Anomaly Detection for the Singulation of Plastic Wastes in Polymer Recycling

Charlotte Goos, Aleksandr Eismont, Dmitrii Seletkov
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology, Germany
{firstname.lastname}@student.kit.edu

Abstract

A major impediment to using deep learning methods for computer vision in practice is unlabeled datasets with a limited amount of data. In this work, we investigate different models for object counting on challenging real-world industrial dataset to detect singulation anomalies. We systematically analyze our dataset and apply copy & paste data augmentation based on the analysis to generate synthetic data and improve our results. Furthermore, we perform the ablation studies depending on generated data, loss function and batch sampler for two primary approaches: classification and instance segmentation. The classification approach ResNet18 with a stratified batch sampler, cross entropy loss with weight, trained on the original and synthetic data, achieves the best performance in terms of accuracy. Finally, we show that the instance segmentation approach Mask R-CNN profits from synthetic data and provides interpretability, despite the poorer performance, compared to the classification approach. Code has been made available at: `https://github.com/yayapa/AnomaliesRecycling`

1. Introduction

In recent years, Deep Neural Networks (DNNs) have achieved great results in various complex computer vision tasks such as classification, object detection, and instance segmentation. However, it is not apparent at the beginning which approach should be preferably applied for some industrial tasks. Furthermore, real-world datasets often are unlabeled, class imbalanced and do not contain much data.

Since the current state-of-the-art DNNs are data-hungry and annotating is computationally expensive, techniques such as Self-Supervised Learning [8], Transfer Learning [15], and Data Augmentation [3, 5] exist.

In this work, we investigate the real-world industrial dataset of plastic wastes in polymer recycling with a small amount of data. Our primary task is anomaly detection for the singulation in the sorting process. This means we perform object counting to guarantee the presence of only a single object under the detectors at a given time in the recycling production line. For this purpose, we investigate classification and instance segmentation methods. We facilitate our approach by generating the corresponding synthetic data using copy and paste data augmentation based on data analysis.

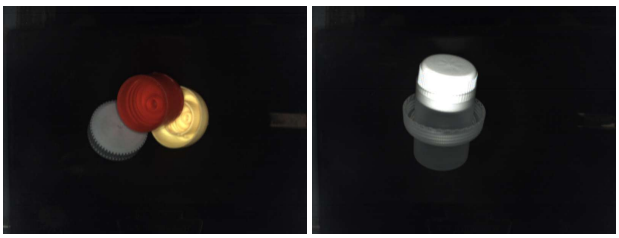


Figure 1: Image from our data set showing plastic lids in black trays.

2. Related works

2.1. Copy & Paste Data Augmentation

State-of-the-art convolutional neural network models for instance segmentation require a lot of data. Due to the lack of large annotated datasets and the expensive and time-consuming annotation process of new datasets, copy & paste data augmentation is used to generate new annotated synthetic data. In [3] new data is generated by selecting an instance image, predicting the foreground mask for the object and pasting it to a scene image. In [5] a copy-paste data augmentation is introduced, with new annotated data generated from an already existing annotated dataset. A random subset of objects is copied from a randomly chosen image and pasted into another randomly selected image. To both images, scale jittering and horizontal flipping are applied beforehand. The ground-truth annotations are adjusted for the new image.

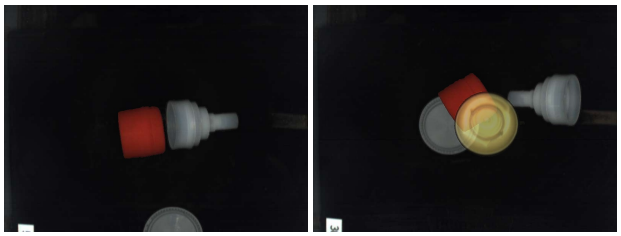


Figure 2: Synthetic images generated with our copy & paste data augmentation method.

2.2. Object Counting

Knowing the number of objects on the image can enable different systems to choose efficient processing pipelines at an early step of the workflow, making the process more intelligent by reducing computational cost. Object counting methods aim to localise objects in a scene and then count them. Recent works [1, 2, 9] have addressed the object counting task as an image segmentation problem by separating the object from the image background. Nevertheless, most of the counting methods are category specifically. Another recent work [16] studies the problem of salient object subitizing (SOS) as multi-class classification. This is the task of counting the number of salient objects in the image (independent of the category) within the subitizing range from 0 to 4 without resorting to any object localisation process.

3. Method

3.1. Dataset

Our real-world industrial dataset consists of 3326 images of black trays containing 0 to 5 plastic lids. The data is unlabeled. Therefore, we manually created image-level labels, which indicate the number of objects in each image. Two example images from our dataset are shown in 1. We further annotate the dataset with different tags to help us evaluate our networks later. The following tags are used:

- *different colors* - tray contains objects with different colors
- *overlapping* - objects are overlapping
- *transparent* - tray contains a transparent object
- *inside* - object is inside another object
- *dark color* - tray contains dark object
- *one color* - tray contains objects with the same color
- *open lid* - tray contains object with open lid
- *edge* - object is on edge of tray

The dataset suffers from class imbalance with 13.6% label 0, 56.7% label 1, 21.4% label 2, 6.4% label 3, 1.5% label 4, 0.4% label 5.

3.2. Copy & Paste Data Augmentation

Our real-world industrial dataset is small, with a significant class imbalance and without annotations for instance segmentation. Therefore, we implement our own copy & paste data augmentation to generate new synthetic data sets.

To generate images with image-level label 0, Gaussian noise is added to images of empty trays in our dataset. Images of empty trays are used as background to create images with an image-level label greater than 0. We insert non-dark objects from images with image-level 1 of our dataset to the background at a random position chosen with the help of a heat map. This heat-map is created from the positions of the object in images with image-level 1 in our real-world dataset. To copy & paste the objects to the background, we first generate a binary mask of the object and an image with the object on black background. We manage this using the contour of the object, calculated by thresholding. Then we apply a bitwise_or operation of background and mask and add the image of the object on a black background to the result. We further improved our copy & paste algorithm by implementing different modes, which can be switched on or off during generation:

- *rotate* - an object is rotated with a random angle
- *color* - an object is colored to a random color with a probability of 15%
- *dark* - an object is changed to a dark object with a probability of 15%
- *transparent* - an object is made transparent with a probability of 25%
- *edge* - an object is inserted on edge of tray with a probability of 5%

Example images generated with our copy & paste data augmentation method are shown in 2.

We automatically create annotations for instance segmentation, while generating our synthetic dataset. For each image, the binary masks of inserted objects are calculated. These masks only display the visible part of the object when a non-transparent object partially conceals it. From each object binary mask, an instance mask annotation is created by calculating the contour and smoothing it with spline interpolation. The annotations are saved in COCO annotation format [10].

3.3. Classification

In this project, we focus on counting lids as salient objects. We consider the task as a multi-class classification

Name	Label 0	Label 1	Label 2	Label 3	Label 4	Label 5	Accuracy	Decision Accuracy
SOS	0.98	0.88	0.06	0.17	0.05	0.00	0.66	0.71
GoogleNet	1.00	0.99	0.85	0.55	0.00	0.00	0.90	0.95
ResNet18	1.00	0.99	0.87	0.58	0.40	0.00	0.92	0.97
ResNet18_Weight	1.00	0.99	0.87	0.63	0.60	0.50	0.93	0.97
ResNet18_Syn	1.00	0.99	0.87	0.70	0.70	0.50	**0.95**	**0.98**

Table 1: Classification results depending on the applying models and training configurations. SOS [12] incorporates GoogleNet, fine-tuned on SOS dataset. GoogleNet and ResNet18 are fine-tuned on our dataset. ResNet_Weight uses the stratified batch sampler and cross entropy loss with weight. ResNet_Syn is ResNet_Weight, but trained on both original and synthetic data generated with all active modes.

problem where the last layer of the network generates confidence scores for each of 0, 1, 2, 3, 4 and 5 lids existing in the input image. We use ResNet18 [7] and modify it for the task of counting lids. Since the model is trained on ImageNet dataset [11] which has 1,000 classes, we load the model and modify the last fully connected layer to have the number of categories that matches the classes of interest. We initialise the network with ImageNet pretrained weight but fine-tune the whole network end-to-end.

For training and testing, we split the dataset into a training set of 2,660 images (80% of the real-world objects dataset) and a testing set of 666 images, using the information about tags to have more accurate results, covering all possible situations. We set the mini-batch size to 32 and fine-tune the model for 100 epochs using stochastic gradient descent (SGD) with a momentum of 0.9. The fine-tuning starts with a learning rate of 0.001, and we multiply it by 0.1 every 10 steps. During training and testing the network, we first resize the images to (256, 256) and apply a random horizontal flip with a probability of 0.5. We normalise the images using the ImageNet mean and standard deviation since we initialise the network with ImageNet pretrained weights. To solve the class imbalance problem, we use two different techniques. The first technique is cross entropy loss with weight to allow the model to pay more attention to examples from the small classes. The second is a stratified batch sampler to keep the original distribution of classes within each mini-batch. We also integrate early stopping with patience step 20 to stop training when a monitored metric stops improving.

3.4. Instance Segmentation

Since the copy and paste data augmentation produces instance masks for each object, we use them to generate a fully synthesized training dataset in COCO annotation format [10]. The training dataset consists of 12,000 images with 2,000 images for each label from 0 to 5. We use two test datasets. The first test dataset is fully synthesized and consists of 2,400 images with 400 images for each label from 0 to 5. This is used to evaluate the power of the selected model in instance segmentation using COCO API Average Precision (AP) metrics at different IoUs such as AP for AP at IoUs 0,50, 0.05, 0.95, AP50 for AP at IoU 0.50 and AP75 for AP at IoU 0.75. The sets of real-world objects used for the generation of training and test synthetic datasets do not overlap and are initially split into 80:20 proportion. The second test dataset consists of 666 real-world images. The same is applied to the classification experiments. This is used for evaluation in the object counting task.

We train Mask R-CNN [6] with backbone ResNet50 [7] and FPN implemented by Detectron2 [13]. The total number of iterations is 5,000. The SGD optimizer with a momentum of 0.9 is used. The learning rate is initialized to 0.001 and multiplied by a factor of 0.1 at iterations 3,000 and 4,000, respectively. The initialized weights of the backbone are pretrained on ImageNet [11].

4. Results

4.1. Evaluation Metrics

For our task, we use the accuracy metric to report classification results. In Tab. 1 and Tab. 2 we also provide accuracy regarding each class. However, this metric does not reflect all the required information for detecting anomalies. Therefore, we define the task-related metric, called decision accuracy, which relies on the number of correctly accepted and rejected trays:

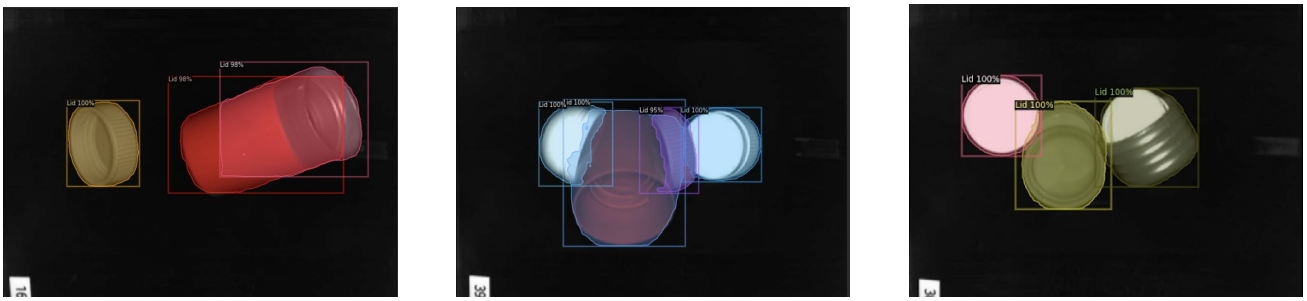
$$Decision\ Accuracy = \frac{Accepted\ Trays + Rejected\ Trays}{All\ Trays} \quad (1)$$

4.2. Classification

We evaluate the discussed approach on different CNN models and in various training settings. Our baseline is the SOS model. This incorporates GoogleNet [12], which was fine-tuned on SOS dataset (set of images from ImageNet [11], COCO [10], VOC07 [4] and SUN [14]). We compare the baseline with the pure GoogleNet and ResNet18 fine-tuned on our dataset. ResNet_Weight uses the stratified batch sampler and cross entropy loss with

Name	Label 0	Label 1	Label 2	Label 3	Label 4	Label 5	Accuracy	Decision Accuracy
Seg_no	1.00	0.94	0.64	0.28	0.20	0.00	0.83	0.89
Seg_edge	1.00	0.94	0.66	0.26	0.10	0.00	0.83	0.89
Seg_edge_dark	1.00	0.95	0.69	0.35	0.30	0.00	0.85	0.90
Seg_all	1.00	0.94	0.74	0.44	0.20	0.00	**0.86**	**0.91**

Table 2: Instance segmentation results depending on the applying data augmentation modes. Seg_no does not use any modes. Seg_edge uses rotate, change color and edge modes. Seg_edge_dark use rotate, change color edge and dark modes. Seg_all use rotate, change color edge, dark and transparent modes.



(a) Label 3. Predicted 3 (b) Label 4. Predicted 4 (c) Label 5. Predicted 3

Figure 3: Examples of the instance segmentation model prediction. Instance segmentation provides interpretability of the results compared to classification

weight. ResNet_Syn is trained on the original and synthetic data generated with all active modes. The results are reported in Tab. 1.

Thereby, we observe that ResNet18 is slightly better than GoogleNet. Also, integrating the stratified batch sampler and providing weight to the loss function improves the accuracy for classes with a small number of images. Our best configuration involves using synthetic data for training. The best decision accuracy score is 98%. We believe that we achieve prediction accuracy comparable to human performance in identifying images with zero or one object.

4.3. Instance Segmentation

We perform an ablation study by generating synthetic datasets using different modes in copy & paste data augmentation. We compare four different synthetic datasets. Seg_no does not use any modes. Seg_edge uses rotate, change color and edge modes. Seg_edge_dark use rotate, change color edge and dark modes. Seg_all use rotate, change color edge, dark and transparent modes. The results are reported in Tab. 2.

Evaluated on the synthetic test datasets, the means of AP, AP50 and AP75 are 82.10, 97.61 and 90.76, respectively. Thus, we conclude the model architecture is selected correctly.

Furthermore, we observe the consequent enhancements in accuracy and decision accuracy on the real-world test dataset used for the classification when we apply additional modes by generating synthetic datasets.

Finally, we observe the performance degradation of the best segmentation model compared to the best classification model. However, the advantage of the segmentation approach is interpretability, illustrated in Fig. 3

5. Conclusion

In this paper, we study the problem of unlabeled and class imbalanced datasets with a limited amount of data on the example of the real-world industrial dataset for object counting. To mitigate those, we develop a copy & paste data augmentation for generating new synthetic data. Based on a systematical analysis of our dataset, we implement and apply different modes for generating synthetic data. We investigate two approaches: classification and instance segmentation. Both methods profit from the synthetic data. The classification with synthetic data and model tuning shows the best performance in terms of accuracy. The instance segmentation performs poorer than classification but has better interpretability.

In future works, we aim to further combine the classification and instance segmentation approaches within one pipeline to boost our performance in terms of accuracy and interpretability. Furthermore, our copy & paste data augmentation should be tested on more complex object contours.

References

[1] Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R Selvaraju, Dhruv Batra, and Devi Parikh. Counting everyday objects in everyday scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1135–1144, 2017.

[2] Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao. Object counting and instance segmentation with image-level supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12397–12405, 2019.

[3] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection.

[4] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[5] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *CoRR*, abs/1902.06162, 2019.

[9] Hui Lin, Xiaopeng Hong, and Yabin Wang. Object counting: You only need to look at one. *arXiv preprint arXiv:2112.05993*, 2021.

[10] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[12] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[13] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[14] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

[15] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014.

[16] Jianming Zhang, Shugao Ma, Mehrnoosh Sameki, Stan Sclaroff, Margrit Betke, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. Salient object subitizing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4045–4054, 2015.