Proceedings of the Joint Austrian Computer Vision and Robotics Workshop 2020 DOI: 10.3217/978-3-85125-752-6-30

The Difficulties of Detecting Deformable Objects Using Deep Neural Networks

Nikola Djukic, Markus Vincze
Automation and Control Institute, TU Wien, Vienna, Austria
{dukic,vincze}@acin.tuwien.ac.at

Walter G. Kropatsch
Pattern Recognition and Image Processing Group, TU Wien, Vienna, Austria
krw@prip.tuwien.ac.at

Abstract. *Object detectors based on deep neural networks have revolutionized the way we look for objects in an image, outperforming traditional image processing techniques. These detectors are often trained on huge datasets of labelled images and are used to detect objects of different classes. We explore how they perform at detecting custom objects and show how shape and deformability of an object affect the detection performance. We propose an automated method for synthesizing the training images and target the real-time scenario using YOLOv3 as the baseline for object detection. We show that rigid objects have a high chance of being detected with an AP (average precision) of 87.38%. Slightly deformable objects like scissors and headphones show a drop in detection performance with precision averaging at 49.54%. Highly deformable objects like a chain or earphones show an even further drop in AP to 26.58%.*

1. Introduction

Object detection in RGB images has received a lot of attention in the previous years due to advances in deep neural networks (DNN) research. Classical techniques usually rely on searching for features in an image that were hand-crafted by a human. Deep neural networks on the other hand use huge datasets of hand-labelled images to learn these features. These labels are either a bounding box of an object or its mask. This approach has shown great efficiency. In general there are two types of DNN based object detectors. The first group performs the detection in a single run through a network. These methods are generally fast and can even run in real-time

Figure 1. Objects used for evaluation

with standard hardware. Second group has a separate region proposal and detection stage, which usually makes the execution of the methods slower but more precise than the first group of methods. Recently, a combination of CBNet and Cascade R-CNN has achieved a new state of the art result on the COCO dataset [9] with an impressive AP50 of 71.9%. [10]

Detecting custom objects is a common problem in robotics. DNN or more precisely Convolutional Neural Networks (CNN) require large amounts of data for training. Having that data hand-labelled by a human is extremely time consuming so there is a lot of research going on in the field of synthesizing training data. This is typically done by first making a 3D reconstruction of the objects and then placing them in a virtual environment which allows the simulation of artificial deformations and the creation of arbitrary synthetic views where labels are taken from the 3D template. However, obtaining a full 3D reconstruction is not possible with all objects, especially

in the case of deformable objects. Objects like folding headphones, scissors, chains, cables can vary in appearance depending on their current usage. This poses a problem for CNN based object detectors. We propose a simple RGB based method for recognition of rigid but also deformable objects and synthesize images for training a neural network. We then test this method by training the YOLOv3 [13] network with the fully synthetic dataset and explore how does the shape of an object, ie. it's symmetry and deformability affect the detection performance.

The contributions of the work include:

- An automated pipeline on synthetic data generation used for detection and recognition of both rigid and deformable objects.
- A novel RGB based method for quick and effortless acquisition of object masks.
- We explore the effect of deformability of an object to its detection performance.

2. Related work

Computer vision tasks depend on large amounts of annotated training data. For the tasks of detecting object classes such as cars or airplanes there are numerous hand-annotated datasets available: COCO [9], PASCAL VOC [3] and Open Image Dataset [7]. These datasets are built by researchers or companies and consist of a large number of images. Each image has annotations of objects of interest. This may be a bounding box only or contain the mask of the object as well. The COCO (Common Objects in Context) dataset consists of over 330 thousand images containing objects that are split into 80 classes. However, sometimes, especially in robotics related tasks, we are interested in detecting a specific object. For example not any mug but the user's favourite coffee mug. The mentioned datasets are of little use in these cases, so there is a necessity for a specialized dataset. Datasets are normally difficult to obtain so there is a lot of research concerning synthesizing datasets.

Jungwoo Huh et al. [6] proposed a method for synthesizing training data that, similarly to ours, relies on obtaining masks of an object. In order to produce the synthetic images they use pure pasting, whereas we use a combination of pasting and Poisson image editing. Additionally they evaluate their method on rigid objects only, for example a baseball bat, a bottle, a toy rifle etc. The only deformable object that they use is an umbrella but they keep it closed during the training and testing so we can consider it as a rigid object in this case. Additionally, they use YOLOv2, which has a lower mAP (Mean Average Precision) than the YOLOv3 while also preserving the ability to process the images in real-time. For obtaining the masks of the objects they use a semi-automatic segmentation method while ours is fully automated and does not involve any manual post-processing.

Debidatta Dwibedi et al. [2] assume that object images, which cover diverse viewpoints, are available. They apply a CNN to obtain a mask of the object. They then randomly place the object into a scene image using Poisson cloning. Next, they train the Faster R-CNN [14] network using the synthetic images and evaluate the method on the GMU-Kitchens dataset [5]. For the evaluation of the method they also use exclusively rigid objects like bottles, detergents, cups, cornflakes packages etc. Although simple, the method achieves an mAP of 88%, which is similar to what we report on detection of rigid objects.

Georgakis et al. [4] propose a method for synthesizing training data that takes into consideration the geometry and semantic information of the scene. They use publicly available RGB-D datasets, the GMU-Kitchens [5] and the Washington Washington RGB-D Scenes v2 [8], as backgrounds for the object images. Using RANSAC they detect planes in the image and artificially place objects on top of them, while also scaling their size according to the distance from the camera. This method produces natural looking images, because instead of being placed randomly in an image, the objects such as a cup or a bottle are placed on a flat desk surface or on the ground. They test their method using SSD and Faster R-CNN [14] and report an mAP between 70% and 85% depending on how much real data they use. Considering the fact that the scenes they use for evaluation are cluttered this is a good results. The objects used for evaluation are a bowl, a cup, a cereal box, a coffee mug and a soda can. These are all non deformable objects.

3. Synthetic Data Generation

Object detection is required in cases such as self-driving cars, unmanned aerial vehicles, robotics etc. Except for detecting rigid objects like cars, chairs or cups it is often needed to detect deformable ob-

jects like chains or cables. Most of the previous work on object detection focuses on detecting rigid objects[3, 15, 6, 2]. Our goal is to expand this research to deformable objects as well. We train an object detector based on CNN to detect both rigid and deformable objects. For this task a big amount of training images is required. Obtaining this data manually is time consuming, therefore we propose a method for synthesizing the training data which includes an RGB based segmentation procedure that is able to handle deformable objects. We then use publicly available datasets as background for the synthetic images and augmentation techniques to increase the variability of the dataset.

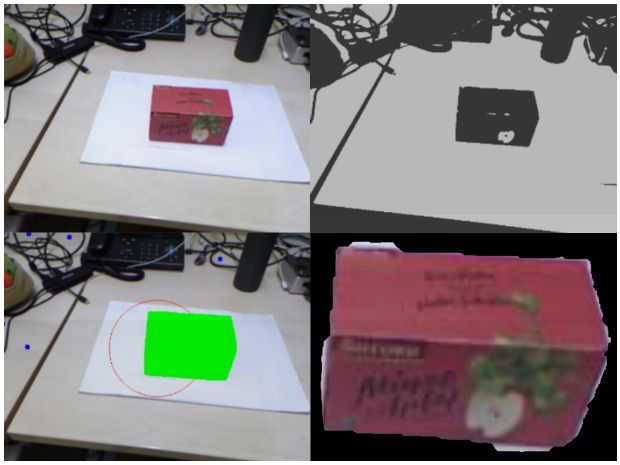

Figure 2. Illustration of the mask acquiring process. Top left image shows the original RGB image. Top right image shows the result of appyling k-means method to the original RGB image. Bottom left image shows the automatically selected contour and the area inside of it colored in green. Bottom right image shows the final extracted object masks.

3.1. Data acquisition

Publicly available datasets which contain annotated objects are suitable for training CNN to detect object classes. However, when it comes to detecting specific objects, a specialized dataset is required. We synthesize a dataset by capturing the images of the objects and develop a method to segment them from the flat surface on top of which they were placed.

For the recording of objects a Kinect camera by Microsoft mounted on a tripod is used. The camera is placed at approximately 30 cm above the flat surface and facing the object at an angle of approximately 45 degrees. During the recording, both the camera and the flat surface are stationary. The flat surface should preferably be unicolor so that the object is clearly distinguishable from it.

After the recording was initiated, the object was manipulated by hand in order to get it to face the camera from all possible viewing angles. The point is to get the object to face the camera in as many unique perspectives as possible. The advantage of this method is that it is able to capture deformable objects by simply changing their shape while they are being recorded.

3.2. Data processing

In order to synthesize images that are needed for training of the network object masks are needed. Obtaining the masks of the object is possible by manually segmenting the object from the background or by using a segmentation method. Manually segmenting objects is inefficient, therefore we devise a simple method for object segmentation that is used for both rigid and deformable objects. For the segmentation of the object from the background a combination of computer-vision based methods is used. It contains the following five steps:

1. Firstly, k-means clustering is applied to the image with the k value of 2. This method is successful at distinguishing the boundaries of interest. Additionally it is computationally more efficient than a possible alternative of using Otsu's Thresholding.

2. After application of k-means, morphological operations like image closing and erosion are applied to the image in order to connect possible discontinuities in the border of the object.

3. Next, contour detection is applied to the whole image and locations of gravity centers of the area inside of the detected contours are determined. A red circle is drawn on the image coming from the Kinect camera, which is shown on the screen, in which the center of the object should be placed in order to automatically start the capturing process.

4. The algorithm then determines if the contour satisfies conditions in terms of its length and distance from the center of the image and, if that is the case, the recording is started. After the capturing process is initiated a predetermined number of object projections is recorded at a regular time interval or per keyboard command. The number of projections recorded is

40, which is usually more than enough to capture the object from all different angles.

5. If the object of interest is deformable the capturing process is paused, to capture a deformed state, and then re-started. Images of segmented objects are then stored on a hard drive for synthesizing training data.

Figure 2 shows the illustration of the mask acquiring process.

3.3. Synthesizing training data

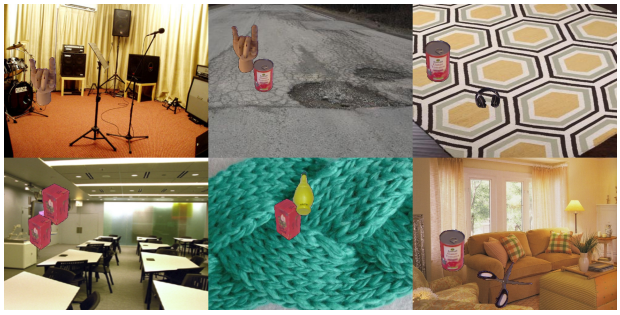

Figure 3. Examples of synthetic images that are used for training the YOLOv3 network

In order to synthesize the training images we used a combination of Poisson image cloning [11] and pure pasting of the segmented objects onto different background images. As background for the synthetic images we used the Indoor Scene Recognition dataset [12] and Describable Textures Dataset (DTD) [1]. We used ten different objects for the evaluation and generated 2500 synthetic images per object. To handle the blur that appears while the objects are moving, we artificially blurred 20% of the images by adding horizontal motion blur between 5 and 15 pixels to the objects. As objects move closer or further away from the camera their relative size changes, so we introduce artificial scaling of the object uniformly distributed between 50% and 125% of its original size. In order to tackle the occlusion problem small patches of textures from the DTD dataset are placed randomly on 10% of the synthetic images. These cover between 0% and 50% of the object surface. Additionally we introduce multiple objects to the image and allow them to occlude each other by a maximum IOU (Intersection Over Union) of 40%.

Figure 3 shows the examples of the synthetic images that are used for training the YOLOv3 network.

4. Evaluation

To evaluate the method, we trained the CNN based object detector YOLOv3 using the synthetic images. A total of ten objects were used, which differ greatly in their shape and deformability. We know already that YOLO performs very well when facing rigid objects. Therefore our aim was to explore to what extend the shape of an object can be deformed. As an example of rigid objects we use a can, two different tea boxes, and a lemon juice bottle. Slightly deformable are headphones, scissors and a human hand model. Extremely deformable objects that we used are earphones, power cable and a piece of chain.

Properties of objects used for evaluation and their detection precision are presented in Table 1.

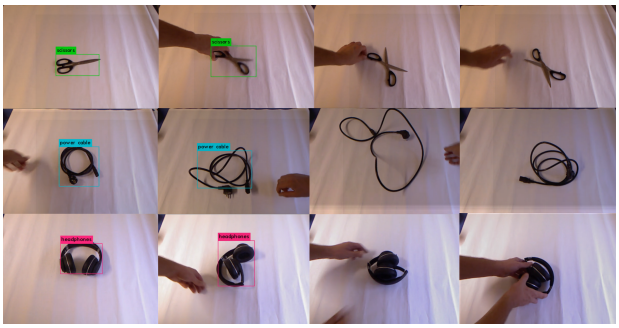

Figure 4. Successful and unsuccessful cases of detection of different deformable objects

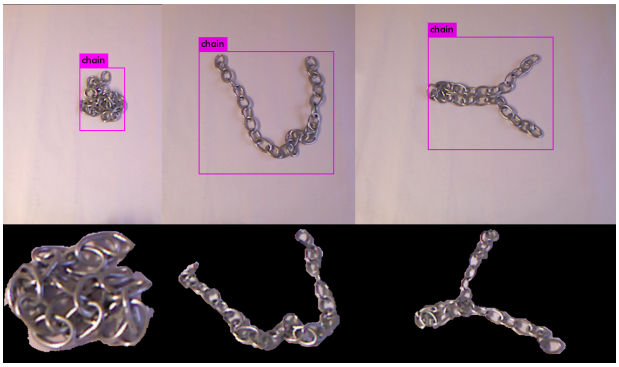

Figure 5. Chain detection success cases

Figure 6. Chain detection failure cases

In order to evaluate the precision of the proposed method two minute videos of each object being manipulated were filmed and every 20th frame extracted

Objects	Deformability	Precision
lemon juice bottle	rigid	89.61
red tomato can	rigid	84.52
red tea box	rigid	87.65
yellow tea box	rigid	87.76
headphones	slightly def.	57.32
scissors	slightly def	54.15
human hand model	slightly def.	37.16
power cable	highly def.	34.66
chain	highly def.	30.24
earphones	highly def.	14.86

Table 1. Object detection performance, def - deformable

and manually annotated. We then ran the YOLO network trained with the synthetic data and calculated precision for each object, taking as ground truth the hand-annotated data. An Intersection Over Union (IOU) of 50% was considered a successful detection.

As shown in previous work rigid objects like a can, a tea box or lemon juice bottle have a very good chance at getting detected with the precision being at close to 90%. These objects do not change greatly in appearance when placed in different positions and it is therefore easy for the network to learn their appearance. We purposely choose that some of the objects have similar color, so that, due to lack a of great number of objects used for evaluation, the detection performance may not be attributed to simple color searching.

Slightly deformable objects that we used were scissors, headphones and a human hand model. We see that in the case of slightly deformable objects the detection performance drops significantly with it being around 55% for the scissors and the headphones. The chances of detecting the human hand model are even lower, being 37.16%.

The last three objects that we evaluate are a chain, a power cable and a pair of earphones. These objects are considered to be highly deformable. Again there is a clear drop in detection performance with the precision of earphones detection being only 14.86%. Chances that a power cable or a piece of a chain will be detected are a bit over 30%.

All of the objects used for evaluation can be seen in Figure 1. Detection of objects used for evaluation using YOLO trained on COCO dataset was unsuccessful for all of the objects except for the scissors with the detection rate of 62.35%, similar to our result. The chain detection success cases can be seen on Figure 5, whereas the chain failure cases are presented in Figure 6.

We can see that the in the cases where chain detection is successful a mask of chain taking a similar structure can be found in the bottom row of Figure 5. In the cases where the chain detection fails there are no masks available that resemble the given chain structure.

We then pose the chain detection problem as single link detection problem and try to detect the structure of the chain by detecting each individual link in the chain. In order to do so, we use our proposed method to segment the link in many different orientations and synthesize the training images. We then connect the individual links into a chain and test detection of individual links while the chain is taking different configurations. The results of a single link detection can be seen on the top row of the Figure 7.

Figure 7. Examples of link detection

We also record 100 images of a chain taking different shapes and manually annotate each of the links on the chain in all of the images and train the YOLOv3 network with those annotated images. The results of a single link detection with the manually annotated links can be seen on the bottom row of the Figure 7.

We took chain as an example of a highly deformable object that is made out of simple rigid elements. These results show that the detection of an deformable object is possible by detecting its elementary parts.

Successful and unsuccessful cases of object detection are presented in the Figure 4. As shown, on the examples of the power cable, the scissors and the headphones, detection is successful in some of the configurations. If the configuration is slightly changed the detection fails. This is due to the big variability in the appearance of these objects which is caused by their deformability. Potentially, modelling of deformable objects such as a power cable or a chain could be used to generate big amounts of dif-

ferent object masks. This would enable the network to learn a bigger amount of object views, than those that a human demonstrator can show in a reasonable time.

Our method works well when facing rigid objects, when the number of unique views is limited. However, when it comes to deformable objects, number of unique views increases dramatically. Therefore, in those cases the efficiency of our method drops significantly.

5. Conclusion

In this paper we intend to highlight open problems of a standard object detector when applied to slightly and highly deformable objects. We specifically trained the YOLOv3 detector to cope with these cases. To reduce the time consuming effort of image annotations, we proposed an automated method for synthesizing the training images. The idea is to show objects on simple background and use a short videos and a few annotations with augmentation of training data to obtain better performance. While this works well for rigid objects with an AP of 87.38%, we show that for slightly deformable objects like scissors and headphones the detection performance drops significantly to 49.54%. The drop is, as expected even more drastic for highly deformable objects like a chain or earphones, down to AP of 26.58%.

Using the example of a chain we show that it is possible to pose the problem of detection of the deformable objects as detection of its elementary rigid element - a link. To further tackle this problem, modelling of deformable objects could be used for synthetic data generation.

Acknowledgment

This research is partially supported by the Vienna Science and Technology Fund (WWTF), project RALLI (ICT15-045 and Festo AG & Co. KG.

References

[1] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[2] D. Dwibedi, I. Misra, and M. Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1301–1310, 2017.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[4] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka. Synthesizing training data for object detection in indoor scenes. *arXiv preprint arXiv:1702.07836*, 2017.

[5] G. Georgakis, M. A. Reza, A. Mousavian, P.-H. Le, and J. Košecká. Multiview rgb-d dataset for object instance detection. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 426–434. IEEE, 2016.

[6] J. Huh, K. Lee, I. Lee, and S. Lee. A simple method on generating synthetic data for training real-time object detection networks. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1518–1522, Nov 2018.

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

[8] K. Lai, L. Bo, and D. Fox. Unsupervised feature learning for 3d scene labeling. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3050–3057. IEEE, 2014.

[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[10] Y. Liu, Y. Wang, S. Wang, T. Liang, Q. Zhao, Z. Tang, and H. Ling. Cbnet: A novel composite backbone network architecture for object detection. *arXiv preprint arXiv:1909.03625*, 2019.

[11] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on graphics (TOG)*, 22(3):313–318, 2003.

[12] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420, June 2009.

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

[14] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[15] K. Wang, F. Shi, W. Wang, Y. Nan, and S. Lian. Synthetic data generation and adaption for object detection in smart vending machines. *arXiv preprint arXiv:1904.12294*, 2019.