Analyzing Airplane Detection Performance with YOLOv4 by using Synthetic Data Domain Randomization

Houssem-Eddine Benseddik[1][a], Ariane Herbulot[1,2][b] and Michel Devy[1]
[1]*LAAS, CNRS, Toulouse, France*
[2]*Univ. de Toulouse, UPS, LAAS, F-31400 Toulouse, France*

Keywords: Airplane Detection, YOLOv4, Deep Learning, Domain Randomization, Object Detection, Synthetic Data.

Abstract: This paper proposes a novel approach to generate a synthetic dataset through domain randomization, to address the problem of real-time airplane detection on airport zones with high accuracy. Most solutions have been employed and developed across satellite images with deep learning techniques. Our approach specifically targets airplane detection on complex airport environment using deep learning approach as YOLOv4. To improve training, a large amount of annotated training data are required for good performance. To address this issue, this study proposes the use of synthetic training data. There is however a large performance gap between methods trained on real and synthetic data. This paper introduces a new method, which bridges this gap based upon Domain Randomization. The approach is evaluated on bounding box detection of airplanes on the FGVC-Aircraft dataset.

1 INTRODUCTION

Since the advent of deep convolutional networks, the capabilities of computer vision systems to solve various problems such as object detection, have advanced considerably in the last few years. The major success of deep learning-based approaches can be attributed to the availability of required computational power to perform the huge amount of calculations, and to the availability of large datasets to enable models to be trained (Agarwal et al., 2019).

Training a deep neural network is a time consuming and expensive task which typically involves collecting and manually annotating a large amount of data for supervised learning. Obtaining such data in the real world is tedious and error-prone. Furthermore, the collection of a large amount of training data with sufficient variety in real word is often expensive or not feasible (Hinterstoisser et al., 2019). As such, researchers have been developing various techniques to overcome this issue and introduce cost saving measures to build a high-quality dataset. One of the most promising solutions that have been investigated for this issue is Domain Randomization (DR) (Nowruzi et al., 2019).

DR is a simple yet powerful technique for generating training data for machine-learning algorithms. The goal is to generate or synthetically improve the data, in order to introduce random variances in the properties of the training environment. These properties are essentially present at the learning task and are not necessarily present at the test task, unlike traditional training data when collected in the same domain as the test data (Valtchev and Wu, 2020). A reassuring method in this direction is to create simulated data for training that is well capable of imitating the test statistics of the real data.

In this paper, we explore multiple ways in this context to extend DR to the task of real-world airplanes detection. By focusing on various synthetic datasets and training the convolutional neural network entirely on synthetic data, our goal is to identify a procedure that addresses the domain shift and the trained network generalizes well to real test data.

In particular, we are interested in answering the following questions:

1. Can DR on synthetic data achieves enormous airplane detection results, on real-world data?
2. Can DR on synthetic data reaches competitive results with those obtained by real data?
3. Can DR with fine-tuning by real data during training improve more the accuracy?

[a] https://orcid.org/0000-0003-4229-0125
[b] https://orcid.org/0000-0002-8377-6474

Benseddik, H., Herbulot, A. and Devy, M.
Analyzing Airplane Detection Performance with YOLOv4 by using Synthetic Data Domain Randomization.
DOI: 10.5220/0011089500003191
In *Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2022)*, pages 425-431
ISBN: 978-989-758-573-9; ISSN: 2184-495X

4. How do the parameters of DR affect results?

By analysing these questions, this work contributes the following:

- Extension of DR to detect different airplane models in front of real-world complex backgrounds;
- Introduction of a new DR component, namely, object motion (rotation and translation), which improves detection accuracy;
- Investigation of the parameters of DR to evaluate their importance for airplane detection;
- A comprehensive metric study to make comparison between synthetic and real data training's;
- Fine-tuning models trained on large synthetic dataset with a small set of real data.

2 PREVIOUS WORKS

Our work is related to airplane detection, synthetic data for machine learning, domain adaptation and domain randomization.

The performance for computer vision algorithms increases logarithmically with training data increasing. However, obtaining a large amount of annotated data in the real-word is a bottleneck for computer vision tasks. One way of dealing with this issue is to use the synthetic data as a cheap and efficient solution to assemble such large datasets (Bousmalis et al., 2018).

The use of synthetic data however introduces what is known as the reality gap, which is the inability for the synthetic environments to fully generate the real-world data, for numerous reasons including textures, physics of materials, lighting and domain distributions (Nowruzi et al., 2019). In an attempt to narrow the reality gap, DR is introduced to simulate a sufficiently large amount of variations such that real-world data is viewed as simply another domain variation. This can include randomization of view angles, textures, shapes, camera localisation, object positions and many other parameters (Valtchev and Wu, 2020). On the other hand, underlying principal of DR is to create enough variance in training data which forces the model to only learn relevant features useful for the task. DR provides a solution to narrow the reality gap, thus by enriching the data generation phase.

To address the reality gap, DR techniques have been explored, including most notably the work of (Tobin et al., 2017), where they synthesized images of basic geometric objects on a table, in an attempt to estimate their 3D positions, such that a robotic arm could pick them up. Their accuracy varied depending on domain parameters, achieving errors as low as 1.5cm on average in terms of object location, showing promise for synthetic data training. Notably, they found that the number of images and the number of unique textures used in the images were the most prominent parameters to model accuracy. Camera positioning and occlusion also had meaningful contributions, while the addition of random noise in images did not. (Tremblay et al., 2018) uses DR for car detection by effectively abandoning photorealism in the creation of the synthetic dataset. The involved work forced the network to learn only the essential features for the task of car detection. Results of this approach are comparable to the Virtual KITTI dataset (Gaidon et al., 2016). One issue with the Virtual KITTI dataset is its limited sample size of 2500 images. This could result in worse performance than the larger datasets.

(Loquercio et al., 2019) used domain randomization to bridge the gap between the artificial world and the real one, in the task of autonomous drone flight. In their work, they synthesized arbitrary race courses for the drone to learn to fly in, and then tested their controller in arbitrary track configurations in the real world. They achieved near perfect course completion scores for many variations including max speed constraints up to 10m/s, and lap totals less than 3.

In (Barisic et al., 2021), a network is trained by texture invariant object representation for aerial object detection. By a technique of randomly assigning atypical textures to UAV models, the obtained results confirm that shape plays a greater role in aerial object detection. The authors proved also that the training by the synthetic dataset outperforms baseline and even real-world data in situations with difficult lighting and distant objects.

On the other hand, supervised-learning-based approaches for airplane detection often require a large amount of training data, for the most important object detection in both military and civil aviation fields. The manual annotation, of an object such as airplane, in large image sets is generally expensive and sometimes unreliable, due to the significant appearance variations of airplane models, the airport area background is often complex and cluttered, and the airplanes can be at multiple scales on images. As a result, it is difficult to achieve accurate detection with training from a small amount of annotated real data. Besides, most research have used satellites as the data feeder. Accordingly, this work targets real-time airplane detection applications in airport areas, using synthetic images collected by domain randomization and processed through deep learning model.

3 METHOD OVERVIEW

Our goal is to detect airplanes in a specific scene and particularly in airport areas. In order to obtain synthetic images appropriate to our case, the SE-SCENARIO tool was used. More details about this tool will be found below. The most important challenge related to this work, as mentioned earlier, is how to ensure generalization of our network trained on synthetic data to unseen real data. To address this challenge, we employ domain randomization in the data generation step to create synthetic data with enough variations, such that the reality gap bridging will exist between the training and test data. In this section, we first describe our scene-specific data generation methodology followed by details about domain randomization.

3.1 Synthetic Data Generation

To generate our training images, we make use of SE-WORKENCH environment proposed by OKTAL-SE [1]. This solution uses physics-based sensor simulation software to generate synthetic dataset. The SE-WORKBENCH environment is a set of rendering of physical properties. Associated to a materials database, the SE-WORKBENCH can give all the information needed to render or perform physical computation on the SE-WORKBENCH database polygon, in a given area. A data generation scenario was created, in visible domain, for Toulouse-Blagnac Airport of Toulouse city in southwest France. Thus, SE-WORKBENCH allows us to produce a large amount of visual synthetic data, and also generates the annotations corresponding to these renderings. Moreover, other parameters can be controlled from this environment, such as atmospheric conditions, time of day, intrinsic parameters of the camera and also its position in the scene. These flexibilities are paramount to our domain randomization setup. The total number of our DR synthetic is about 6500 images with the resolution of 1280x960 pixels.

3.2 Domain Randomization

Domain Randomization attempts to generate a rich distribution of training images by introducing randomness into the data. When the synthetic data contains enough variability, the real world may appear to the model as yet another variant of what it has seen during training. The key idea here is to force a model to generalize to real-world data. Within a domain, various attributes can be parametrically randomized to produce data samples from a large variation of the possible image space. The resulting images, with automatically generated ground truth airplane labels (e.g., bounding boxes), are used for neural network training.

To conduct the training, the synthetic data were generated by randomly varying the following aspects of the scene:

3.2.1 Object Variation

A random number of positions are applied to the airplane model placed in different areas on the Toulouse-Blagnac airport. The airplane model used during the whole data generation is an Airbus-A320. To achieve airplane shape and size variations we have created two kinds of scenarios. The first one, when the airplane is placed on the airport tarmac and the camera is pointed at the airplane when it moves straight, as shown in Figs. [1-(a), 1-(b)]. Eight images Collection Points CPs were established every 50 meters until the airplane find the distance of 400 meters from the camera. At each CP, a rotation in the range of 0^o to 360^o around the Z axis with an angular spacing of 10^o were applied to the airplane. The second one, the airplane is positioned at 70m above the tarmac and 3-D rotations around the X-Y-Z were considered for the airplane. In fact, we took a single collection point, and the airplane was rotated in the range of 0^o to 360^o around each axis with an angular step of 10^o to obtain all the configuration of distinct 3-D orientations, as shown in Figs. 1-(f), 1-(g) and 1-(h)]. Since the environment is synthetic, these rotations were also complied with any airplane mechanical constraints.

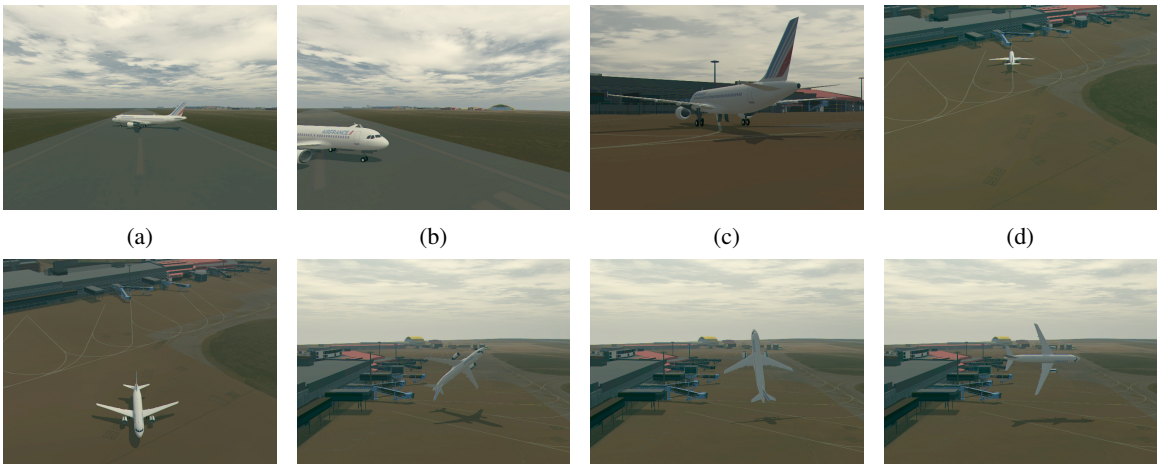
3.2.2 Background Variation

The background variation is achieved by varying the CPs locations at the airport. Indeed, two different locations were considered for the acquisitions. The first location is on the airstrip in an open area 1-(a). The second location is near to the airport terminal, where building textures are ubiquitous in rendered images 1-(c). Furthermore, we apply also light conditions varying by changing the time of day.

3.2.3 Viewpoint Variation

As a third variation, we randomized the position of the camera in the 3D environment in such a way that it is always placed around the airplane. For each position, the optical axis of the camera was pointed towards the airplane, and along this direction a translation motion was applied to the camera to obtain near and far

[1] Software Editor company expert in the development of ElectroOptics, RADAR and GNSS rendering simulation tools.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 1: Sample images generated by DR approach. Note that DR images yet contain more variety to force the deep neural network to focus on the structure of the objects of interest.

views, as illustrated in Figs 1-(c), 1-(d) and 1-(e). By using this variation, we intend to make the trained network robust to any change of perspective. Furthermore, to eliminate the need to collect new training data that only contains the airplane parts, a change in the orientation of the camera around its Z axis was involved. This forces the network to learn to deal with partial occlusion of the object of interest, as shown in Figs 1-(f), 1-(g) and 1-(h).

3.3 Detector Model Architecture and Training

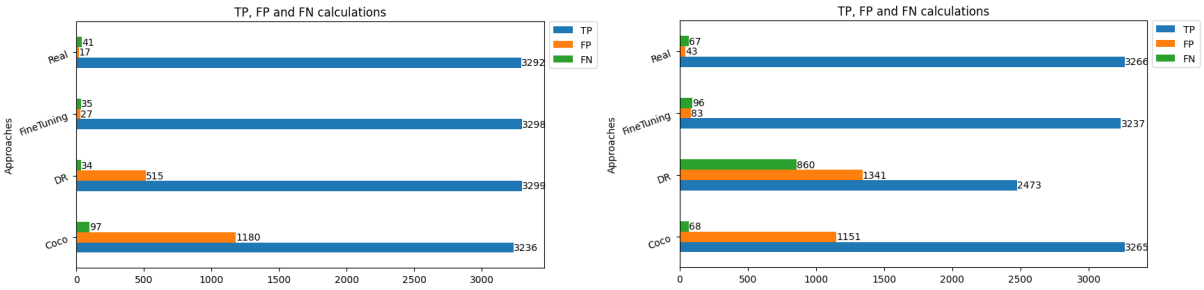
We parametrize our object detector with a deep convolutional neural network. The object detector used in this work comes from a well-established family of one-stage detectors called You Only Look Once (YOLO) (Redmon et al., 2016), (Redmon and Farhadi, 2017) and (Redmon and Farhadi, 2018). YOLO detectors treat object detection as a regression problem and use features from the entire image to detect objects. In this study, to efficiently and accurately inspect the DR for airplane detection with a real-time speed, the recent developed one-stage object detection framework, YOLO-v4 (Bochkovskiy et al., 2020), is selected. YOLO-v4 is a high-precision and real-time object detection algorithm based on regression proposed in 2020, which integrated the characteristics of YOLO-v1, YOLO-v2 and YOLO-v3.

YOLO-v4 object detection algorithm consists of three structures: the backbone, the neck, and the heads. The main function of the backbone block is the feature extraction process. Selection of the backbone is one of the most important steps to increase performance of the object detection algorithm. The purpose of the neck block is to add extra layers between the backbone and the head so that feature maps from different stages can be combined. Usually, a neck consists of several bottom-up paths and several top-down paths. The final stage, the head block in single-stage detectors, performs the final prediction. This prediction consists of a vector containing the coordinates of the predicted bounding box, the confidence score, and the label of the prediction (Pacal and Karaboga, 2021).

For experiments, we used the pretrained model yolov4.conv.137 and trained for 2000 epochs. The detector was modified in way to accommodate training and testing for one class. Each input image is resized to 512x512 before passing into YOLOv4 architecture. All the experiments are carried out by using 512x512 input image size. The reason behind choosing 512 image patch is to make computations more simple and the number of images per batch is limited to 64. The learning rate and the momentum are set to 0.0001 and 0.9, respectively. The weights are saved after every 1000 epochs so that we can calculate mAP results to make sure that the model is learning well.

4 EVALUATION

We benchmark our methodology on the FGVC-Aircraft dataset (Maji et al., 2013). This dataset contains 100 example real images for each of the 100 model variants for aircraft. Therefore, the dataset contains 10,000 annotated images and their resolution is about 1-2 Mpixels. Images are equally divided into training, validation, and test subsets, so that FGVC-Aircraft dataset has 34:33:33 split as the training, val-



(a) AP@0.5. (b) AP@0.75.

Figure 2: Results of True Positive (TP), False Positive (FP) and False Negative (FN) on real-FGVC-Aircraft test dataset of YOLOv4. The models training's established on COCO dataset, our DR data, fine-tuning and FGVC-Aircraft real images, for comparison.

idation and test set. We perform evaluations on test subset and the 3333 images, thus, constitutes our real data for evaluations. The bounding box information has ben used for validating our airplane detector algorithm.

For all our experiments, we compare four models.

- COCO: YOLOv4 is trained on COCO dataset (Veit et al., 2016)
- DR: initialized with YOLOv4 random weights and trained on domain randomized synthetic data.
- DR+finetune : initialized with DR and fine-tuned on 10% FGVC-Aircraft real training data, about 333 images.
- Real: initialized with YOLOv4 random weights and trained on the entire FGVC-Aircraft training data.

Quantitative Analysis: Table 1 compares the performance of the YOLOv4 when trained on COCO-dataset, FGVC-aircraft dataset versus our DR dataset. Our DR and DR+fine tuning achieves the highest scores compared to the real COCO-dataset for AP=0.5. We hypothesize this is due to overfitting on limited real data which is avoided by design in domain randomization. Besides, the individual parameters for the True Positive(TP), False Positive(FP) and False Negative(FN), who participate directly on the calculation of Precision, Recall and F1-score metrics, are shown in Fig 2. This helps to explain the metric results of Table 1. The necessary conclusion to draw from all these results, is that our model trained only on synthetic data is competitive to the model trained on the entire real data at IoU=0.5. DR+finetune outperforms the off-shelf baseline COCO and DR, which illustrates the benefits of training on real images after first training on synthetic images. Furthermore, even though our DR network has never seen a real image, it is able to successfully detect different models of airplane present on FGVC-aircraft dataset. This surprising result illustrates the power of such a simple technique as DR for bridging the reality gap. Also, we found that the bounding boxes of DR and mostly DR+finetune are more accurate than COCO-dataset. We argue that this is because DR helps to mitigates this problem by randomizing the dimensions and shapes of the airplane during data generation, thus directly providing enough variance in the bounding box distribution.

Consequently, by the use of randomization techniques, we achieve an improvement for real airplanes detection, outperforming the results without DR. The weights of pretrained network on Coco-dataset is at a local minimum at the beginning of the training. The bigger variance in the randomized images might help to force the network out of its local minimum. However, the non-randomized is likely to stay close to its

Table 1: Detection results tested on FGVC-Aircraft dataset.

	AP@0.5				AP@0.75			
Approaches	Precision	Recall	F1-score	aIoU(%)	Precision	Recall	F1-score	aIoU(%)
Coco	0.4	0.97	0.57	36.53	0.41	0.98	0.58	37.12
DR	0.86	0.99	0.92	69.29	0.65	0.74	0.69	54.2
Fine-tuning	0.99	0.99	0.99	87.96	0.98	0.97	0.97	88.90
Real	0.99	0.99	0.99	93.04	0.99	0.98	0.98	92.5

Figure 3: Examples of detection results. We compare the performance of COCO (first column), DR (second column) and DR+finetune (third column) with real FGVC-Aircraft data, on three different views. Each row contains one of theme .

local minimum. As a result, the randomization techniques are a very fundamental tool for achieving better accuracy, when applied to pretrained networks.

Qualitative Analysis: A qualitative comparison between the purely DR, DR+finetune trained detector and the detector trained on COCO and FGVC-Aircraft real images, is shown in Fig. 3. In the presented images obtained from free video of airplane landing, a detector trained on real images fails to detect one airplane due to the challenging conditions of distant-objects and difficult illumination. COCO fails in cases of false positive and false negative detection of other object classes. In contrast, for purely synthetic data and also with finetune, the detector can successfully and accurately detect the airplane in three different view images. These results show that by DR the real gap can be bridged and that the accuracy of the real data can even be surpassed. Thus, the DR is a fundamental tool for achieving high accuracy with synthetic training data. They benefit from randomization in particular, because the increased variance in the images forces the networks to ascent out of local minimum.

5 CONCLUSIONS

This work was motivated by real-time airplane detection on airport zones in a context where acquiring large amount of annotated images is not easily accessible. To tackle this challenge, we propose a solution that uses Domain Randomization to train the YOLO-v4 architecture, based upon synthetic images. The results of the experimental evaluation of the synthetically generated dataset confirm that DR is an effective technique to bridge the reality gap, to accomplish the task of airplane detection. We demonstrated also that with fine-tuning on quiet amount of real images, the DR can outperforms the real datasets and our method achieves higher detection accuracy than state-of-the-art baselines and even real-world data in situations with difficult lighting and distant objects. Thus, using DR for training deep neural networks is a promising approach to bridge the reality gap between simulation and the real world and other direction can be explored to generalize this technique to solve other issues on computer vision.

Future directions that should be explored include using several types of airplanes, different weather and light conditions, applying the technique to motion and/or 3D depth estimation, and further investigating the mixing of synthetic and real data to leverage the benefits of both.

ACKNOWLEDGEMENTS

We would like to thank OKTAL-SE for providing us SE-SCENARIO and SE-TOOLKIT tools to generate synthetic data.

REFERENCES

Agarwal, S., Terrail, J. O. D., and Jurie, F. (2019). Recent advances in object detection in the age of deep convolutional neural networks.

Barisic, A., Petric, F., and Bogdan, S. (2021). Sim2air-

synthetic aerial dataset for uav monitoring. *arXiv preprint arXiv:2110.05145*.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Bousmalis, K., Irpan, A., Wohlhart, P., Bai, Y., Kelcey, M., Kalakrishnan, M., Downs, L., Ibarz, J., Pastor, P., Konolige, K., Levine, S., and Vanhoucke, V. (2018). Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4243–4250.

Gaidon, A., Wang, Q., Cabon, Y., and Vig, E. (2016). Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349.

Hinterstoisser, S., Pauly, O., Heibel, H., Marek, M., and Bokeloh, M. (2019). An annotation saved is an annotation earned: Using fully synthetic training for object instance detection.

Loquercio, A., Kaufmann, E., Ranftl, R., Dosovitskiy, A., Koltun, V., and Scaramuzza, D. (2019). Deep drone racing: From simulation to reality with domain randomization. *IEEE Transactions on Robotics*, 36(1):1–14.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. (2013). Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

Nowruzi, F. E., Kapoor, P., Kolhatkar, D., Hassanat, F. A., Laganiere, R., and Rebut, J. (2019). How much real data do we actually need: Analyzing object detection performance using synthetic and real data.

Pacal, I. and Karaboga, D. (2021). A robust real-time deep learning based automatic polyp detection system. *Computers in Biology and Medicine*, page 104519.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Redmon, J. and Farhadi, A. (2017). Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE.

Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., and Birchfield, S. (2018). Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977.

Valtchev, S. Z. and Wu, J. (2020). Domain randomization for neural network classification. *Journal of Big Data*.

Veit, A., Matera, T., Neumann, L., Matas, J., and Belongie, S. (2016). Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*.