Influence of Synthetic Image Datasets on the Result of Neural Networks for Object Detection

Aleksandr Kniazev[1, 2], Pavel Slivnitsin[1, 2], Leonid Mylnikov[1], Stefan Schlechtweg[2] and Andrey Kokoulin[1]
[1]Perm National Research Polytechnic University, Komsomolsky avenue 29, 614990 Perm, Russian Federation
[2]Anhalt University of Applied Sciences, , Bernburger Str. 57, 06366 Köthen, Germany
knxandr@rambler.ru, slivnitsin.pavel@gmail.com, leonid.mylnikov@pstu.ru, stefan.schlechtweg@hs-anhalt.de, a.n.kokoulin@gmail.com

Keywords: Image Recognition, Object Detection, Neural Network, Synthetic Dataset, Data Generation.

Abstract: The goal of the article is research of ways to improve the quality of neural networks object detection. To achieve this goal we suggest to use synthetic image datasets. The algorithm of generating synthetic images, which uses the environment of the detected object, is described in the article. That algorithm could be applied in the control algorithm of the robotic system for luminaire replacement that is based on target object detection. 3D models and 3D camera images of detected objects, backgrounds, noise objects and different effects are used to create realistic images that will increase the quality of predictions. Quality tests were made with synthetic and real datasets. Results show that quality could be increased up to 16%. Ratio of real and synthetic data is 1:4.

1 INTRODUCTION

Training is a very important part of neural network creation. Less datasets leads to undertraining, while huge datasets leads to overtraining. Even optimal size of dataset can lead to bad results if objects for detection would be captured from one view or/and on the same background. Moreover false positive detection can appear. The order of the training dataset is also an important thing in the training process [1]. In case of object detection images annotated with coordinates of objects are elements of training datasets.

Manual annotation is a very popular way to annotate images presently. Scientists have to define bounding boxes of objects by hand in special programs (for example LabelImg - https://github.com/tzutalin/labelImg). Example of manual annotation is shown on Figure 1. This process is very time consuming.

Community of scientists created a huge amount of annotated datasets for the last 20 years [2]. These datasets can be easily found and have free access to use in tasks of object detection. However, all these datasets are applicable for a small range of object detection tasks and cannot be applied in other tasks. Luminaire detection is one of these tasks which demand dataset creation. Neural network for luminaire detection can be applied to a robot which replaces broken luminaires [3] with a special connector [4].

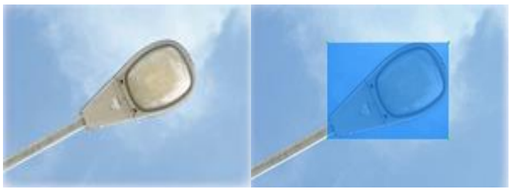


Figure 1: Manually annotated image with LabelImg.

There are some articles about automatic generation of training datasets in literature. Some simple methods use a sliding window to capture movable objects [5]. Other methods use combinations of elements to generate images. For example, 3D models of objects for detection are used with different backgrounds [6-8]. Moreover, additional noise effects can be applied to simulate different factors which can influence image quality [9]. This can improve precision of detection.

To generate datasets in [6-7] each 3D scene has to be manually set in Blender 3D. Complex algorithms of calculating horizontal planes are used in [10]. After that, scientists have to manually remove false regions. Finally, an image would be generated. All these factors strongly influence the speed of generation.

Synthetic datasets can influence the quality of neural networks results. To measure this influence it is necessary to use methods of assessing the accuracy of object detection. Intersection over Union (IoU) also known as Jaccard index [11] is a widespread method. This method compares two shapes. That is why IoU is invariant to the scale of the object in the image. Due to this property the precision of the detected object is measured [12, 13].

2 METHODOLOGY

To generate a training dataset we will use a combination of background image, image of the detected object and some noise objects. A small amount of images could be enough to create big datasets. Random position, size of detected object, different position, gamma and blur value of noise objects allow to create a lot of various datasets.

To obtain an image of the detected object we use two approaches: 3D model and depth image from a 3D camera. Figure 2 shows an example of the algorithm of generation synthetic images.

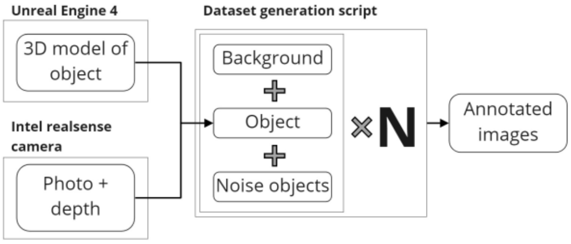


Figure 2: Dataset generation algorithm (N - is a number of images).

The advantage of this approach is that we know positions of detected objects and can automatically annotate images. Blender 3D is used to obtain images of the detected object. Firstly, a 3D model has to be created. Then the 3D model is rotated by Z-axes and rotation is recorded as animation. Animation then should be saved as a set of images. To calculate object position background should have a color which is contrast to the detected object.

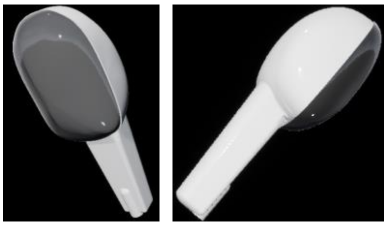
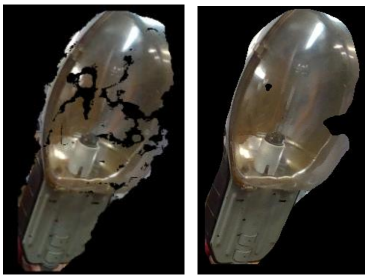


Figure 3: Examples of images of objects obtained from 3D models.

Another approach is to use a 3D camera to obtain images of detected objects. We used an Intel RealSense camera which can capture simple RGB images and in addition it captures depth of images. Depth image is used as mask to separate object and background. Then mask have to be written to the alpha channel of the PNG image. It allows to combine object image and background.

Such images could have some defects due to transparent parts of objects. For example, a light bulb of the luminaire. Figure 4.a, shows these defects.

To fix these defects we use the Graph Cut method [13]. Example is shown on Figure 4.b.



a) b)

Figure 4: Examples of object images obtained with 3D camera: a) image of object directly from 3D camera, b) image of object after Graph Cut method.

Generated images of objects should be placed on background images in a realistic way. It means that the background should correspond to places in the real world where objects could be placed. Noise objects then placed in addition. In case of luminaires, it could be tree branches, which can cover luminaire, rain, low light, camera defects. These noise effects make the resulting image more realistic. To make a more realistic result we add some blur to noise images.

Mask is calculated to combine object image and background. Coordinates of the object set randomly from predefined parameters. After that, the object image is put on the result image. Moreover, gamma and size changes can be applied. Noise objects are preprocessed the same way. In addition, noise objects can be flipped.

Object mask is also used to annotate images. As a result “xml” file is obtained (example is shown in Listing 1) with coordinates of object bounding box. This box is used in neural network training.

Listing 1. Annotation of an object on the image in the form of an “xml” file.

```
<annotation>
  <folder>train</folder>
  <filename>1608936293.0034409.jpg</filename>
  <path>test_saves/1608936293.0034409.jpg</path>
  <source>
    <database>LumAutoGenDataset</database>
    </source>
    <size>
        <width>416</width>
        <height>416</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>luminaire</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>250</xmin>
            <ymin>117</ymin>
            <xmax>320</xmax>
            <ymax>264</ymax>
        </bndbox>
    </object>
</annotation>
```

This approach allows to obtain a huge amount of unique images. Examples of generated images are shown on Figure 5.

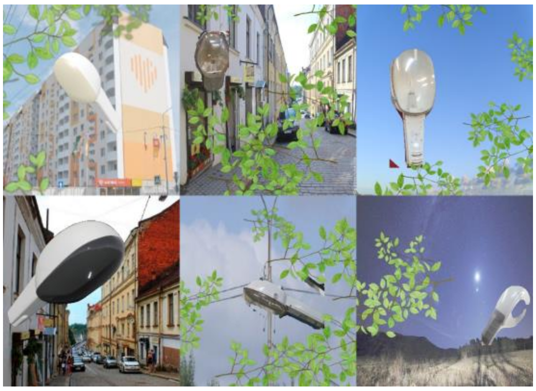


Figure 5: Examples of images from generated datasets.

3 EXPERIMENTS

YOLOv3-Tiny [14] was chosen to test the influence of synthetic data on neural network results. This neural network model demands less time on training and testing. It allows to make more experiments and compare results. Model structure is simpler than other models [15]. We can suggest that the influence of synthetic datasets will show and it could appear on other models.

We used mixed datasets of real photos and synthetic images.

Generation of synthetic datasets was made by the algorithm on Figure 6.

Step 1: Setting the required number of generated images (*N*);
Step 2: Loading of *N* random backgrounds from predefined set of 2D images;
Step 3: Loading of object images from predefined set of images with 3D models and images from 3D camera for each background;
Step 4: Random changes of gamma, size and position of objects;
Step 5: Add object images on backgrounds by mask;
Step 6: Calculation of object bounding box and saving to "xml" file;
Step 7: Loading of noise objects k times for each background ($k \in (0; m)$, m – is the maximum number of noise objects on en image). Calculation of masks of noise images;
Step 8: Random change of gamma, size and position of each k noise object for each background;
Step 9: Add k noise objects to backgrounds by masks;
Step 10: Save all generated images.

Figure 6: Algorithm of generation synthetic datasets for training neural networks (to work with random numbers we use uniform distribution law).

Since while preparing the training data we do not have information about the location of the target object and do not take it into account, we will assume that if the data is mixed evenly, we can avoid a drastic change in weights during neural network training.

The learning quality in this case depends on how evenly our data is shuffled and how diverse the data is generated. When these operations are performed optimally, the result can be expected to be non-random according to the central limit theorem [16].

After synthetic data generation, the parameters that affect the quality of the model are: 1) ratio of synthetic data and real photos in the training dataset; 2) size of training dataset.

This results in two criteria:

1) quality criterion $\mathrm{IoU}(\Omega(x, y)) \rightarrow \max$,

2) time criterion, $T(\Omega(x, y)) \rightarrow \min$, where Ω is training dataset, x is the number of real data, y – is the number of synthetic data, T – training time.

The criteria are differently oriented. As a result of empirical research (using different data ratios of 1:4, 1:8 and dataset sizes of 1000 and 2000 images), a real to synthetic data ratio of 1:4 and a dataset size of 1000 images were chosen.

For the completeness of the research, we tried to train the model using various combinations of real and synthetic data: training using synthetic data with 3D model images and validation on similar data; training using synthetic data with 3D model images and validation on real images; training using mixed data (3D models + real photos) and validation on real images; training using 3D camera images and validation on similar data; training using 3D camera images and validation with real data.

Table 1: Synthetic datasets for training.

№	Number of real photos	Number of generated images
1	200 + 50 (training, validation)	0
2	0	800 + 200 (training, validation) (3D model)
3	250 (validation)	1000 (training) (3D model)
4	100 + 150 (training, validation)	1000 (training) (3D model)
5	100 + 150 (training, validation)	1800 + 200 (training, validation) (3D model)
6	250 (validation)	1000 (training)
7	100 + 150 (training, validation)	1000 (training) (3D camera)
8	250 (validation)	1000 (training) (3D camera + Graph Cut)
9	100 + 150 (training, validation)	1000 (training) (3D camera + Graph Cut)
10	0	800 + 200 (training, validation) (3D camera + Graph Cut)
11	0	2400 + 600 (training, validation) (3D model, 3D camera, 3D camera + Graph Cut)

All models were trained in the same way, only the input data was changed. Model YOLOv3-Tiny pre-trained on the COCO trainval dataset provided by the developers on the official website [17] was used for experiments.

The training consisted of two stages, the first one with frozen weights of all layers except the last two ones, responsible for object detection. This was performed to obtain stable losses to reduce the impact of the initial high losses on the weights in the main part of the model. After the losses stabilisation within 50 epochs, all weights were unfrozen and training continued.

After first attempts to train the neural network on synthetic data obtained using 3D model, we have observed that training and validation of the model using only synthetic data gives poor quality of object detection in real photos, an example is shown in Figure 7.

For this reason, we decided to also train models on synthetic data with validation on real data and on mixed data.

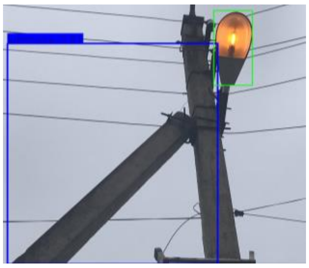


Figure 7: An example of poor object detection with a model trained on only synthetic data.

4 RESULTS AND DISCUSSION

After training the models with the datasets described in Table 1 and evaluating their performance on IoU metric (see Tables 2 and 3), we can say that there is a 0.002-0.16 improvement in neural network performance for 5 experiments and 0.013-0.146 for 10 experiments using datasets 5, 6, 7, 9 compared to 1. The other datasets had no positive effect on the performance of object detection in the image. Increasing the number of experiments, there is a slight fluctuation in the dispersion, which indicates reproducibility and result stabilisation despite the stochastic elements used in the models according to the central limit theorem.

The lowest quality is shown by training the model using 3D models with validation on real photos. It can be caused by the fact that the 3D models we use have technical inaccuracies (models may not look enough detailed and believable from some angles). The validation data were realistic and detailed, that may have reduced the detection quality of the model.

Figures 8 to 10 show the loss curves for models (2), (3), (9) compared with model (1).

The curves show the difference in the effect of validation data on learning. For example, model (3), which was trained using real photos for validation, significantly reduced its performance after unfreezing the main part of the YOLOv3-Tiny neural network weights. Whereas the learning curve of model (2), which was validated on the same type of data as the training data, did not show such a sharp increase in losses. This allows us to see that a network trained on 3D model images has learned to detect luminaires in the generated images (3D model), but it will not be able to detect luminaires in real photos.

Table 2: Average IoU value and dispersion of IoU values for 5 experiments.

№	Average IoU value	Dispersion of IoU values
1	0,427	0,136
2	0,081	0,042
3	0,217	0,076
4	0,429	0,142
5	0,429	0,146
6	**0,488**	0,141
7	**0,587**	**0,098**
8	**0,431**	**0,112**
9	**0,592**	**0,091**
10	0,413	0,11
11	0,429	0,137

Table 3: Average IoU value and dispersion of IoU values for 10 experiments.

№	Average IoU value	Dispersion of IoU values
1	0,444	0,131
2	0,059	0,032
3	0,249	0,086
4	0,413	0,145
5	**0,457**	0,148
6	**0,477**	0,146
7	**0,59**	**0,101**
8	0,419	0,111
9	**0,586**	**0,091**
10	0,428	0,113
11	0,434	0,14

The lack of a sharp increase in losses after weights unfreezing can be observed in Figure 10 (model 9). This is due to the fact, that the validation was performed using both synthetic images and real photos. This improved quality of object detection in cases (7) and (9).

Neural network training based on the generated dataset using Intel RealSense 3D camera images showed better results compared to the 3D model images. The combination of synthetic images and a small number of real photos improved the quality of object detection compared to models trained on only a small number of real photos. This shows that this approach can be used to improve the quality of object detection.

The research described in the paper focused on the detection tasks of static objects. However, we assume that since moving objects can be represented by a set of sequential images, the proposed approach can be extended to moving images as well. This is possible by using an algorithm that makes the necessary corrections for false positives or false negatives of the base algorithm on single images.

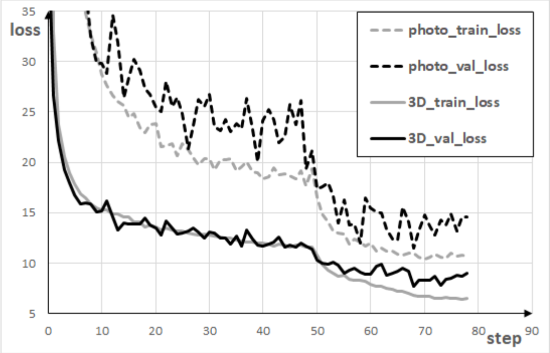


Figure 8: Loss curves for the model trained on synthetic dataset 2 (see Table 1).

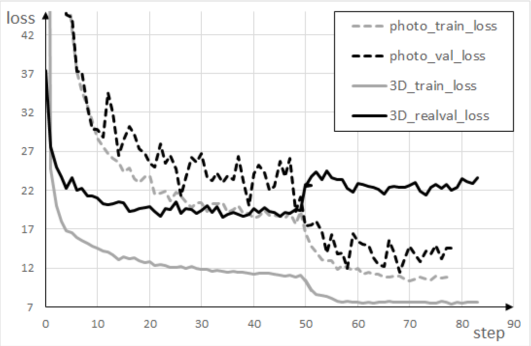


Figure 9: Loss curves for the model trained on synthetic dataset 3 (see Table 1).

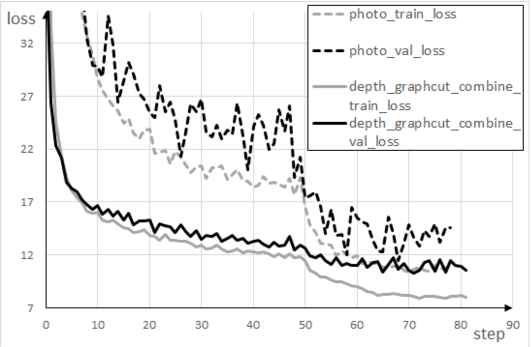


Figure 10: Loss curves for the model trained on synthetic dataset 9 (see Table 1).

5 CONCLUSION

Data sets of 1,000 images each were used in the experiments. As a result, we have found that it is possible to create datasets on synthetic data, but it is also necessary to dilute this synthetic image dataset with a small number of real photos (with a ratio of real to synthetic data approximately ¼). This solves the problem of creating large annotated datasets, required for training neural networks to improve the quality of object detection. This paper shows the effect of different combinations of synthetic and real data on the performance of a neural network for object detection. In our paper, we have tried to perform as many experiments as possible to get the broadest possible overview of the impact of synthetic data on neural network performance.

The proposed approach differs from existing approaches by using a combination of 3D models, fragments of real photographs and noise effects. In addition, this approach does not use algorithms that calculate the position of objects in 3D space and algorithms that calculate the possible position of detection objects. In our example (luminaire detection), the object can be located in any part of the image. This reduces the required time to create a single image for a dataset.

We expect that works intended to produce more realistic images, for example containing elements such as corrosion and deformation effects, and failures will contribute to further improvements in detection quality.

REFERENCES

[1] L. Mylnikov, “Statistical methods of intelligent data analysis,” St. Petersburg: BHV-Petersburg, 2021, 240 p.

[2] Z. Zou, Z. Shi, Y. Guo, and J. Ye, “Object Detection in 20 Years: A Survey,” arXiv, pp. 1-39, May 2019.

[3] P. Slivnitsin, A. Bachurin, and L. Mylnikov, “Robotic system position control algorithm based on target object recognition, ” in Proceedings of International Conference on Applied Innovation in IT, vol. 8, no. 1, pp. 87-94, 2020.

[4] P. A. Slivnitsin and A. A. Bachurin, “A modern way of outdoor lighting maintenance, ” in Journal of Physics: Conference Series, vol. 1415, no. 1, 2019.

[5] T. Anwar, “Training a Custom Object Detector with DLIB & Making Gesture Controlled Applications,” 2020 [Online]. Available: https://www.learnopencv.com/training-a-custom-object-detector-with-dlib-making-gesture-controlled-applications/ [Accessed: 07-Dec-2020].

[6] J. Li, P. L. Götvall, J. Provost, and K. Åkesson, “Training Convolutional Neural Networks with Synthesized Data for Object Recognition in Industrial Manufacturing,” IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA, vol. 2019-Septe, pp. 1544-1547, 2019.

[7] M. Andulkar, J. Hodapp, T. Reichling, M. Reichenbach, and U. Berger, “Training CNNs from Synthetic Data for Part Handling in Industrial Environments,” IEEE Int. Conf. Autom. Sci. Eng., vol. 2018-Augus, pp. 624–629, 2018.

[8] D. Mas Montserrat, Q. Lin, J. P. Allebach, and E. J. Delp, “Scalable Logo Detection and Recognition with Minimal Labeling,” Proc. - IEEE 1st Conf. Multimed. Inf. Process. Retrieval, MIPR 2018, pp. 152-157, 2018.

[9] G. Volk, S. Muller, A. Von Bernuth, D. Hospach, and O. Bringmann, “Towards Robust CNN-based Object Detection through Augmentation with Synthetic Rain Variations,” 2019 IEEE Intell. Transp. Syst. Conf. ITSC 2019, pp. 285-292, 2019.

[10] G. Georgakis, A. Mousavian, A. C. Berg, and J. Košecká, “Synthesizing training data for object detection in indoor scenes,” Robot. Sci. Syst., vol. 13, 2017.

[11] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019, vol. 2019-June, pp. 658-666.

[12] G. T. U. A. Colleges, et al., “Microsoft COCO,” Eccv, no. June, pp. 740-755, 2014.

[13 M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” Int. J. Comput. Vis., vol. 88, no. 2, pp. 303-338, 2010.

[14] J. Redmon and A. Farhadi, “YOLO v.3,” Tech Rep., pp. 1-6, 2018.

[15] W. He, Z. Huang, Z. Wei, C. Li, and B. Guo, “TF-YOLO: An improved incremental network for real-time object detection,” Appl. Sci., vol. 9, no. 16, 2019.

[16] P. G. Doyle, “Grinstead and Snell's Introduction to Probability,” 2006, American Mathematical Society. 518 p

[17] J. Redmon and A. Farhadi, “YOLO: Real-Time Object Detection,” 2018 [Online]. Available: https://pjreddie.com/darknet/yolo/. [Accessed: 25-Nov-2020].