Parcel3D: Shape Reconstruction from Single RGB Images for Applications in Transportation Logistics

Alexander Naumann, Felix Hertlein, Laura Dörr and Kai Furmans

FZI Research Center for Information Technology, Karlsruhe, Germany and
Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

{anaumann, hertlein, doerr}@fzi.de kai.furmans@kit.edu

Abstract

We focus on enabling damage and tampering detection in logistics and tackle the problem of 3D shape reconstruction of potentially damaged parcels. As input we utilize single RGB images, which corresponds to use-cases where only simple handheld devices are available, e.g. for postmen during delivery or clients on delivery. We present a novel synthetic dataset, named Parcel3D, that is based on the Google Scanned Objects (GSO) dataset and consists of more than 13,000 images of parcels with full 3D annotations. The dataset contains intact, i.e. cuboid-shaped, parcels and damaged parcels, which were generated in simulations. We work towards detecting mishandling of parcels by presenting a novel architecture called CubeRefine R-CNN, which combines estimating a 3D bounding box with an iterative mesh refinement. We benchmark our approach on Parcel3D and an existing dataset of cuboid-shaped parcels in real-world scenarios. Our results show, that while training on Parcel3D enables transfer to the real world, enabling reliable deployment in real-world scenarios is still challenging. CubeRefine R-CNN yields competitive performance in terms of Mesh AP and is the only model that directly enables deformation assessment by 3D mesh comparison and tampering detection by comparing viewpoint invariant parcel side surface representations. Dataset and code are available at https://a-nau.github.io/parcel3d.

1. Introduction

Transportation logistics and warehousing are a central part of every supply chain and play an important strategic role in the Industry 4.0 era [1]. However, several challenges need to be faced by companies working in the logistics sector: clients demand cheaper, faster and more precisely scheduled deliveries while at the same time, cities and highways are congested, and environmental concerns are of rising importance. To tackle these challenges, process automation has huge potential [2]. Key processes for automation in logistics are identification, digital measurement, damage detection and tampering recognition of packaging units, all of which we work towards with the approach presented in this work. Identification is necessary for process documentation and parcel tracking. Damage and tampering detection can be utilized to increase the safety and security along the supply chain [3]. Finally, digital measurement and volume estimation are essential for the optimization of vessel capacity usage [4].

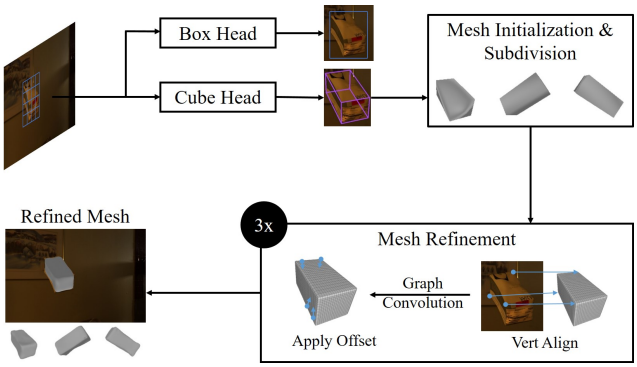


Figure 1. We take an RGB image as input and use Cube R-CNN's *Cube Head* [5] to estimate a 3D bounding box. This bounding box is subdivided and serves as initial mesh, which is refined by an iterative mesh refinement as proposed in [6]. For training and evaluation we present Parcel3D, a novel dataset of normal and damaged parcels with full 3D annotations.

Before introducing our key ideas, we shortly present important features of the logistics domain, since these characteristics influence our dataset and architecture design decisions. In logistics, packaging is usually used to handle,

transport and store goods in a safe and efficient way [7]. The most common choice are cuboid-shaped packaging units. Moreover, transportation logistics is a constrained environment, with a lot of standardizations: Loading devices, such as the EPAL Euro pallet[1], or standardized labels, such as GS1 STILL[2], to name a few examples. Finally, due to the ubiquity of logistics processes in manufacturing businesses and beyond, it is crucial to develop flexible and easy to use solutions. In fact, a study by Noceti *et al.* [3] showed, that easy integration of novel automated processes such as damage detection, is a crucial factor for technology adoption.

In this work, we present an approach for the automation of localization and shape reconstruction in logistics, which is outlined in Figure 1. We focus on the detection and shape reconstruction for intact (cuboid-shaped) and damaged parcels. By leveraging the standardizations mentioned before, i.e. by using objects with known sizes as references, length measurements are also possible with monocular images. We prioritize flexibility and thus, refrain from using expensive sensors or even multi-sensor setups. Instead, our approach solely relies on a single RGB image as input. Since RGB cameras are already integrated into many handheld devices that are used in transportation logistics, our approach is suitable for various scenarios where high flexibility is needed, such as for postmen during delivery or for clients on delivery.

There has been active research in the area of single image 3D reconstruction [6], [8]. We use Cube R-CNN [5] as base architecture, since a 3D bounding box closely describes cuboid-shaped parcels and yields a suitable shape-prior for damaged parcels. We extend Cube R-CNN by an iterative mesh refinement as used in [6], [9]. This enables us to (1) leverage a strong prior as starting point for the mesh refinement process and (2) it simultaneously estimates the original shape of parcels in form of the 3D bounding box and their potentially deformed current state. The latter enables a direct comparison between 3D meshes for damage quantification and tampering detection by comparing viewpoint invariant parcel side surface representations [3]. One important issue with 3D reconstruction, however, is the availability of suitable datasets, which are scarce due to the excessive annotation costs. Thus, most approaches are trained on synthetic data or data for specific domains (e.g. Pix3D [10]), however, no suitable dataset in the area of transportation logistics exists. To overcome this, we introduce the new synthetic dataset Parcel3D, that is built by automatically selecting suitable Google Scanned Objects (GSO) [11] models and generating damaged parcel models through simulation with Blender[3]. We employ a flexible rendering pipeline that includes varying camera parameters, lighting and scene contexts. Since the parcel texture is crucial for rendering realistic images, we also present a small synthetic cardboard texture dataset. In addition, we use a real dataset of intact parcels for evaluation and report a Box AP of up to 82.1 and Mesh AP_{50} of 32.3, which confirms the suitability of our synthetic dataset for applications in logistics. Note, that due to the lack of suitable datasets we do not evaluate damage pattern recognition and tampering detection.

The main contributions of this work are:

- we introduce Parcel3D, a novel synthetic dataset of intact and damaged parcel images with full 3D annotations that allows transfer to real images, and
- we present CubeRefine R-CNN, a novel architecture targeting single image 3D reconstruction for applications in transportation logistics, which combines 3D bounding box estimation with an iterative mesh refinement.

This work is structured as follows. In Section 2 we present an overview of related literature. Section 3 outlines the dataset generation and Section 4 our novel neural network architecture. Section 5 evaluates our approach on synthetic and real data, and Section 6 concludes the paper.

2. Related Work

To the best of our knowledge there is no prior work on shape reconstruction from single images in transportation logistics and warehousing. We review literature on applications in logistics, cuboid reconstruction from RGB images and finally, 3D reconstruction of arbitrary objects from single images in the following.

Applications in Logistics. There is work on 2D segmentation of parcels [12], [13], packaging units [14], [15] and packaging structure recognition [16], [17]. Moreover, there has been research on 3D reconstruction from RGBD images [18]–[21] and from multiple views [3]. 3D reconstruction by using RFID technology has been explored in [22]. Damage and tampering detection has been tackled by Noceti *et al.* [3] in a constrained multi-camera setup. Tampering is detected by comparing normalized parcel side surfaces and damage detection by fitting a parallelepiped across multiple views. For an in-depth review on computer vision applications in logistics, we refer to Naumann *et al.* [23].

Cuboid reconstruction. Cuboid reconstruction from single RGB images by identifying its 8 corner points in 2D has been tackled in the literature. Approaches are class agnostic, meaning that diverse object categories are considered as either cuboid or not. Xiao *et al.* [24] present such an approach in the pre-deep learning era that leverages corner

[1] See https://www.epal-pallets.org.
[2] See https://www.gs1si.org.
[3] See https://www.blender.org/.

and edge detection techniques. After the rise of deep learning, also cuboid reconstruction was tackled with Artificial Neural Networks (ANNs). Dwibedi *et al.* [25] present an approach to estimate the position of the 8 cuboid keypoints using deep learning. A similar line of work is concerned with 3D bounding box estimation for cars [26]–[28], which is reviewed in-depth by Ma *et al.* [29]. Note, that by assuming that cars are driving on the road, rotation estimation can be reduced to yaw estimation. Approaches leverage geometric priors by requiring consistent vanishing points [30] and by imposing 2D/3D consistency [31]. Recently, Brazil *et al.* [5] introduced a large benchmark for 3D object detection, which combines several existing datasets. Moreover, they present a simple and effective model for 3D object detection, called Cube R-CNN.

Single RGB image 3D reconstruction. There are many approaches for general image-based 3D reconstruction without a confinement to an object type. While the input for many approaches is a single RGB image, the output varies: representations based on voxels [8], [32], [33], meshes [9], [34], [35] and pointclouds [36], [37] are common. In addition to that, implicit representations [38], [39] have been introduced. Most reconstruction approaches focus on single instances, either by considering only images with a single instance or by employing 2D segmentation. More recently, also NeRFs [40] have been used to tackle single-view reconstruction [41]. Apart from supervised approaches, there has been work on 3D reconstruction from 2D supervision [34], unpaired image collections [42] and unsupervised reconstruction [43]–[46], since training data with ground truth 3D annotations is difficult and costly to obtain. Han *et al.* [47] present an overview of approaches from the deep learning era that leverage either single or multiple RGB images for 3D reconstruction. The reviews of Fu *et al.* [48] and Khan *et al.* [49] focus explicitly on single image 3D reconstruction.

We introduce the new dataset Parcel3D to enable research on image-based 3D reconstruction in the domain of logistics. Furthermore, we leverage the existing general 3D object detection architecture Cube R-CNN [5] and extend it by an iterative mesh refinement. Adding the iterative mesh refinement is necessary, since 3D object detection approaches are not suitable for damage detection and analysis. In contrast to other 3D reconstruction approaches, CubeRefine R-CNN directly enables comparing the original shape of a cuboid-shaped object with its current state, which is crucial for damage quantification.

3. Synthetic Dataset: Parcel3D

We present details on the generation of our synthetic dataset Parcel3D and start by describing the automatic selection process for suitable GSO [11] object models in Section 3.1. Next, the approaches to generate data for damaged parcels and for new textures are presented in Section 3.2 and Section 3.3, respectively. Finally, we present details on the rendering in Section 3.4.

3.1. Model Selection

We use GSO as a base dataset, since it has a wide variety of realistic 3D models. We create a new subset of the GSO dataset that is tailored towards our use-case in transportation logistics and warehousing by automatically selecting relevant models based on their shape. This filtering is done by evaluating each model's similarity with a surrounding cuboid. We initialize a template mesh from the surrounding cuboid and use the Chamfer Distance d_{cham} and Normal Consistency c_{norm} between this template mesh and the model mesh for comparison.
We divide the models in three categories using empirically determined thresholds for both similarity metrics. Models with $d_{cham} \leq 0.1$ and $c_{norm} \geq 0.9$ are chosen as cuboid models due to their high resemblance with the desired shape. We refer to these picked models by $\mathbb{M}_{\text{Pick}}$. The second threshold of $d_{cham} \leq 0.5$ and $c_{norm} \geq 0.8$ identifies objects that are not closely related to a cuboid in shape, yet similar. These models are denoted $\mathbb{M}_{\text{Rem}}$. All other models are referred to by $\mathbb{M}_{\text{Distr}}$. We use models from $\mathbb{M}_{\text{Distr}}$ as distractor objects, which we also render into images to prevent overfitting on rendering artifacts [50]. The models from $\mathbb{M}_{\text{Rem}}$ are not used as distractors, since their resemblance in shape with a cuboid might be confusing. The subset $\mathbb{M}_{\text{Distr}}$ contains 750 models, $\mathbb{M}_{\text{Rem}}$ contains 71 models and $\mathbb{M}_{\text{Pick}}$ contains 209 models. Exemplary instances for each of the three categories are visualized in Figure 2.

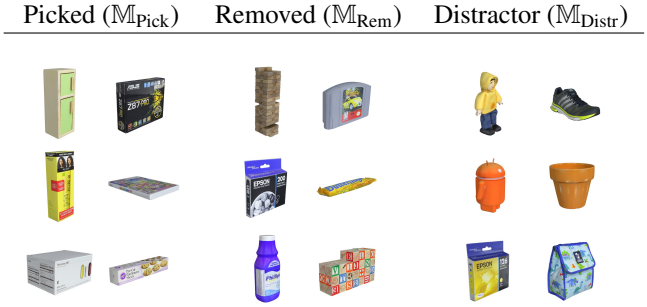


Figure 2. Samples of the three object model subsets of the GSO dataset [11] that were generated based on the models' similarity with a cuboid.

Since there are very similar models within $\mathbb{M}_{\text{Pick}}$, we combine the models into 66 groups. The grouping is done automatically by using brand and category names, since the

GSO dataset contains similar object models as seen in Figure 3 for the example of Pepsi cartons.

Figure 3. Visualization of the similarity between certain models.

3.2. Model Generation

Since we obtain only 209 suitable models from the GSO dataset, we generate 10 scaled versions for each of them. The scaling is done for each of the three dimensions separately by sampling a scaling factor from a triangular distribution with lower limit 0.5, upper limit 2 and mode 1. These models make up the subset of intact boxes.
This method for dataset generation is suitable for intact parcel recognition, however, automatically identifying suitable models for damaged boxes within the GSO or other datasets such as ShapeNet [51], is difficult. Thus, we automatically generate models for damaged boxes using physics-based simulation in Blender. For each simulation, we start by randomly sampling a base model from the previously generated subset of intact boxes. The chosen model is then simulated to be falling onto a rigid ground as seen in Figure 4. Soft body simulation is used to allow deformations during the collision. We sample falling height, angle and soft body physics parameters randomly within empirically determined ranges to obtain a wide variety of deformations. Only models from timesteps that have between 75% and 90% of their original volume are chosen as suitable models for damaged parcels. These thresholds ensure that models have at least a certain degree of deformation, while not allowing extreme changes in appearance. Furthermore, we use a RANSAC algorithm [52] to find the best rigid transformation between the original, cuboid-shaped model and the deformed model during simulation, to track the position and rotation of the object. Note, that this is necessary, since Blender does not incorporate the tracking of objects during a soft body simulation. Using this information we are able to identify the area of impact with the strongest deformation, which allows us to render damaged parcels such that the impacted area is visible. Finally, we apply a smoothing filter in Blender to the selected models.

3.3. Texture Generation

In order to obtain more variance in the textures of the models and to bias the training data towards cardboard, we generate new textures. We use a cardboard shader in Blender[4], to generate a dataset of 230 cardboard textures.

[4]See https://blendermarket.com/products/cardboard.

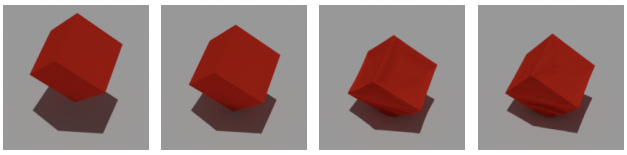


Figure 4. Visualization of the collision for damaged parcels using soft body simulation in Blender.

These textures replace the original texture of the model with a probability of 0.6, and an example is shown in Figure 5a. When the original texture is used for a damaged parcel, texture mapping is not trivial and we need to extrapolate the texture image. This extrapolation is done using pixel-wise nearest neighbor averaging and an exemplary result can be seen in Figure 5b. In addition, we randomly add to each texture

- 0-3 logos from the Large Logo Dataset (LLD) [53]
- 1 shipping label from a mix of 30 labels from [54] and 65 labels found online, with a probability of 0.6
- 0-2 fragile labels from 16 labels found online, with a probability of 0.4

An example for a final cardboard texture with labels and logos is visualized in Figure 5c.

(a) Cardboard (b) Extrapolation (c) Labels & Logos

Figure 5. Examples for generated textures: (a) Plain cardboard texture, (b) extrapolation of existing textures for damaged parcels and (c) cardboard texture with labels and logos.

3.4. Rendering Details

We sample 200 models randomly for each of the 66 groups, yielding more than 13 000 scenes, which we render with 1080×720 resolution. Damaged models and cuboid-shaped models, respectively, are sampled with a probability of 50% and textures are generated as described before. We add 0-3 randomly sampled distractor models from $\mathbb{M}_{\text{Distr}}$ to the scene and use environment maps from Gardner *et al.* [55] for realistic scene contexts. We permit an occlusion of up to 30% of the model of interest and generate a new image composition if the criteria is not met.

All assets that were used follow a 0.7, 0.15, 0.15 split between training, validation and test data. These splits were

respected in the generation of the rendered images. To have realistic poses of the objects we restrict the elevation angle to lie between $20°$ and $60°$ degrees. The azimuth angle is sampled freely for intact and between $-30°$ and $30°$ degrees for deformed models, such that the damage is visible and not self-occluded. We add small random rotations to the *lookat* configuration resulting from azimuth and elevation angle and vary the focal length slightly at random.

4. Approach

We present our novel model architecture CubeRefine R-CNN that is targeted towards reconstructing potentially deformed cuboid objects such as parcels in Section 4.1. Furthermore, we present details on our training procedure in Section 4.2.

4.1. Neural Network Architecture

Our model CubeRefine R-CNN extends Cube R-CNN [5] by adding an iterative mesh refinement (cf. Figure 1). Cube R-CNN is a general architecture that combines 2D detection with 3D bounding box estimation. Its architecture consist of a backbone network for feature extraction, which is followed by a Region Proposal Network (RPN) [56]. We follow the original work and use a DLA-34-FPN [57], [58] as backbone. The generated region proposals are then passed on to two different branches. The first branch is a *Box Head*, which outputs a 2D bounding box and the category label. The second branch estimates the 3D bounding box and is called *Cube Head*. It takes 7×7 feature maps pooled from the region-aligned backbone features and passes them to two fully connected layers with hidden dimension 1024. A final fully connected layer predicts 13 parameters which represent the 3D bounding box. Note, that this architecture could be easily extended to encompass a full Mask R-CNN [59] by adding segmentation. For details, we refer to Brazil *et al.* [5].

For the mesh refinement, we extend the *Cube Head* by subdividing its 8-point mesh triangulation output four times to obtain an initial mesh prediction of sufficient granularity. Note, that without the iterative subdivision, the mesh representation would be too coarse to accurately represent parcel deformations. The subdivided mesh is then passed on to the mesh refinement stage. We follow Gkioxari *et al.* [9], and use three refinement stages with three graph convolutions each. In each stage, image features from the backbone are aligned with the vertices of the current mesh version and graph convolutions are applied to compute a positional offset for each vertex in the mesh. These mesh offsets should morph the current mesh representation such that the mesh closely depicts the real parcel shape. We experimented with different options for message passing within the graph such as Residual Gated Graph Convolution [60], EG [61] and GATv2 [62]. Since no significant improvements were observed, we stick to the original architecture.

CubeRefine R-CNN leverages a cuboid prior, which is a valid assumption for both cuboid-shaped and most damaged parcels. Compared to Mesh R-CNN, the *Cube Head* is more lightweight than the *Voxel Head*. Moreover, our model predicts both, the original shape of the parcel and the possibly deformed current shape of the parcel at the same time. We discuss the advantages of this in more detail in Section 5.3.

4.2. Training Procedure

We follow the same training procedure for all our training runs. We choose a batch size of 16, use Stochastic Gradient Descent with Momentum (SGD+M) with a base learning rate of 0.02. The learning rate increases linearly from 0.002 over the first 1500 iterations. Subsequently, we divide the learning rate by four in iterations 7500, $12\,500$ and $17\,500$. The maximum number of iterations is set to $20\,000$.

During our experiments, we consider two different backbones, namely a ResNet-50 [63] and a DLA-34 [57], both in combination with a Feature Pyramid Network (FPN) [58]. We freeze the backbone weights at stage four and initialize them using pre-trained weights from Gkioxari *et al.* [9] and Brazil *et al.* [5].

5. Evaluation

In the following, we present our evaluation of 2D bounding box detection, 3D bounding box detection and shape reconstruction on synthetic and real data. Due to the lack of annotated real data of damaged parcels, the quantitative real-world evaluation only presents results on cuboid-shaped parcels. We benchmark our model against Pix2Mesh [6][5], Mesh R-CNN [9] and Cube R-CNN [5] by training and evaluating on the respective splits of Parcel3D. Unless stated otherwise, we use the same DLA-34-FPN backbone and three mesh refinement stages with three graph convolutions each, to enable a direct comparison between approaches. We present results for the original version of Mesh R-CNN with a ResNet-50-FPN backbone, however, focus on the comparable results in the following.

All results are summarized in Table 1 and Table 2, and we present details on the evaluation for synthetic data in Section 5.1 and for real data in Section 5.2. Finally, we summarize the findings focusing on the real-world applicability in Section 5.3.

5.1. Synthetic Data

We consider the case of intact parcels and damaged parcels separately by evaluating only on the respective subsets of the Parcel3D test dataset. The performance for 2D bounding box detection is very high for all models on our

[5] We use the implementation of Gkioxari *et al.* [9].

Model	Dataset	Box AP	Box AP_{50}	Box AP_{75}	Mesh AP_{50}	Mesh AP_{75}	Chamfer Distance (↓)	Normal Consistency
Pix2Mesh [6]	Intact	96.0 (0.5)	98.9 (0.5)	98.7 (0.2)	89.6 (1.0)	48.5 (1.6)	0.311 (0.086)	0.901 (0.001)
Mesh R-CNN (RN50) [9]	Intact	93.2 (0.4)	98.3 (0.3)	97.9 (0.3)	82.9 (1.2)	42.6 (1.1)	1.924 (1.214)	0.886 (0.001)
Mesh R-CNN [9]	Intact	95.9 (0.5)	98.9 (0.5)	98.7 (0.2)	**92.9 (1.6)**	67.0 (2.8)	0.225 (0.088)	0.914 (0.001)
Cube R-CNN [5]	Intact	97.1 (0.1)	**99.0 (0.0)**	**99.0 (0.0)**	92.0 (0.3)	74.4 (2.0)	0.159 (0.016)	0.925 (0.001)
CubeRefine R-CNN (ours)	Intact	**97.1 (0.0)**	**99.0 (0.0)**	**99.0 (0.0)**	92.8 (0.2)	**77.2 (1.2)**	**0.128 (0.002)**	**0.929 (0.001)**
Pix2Mesh [6]	Damaged	95.1 (0.6)	**99.8 (0.1)**	98.8 (0.2)	84.3 (1.2)	12.4 (1.4)	0.750 (0.553)	0.866 (0.002)
Mesh R-CNN (RN50) [9]	Damaged	92.1 (0.4)	99.6 (0.1)	98.9 (0.4)	78.8 (0.7)	9.0 (0.4)	0.599 (0.322)	0.859 (0.001)
Mesh R-CNN [9]	Damaged	94.6 (0.5)	99.2 (0.5)	98.8 (0.3)	**91.1 (0.5)**	**26.1 (1.9)**	0.860 (0.436)	**0.880 (0.002)**
Cube R-CNN [5]	Damaged	95.0 (0.2)	99.0 (0.0)	**99.0 (0.0)**	32.6 (0.5)	0.1 (0.0)	0.494 (0.004)	0.806 (0.000)
CubeRefine R-CNN (ours)	Damaged	**95.2 (0.1)**	99.0 (0.0)	**99.0 (0.0)**	70.7 (0.7)	4.1 (0.2)	**0.293 (0.003)**	0.861 (0.000)
Pix2Mesh [6]	Real	74.4 (1.9)	93.4 (1.7)	89.3 (2.4)	27.8 (2.1)	2.3 (0.6)	2.112 (0.060)	0.744 (0.006)
Mesh R-CNN (RN50) [9]	Real	**82.1 (0.7)**	**99.0 (0.0)**	**97.8 (0.1)**	32.0 (0.4)	5.0 (1.0)	1.965 (0.050)	0.756 (0.002)
Mesh R-CNN [9]	Real	70.6 (5.0)	89.2 (5.9)	84.4 (5.7)	29.4 (2.7)	4.9 (1.5)	2.153 (0.073)	0.742 (0.008)
Cube R-CNN [5]	Real	43.4 (6.9)	52.8 (8.3)	49.9 (7.3)	30.1 (5.8)	**13.3 (4.3)**	0.875 (0.041)	0.808 (0.003)
CubeRefine R-CNN (ours)	Real	41.5 (5.8)	50.3 (6.6)	47.6 (6.5)	**32.3 (4.2)**	13.1 (3.0)	**0.814 (0.062)**	**0.828 (0.006)**

Table 1. Quantitative performance analysis of mesh reconstruction on different datasets. The Mesh AP is the mean area under the Precision-Recall curve for F1@0.3$> x$, as in [9]. We repeated all trainings five times and report mean values with standard deviations in parentheses. The best mean performance for each dataset type is highlighted.

Model	Dataset	AP3D	$AP3D_{15}$	$AP3D_{25}$
Cube R-CNN [5]	Intact	69.5 (0.8)	81.6 (1.1)	74.4 (1.4)
CubeRefine R-CNN (ours)	Intact	69.3 (0.6)	80.9 (0.5)	74.1 (1.1)
Cube R-CNN [5]	Damaged	86.6 (0.3)	94.4 (0.6)	89.9 (0.8)
CubeRefine R-CNN (ours)	Damaged	86.5 (0.6)	94.6 (0.6)	89.7 (0.6)
Cube R-CNN [5]	Real	53.3 (8.6)	53.8 (8.7)	53.8 (8.7)
CubeRefine R-CNN (ours)	Real	50.6 (6.8)	51.1 (6.8)	51.1 (6.8)

Table 2. Quantitative performance analysis of 3D object detection for Cube R-CNN and CubeRefine R-CNN on different datasets. The average precision for 3D IoU (AP3D) is computed as in [5]. We repeated all trainings five times and report mean values with standard deviations in parentheses.

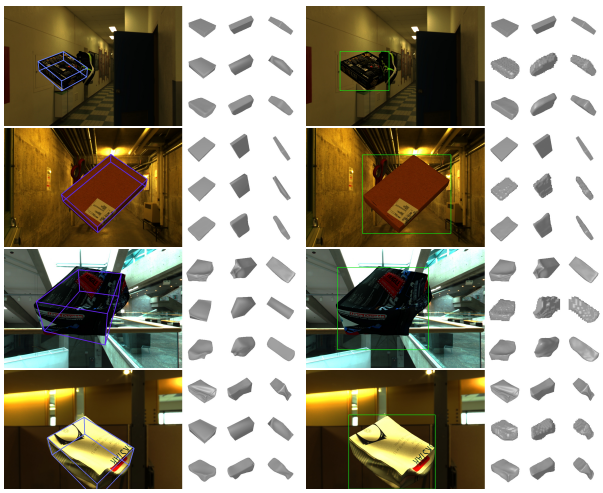
presented synthetic dataset Parcel3D with the lowest observed Box AP being 92.1 (cf. Table 1).

Considering 3D bounding box detection in the case of cuboid-shaped parcels, Cube R-CNN and CubeRefine R-CNN perform best w.r.t. Mesh AP_{75}, Chamfer Distance and Normal Consistency, since they explicitly model cuboid-shaped objects. Our additional mesh refinement increases performance compared to the base model Cube R-CNN by 2.8 percentage points in Mesh AP_{75}. Mesh R-CNN still performs competitively, and the qualitative inspection (cf. Figure 6) suggests that differences mainly stem from difficulties in reconstructing the nonvisible, (self-)occluded parts of objects. Cube R-CNN and CubeRefine R-CNN do not suffer from this problem as much, since symmetry is imposed by the predicted 3D bounding box.

Considering only damaged parcels, we observe that predicting a voxel occupancy grid as done in Mesh R-CNN is advantageous. Mesh R-CNN performs best in Mesh AP and Normal Consistency. Despite high-quality 3D object detection, as suggested by the results in Table 2, CubeRefine R-CNN has difficulties to adopt to the fine-grained meshes of damaged parcels. This is observed in the considerably lower Mesh AP. However, the better Chamfer Distance suggests that general alignment with the ground truth is very high for CubeRefine R-CNN. This can also be observed in qualitative samples as visualized in Figure 6 and might be caused by the symmetry the 3D bounding box imposes for (self-)occluded object parts. Cube R-CNN performs poorly, as it only predicts 3D bounding boxes and thus, cannot take the damages into account.

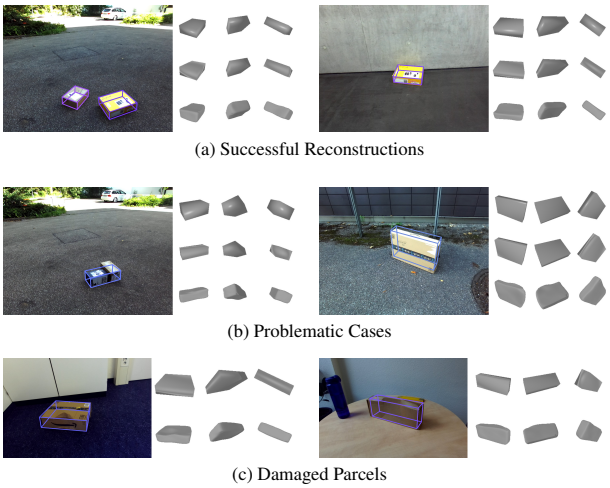
5.2. Real Data

For the evaluation of the usability of our approach in real-world applications, we use a dataset of parcels photos



(a) CubeRefine R-CNN (ours) (b) Mesh R-CNN [9]

Figure 6. Exemplary qualitative results for synthetic intact (row 1, 2) and damaged parcels (row 3, 4) for (a) CubeRefine R-CNN and (b) Mesh R-CNN. Per model, the input image with the detected 2D or 3D bounding box is shown on the left, and a 3×3 grid of mesh reconstructions on the right. Each column of the grid shows a different viewing angle, and the rows contain ground truth, 3D bounding box or voxelization (depending on the model) and refined mesh, respectively.



(a) Successful Reconstructions

(b) Problematic Cases

(c) Damaged Parcels

Figure 7. Exemplary qualitative results for real parcels using CubeRefine R-CNN. We show the input image with the projected 3D bounding box on the left, and a 3×3 grid of mesh reconstructions on the right. Each column shows a different viewing angle, and the rows contain ground truth, 3D bounding box and refined mesh, respectively. Note, that for damaged parcels no ground truth is available.

in various environments [13]. The dataset was generated using a custom camera rig to capture images with a depth and a stereo camera at the same time. The depth information is then used to automatically generate annotations, which can be projected onto the stereo images. The validation dataset comprises 96 and the test dataset 297 images. Note, that it contains only normal parcels, since the annotation generation process was automated using the assumption of a cuboid shape.

Shape reconstruction on real images of cuboid-shaped parcels is more challenging due to the reality gap, as can be seen from the generally lower performance in Table 1. CubeRefine R-CNN performs best despite having a low 2D bounding box detection precision compared to Mesh R-CNN. Note, that Mesh AP, Chamfer Distance and Normal Consistency were computed on meshes normalized within a unit cube, due to the scale ambiguity. While Cube R-CNN is able to estimate scale, our synthetic training data is generated randomly, and thus, does not allow a scale transfer to the real world.

We present qualitative samples in Figure 7 and observe accurate reconstructions, when the object is localized correctly. However, common error cases include not being able to distinguish nearby positioned parcels and inaccurate or missing localizations (cf. Figure 7b). Since there are no real-world datasets with full 3D annotations, we focus on brief insights into our qualitative inspection of damaged parcels. The simulated deformation process that was presented in Section 3.2 does not seem to represent the great variance of real-world deformations closely enough. Thus, performance on real-world data is still limited as can be seen in Figure 7c.

5.3. Applicability Summary

We summarize the advantages and limitations of our approach, and present brief insights into using damage quantification and tampering detection in practice.

Advantages. We argue, that while Mesh R-CNN performs best in the case of damaged parcels, our approach is still advantageous for real-world application due to the following reasons: (1) our approach is more lightweight and predicts both the current, potentially deformed shape of an object and its original shape at the same time. This allows a direct 3D mesh comparison between the original and the deformed shape for damage quantification. (2) The lower Mesh AP and better Chamfer Distance compared to Mesh R-CNN suggest that our model represents the overall damage pattern well, however, is not as detailed as Mesh R-CNN. We argue, that this is sufficient for damage pattern recognition in 3D, which is only enabled by our model. (3) 3D bounding box detection enables using viewpoint invariant parcel side surface representations for tampering detection, as will be explained in the respective paragraph.

Limitations. While CubeRefine R-CNN has important advantages for real-world use-cases, enabling reliable deployment in real-world scenarios is still challenging, presumably due to the constrained variance of deformations within Parcel3D and the domain shift caused by our training on synthetic data. Furthermore, it is important to note that we focus on deformations of the packaging and do not treat other types of damages which frequently occur in practice (e.g. water damage). It is also not possible to reliably infer the impact of packaging deformations on the state of the transported good. This information is essential to estimate economic damages.

Damage Quantification. To utilize our model for automated deformation quantification and pattern recognition, metrics for 3D mesh comparison are necessary. The change in volume between the original and current shape constitutes a simple metric that can be readily computed and interpreted. However, mere volume analysis does not take the deformation location into account. To remedy this, extending the axis-aligned pointcloud representation of the original 3D model by the per-point distance to the nearest neighbor of its potentially deformed version, and clustering in this 4D space can help to identify areas that underwent the strongest deformations. Further clustering across parcel instances can provide insights into damage patterns. Moreover, normalized voxel grid occupancy differences can be analyzed by considering the union of the voxelized meshes and subtracting their intersection.

Tampering Detection. From the 3D bounding box output of CubeRefine R-CNN we can infer the visible parcel side surfaces and project them back onto the image. For each such parcel side surface, a perspective transformation can be applied to obtain normalized fronto-parallel views. These representations have already been successfully used for tampering detection [3] and re-identification [30]. For tampering detection, recent advances in change detection [64] could be leveraged.

6. Conclusion

In this work, we present an approach for simultaneous detection and shape reconstruction of intact and damaged parcels from single RGB images, called CubeRefine R-CNN. We extend Cube R-CNN [5] by an iterative mesh refinement to benefit from a cuboid prior, while at the same time enabling adjustments for damages in parcels. To overcome the lack of existing datasets, we also introduce Parcel3D, a novel synthetic dataset of intact and damaged parcel images with full 3D annotations that is suitable for applications in transportation logistics and warehousing. To generate the dataset, we leverage selected data from the Google Scanned Objects (GSO) dataset [11]. We combine these with models of damaged parcels that were generated using physics-based simulations and a new dataset of synthetic cardboard textures.

Our approach outperforms existing baselines for intact parcels and performs competitively for damaged parcels. While Mesh R-CNN [9] yields the best results for the case of damaged parcels, our approach is the only one directly enabling deformation assessment and tampering detection. The results of our approach are promising to help identifying systematic mishandling of parcels, especially in scenarios where only simple sensor data is available, as during last mile delivery to a client for example. However, the reliable deployment in real-world scenarios is still challenging. More diverse and realistic shape deformation types within the dataset and real-world training data are promising improvements.

Acknowledgement: We thank Zeyu Wang for helping with the implementation of the rendering pipeline and for fruitful discussions in the early stages of the project.

References

[1] C. S. Tang and L. P. Veelenturf, "The strategic role of logistics in the industry 4.0 era," *Transportation Research Part E: Logistics and Transportation Review*, 2019.

[2] M. Woschank, E. Rauch, and H. Zsifkovits, "A Review of Further Directions for Artificial Intelligence, Machine Learning, and Deep Learning in Smart Logistics," *Sustainability*, 2020.

[3] N. Noceti, L. Zini, and F. Odone, "A multi-camera system for damage and tampering detection in a postal security framework," *EURASIP Journal on Image and Video Processing*, 2018.

[4] X. Zhao, J. A. Bennell, T. Bektaş, and K. Dowsland, "A comparative review of 3D container loading algorithms: A comparative review of 3D container loading algorithms," *International Transactions in Operational Research*, 2016.

[5] G. Brazil, A. Kumar, J. Straub, N. Ravi, J. Johnson, and G. Gkioxari, "Omni3D: A Large Benchmark and Model for 3D Object Detection in the Wild," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2023.

[6] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images," in *European Conference on Computer Vision (ECCV)*, 2018.

[7] M. Saghir, "The concept of packaging logistics," *World Conference on Production and Operations Management Society*, 2004.

[8] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction," in *European Conference on Computer Vision (ECCV)*, 2016.

[9] G. Gkioxari, J. Malik, and J. Johnson, "Mesh R-CNN," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[10] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, *et al.*, "Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, *et al.*, "Google Scanned Objects: A High-Quality Dataset of 3D Scanned Household Items," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022.

[12] A. Naumann, L. Dörr, N. O. Salscheider, and K. Furmans, "Refined Plane Segmentation for Cuboid-Shaped Objects by Leveraging Edge Detection," in *International Conference On Machine Learning And Applications*, 2020.

[13] A. Naumann, F. Hertlein, B. Zhou, L. Dörr, and K. Furmans, "Scrape, cut, paste and learn: Automated dataset generation applied to parcel logistics," in *IEEE Conference on Machine Learning and Applications (ICMLA)*, 2022.

[14] C. Mayershofer, T. Ge, and J. Fottner, "Towards Fully-Synthetic Training for Industrial Applications," *10th International Conference on Logistics, Informatics and Service Sciences (LISS)*, 2020.

[15] C. Mayershofer, D.-M. Holm, B. Molter, and J. Fottner, "LOCO: Logistics objects in context," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2020.

[16] L. Dörr, F. Brandt, M. Pouls, and A. Naumann, "Fully-Automated Packaging Structure Recognition in Logistics Environments," in *International Conference on Emerging Technologies and Factory Automation*, 2020.

[17] L. Dörr, F. Brandt, A. Naumann, and M. Pouls, "TetraPackNet: Four-Corner-Based Object Detection in Logistics Use-Cases," in *DAGM German Conference on Pattern Recognition*, 2021.

[18] X. Li, I. Y.-H. Chen, S. Thomas, and B. A. MacDonald, "Using Kinect for monitoring warehouse order picking operations," *Proceedings of Australasian Conference on Robotics and Automation*, 2012.

[19] C. Prasse, J. Stenzel, A. Böckenkamp, B. Rudak, K. Lorenz, F. Weichert, *et al.*, "New Approaches for Singularization in Logistic Applications Using Low Cost 3D Sensors," in *Sensing Technology: Current Status and Future Trends IV*, 2015.

[20] N. T. Son, B. N. Anh, T. Q. Ban, and T. B. Duong, "A Method to Construct Automatic Object Bounding-Box Estimation System using 3D Cameras," *International Journal of Science and Research (IJSR)*, 2017.

[21] P. Arpenti, R. Caccavale, G. Paduano, G. Andrea Fontanelli, V. Lippiello, L. Villani, *et al.*, "RGB-D Recognition and Localization of Cases for Robotic Depalletizing in Supermarkets," *IEEE Robotics and Automation Letters*, 2020.

[22] Y. Bu, L. Xie, J. Liu, B. He, Y. Gong, and S. Lu, "3-Dimensional Reconstruction on Tagged Packages via RFID Systems," in *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2017.

[23] A. Naumann, F. Hertlein, L. Doerr, S. Thoma, and K. Furmans. (2023). Literature Review: Computer Vision Applications in Transportation Logistics and Warehousing, [Online]. Available: http://arxiv.org/abs/2304.06009.

[24] J. Xiao, B. Russell, and A. Torralba, "Localizing 3D cuboids in single-view images," *Conference on Neural Information Processing Systems*, 2012.

[25] D. Dwibedi, T. Malisiewicz, V. Badrinarayanan, and A. Rabinovich. (2016). Deep Cuboid Detection: Beyond 2D Bounding Boxes, [Online]. Available: http://arxiv.org/abs/1611.10010.

[26] J. Fang, L. Zhou, and G. Liu. (2019). 3D Bounding Box Estimation for Autonomous Vehicles by Cascaded Geometric Constraints and Depurated 2D Detections Using 3D Results, [Online]. Available: http://arxiv.org/abs/1909.01867.

[27] Y. Liu, Y. Yixuan, and M. Liu, "Ground-aware Monocular 3D Object Detection for Autonomous Driving," *IEEE Robotics and Automation Letters*, 2021.

[28] A. Kumar, G. Brazil, E. Corona, A. Parchami, and X. Liu, "DEVIANT: Depth EquiVarIAnt NeTwork for Monocular 3D Object Detection," in *Computer Vision – ECCV 2022*, 2022.

[29] X. Ma, W. Ouyang, A. Simonelli, and E. Ricci. (2022). 3D Object Detection from Images for Autonomous Driving: A Survey, [Online]. Available: http://arxiv.org/abs/2202.02980.

[30] Z. Rui, G. Zongyuan, D. Simon, S. Sridha, and F. Clinton, "Geometry-Constrained Car Recognition Using a 3D Perspective Network," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[31] P. Li, H. Zhao, P. Liu, and F. Cao, "RTM3D: Real-Time Monocular 3D Detection from Object Keypoints for Autonomous Driving," in *Computer Vision – ECCV 2020*, 2020.

[32] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang, "Pix2Vox: Context-aware 3D Reconstruction from Single and Multi-view Images," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[33] S. Yang, M. Xu, H. Xie, S. Perry, and J. Xia, "Single-View 3D Object Reconstruction from Shape Priors in Memory," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[34] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning Category-Specific Mesh Reconstruction from Image Collections," in *European Conferenceon Computer Vision (ECCV)*, 2018.

[35] C. Wen, Y. Zhang, Z. Li, and Y. Fu, "Pixel2Mesh++: Multi-View 3D Mesh Generation via Deformation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[36] H. Fan, H. Su, and L. Guibas, "A Point Set Generation Network for 3D Object Reconstruction from a Single Image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[37] M. Gadelha, R. Wang, and S. Maji, "Multiresolution Tree Networks for 3D Point Cloud Processing," *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[38] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy Networks: Learning 3D Reconstruction in Function Space," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[39] S. Zakharov, R. A. Ambrus, V. C. Guizilini, D. Park, W. Kehl, F. Durand, *et al.*, "Single-Shot Scene Reconstruction," in *5th Annual Conference on Robot Learning*, 2021.

[40] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis," in *Computer Vision – ECCV 2020*, 2020.

[41] N. Muller, A. Simonelli, L. Porzi, S. R. Bulo, M. Niessner, and P. Kontschieder, "AutoRF: Learning 3D Object Radiance Fields from Single View Observations," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[42] S. Duggal and D. Pathak, "Topologically-Aware Deformation Fields for Single-View 3D Reconstruction," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[43] E. Insafutdinov and A. Dosovitskiy, "Unsupervised Learning of Shape and Pose with Differentiable Point Clouds," in *Conference on Neural Information Processing Systems (NeuRIPS)*, 2018.

[44] K. L. Navaneet, A. Mathew, S. Kashyap, W.-C. Hung, V. Jampani, and R. Venkatesh Babu, "From Image Collections to Point Clouds With Self-Supervised Shape and Pose Networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[45] S. Wu, C. Rupprecht, and A. Vedaldi, "Unsupervised Learning of Probably Symmetric Deformable 3D Objects From Images in the Wild," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[46] T. Hu, L. Wang, X. Xu, S. Liu, and J. Jia, "Self-Supervised 3D Mesh Reconstruction from Single Images," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[47] X.-F. Han, H. Laga, and M. Bennamoun, "Image-based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[48] K. Fu, J. Peng, Q. He, and H. Zhang, "Single image 3D object reconstruction based on deep learning: A review," *Multimedia Tools and Applications*, 2021.

[49] M. S. U. Khan, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Three-Dimensional Reconstruction from a Single RGB Image Using Deep Learning: A Review," *Journal of Imaging*, 2022.

[50] D. Dwibedi, I. Misra, and M. Hebert, "Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[51] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, *et al.* (2015). ShapeNet: An Information-Rich 3D Model Repository, [Online]. Available: http://arxiv.org/abs/1512.03012.

[52] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," in *Readings in Computer Vision*, 1987.

[53] A. Sage, E. Agustsson, R. Timofte, and L. Van Gool. (2017). LLD - large logo dataset - version 0.1, [Online]. Available: https://data.vision.ee.ethz.ch/cvl/lld.

[54] L. Dörr, F. Brandt, A. Meyer, and M. Pouls, "Lean Training Data Generation for Planar Object Detection Models in Unsteady Logistics Contexts," in *IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019.

[55] M.-A. Gardner, K. Sunkavalli, E. Yumer, X. Shen, E. Gambaretto, C. Gagné, *et al.*, "Learning to predict indoor illumination from a single image," *ACM Transactions on Graphics*, 2017.

[56] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[57] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep Layer Aggregation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[58] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[59] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[60] X. Bresson and T. Laurent. (2018). Residual Gated Graph ConvNets, [Online]. Available: http://arxiv.org/abs/1711.07553.

[61] S. A. Tailor, F. Opolka, P. Lio, and N. D. Lane, "Do We Need Anisotropic Graph Neural Networks?" In *International Conference on Learning Representations*, 2022.

[62] S. Brody, U. Alon, and E. Yahav, "How Attentive are Graph Attention Networks?" In *International Conference on Learning Representations*, 2022.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[64] W. Shi, M. Zhang, R. Zhang, S. Chen, and Z. Zhan, "Change Detection Based on Artificial Intelligence: State-of-the-Art and Challenges," *Remote Sensing*, 2020.