Dataset Expansion by Generative Adversarial Networks for Detectors Quality Improvement

Alexander Kostin, Vadim Gorbachev

Federal State Unitary Enterprise «State Research Institute Of Aviation Systems» (GosNIIAS), Moscow, Russia
{akostin,vadim.gorbachev}@gosniias.ru

Abstract. Modern neural network algorithms for object detection tasks require large labelled dataset for training. In a number of practical applications creation and annotation of large data collections requires considerable resources which are not always available. One of the solutions to this problem is creation of artificial images containing the object of interest. In this work the use of generative adversarial networks (GAN) for generation of images of target objects is proposed. It is demonstrated experimentally that GAN's allows to create new images on the basis of the initial collection of real images on background images (not containing objects), which simulate real images accurately enough. Due to this, it is possible to create a new training collection containing a greater variety of training examples, which allows to achieve higher precision for detection algorithm. In our setting, GAN training does not require more data than is required for direct detector training. The proposed method has been tested to teach a network for detecting unmanned aerial vehicles (UAVs).

Keywords: Object Detection, GAN, Domain Adaptation, UAV, Drone.

1 Introduction

The majority of modern object detection systems and computer vision algorithms are based on machine learning, primarily neural networks. They have proven their reliability and quality in a wide range of tasks. The main disadvantage of such algorithms is the requirement of large (or even super large) annotated training datasets. Thus, the problem of lack of such data is usually faced in applied tasks. For example, in case of training the detector for a specific object that is not represented in large public annotated data collections, or the need to work in specific conditions.

The problem of development of visual indoor UAV positioning system [1] is one of such cases. In the absence of data from satellite navigation systems and requirements for the absence of additional radio wave sources, the use of passive sensors such as video cameras with detection algorithm is extremely useful.

Fig. 1. An example of an image in which it is necessary to detect a mini UAV.

A massive training set is required to train a highly accurate detection network, while the amount of labelled data is severely limited due to limited resources for markup. The authors had at their disposal about 900 images with the drone taken from only 6 different angles. It will be shown below that this number is not sufficient to provide training for a robust detector. For comparison, the standard data collections for object detection tasks include a huge number of images. For example, data set ImageNet [2] contains more than 14 million images, MS COCO [3] - 328000.

It is standard practice for tasks with small data to use augmentations, but in our case they have not been effective enough. In order to achieve high accuracy of detector training in conditions of very limited training set, we have investigated the possibility of using generative adversarial neural networks (GAN) for creation of synthetic training images. They were created by drawing drones with neural network in different areas of the background. With this approach it is possible to achieve enriching the dataset with new object-background combinations and automatic annotations and to raise the detector quality.

Adversarial algorithms are learning method in which two agents (a generator and a descriptor) are created inside a neural network that pursue opposite goals and have corresponding loss functions. The generator tries to draw an artificial image that the discriminator cannot distinguish from the real one, while the discriminator tries to learn to distinguish the imitation from real images. This method makes it possible to create a generative network, the output of which will simulate the distribution of available data accurately enough. Such algorithms show that it is possible to create highly realistic artificial images. For example, it is possible to train a network to generate plausible images from noise [4] or a network to transform the data domain with or without a teacher [5,6].

2 Related work

The problem of data lack in neural network training is well known. There are various approaches to expanding training samples without additional manual markup. The main approach is augmentation of the original collection of marked images. Augmentation consists in turns, reflections and distortions of color channels, image noise and so on [7].

Another approach is the so-called "transfer learning", i.e., teaching the neural network on large available collections of similar data with additional training directly on the target data [8]. The approach is generally accepted, but it does not solve the problem of lack of target data until the end.

One more possible approach is to add undetected data to a training sample and then mark it up using a training model [9]. Thus, we obtained pseudo labels, which will be the markup for added data on the next epoch of neural network training. But the quality of the bounding box regression task could not be significantly improved with this approach.

An alternative approach is to create and use in training synthetic images obtained by rendering 3D models of objects [10]. At such approach annotations of objects are generated automatically, and the volume of received data is theoretically not limited. The disadvantage of this method is that the synthesized images do not always simulate real ones accurately enough, and neural networks are highly sensitive to the data distribution fluctuations. In the article [1] this very approach was used to extend the training dataset and train the detector. The data were synthesized on with an existing 3D drone model. The drone model was drawn in the 3D modeling system with different angles. Then random transformations were applied to the image and its mask: rotation, scaling, displacement, reflection, etc. After that the image of the object by its mask was inserted into arbitrary background. This approach showed a good result in case the drones on the images looked relatively large and contrasting, but when switching to higher resolution images with smaller objects (near-realistic conditions) proved to be ineffective.

In the article [11] GAN was used for creating additional training samples. The key feature of this approach is that an attempt was made to obtain feedback (in the form of gradients in training network) for the generator from the detector. The network was arranged in such a way that the input of the generator was fed with a background image in RGB format and a window representing a rectangle of ones against the background of zeros, which indicated the place to which the generator should overlay the detectable object. The entire image was created by the generator. In the application task under consideration such architecture did not work, as the detectable objects were much smaller than the image itself. In contrast to DetectorGAN, in our approach images do not generated entirely, but only a small square containing the object to be detected, which is then inserted back into the high-resolution image.

3 Detection algorithm

The algorithms of object detection based on neural networks can be divided into single-stage [12] and two-stage [13]. Detectors from the first group are faster, but in general they are inferior in accuracy to detectors from the second group. Since in the application task under consideration it was required to provide real-time processing of image stream from 6 cameras, it was decided to use single stage detectors. RetinaNet [14] with PeleeNet [15] backbone was chosen as the detection network. As the specificity of the task consists in detecting small objects, the network has been modified accordingly. The initial version of PeleeNet received a 304x304 resolution image for input and detected objects using 5 different scales. The initial image was divided into grids of sizes from 1x1 to 19x19 cells. The 19x19 grids were not enough to provide a sufficiently dense coverage of the image with anchor boxes. Therefore, in this study the network was modified in such a way as to accept an image of 608x608 resolution at the input and split it on a 38x38 mesh grid.

4 Data generation algorithm

The main objective of our work was to create an algorithm that draw target objet on a given background image. To solve this problem, the principle of domain transfer was applied. It bases on a render image of 3D object model which is pasted into some background image. This image is processed by the neural network, which transforms the synthetic image of the object to a more realistic one. The generative adversarial network (GAN) was used as such transformation network.

Fig. 2. The general scheme of the proposed pipeline. From left to right: background image, background image with pasted render of 3D model of the object, transformed image, whole image with inserted fragment (fragment is highlighted with red square)

In spite of the fact that in general the scheme is quite simple, in practice a number of problems arise in its implementation, because of which synthesized data cannot be an effective substitute for real data. The main difficulty with this approach is a sharp boundary (Fig. 3.), which appears at the place where the modified fragment is inserted into the original image. Usually, image transformation algorithms change not only the object itself and its domain, but also partially change the background. When a fragment is inserted back into the original image, a sharp non-uniform border appears. Such an

artifact may be "learned" by the detection algorithm at the training stage as an important informative feature, which prevents it from working correctly on real data.

Fig. 3. An example of generated images containing a clear boundary at the drone generation site

In order to get rid of this boundary effect, Attention Guided GAN was taken as a generative model [16]. Its learning algorithm is similar to classical CycleGAN [3], but the principle of image generation is different. The generator receives the input image from the original (synthetic) domain, the encoder extracts latent features, and then the decoder creates a mask and some image, which is pasted into the original image by the mask. This changes only the part of the image that is directly related to the domain of the image. The border of the changed part of the picture turns out to be smooth and it is possible to insert a fragment back in picture without obvious artifacts, which in training may mislead the detector. Such generator produces 9 masks and 9 images. To train it, two sets of images from different domains are required: a set of background images with pasted drone renderers into them and a set of real drone images cropped from training dataset. This network will learn how to convert images from one domain to another like CycleGAN. The proposed data extension algorithm consists of 3 stages:

The input of the algorithm contains a picture and a parameters of a bounding box inside which the target object is located. A square fragment is cut out of the picture with a center equals to a center of the box and fixed by height and width. The values of these parameters depend on the data and should be large enough that the resulting fragment can hold an object in the training dataset with the largest bounding rectangle. The fragment's side was set to 152 pixels in our experiment.

An arbitrary image of the rendered drone is pasted into the cut out fragment, after which it is fed to the input of the trained generator, which performs the transition from the domain of synthetic data to the domain of realistic images.

The converted image is inserted back into the original picture. As a ground true bounding box for the obtained data bounding box with centers corresponding to the centers of squares, and sizes equal to the largest of the marked data are taken.

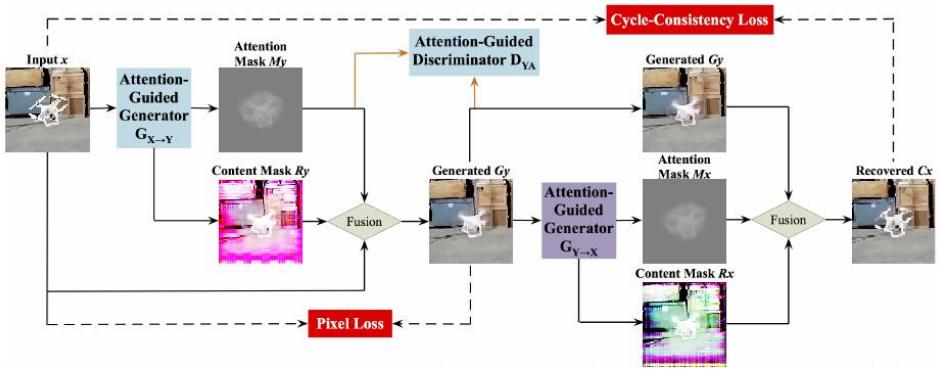

Fig. 4. Architecture of the algorithm

5 Experiments and results

The experiments were carried out on the data, which is a footage of the drone flight in a hangar. The footage was carried out with six tripod-mounted cameras at different angles (Fig.5). To get rid of the need for manual annotation of video files, the following algorithm of automatic annotation [1] was used to create a training collection. Optical stream maps were calculated for each video frame. The area with the maximum magnitude of the optical stream was selected on the maps. It was believed that this area corresponds to the drone. However, due to the presence of other moving objects, shadows and segmentation inaccuracies, such a mark-up can't be considered completely true. Images from 4 cameras were taken for training of the generator and detector model, images from another two cameras were taken as validation and test datasets.

Fig. 5. An example of an image from the available data. The drone is highlighted with a red rectangle

To establish the effectiveness of the proposed algorithm three experiments were carried out. The first consisted in training the detector on a raw training set of 900 images, which is the hangar footage from 4 angles (4 training backgrounds). The second experiment repeated the method proposed in the article [1]. The data for this experiment were extended by placing drone renders in arbitrary places. The third was to teach the detector on the same data, extended by new pictures created by the trained generator. In the process of data extension, drones were applied to each background by a uniform grid with 30 and 20 pixel steps in the x and y axis respectively. All manipulations were performed with images of original resolution 1280x720 pixels. The sizes of datasets in the second and third experiment were equal. Since the source data are very poor in variety of backgrounds (there are only 4 angles of the same hangar in the training set), it was decided to add to the dataset random images that do not contain detectable objects. This solution expands the variability of backgrounds, which increases the discriminatory ability of the network and improves precision of the detector.

5.1 Details of AGGAN training

To teach the generator, a square with the side of 152 pixels and the center coinciding with the center of the limiting rectangle was cut out from each image in the training sample. The data obtained by this procedure formed domain A. Train dataset was parsed in order to find empty square for each corresponding square in domain A. Such empty pictures formed domain B. In this way, pairs of images from different domains were obtained. In order to introduce variability into the generated data, it was decided to put drone renderers on the images from domain B. It was assumed that the generator would make the transition between domains by increasing the visual likelihood of pasted drones. In this case, the data could be expanded by overlaying new renders from different angles. There were a total of 15 images of drone renderers. Attention Guided GAN training was running on the data obtained in this way for 200 epochs. Adam with parameters lr=0.0002, beta1=0.5 and beta2=0.999 was used to optimize this network. After passing 100 epoch, learning rate began to decrease linearly to zero. Learning rate was decreasing linearly to zero after 100 epochs passed.

5.1 Detector training details

In all experiments the detector was trained for 100 epochs. As the optimizer Adam with parameters lr=0.001, beta1=0.9, beta2=0.999 was chosen. Since it is known that the image will always contain no more than one object (this is the specificity of the applied task), as the network output was taken only prediction with the greatest confidence.

Fig. 6. Examples of generated drones. The first and third rows shows source images with pasted drone render, the second and fourth rows shows corresponding transformed images

5.2 Results

To establish the effectiveness of the proposed algorithm three experiments were carried out. The first consisted in training the detector on a raw training set of 900 images, which is the hangar footage from 4 angles (4 training backgrounds). The second experiment repeated the method proposed in the article [1]. The data for this experiment were extended by placing drone renders in arbitrary places. The third was to teach the detector on the same data, extended by new pictures created by the trained generator. In the process of data extension, drones were applied to each background by a uniform grid with 30 and 20 pixel steps in the x and y axis respectively. All manipulations were performed with images of original resolution 1280x720 pixels. The sizes of datasets in the second and third experiment were equal. Since the source data are very poor in variety of backgrounds (there are only 4 angles of the same hangar in the training set), it was decided to add to the dataset random images that do not contain detectable objects. This solution expands the variability of backgrounds, which increases the discriminatory ability of the network and improves precision of the detector. The f1 metric curves for all three experiments (Fig.7, Fig.8 and Fig.9) and the table with the results on the test dataset are presented in Table 1. Figure 7 shows instability of learning curve

on raw real data, that witnesses insufficient amount of training data. The final experimental results (Table 1) prove that the addition of artificial data to training set is useful, but proposed method of image transformation is of most effectiveness. The amount of real images used in each experiment was equal, so artificial expansion of dataset by our method is a solution to lack of data problem.

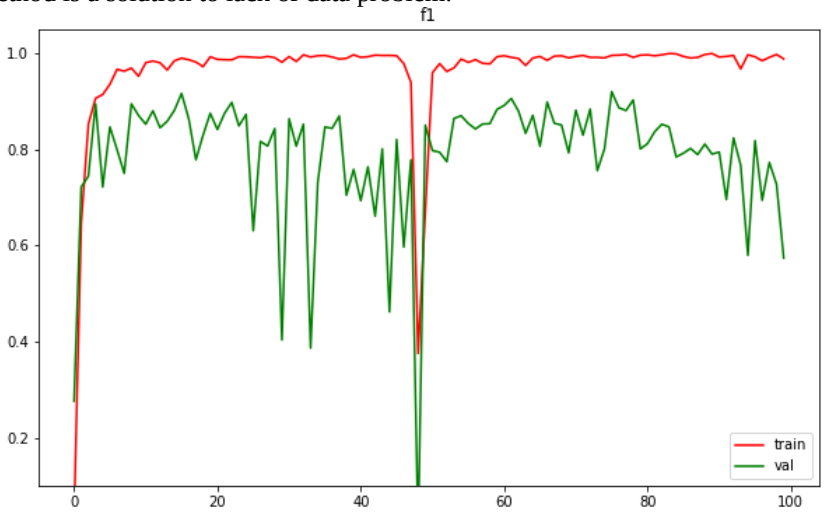

Fig. 7. Graph of the metric f1 while teaching only on real data

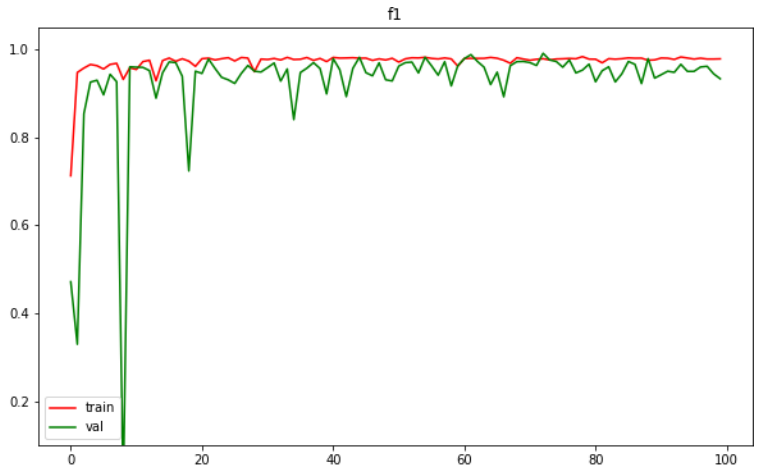

Fig. 8. Graph of f1 metric while training on data extended by drone renders

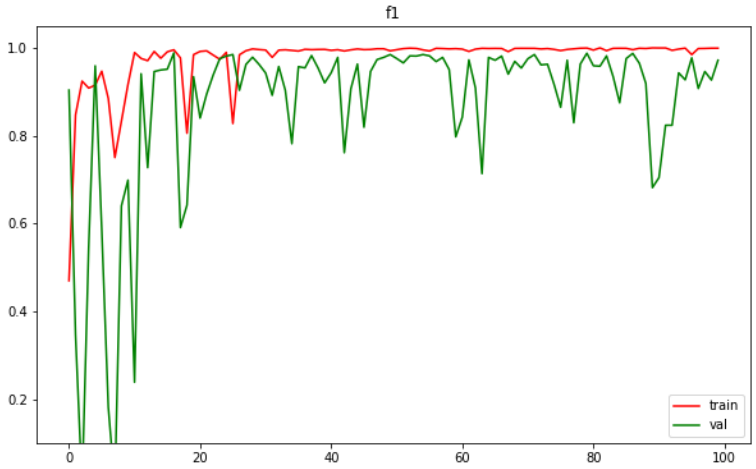

Fig. 9. Graph of f1 metric while training on data extended with the proposed method of image transformation

Table 1. Values of metrics on the test sample in various experiments

Dataset	f1	recall	precision
Raw data	0.9723	0.9489	1.0
Raw data + renders	0.9813	0.9659	1.0
Raw data + GAN	0.9909	0.9830	1.0

6 Conclusion

The problem of artificial expansion of the training dataset via GAN for object detection neural network was solved in this work. Input data of the proposed algorithm is background images containing renders of 3D model of target object. Proposed algorithm first pastes the object model render on the background image, then the neural network performs domain transfer for the local fragment of the image containing the object. It is shown in experiments that such synthesized images can be successfully used for learning detectors and allow to significantly improve their quality in comparison with the use of both only raw real images and a mixture of real images with 3D model renders.

References

1. Technology for the Visual Inspection of Aircraft Surfaces Using Programmable Unmanned Aerial Vehicles Blokhinov, Yu. B.; Gorbachev, V. A. ; Nikitin, A. D.; Skryabin, S. V. // Journal of computer and systems sciences international N 58 V 6 P 960-968, 2019
2. O. Russakovsky et al., “ImageNet Large Scale Visual Recognition Challenge,” Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.0575.
3. T.-Y. Lin et al., “Microsoft COCO: Common Objects in Context,” May 2014, [Online]. Available: http://arxiv.org/abs/1405.0312.
4. A. Brock, J. Donahue, and K. Simonyan, “Large Scale GAN Training for High Fidelity Natural Image Synthesis,” Sep. 2018, [Online]. Available: http://arxiv.org/abs/1809.11096.
5. P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” Nov. 2016, [Online]. Available: http://arxiv.org/abs/1611.07004.
6. J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” Mar. 2017, [Online]. Available: http://arxiv.org/abs/1703.10593.
7. Perez, Luis and Jason Wang. “The Effectiveness of Data Augmentation in Image Classification using Deep Learning.” ArXiv abs/1712.04621 (2017)
8. S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.
9. S. Kim, J. Choi, T. Kim, and C. Kim KAIST, “Self-Training and Adversarial Background Regularization for Unsupervised Domain Adaptive One-Stage Object Detection.” Aug. 2019, [Online]. Available: http://arxiv.org/abs/1903.12296.
10. Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization. Tremblay, Jonathan, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon and Stanley T. Birchfield. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2018): 1082-10828.
11. L. Liu, M. Muelly, J. Deng, T. Pfister, and L.-J. Li, “Generative Modeling for Small-Data Object Detection,” Oct. 2019, [Online]. Available: http://arxiv.org/abs/1910.07169.
12. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.02640.
13. S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.01497.
14. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection,” Aug. 2017, [Online]. Available: http://arxiv.org/abs/1708.02002.
15. R. J. Wang, X. Li, and C. X. Ling, “Pelee: A Real-Time Object Detection System on Mobile Devices,” Apr. 2018, [Online]. Available: http://arxiv.org/abs/1804.06882.
16. H. Tang, D. Xu, N. Sebe, and Y. Yan, “Attention-Guided Generative Adversarial Networks for Unsupervised Image-to-Image Translation,” Mar. 2019, [Online]. Available:http: http://arxiv.org/abs/1903.12296.