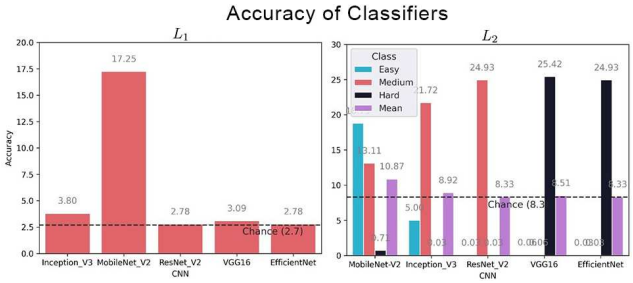

Figure 15. Evaluation results on L_1 (left) and L_2 (right) for five different CNNs and their accuracy across the entire datasets.

Generally, L_2 is more challenging to learn for CNNs than L_1. Even the best performing CNN is only 2.53% above chance, where this margin for L_1 was at about 14.5%. This is explainable with the high intra-class similarity of L_2 – objects of one class look very similar to each other and only vary in a small detail, which might be only observable from certain views, hence will be confused with each other. The L_1 dataset, on the other side, has a low inter-class similarity – the appearance of objects varies between classes. A closer look at the results of L_2 reveals that more extensive networks (VGG16 and EfficientNet-B7) were able to learn objects of class "Hard" of L_2; however, they could not learn "Medium" and "Easy" Objects. The smaller networks, on the other hand (MobileNet-V2 and Inception-V3), were able to learn "Easy" and "Medium" objects but not "Hard." Except for MobileNet-V2, all networks have problems to learn the "Easy" Objects. See Fig. 15 for details.

Regarding the connection between classification accuracy and amount of self-occlusion, it can be generally said that the classification accuracy goes down if self-occlusion increases. We have chosen the three best-performing CNNs to analyze this connection and grouped L_1 and L_2 from 50% to 85% self-occlusion in 5% intervals. $< 50\%$ captures viewpoints with a self-occlusion of less than 50%. $> 85\%$ includes images with more than 85% (Fig. 16).

Furthermore, we also investigated the connection between the viewpoint mapped to an octahedral viewing-sphere and accuracy. As can be seen in the example of L_1 and MobileNet-V2, the viewpoint does play a vital role and can result in an increase of accuracy performance by $13.28 \hookrightarrow 22.31 = 67.99\%$. Across L_1 and L_2 the octahedral viewpoint resulting in the best performance was oh_5. This can be explained with that all objects share a common coordinate system and shows once more that the viewpoint matters and, even more, that an *ideal viewpoint* can exist.

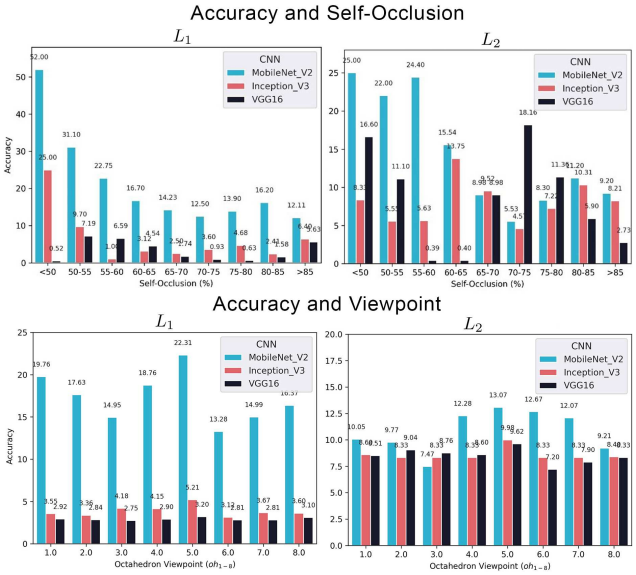

Figure 16. Evaluation for the three top-performing CNNs. Top: Accuracy across the entire datasets with respect to self-occlusion. Bottom: Accuracy and how it is affected by the chosen viewpoint.

Further, even though the CNNs are trained and validated on the entire data set, their best performance can be seen at lower self-occlusion rates, which shows the vital role of self-occlusion for object classification performance.

7. Conclusion and Future Directions

In this work, we have presented a novel 3D blocks world dataset that focuses on the geometric shape of 3D objects and their omnipresent challenge of self-occlusion. We have created two data sets, L_1 and L_2, including hundreds of high-resolution, realistic renderings from known camera angles. Each data set also comes with rich annotations.

Further, we have presented a simple but precise measure of self-occlusion and were able to show how self-occlusion challenges the classification accuracy of modern CNNs and the viewpoint can benefit the classification. Lastly, in our baseline evaluation, we have presented that CNNs cannot learn *TEOS*, leaving room for future work improvements.

We hope to have paved a way to explore the relationship between object classification, viewpoint, and self-occlusion with this work. Specifically, we hope that *TEOS* is useful for research in the realm of active vision – to plan and reason for the next-best-view seems to be crucial to increase object classification performance.

References

[1] Alexander Andreopoulos and John K. Tsotsos. 50 Years of object recognition: Directions forward. *Computer Vision and Image Understanding*, 117(8):827–891, 2013. 2

[2] Blender. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2020. 5

[3] Ujwal Bonde, Vijay Badrinarayanan, and Roberto Cipolla. Robust instance recognition in presence of occlusion and clutter. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8690 LNCS(PART 2):520–535, 2014. 4

[4] Eric Brachmann, Alexander Krull, Frank Michel, Stefan Gumhold, Jamie Shotton, and Carsten Rother. Learning 6D object pose estimation using 3D object coordinates. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8690 LNCS(PART 2):536–551, 2014. 2, 3

[5] John B Carroll. *Human cognitive abilities: A survey of factor-analytic studies*. Cambridge University Press, 1993. 1

[6] Maxwell B Clowes. On seeing things. *Artificial intelligence*, 2(1):79–116, 1971. 1

[7] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761, 2012. 3

[8] Alexey Dosovitskiy et al. FlowNet: Learning Optical Flow with Convolutional Networks Alexey. In *Proc. ICCV*, pages 2758–2766, 2015. 2

[9] Andreas Ess, Konrad Schindler, Bastian Leibe, and Luc Van Gool. Improved multi-person tracking with active occlusion handling. In *ICRA Workshop on People Detection and Tracking*, volume 2. Citeseer, 2009. 3

[10] Mark Everingham, Luc Van Gool, Christopher K I Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 5

[11] Vincent Gay-Bellile, Adrien Bartoli, and Patrick Sayd. Direct estimation of nonrigid registrations with image-based self-occlusion reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):87–104, 2010. 6

[12] Ross Girshick, Pedro Felzenszwalb, and David McAllester. Object detection with grammar models. *Advances in neural information processing systems*, 24:442–450, 2011. 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the IEEE international conference on computer vision*, 2015. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 7

[15] 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 *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7724 LNCS, pages 548–562, 2013. 3

[16] Edward Hsiao, Alvaro Collet, and Martial Hebert. Making specific features less discriminative to improve point-based 3D object recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2653–2660, 2010. 2

[17] Edward Hsiao and Martial Hebert. Occlusion reasoning for object detectionunder arbitrary viewpoint. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(9):1803–1815, 2014. 2, 3

[18] David A Huffman. Impossible object as nonsense sentences. *Machine intelligence*, 6:295–324, 1971. 1

[19] Eddy Ilg et al. Occlusions, Motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *Proc. ECCV*, pages 614–630, 2018. 2

[20] Mona Jalal et al. SIDOD: A synthetic image dataset for 3D object pose recognition with distractors. In *Proc. CVPR-W*, pages 475–477, 2019. 2

[21] Justin Johnson et al. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proc. CVPR*, volume 2017, pages 2901–2910. IEEE, 2017. 2

[22] Rishabh Kabra, Chris Burgess, Loic Matthey, Raphael Lopez Kaufman, Klaus Greff, Malcolm Reynolds, and Alexander Lerchner. Multi-Object Datasets. https://github.com/deepmind/multi-object-datasets/, 2019. 5

[23] Lefteris M Kirousis and Christos H Papadimitriou. The complexity of recognizing polyhedral scenes. *Journal of Computer and System Sciences*, 37(1):14–38, 1988. 2

[24] Gregor Koporec and Janez Pers. Deep learning performance in the presence of significant occlusions - An intelligent household refrigerator case. *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, pages 2532–2540, 2019. 1

[25] Adam Kortylewski, Qing Liu, Angtian Wang, Yihong Sun, and Alan Yuille. Compositional Convolutional Neural Networks: A Robust and Interpretable Model for Object Recognition Under Occlusion. *International Journal of Computer Vision*, (Economist 2017), 2020. 4

[26] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The Open Images Dataset V4: Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 1

[27] Chi Li, M. Zeeshan Zia, Quoc Huy Tran, Xiang Yu, Gregory D. Hager, and Manmohan Chandraker. Deep Supervision with Intermediate Concepts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1828–1843, 2019. 4

[28] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel

bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017. 7

[29] Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5):740–755, 2014. 5

[30] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dSprites: Disentanglement testing Sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017. 5

[31] Nikolaus Mayer et al. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proc. CVPR*, pages 4040–4048, 2016. 2

[32] David Meger, Christian Wojek, James J Little, and Bernt Schiele. Explicit Occlusion Reasoning for 3D Object Detection. In *BMVC*, pages 1–11. Citeseer, 2011. 3

[33] Wanli Ouyang and Xiaogang Wang. A discriminative deep model for pedestrian detection with occlusion handling. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3258–3265, 2012. 2, 3

[34] P Parodi, R Lancewicki, A Vijh, and J K Tsotsos. Empirically-derived estimates of the complexity of labeling line drawings of polyhedral scenes. *Artificial Intelligence*, 105:47–75, 1998. 2

[35] Bojan Pepikj, Michael Stark, Peter Gehler, and Bernt Schiele. Occlusion patterns for object class detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3286–3293, 2013. 3

[36] Ibrahim Radwan, Abhinav Dhall, and Roland Goecke. Monocular image 3D human pose estimation under self-occlusion. *Proceedings of the IEEE International Conference on Computer Vision*, pages 1888–1895, 2013. 3

[37] N Dinesh Reddy, Minh Vo, and Srinivasa G Narasimhan. Occlusion-net: 2d/3d occluded keypoint localization using graph networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7326–7335, 2019. 3, 4

[38] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963. 2

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1

[40] Caner Sahin, Guillermo Garcia-Hernando, Juil Sock, and Tae Kyun Kim. Instance- and Category-Level 6D Object Pose Estimation. In *Advances in Computer Vision and Pattern Recognition*, pages 243–265. 2019. 4

[41] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 7

[42] Roger N. Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971. 2

[43] Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 2019. 7

[44] Guang Shu, Afshin Dehghan, Omar Oreifej, Emily Hand, and Mubarak Shah. Part-based multiple-person tracking with partial occlusion handling. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1815–1821. IEEE, 2012. 3

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[46] John Slaney and Sylvie Thiébaux. Blocks World revisited. *Artificial Intelligence*, 125(1-2):119–153, 2001. 2

[47] Markus D. Solbach, Stephen Voland, Jeff Edmonds, and John K. Tsotsos. Random polyhedral scenes: An image generator for active vision system experiments, 2018. 2

[48] Richard P Stanley. Differential posets. *Journal of the American Mathematical Society*, 1(4):919–961, 1988. 5

[49] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2016. 7

[50] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 7

[51] John K. Tsotsos, Yueju Liu, Julio C. Martinez-Trujillo, Marc Pomplun, Evgueni Simine, and Kunhao Zhou. Attending to visual motion. *Computer Vision and Image Understanding*, 100(1-2 SPEC. ISS.):3–40, 2005. 1

[52] Andrea Vedaldi and Andrew Zisserman. Structured output regression for detection with partial truncation. *Advances in neural information processing systems*, 22:1928–1936, 2009. 3

[53] Tao Wang, Xuming He, and Nick Barnes. Learning structured hough voting for joint object detection and occlusion reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1790–1797, 2013. 3

[54] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An HOG-LBP human detector with partial occlusion handling. In *2009 IEEE 12th international conference on computer vision*, pages 32–39. IEEE, 2009. 3

[55] Bo Wu and Ram Nevatia. Detection and segmentation of multiple, partially occluded objects by grouping, merging, assigning part detection responses. *International journal of computer vision*, 82(2):185–204, 2009. 3

[56] Junliang Xing, Haizhou Ai, and Shihong Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1200–1207. IEEE, 2009. 3