Improving CNN-based Planar Object Detection with Geometric Prior Knowledge

Jianxiong Cai¹, Hongyu Chen¹, Laurent Kneip² and Sören Schwertfeger²

Submitted to:
IEEE Conference on Robotics and Automation (ICRA) 2020

This is a publication from the Mobile Autonomous Robotic Systems Lab (MARS Lab), School of Information Science and Technology (SIST) of ShanghaiTech University. For this and other publications from the MARS Lab please visit: https://robotics.shanghaitech.edu.cn/publications

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Improving CNN-based Planar Object Detection with Geometric Prior Knowledge

Jianxiong Cai¹, Hongyu Chen¹, Laurent Kneip² and Sören Schwertfeger²

Abstract—In this paper, we focus on the question: how might mobile robots take advantage of affordable RGB-D sensors for object detection? Although current CNN-based object detectors have achieved impressive results, there are three main drawbacks for practical usage on mobile robots: 1) It is hard and time-consuming to collect and annotate large-scale training sets. 2) It usually needs a long training time. 3) CNN-based object detection shows significant weakness in predicting location. We propose a novel approach for the detection of planar objects, which rectifies images with geometric information to compensate for the perspective distortion before feeding it to the CNN detector module, typically a CNN-based detector like YOLO or MASK RCNN. By dealing with the perspective distortion in advance, we eliminate the need for the CNN detector to learn that. Experiments show that this approach significantly boosts the detection performance. Besides, it effectively reduces the number of training images required. In addition to the novel detection framework proposed, we also release an RGB-D dataset for hazmat sign detection. To the best of our knowledge, this is the first public-available hazmat sign detection dataset with RGB-D sensors.

I. INTRODUCTION

Affordable RGB-D sensors (such as Microsoft Kinect and Intel Realsense) are becoming more and more common in the modern robotics, due to their cheap price and portable size. With the increasing attention on semantic understanding for mobile autonomous robots, we ask ourself the question: how might mobile robots take advantage of the depth information (RGB-D) for object detection in real-world robotics applications. In this work, we take hazmat sign (see Fig. 1) detection as an example task for using geometric image rectification to aid a CNN-based object detector.

A. Background

For modern robotics, image based object detection methods can be categorized into two main approaches: Feature-based matching [1] [2] [3] and convolutional neural network (CNN) [4] [5] [6]. Since the initial proposal of RCNN [7], state-of-art CNN-based approaches [4] [5] have achieved impressive results on large-scale standardized datasets[8] [9].

A common approach for current state-of-art object detection networks is to feed images into the CNN detection network directly, as an end-to-end solution. This eliminates the need for feature engineering and enables CNN networks to learn features automatically. With sufficient training images, state-of-art detection networks can thus distinguish different signs with good performance.

However, feeding images directly to CNN detector has two main drawbacks, in terms of mobile robotics applications. First, the CNN detector needs a large number of images for training, but creating such a training set is considerably challenging in most cases. In practice, collecting the training set turns out to be very time-consumption. As we expect the CNN detector to automatically learn how to deal with geometrical information for perspective distortion and multiple scales, the network needs training data on different points of view. Besides, given the huge amount of human resource and time needed for dataset collection, it is more difficult for real-world mobile robotics.

Second, even with a nice training set, there is no guarantee that the detector can correctly learn perspective distortion. One key issue is that an end-to-end solution requires deeper and more complicated neural networks. This is because the network needs more layers for viewpoint angle estimation and perspective distortion. In some cases, this might lead to an over-fitting situation, where the network fails to learn useful features.

Moreover, previous research work has shown that it is hard for CNN detection framework to learn from raw RGB-D images automatically[10]. Perspective distortion is introduced when the image is not captured in a canonical view of the object. Intuitively, parallel 3D lines are no longer parallel on the 2D image. This results in differences between images from different viewing angles. For planar objects, a homography matrix can be used to transfer between those images.

In this work, we propose to utilize the depth information for rectifying RGB information with homography matrix. In short, a homography matrix is calculated from depth information to transform input RGB images to the canonical view. The CNN detector then takes rectified RGB images as input to perform detection. The final detection result on rectified RGB images is transformed back the original images in the end.

The proposed method provides two key advantages. First, the image rectification simplifies the problem for CNN
detector after image rectification. Typical CNN detectors suffer from multi-scale input images and show noticeable weakness in bounding box regression accuracy. Image rectification can avoid the multi-scale problem to some extent. For the bounding box accuracy, as all images have been rectified to the canonical view, the problem gets slightly easier for CNN. Secondly, because the CNN detector only takes canonical-view images as input, the proposed method requires a smaller training set, which reduces the workload for practical deployment with mobile robotics.

What should be learned? Perspective distortion is something for which exact models are known, so we should not have to learn this. This combination of the geometric and semantic model is motivated by having a smaller, more efficient network that focuses on the part of the problem which they are best and most needed for.

B. Contribution

Our contributions are:

- We propose a feasible way of combining geometric information with CNN detectors to improve detection performance.
- Our approach shows good tolerance towards noise in depth images, with homography based image rectification.
- We successfully reduce the number of images needed for training the CNN detector, because perspective distortion has been dealt with in advance. It is especially meaningful for practical usage in mobile robotics when facing a new environment or target object.
- We release a new hazmat sign detection dataset. To the best of our knowledge, it is the first RGB-D hazmat sign detection dataset.

C. Paper Organization

Section II discusses related work while Section III introduces the new detection framework with the homography-based rectification. Section IV presents the new RGB-D hazmat sign detection dataset. Experiments results are shown in Section V and conclusions are drawn in Section VI.

II. RELATED WORK

A. Hazmat Sign Detection

Hazmat sign detection has been studied by the robotics community for a long time. It is still challenging because of detection speed, illumination changes, background similarity, size variety, and inter-class variety. Due to the importance of hazmat sign detection for rescue robots, the RoboCup Rescue League competition has the task of detecting hazmat signs on system inspection stage and exploration tasks[11][12].

Existing sign detection algorithms can be categorized into 4 types: 1) color-based 2) shape-based 3) saliency-based 4) Deep neural network-based. Typical color-based methods use color as strong prior knowledge and key features, like the color histogram [13]. However, color-based methods can be easily affected by illumination changes. Shaped-based methods usually use feature detection and matching, like SIFT or SURF [13][14]. Those methods, however, can be affected by perspective distortion from different viewpoints. Saliency-based methods use saliency detection to speed up the detection process and get the region of interest where the hazmat sign might locate in [15][16]. Recent research focused on CNN methods, such as [17][18].

B. Image Rectification

There are two main methods for image rectification, 3D reprojection or 2D homography matrix transformation. For 3D reprojection, it first generates a 3D point cloud from input (RGB-D) and then reprojects the points to the new image plane. In the 2D homography matrix transformation approach we directly apply a 3 by 3 matrix \((H)\) on the input RGB image.

\[
I_2(x) = I_1(Hx)
\]

Where \(I_1\) and \(I_2\) are images, \(x\) are the homogenous coordinates.

In the proposed approach, rectification via homography matrix is used, due to its robustness. Low-cost RGB-D cameras (like Intel Realsense) tend to have a quite noisy depth map. Through our experiments, we found that the pointcloud recovered from the depth map is too noisy for the other method. As it does not operate on 3D space, the homography method only requires the pointcloud to be accurate enough for plane extraction, which can be easily met via RANSAC [19].

C. Combining geometry and CNN

With the impressive performance of CNN networks, recent works mainly focus on end-to-end CNN solutions [20][6]. Unfortunately, it remains unclear about how to take good advantage from the additional depth information from RGB-D. Feeding raw RGB-D images into CNN will not help much, as the depth information with its high variance makes the learning process of CNN detector even harder [21].

Some work instead utilizes depth information geometrically. [22] combines CNN with geometry priors by using CNN for image-based 2D part location estimates and assumes the geometry model for 3D pose reconstruction. [23] utilize the geometric constraints on translation imposed by the 2D bounding box to recover a stable and accurate 3D object pose. [24] is using geometric shape features to boost the performance of neural networks. [25] uses depth for proposal generation with contour detection. It also encodes the depth information before forwarding it to the CNN detector. One close work to this paper is [26]. They first do contour detection on RGB images and then use depth for rectifying the proposal to canonical view, so that it becomes easier for feature matching. However, their experiments were performed on conventional feature descriptors, such as SIFT [1] and ORB [3]. Different from [26], our system does not require contour detection. This extends the usage to complicated planar objects, where contour detection might fail. To the best of our knowledge, there is no concrete research on use image rectification for CNN-based object detectors, especially with the consideration of mobile robotics.
Fig. 2: The proposed approach rectifies the image from the original viewpoint to a virtual viewpoint (canonical view).

III. METHOD

A. Overview Framework

The detection framework takes 3 inputs: 1) RGB images 2) point cloud 3) Camera intrinsic matrix. The Intel Realsense Driver provides the point cloud as output, so we use it as the input. This could be easily substituted with using depth images in the implementation.

B. Rectification Pipeline

The goal of the algorithm is to rectify the input images based on planes detected in the 3D data, as shown in Figure 2. The overall pipeline contains the following stages and is shown in Figures 4 and 5. The input image is rectified to multiple parallel viewpoints, so the rectification module needs to compute a set of homography matrices as the output.

1) Estimate plane segmentation from 3D point cloud
2) Calculate virtual canonical viewpoint
3) Compute the initial homography matrix for image rectification (rectify to the virtual canonical viewpoint)
4) Refine the homography matrix by applying translation matrix (sliding through the image)
5) Apply rectification matrices to get rectified images

C. Plane Segmentation

1) Plane Estimation: RANSAC [19] is used to estimate plane parameters from the input point cloud. The usage of RANSAC shows a good robustness for plane estimation on noisy inputs, which then improve the robustness of the whole detection framework. In this step, we obtain the major planes in the scene. For each plane, the following are calculated (with respect to the original viewpoint, \(v_1\)):

- the plane parameters \(\pi\).

\[
\pi = (n^T, d)
\]

where \(n^T\) is the normal of the plane, \(\|n\| = 1\), \(d\) is the distance from viewpoint to plane.

- centroid point \(P_c\)
- boundary points \(X_{i,j}\) where \(i \in \mathbb{Z}^+\)

2) Unique Normal: Each plane has two normals, in opposite directions. In order to calculate the new viewpoint, which has to be correctly aligned with the original viewpoint, it is necessary to use one unique normal for each plane. The unique normal is defined as the one not facing toward the origin.

\[
P_c \cdot n^T > 0 \quad (1)
\]

D. Calculating the Virtual Canonical Viewpoint

In order to transform images to the canonical view, a virtual canonical viewpoint needs to be calculated from the plane parameters. We set the virtual viewpoint \((v_2)\) at a fixed distance away from the plane centroid. In our experiments, we set this to 1.2 meter to align with the training set, as the training set is collected with the distance of 0.8 and 1.2 meter.

The new viewpoint is calculated as such:

\[
T_2 \cdot T_1 = \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & t \\ 0 & 1 \end{bmatrix} \quad (2)
\]

\[
t = P_c - d n^T \quad (3)
\]

\[
R = \begin{bmatrix} 0 & 0 & n^T \end{bmatrix} \quad (4)
\]

Where \(T_2 \cdot T_1\) is a 4 * 4 matrix denotes the position and orientation of \(v_2\) in \(v_1\). \(n^T\) is the normal of the plane, \(\|n\| = 1\), \(d\) is the distance from viewpoint to plane.

Fig. 3: The middle image V2 shows the partial loss of the plane after applying \(H_2^1\). Result image size is kept same as input. Green lines denotes the border of a plane, solid rectangular are image patches, blue bounding box are the bounding box of the plane transformed to V2. The last image V3 is the visualization of system output matrices after applying the transformation. The resulting rectified images are with 50\% of overlapping with its neighbors. The result images are the rectangular boxes in black, orange and blue (img1, img2 and img3)
3D PointCloud
RGB Images
Camera Model

Rectification Module

Rectified Images (canonical view)  → CNN detector  → Detection result (on rectified images)  → Warp back  → Final detection result

Homography Matrices

Fig. 4: Proposed detection framework (testing pipeline)

RGB Images Rectification Module

Rectified Images
(canonical view)

CNN detector
Detection result
(on rectified images)
Warp back
Final detection result

Homography Matrices

3D PointCloud
Camera Model

Plane Extraction (RANSAC)

3D Points

Calculate Viewpoints

Virtual Viewpoints

Calculate Homography Matrix (2D)

Initial Homography Matrix Io

RGB Images

Rectified Images

Corresponding Homography Matrices Hi

Fig. 5: Overall Geometric Image Rectification Pipeline (Rectification Module)

E. Calculate Homography Matrix for Virtual Viewpoint (2D)

Once the virtual viewpoint is calculated, it is easy to calculate an equivalent homography matrix that denotes the transformation between \( v_1 \) and \( v_2 \). Non-robust DLT (Direct Linear Transform) [27] is used to compute the homography.

For homography calculation, four points are sampled from the 3D plane, denoted as \( X_i \). On both camera viewpoints (the original viewpoint and the new one), we project the 3D points to 2D images, denoted as \( x^1_i \) and \( x^2_i \). As shown in Figure 2.

As the plane is an ideal infinite plane, and since the 2D image points are all obtained from re-projection, non-robust DLT (Direct Linear Transform) [27] is used to compute the homography.

\[
I_2(x) = I_1(Hx)
\]

(5)

where \( x \) is homogenous coordinates.

F. Refine Homography Matrix

At this point, we have a homography matrix which transforms side view images to the canonical view. However, there is still one minor issue: Some pixels go out of view, due to the FoV (field of view) of the camera. As shown in Figure 3, in some case, only part of the pixels are included in the resulting image.

1) Bounding box around the plane: In our case, as the plane is a finite plane defined by boundary points, the first step is to calculate the tight bounding box around the reprojected plane in \( v_2 \), as is shown in blue in Figure 3.

\[
x^2_b = P * T_2^2 * X_b
\]

(6)

Where \( X_b \) denotes points around plane boundary, \( P \) denotes projection matrix.

From \( x^2_b \), we can easily calculate a tight bounding box by taking the minimum and maximum. Specifically, we calculate the top-left corner \( x_{top,\text{left}} \), the bounding box height \( H_{bbox} \) and the width \( W_{bbox} \).

2) Refinement Algorithm: This next step is to calculate the final rectification matrices which produce the final resulting images. As is shown in Equation 7 we slide the camera window through the reprojected plane on \( v_2 \). Every two sliding images have 50% of overlap either horizontally or vertically, so that hazmat signs around the border will present in the center of its sliding image. See Figure 3 for an example.

Denote \( x_{top,\text{left}} = (x_{top,\text{left}}, y_{top,\text{left}}) \) as the top-left corner of the plane bounding box on \( v_2 \), \( H_{bbox}, W_{bbox} \) as the bounding box height and width. \( H_{img}, W_{img} \) as height and width of the resulting image.

\[
H = \begin{bmatrix}
1 & 0 & \frac{H_{img}}{W_{img}} * (i - 1) \\
0 & 1 & \frac{W_{img}}{2} * (j - 1) \\
0 & 0 & 1
\end{bmatrix} \cdot \begin{bmatrix}
1 & 0 & -x_{top,\text{left}} \\
0 & 1 & -y_{top,\text{left}} \\
0 & 0 & 1
\end{bmatrix}
\]

(7)

where \( i = 1 \) to \( 2 * \left[ \frac{H_{bbox}}{H_{img}} \right] - 1 \), \( j = 1 \) to \( 2 * \left[ \frac{W_{bbox}}{W_{img}} \right] - 1 \)

The final step is to apply the refined homography matrix to obtain the final rectified images. The warping is done with bilinear interpolation. By doing that, object patches from non-canonical images are transformed into canonical view. Besides, rectified images are on the same distance, as we manually set it to a fixed distance (1.2 meters in this case). This avoids the multi-scale problem, which is challenging for CNN networks.
IV. DATASET

Currently, there are only very few publicly available hazmat sign detection datasets. [14] published their high-resolution RGB hazmat detection dataset. However, the dataset was collected with a hand-held single-lens camera, containing only RGB images. On the contrary, our dataset contains images from an affordable RGB-D sensor (Intel Realsense RGB-D Camera). With the additional depth information, we are able to provide geometric information.

We provide a high-resolution RGB-D hazmat dataset with 13 labels in this paper, which can be found here. It contains both 360 RGB images and 360 depth images with a resolution of 1280 * 720. Ground truth label information of the RGB images is also provided. 130 of the RGB and depth images contain only one of the 13 types of hazmat label. Each of these images contains two backgrounds (plain and plywood) and five positions (top left, top right, center, bottom left, bottom right). For the rest of 216 images, each image contains 13 types of hazmat labels. Nine different angles ($\pm 75^\circ$, $\pm 60^\circ$, $\pm 45^\circ$, $\pm 30^\circ$, $0^\circ$) with three distances ($1.25m$, $1.50m$, $1.75m$) are included in these images.

V. EXPERIMENT AND RESULTS

A. Evaluation Metric

In our detection framework, we use a training set containing canonical-view images only to train the CNN detector. It effectively reduces the size of the training set, thus lowering the difficulty for collecting a good training set for mobile robotics applications. In the testing stage, each image first goes through the rectification system to get rectified images in the canonical view, as is shown in Figure 4. Then CNN detector is used for performing the actual detection. Finally, all detection results get warped back to original images to get final detection results as the output of the system.

We use the MSCOCO[8] object detection evaluation matrix to evaluate the detection performance. Two main metrics are IoU (Intersection of Union) and mAP (mean Average Precision). We propose an extend NMS method to select final bounding boxes. Since each image is split into a series of images, we can convert the bounding boxes back to origin image by utilizing the homography matrix we calculate before. For each splitted image we utilize the homography matrix to recover final bounding boxes. The implementation of our method is provided here.

B. Experiments Setup

Most experiments are performed on our own self-collected dataset, for there is no public available RGB-D hazmat detection dataset. To show that our approach can effectively reduce the difficulty for the CNN detector by dealing with perspective distortion in advance, only images from canonical views are used for training the network. The test sets include images from various angles (-75° to 75°).

TABLE I: Experiment result with / without geometry rectification.

|                | mAP (IoU=0.50) | mAP (IoU=0.75) | mAP (IoU=0.50:0.05:0.95) | AR (IoU=0.50:0.05:0.95) |
|----------------|---------------|----------------|--------------------------|------------------------|
| baseline       | 0.236         | 0.053          | 0.088                    | 0.15                   |
| our            | 0.53          | 0.193          | 0.246                    | 0.351                  |

TABLE II: Experiment result with / without geometry rectification in different angles.

| Angle   | mAP (IoU=0.50) | mAP (IoU=0.75) | mAP (IoU=0.50:0.05:0.95) | AR (IoU=0.50:0.05:0.95) |
|---------|---------------|----------------|--------------------------|------------------------|
| -75°    | baseline      | 0.009          | 0                         | 0.001                  |
|         | our           | 0.132          | 0.006                    | 0.034                  |
|         |               | 0.006          |                          | 0.004                  |
| -60°    | baseline      | 0.184          | 0.016                    | 0.054                  |
|         | our           | 0.477          | 0.055                    | 0.158                  |
|         |               | 0.055          |                          | 0.216                  |
| -45°    | baseline      | 0.329          | 0.099                    | 0.139                  |
|         | our           | 0.645          | 0.274                    | 0.303                  |
|         |               | 0.274          |                          | 0.401                  |
| -30°    | baseline      | 0.445          | 0.149                    | 0.188                  |
|         | our           | 0.679          | 0.365                    | 0.362                  |
|         |               | 0.365          |                          | 0.456                  |
| 0°      | baseline      | 0.538          | 0.222                    | 0.263                  |
|         | our           | 0.665          | 0.386                    | 0.375                  |
|         |               | 0.386          |                          | 0.492                  |
| 30°     | baseline      | 0.434          | 0.09                    | 0.169                  |
|         | our           | 0.632          | 0.367                    | 0.364                  |
|         |               | 0.367          |                          | 0.455                  |
| 45°     | baseline      | 0.287          | 0.041                    | 0.098                  |
|         | our           | 0.665          | 0.354                    | 0.364                  |
|         |               | 0.354          |                          | 0.46                   |
| 60°     | baseline      | 0.116          | 0.015                    | 0.029                  |
|         | our           | 0.586          | 0.179                    | 0.269                  |
|         |               | 0.179          |                          | 0.351                  |
| 75°     | baseline      | 0.026          | 0                        | 0.004                  |
|         | our           | 0.55           | 0.114                    | 0.222                  |
|         |               | 0.114          |                          | 0.281                  |

We use yolov3-tiny [4] as the CNN detector. We choose yolov3-tiny because 1) We take it as an typical example of off-the-shelf CNN-based detection network and 2) because it is small and fast enough for real-world deployment on mobile robots. The training time takes about one and a half hours with our computer (Intel Core i7-6700 CPU, GeForce GTX 1080, 8 GiB Memory). We trained our model from scratch with a batch size of 32, momentum 0.9, subdivisions 2, burn_in 2000, max_batches 8000, learning_rate 0.1 and the learning_rate will be multiplied by 0.1, when the number of batches is 3000, 4000, 5000, 6000, 7000.

C. Rectification Parameters

For the plane segmentation we assume that 90% of the points of each frame are from planes. As a result, the RANSAC keeps extracting planes until less than 10% of total points are in the remaining set. We set the number of maximum planes per image to 1, because it is known that the test set only contains one plane per test image. The distance from the virtual viewpoints to the plane is set as 1.2m. Because the training dataset is collected with distances between 1m and 1.5m, it is reasonable to assume that the CNN detector will have better performance on 1.5m or 1m than others. The training dataset contains two main parts, images that contain one hazmats and images with 13 hazmats, in order to prevent overfitting to the background. The homography matrix can be calculated in a closed form solution by using planar homography, but we are using DLT to compute the homography matrix. In the future we plan to

1https://robotics.shanghaitech.edu.cn/datasets/MARS-Hazmat-RGBD
2Will be available in the final version.
move to the closed form solution.

D. Results

We compare the detection performance with and without geometry rectification. The results are shown in Table I. Baseline means without geometry rectification. From Table I, we can see that, after geometry rectification, the performance is much better than before. \( mAP(IoU = 0.50) \) increases nearly 20% after geometry rectification while \( mAP(IoU = 0.50 : 0.05 : 0.95) \) increases 20.1%.

Table II shows the performance of our approach at different angles. The test dataset contains nine angles \( (\pm 75^\circ, \pm 60^\circ, \pm 45^\circ, \pm 30^\circ, 0^\circ) \). From Table II, we can see even on very large angles such as \( \pm 75^\circ \) our approach can still detect some hazmat signs. In our approach the \( mAP(IoU = 0.5) \) is 0.132 while without geometry rectification \( mAP(IoU = 0.5) \) is 0.009. On other angles such as \( \pm 60^\circ, \pm 45^\circ, \pm 30^\circ \), according to the results shown in Table II, our results are better than the previous results. \( mAP(IoU = 0.5) \) increases 47% at angle 60°, which is a huge improvement.

Besides perspective distortion, our proposed detection approach helps the CNN detector module to avoid dealing with multi-scale detection problems by explicitly rectifying the target patch to an ideal scale, as the distance from the plane to virtual viewpoints is fixed. Previous research has shown that the scale problem is challenging for CNN networks[28]. As the result, in Table II, we have better performance even at 0°, where there is no perspective distortion. For 0° images, the mAP performance gets improved from 0.263 to 0.375.

The top images on Figure 6 and Figure 7 show examples of the detection results of baseline algorithm without geometry while the bottom images in Figure 6 and Figure 7 show the detection results of our approach in the same images. We can see that on both Figure 6 and Figure 7 our approach performs much better than baseline approach. Our approach can detect more hazmat signs at very large angles. Also, the accuracy of bounding boxes is more precise in our approach.

Running single-threaded on a CPU, our algorithm needs about 5.7 seconds per image, a value that should be improved, since it is too slow to run live on a robot.

VI. CONCLUSIONS

In this work, we showed a simple but effective way to combine geometric information with an off-the-shelf CNN-based detector. By doing image rectification explicitly in advance of the CNN detector, we take full advantage of available geometric information from RGB-D images to 1) reduce the time for training stage; 2) reduce the size of training set required; 3) improve the performance of overall detection system; and 4) produce more accurate detection result (more tight bounding box). This approach also features a high robustness towards noisy depth information input (noisy point cloud), as the depth is just used to estimate the plane parameters, where RANSAC is effective even with noisy input.

For the mobile robotics application, especially hazmat sign detection in rescue robotics, our approach lowers the work required to create a nice training dataset, because fewer training images are needed. In the interest of reproducible science we provide the dataset used in the paper as well as our code to the public.
REFERENCES

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, “Orb: An efficient alternative to sift or surf,” in *IJCV*, vol. 11, no. 1. Citeseer, 2011, p. 2.

[4] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[6] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, “Microsoft coco captions: Data collection and evaluation server,” *arXiv preprint arXiv:1504.00325*, 2015.

[7] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Mallocci, T. Duerig, et al., “The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale,” *arXiv preprint arXiv:1811.00982*, 2018.

[8] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning rich features from RGB-D images for object detection and segmentation,” in *ECCV 2014*, vol. 8695 LNCS, no. PART 7, 2014, pp. 345–360.

[9] R. Sheh, S. Schwertfeger, and A. Visser, “16 years of robocup rescue,” *KI-Künstliche Intelligenz*, vol. 30, no. 3–4, pp. 267–277, 2016.

[10] R. Sheh, T. Kimura, E. Mihankhah, J. Pellenz, and B. Mertsching, “Seeing Signs of Danger: Attention-Accelerated Hazmat Label Detection,” in *2018 IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2018*, 2018.

[11] B. Zhao, A. Parra, and E. J. Delp, “Mobile-based hazmat sign detection and recognition,” in *2013 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013 - Proceedings*, no. 2009. IEEE, 2013, pp. 735–738.

[12] A. Parra, B. Zhao, A. Haddad, M. Boutin, and E. J. Delp, “Hazardous material sign detection and recognition,” in *2013 IEEE Conference on Image Processing, ICIP 2013 - Proceedings*. IEEE, 2013, pp. 2640–2644.

[13] R. Edlinger, G. Zauner, and M. Zauner, “Hazmat label recognition and localization for rescue robots in disaster scenarios;” *Electronic Imaging*, vol. 2019, no. 7, pp. 463–1, 2019.

[14] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, “Traffic-sign detection and classification in the wild,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2110–2118.

[15] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[16] A. Handa, M. Bloesch, V. Pătraşcu, S. Stent, J. McCormac, and A. Davison, “gynn: Neural network library for geometric computer vision,” in *European Conference on Computer Vision*. Springer, 2016, pp. 67–82.

[17] C. Couplie, C. Farabet, L. Najman, and Y. LeCun, “Indoor semantic segmentation using depth information,” in *International Conference on Learning Representations (ICLR2013)*, April 2013, 2013.

[18] Z. Zhou, M. Zhu, G. Pavlakos, S. Leonardos, K. G. Derpanis, and K. Daniilidis, “Monocap: Monocular human motion capture using a cnn coupled with a geometric prior;” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 4, pp. 901–914, 2019.

[19] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, “3d bounding box estimation using deep learning and geometry,” pp. 5632–5640, 2017.

[20] C. Wang and K. Siddiqi, “Differential geometry boosts convolutional neural networks for object detection,” pp. 1006–1013, 2016.

[21] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning rich features from rgb-d images for object detection and segmentation,” in *European conference on computer vision*. Springer, 2014, pp. 345–360.

[22] J. P. S. do Monte Lima, F. P. M. Simões, H. Uchiyama, V. Teichrieb, and E. Marchand, “Depth-assisted rectification for real-time object detection and pose estimation,” *Machine Vision and Applications*, vol. 27, no. 2, pp. 193–219, 2016.

[23] I. E. Sutherland, “Three-dimensional data input by tablet,” *Proceedings of the IEEE*, vol. 62, no. 4, pp. 453–461, 1974.

[24] B. Singh and L. S. Davis, “An analysis of scale invariance in object detection snip,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3578–3587.