ColANet: A UAV Collision Avoidance Dataset

Dário Pedro[1,3](✉), André Mora[3], João Carvalho[2,3], Fábio Azevedo[2],
and José Fonseca[3]

[1] PDMFC Research Group, Lisbon, Portugal
dario.pedro@pdmfc.com
[2] Beyond Vision Research Group, Ílhavo, Portugal
{joao.m.carvalho, fabio.azevedo}@beyond-vision.pt
[3] CTS/Uninova, FCT, NOVA University of Lisbon, Caparica, Portugal
{atm, jmf}@uninova.pt

Abstract. Artificial Intelligence is evolving at an accelerating pace alongside the increasing number of large datasets due to vast number of image data on the Internet. Unnamed Aircraft Vehicles (UAVs) are also a new trend that will have a huge impact over the next years. The use of UAVs arises some safety issues, such as collisions with dynamic obstacles like birds, other planes, or random thrown objects. Those are complex and sometimes impossible to avoid with state-of-the-art algorithms, representing a threat to the applications. In this article, a new video dataset of collisions, entitled ColANet, aims to provide a base for training new Machine Learning algorithms for handling the problem of avoiding collisions with high efficiency and robustness. It is also shown that using this dataset is easy to build new neural network models and test them.

Keywords: UAS · UAV · Safety · Artificial Intelligence · Machine Learning · Neural network · Dataset · Collision avoidance

1 Introduction

Images contain a high amount of information in a relatively concise way [1]. However, their processing is hard and resource-consuming due to the infinite type of variations that might appear to represent the same structure. Adding a time reference and sequencing the images we can build videos. Due to the fast development of cameras, CPUs, and image and video processing algorithms [2], this kind of data source is becoming widely used. For example, on YouTube [3], are uploaded approximately 72 h of videos in every minute, being expected that, by the end of 2020, online video will be responsible for four-fifths of global internet traffic.

Unnamed Aircraft Vehicles (UAVs) have as main objective aiding or making possible the execution of difficult or impossible tasks by the human being [4, 5]. Nowadays, UAVs are not a strange but instead a trendy tool used in the industry market. Despite the advantages, UAVs' operation might be challenging when operating in complex or confined environments [6, 7]. The presence of obstacles is dangerous and requires the use of obstacle detection and collision avoidance algorithms. For the static obstacles, there are already algorithms that deal relatively well with them, being capable of generating safe paths to the desired positions [8]. However, in real applications,

Published by Springer Nature Switzerland AG 2020
L. M. Camarinha-Matos et al. (Eds.): DoCEIS 2020, IFIP AICT 577, pp. 53–62, 2020.
https://doi.org/10.1007/978-3-030-45124-0_5

it is very frequent to find dynamic moving obstacles. The latter can be detected by using the video stream provided by the onboard cameras. Video processing is computationally heavy [9] and collision avoidance algorithms need to be fast enough to retrieve solutions without colliding. Therefore, a neural network trained for this kind of situations is useful. The problem of the convolutional neural networks is that they are generally data-hungry, turning them extremely hard to train with small datasets, which leads to a very likely memorization of the dataset [10].

Given this, the objective of this article is to present and make available a new open source dataset to accelerate and facilitate the development of new collision avoidance algorithms, increasing safety and performance of UAVs. This dataset, will also be one module of a bigger framework for designing safer UAVs.

The remaining of this article is structured in the following sections: Sect. 2 links the article with the PhD student work and technology for life improvement, Sect. 3 provides an overview of the existing datasets and their usage to build Machine Learning (ML) models. Section 4 introduces the ColANet dataset. In Sect. 5 it's presented the results of an experimental deep neural network model that was trained using the ColANet dataset. The article is then concluded, and the future work is discussed in Sect. 6.

2 Contribution to Life Improvement

The growth of the commercial UAV industry emphasises the need of a conceptual framework for UAV design and arrises the following research question:

> *What could be an adequate conceptual framework for designing generic autonomous vehicles that characterizes individual aircrafts behavior and aggregates them in a network, ensuring reliability and safety in the flight, regardless of the world conditions and unexpected events?*

A possible hypothesis for this question would be that the desired framework can be build utilizing different nodes, that are based on SoA architectures, handling atomic tasks separately but performing complex tasks in collaboration. Furthermore, new complex blocks that don't exist can be developed utilizing ML technology and then integrated into the developed framework.

A good implementation of this framework with all the relevant components working in symbiosis would increase commercial UAV safety by decreasing the number of UAV accidents in urban areas. UAVs are tackling everything in from disease control to vacuuming up ocean waste or even delivering pizza [11], which are all areas that safety is important, and such framework would improve their feasibility, ultimately improving daily lives. This article is a step towards the construction of such framework and the novel collision avoidance ML based blocks.

3 Related Datasets

In the last decade, there has been an increasing number of publications of datasets that are enabling the development of new ML models and solutions [12, 13]. In this section it's presented a review of existing datasets, highlighting their relevance and impact in the field. These datasets were selected using both a criteria of usefulness for the community (novel data or utility scenarios) and relevance (amount of citations reports, usage on benchmarks zoos, scientific quality extrapolated by top-tier conferences and journals). As can be found in other works [14], this paper related work review will be split according to their data representation, 2D or RGB datasets, 2.5D or RGB-Depth datasets and 3D or video (volumetric) datasets.

3.1 2D Datasets

Most of vision ML algorithms were developed using 2D datasets that tried to understand the correlation between pixels for image classification. In this section we describe 2D image datasets, whether they are RGB or grayscale.

Probably the most revolutionary that waken CNNs (Convolutional Neural Networks) is MNIST [15], which is a database of handwritten digits, has a training set of 60,000 examples, and a test set of 10,000 examples. As it is being used as an example in several papers and courses in this area, it is one of the most used worldwide.

Other popular 2D dataset are ImageNet [16, 17], LabelMe [18], Microsoft Common Objects in Context (COCO) [19] and Pascal Visual Object Classes (VOC) [20].

3.2 RGB-Depth Datasets

Recently, cameras with multiple sensors that capture both RGB and depth are becoming more popular due to their decrease in price and increase of applications. SUNRGBD [21] dataset is a good example within this category. It was captured with multiple RGB-D sensors, containing more than 10.000 images. It merges data from multiple other datasets such as NYU depth v2 [22], Berkeley B3DO [23] and SUN3D [24]. It is highly annotated with polygons, bounding boxes, layout info and categories, being excellent for scene understanding tasks.

Another similar datasets in content type are Objectnet3D [25], ScanNet [26] and others [24, 27–29].

3.3 Video Datasets

Three-dimensional databases are more unusual but are the ones that can be easily compared with the dataset introduced by our article. They are either point-clouds or videos, which are costly to store and difficult to segment and annotate.

From the point-cloud type, we have parts of databases such as Objectnet3D [25] used for 3D object recognition with 100 categories, 90,127 images, 201,888 objects in these images and 44,147 3D shapes. The objects in the 2D images are aligned with the 3D shapes, and the alignment provides both 3D pose annotation and the closest 3D shape annotation for each 2D object.

From the video type, a good example is ScanNet [26] which consists of a video dataset that contains 2.5 M views in 1513 scenes annotated with 3D camera poses, surface reconstructions, and semantic segmentations. The data was collected using a scalable RGB-D capture system that includes automated surface reconstruction and crowdsourced semantic annotation.

Other datasets can be found in [28–30] and cover more complex use cases, like occlusions or unstructured point clouds.

4 ColANet Dataset

The ColANet can be summarized as a Collision Avoidance Video Dataset. This dataset is an open repository of UAV collisions and intends to be an initial step towards safer UAV's operations without collisions.

The ColANet dataset has the particularity to let the user to upload a video, allowing an easier annotation of a video, frame by frame with an escape vector, like it's presented in Fig. 1.

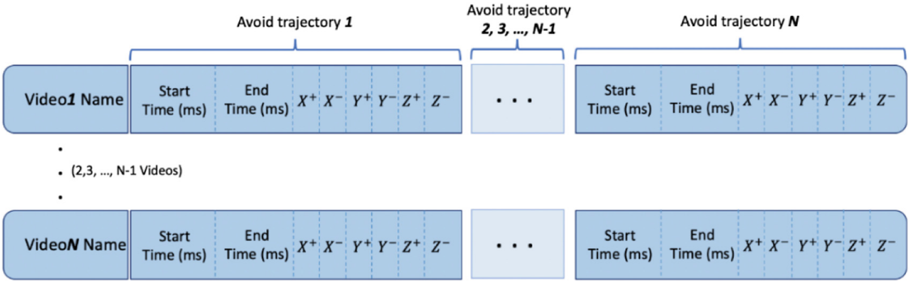

Fig. 1. File structure used on dataset annotated frames generation.

As Fig. 1 illustrates, each row consists of four elements:

- Video name.
- Start Time (in milliseconds).
- End Time (in milliseconds).
- Escape Vector $(X^+, X^-, Y^+, Y^+, Z^+, Z^-)$.

The start time, end time and escape vector can have multiple occurrences per row, representing multiple potential collision situations where the escape vectors might differ.

With this information, the server iterates over all videos (one per row) and generates a labeled set of images that are extracted from the video frames. For this purpose, the algorithm opens the video, retrieve the frames per second information, and then generates one image per frame, and a text file containing the annotations. The directions of the escape vector are illustrated in Fig. 2, where is possible to see a UAV with the directions vector overlaid.

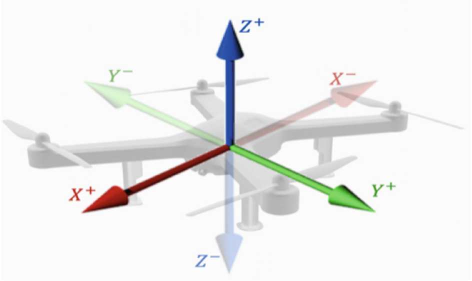

Fig. 2. UAV with escape vector.

The software written in python is open source and can be found alongside the dataset[1]. Note that this also gives the freedom to fine-tune the working dataset, since the user can run normalization and regularization when passing the data from a video to a temporal labeled set of images.

The provided version of the dataset already contains 100 videos of drone collisions that were recorded during flights of different models, with different environment conditions (sunny days, cloudy days and during the night), and some examples are represented in Fig. 3. In most videos, the drones are flying freely until the moment of collision. This videos are already labeled with escape vectors, and represent a total of over 2000 collision frames and 6000 free flying frames.

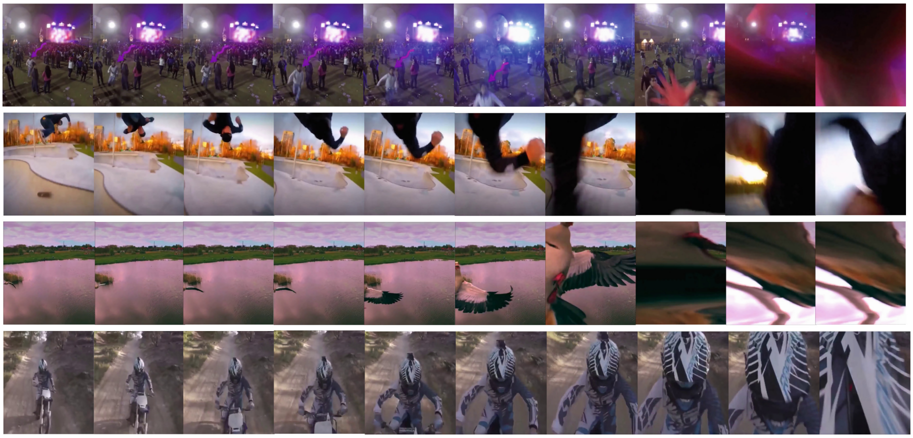

Fig. 3. Visualization of 10 frames from 4 different collision videos on ColANet.

To train an algorithm that classifies the current instant (frame) as collision or no collision, all it has to do is check if all the numbers on the row are 0, and he can label it

[1] The dataset can be downloaded at https://colanet.qa.pdmfc.com/.

as ‘no collision’. If the researcher intends to directly estimate the escape route or trajectory, he can use the escape vector directly and from the algorithm output, take actions for avoiding the collision.

5 Experimental Neural Network Using ColANet

To test the dataset presented in this article, a model based on VGG16 [31] (Fig. 4) was trained. For the sake of simplification, the output of the neural network was translated from an escape vector to two labels (collision or no collision). The escape vector file was iterated and the frames whose all 6 escape values equaled 0, were labelled as ‘no collision’; all the others received the label ‘collision’.

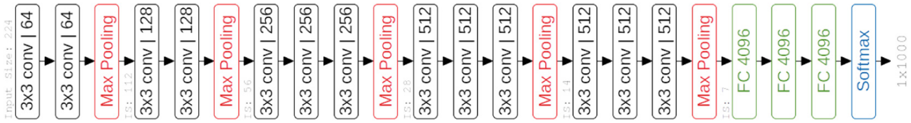

Fig. 4. VGG16 architecture.

As stated in Fig. 4, the output has the form of a 1 × 1000 probability vector, which is not what is pretended (only 2 classes). To overcome that, the VGG16 model was adapted, removing the classifier block (Fig. 5) and adding a new dense layer with 1024 neurons and a SoftMax activation layer (Fig. 6). The output reduced to a 1 × 2 probability vector.

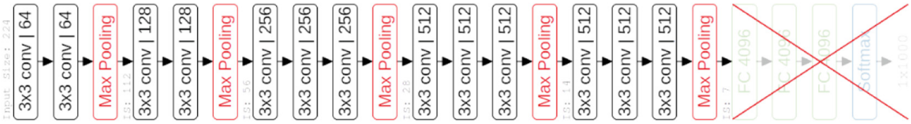

Fig. 5. VGG16 without the classifier block.

The newly added layers are composed of a Flatten layer (because the input is from a convolutional layer), a Dense layer with a ReLU activation function, a Dropout layer with a dropout of 0.5 to prevent overfitting and, finally, a Dense layer with a SoftMax activation function.

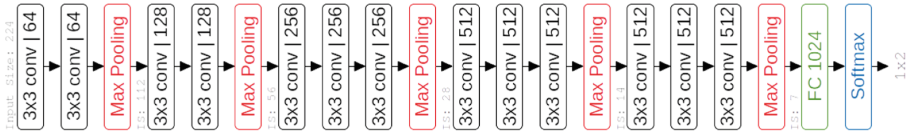

Fig. 6. Our model based on the VGG16 architecture.

An epoch usually corresponds to a complete processing of the training-set. However, the data-generator used from TensorFlow produces batches of training-data for eternity. Therefore, it's required to define the number of steps we want to run for each *epoch*, that will be multiplied by the *batch-size* defined. In the illustrated mode, it's used 100 steps per *epoch* and a *batch-size* of 20. So a *pseudo-epoch* consists of 2000 random images from the training-set. It was run those *pseudo-epochs* 20 times.

These values were chosen empirically, because they were enough to complete the training with this model and dataset, taking around 12 h on an Intel i7 8700 with a Nvidia RTX 2070. The results also contain 20 data-points (one for each *pseudo-epoch*) which can be plotted afterwards. It is also worth noting that the input frames are all normalized in order to have values that range from 0 to 1.

Using Transfer Learning techniques [32], it is possible to check the compatibility and interoperability of the presented dataset. For that, it was used the network weights obtained in the ImageNet [16, 17]. For getting most of this pre-trained network without compromising the results of the classifier block (last layers of the model), the training was split into two phases. In the first phase, the layers of the default VGG16 model were freeze, having only the newly added layers released for training. After 20 *pseudo-epochs*, all the network model is released and the training proceeds, but now adjusting the weights of all the layers. This technique takes advantage of the pre-trained weights of the model with another dataset to calculate the initial weights of the new layers. Releasing all the layers for training in the last step can be considered as a fine-tuning of the weights calculated initially.

5.1 Training Results

The ColANet dataset was used to train the presented model. The training and test accuracies were measured during all the training procedure. In order to prove that the network must be trained, the test set was evaluated before any training. Then the results were evaluated both after training the classifier (keeping the default ImageNet weights) and after the fine-tuning. The results are shown in Table 1.

Table 1. Training classifier results.

VGG – Pre train	VGG – Trained classifier		VGG – Fully trained	
Test	Train	Test	Train	Test
58,18%	93,89%	92,73%	96,53%	94,55%

During the model training, on the first phase, it's able to reach an accuracy of 92,73% with the test data when training. After the fine-tuning, the final calculated weights allowed to get an accuracy of 94,55%. Due to the similarity with the training accuracies, we can assume that the model wasn't memorizing nor overfitting the training data. In Figs. 7 and 8 is depicted the evolution of the accuracy and loss values over all the *pseudo-epochs*.

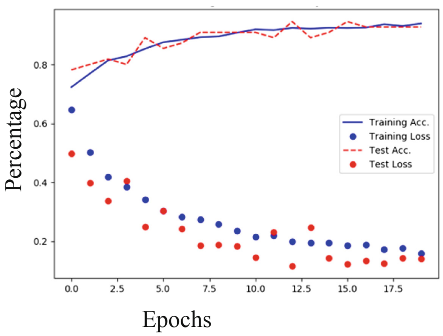

Fig. 7. Training classifier.

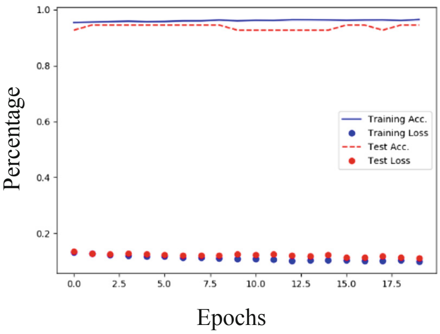

Fig. 8. Full training.

6 Conclusions

This article introduces a new video dataset with the purpose of training safer and reliable UAV models. On this paper was proposed both a module that consumes a meta-data file to generate per frame annotations with escape vectors, indicating how to avoid collisions, and also a new dataset of 100 UAV collision videos.

In addition, it was presented a prototype of a collision detector model based on VGG-16. The experimental results showed that is possible to train such a deep neural network, and reuse other datasets using Transfer Learning techniques. This work intends to be a starting point for new UAV architectures and new ML algorithms that leverage the dataset annotations.

Future work will consist of enlarging the dataset with more variety and different videos of UAV collisions, and also the development of new neural networks that leverage not only the spatial frame information (like the one presented on this article), but also the temporal information of the video (like Long Short-Term Memories); to produce better results on real case scenarios. New spatio-temporal algorithms are also envisioned, which combine CNNs and RNNs approaches.

Acknowledgments. This work was partially funded by FCT Strategic Program UIDB/00066/2020 of the Center of Technologies and System (CTS) of UNINOVA - Institute for the Development of new Technologies.

This work was supported by the European Regional Development Fund (FEDER), through the Regional Operational Programme of Lisbon (POR LISBOA 2020) and the Competitiveness and Internationalization Operational Programme (COMPETE 2020) of the Portugal 2020 framework [Project 5G with Nr. 024539 (POCI-01-0247-FEDER-024539)], and also SECREDAS, which received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement nr.783119.

The authors would like to thank the PDMFC Research Group, in particularly Luis Miguel Campos.

References

1. Berg, A.C., et al.: Understanding and predicting importance in images. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2012)
2. Akyildiz, I.F., Melodia, T., Chowdury, K.R.: Wireless multimedia sensor networks: a survey. IEEE Wirel. Commun. **14**, 32–39 (2007)
3. Stewart, P.: YouTube. In: The Live-Streaming Handbook (2019)
4. Amazon.com Inc.: Determining safe access with a best-equipped, best-served model for small unmanned aircraft systems. In: NASA UTM 2015: The Next Era of Aviation (2015)
5. Hartmann, K., Giles, K.: UAV exploitation: a new domain for cyber power. In: International Conference on Cyber Conflict (CYCON) (2016)
6. Ryan, A., Zennaro, M., Howell, A., Sengupta, R., Hedrick, J.K.: An overview of emerging results in cooperative UAV control (2008)
7. Pedro, D., et al.: Localization of static remote devices using smartphones. In: IEEE Vehicular Technology Conference (2018)
8. Matos-Carvalho, J.P., Pedro, D., Campos, L.M., Fonseca, J.M., Mora, A.: Terrain classification using W-K filter and 3D navigation with static collision avoidance. In: Bi, Y., Bhatia, R., Kapoor, S. (eds.) IntelliSys 2019. AISC, vol. 1038, pp. 1122–1137. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-29513-4_81
9. Waizenegger, W., Feldmann, I., Schreer, O.: Real-time patch sweeping for high-quality depth estimation in 3D video conferencing applications. In: Real-Time Image and Video Processing 2011 (2011)
10. Zhao, B., Wu, B., Wu, T., Wang, Y.: Zero-shot learning posed as a missing data problem. In: Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW 2017) (2018)
11. PwC: How Drones Will Impact Society: From Fighting War to Forecasting Weather, UAVs Change Everything. CB Insights Research (2020). https://www.cbinsights.com/research/drone-impact-society-uav/. Accessed 05 Jan 2020
12. Hutter, F.: Automated Machine Learning (2019)
13. Wu, C.J., et al.: Machine learning at Facebook: understanding inference at the edge. In: Proceedings of the 25th IEEE International Symposium on High Performance Computer Architecture (HPCA 2019) (2019)
14. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P., Garcia-Rodriguez, J.: A survey on deep learning techniques for image and video semantic segmentation. Appl. Soft Comput. J. **70**, 41–65 (2018)
15. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**, 2278–2324 (1998)
16. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (2009)
17. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
18. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. Int. J. Comput. Vis. **77**, 157–173 (2008)
19. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48

20. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: a retrospective. Int. J. Comput. Vis. **111**, 98–136 (2014). 10.1007/s11263-014-0733-5
21. Song, S., Lichtenberg, S.P., Xiao, J.: SUN RGB-D: a RGB-D scene understanding benchmark suite. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2015)
22. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7576, pp. 746–760. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33715-4_54
23. Janoch, A., et al.: A category-level 3-D object dataset: putting the Kinect to work. In: Proceedings of the IEEE International Conference on Computer Vision (2011)
24. Xiao, J., Owens, A., Torralba, A.: SUN3D: a database of big spaces reconstructed using SfM and object labels. In: Proceedings of the IEEE International Conference on Computer Vision (2013)
25. Xiang, Y., et al.: ObjectNet3D: a large scale database for 3D object recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 160–176. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_10
26. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: richly-annotated 3D reconstructions of indoor scenes. In: Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017) (2017)
27. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view RGB-D object dataset. In: Proceedings of the IEEE International Conference on Robotics and Automation (2011)
28. Hackel, T., Wegner, J.D., Schindler, K.: Contour detection in unstructured 3D point clouds. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2016)
29. Quadros, A., Underwood, J.P., Douillard, B.: An occlusion-aware feature for range images. In: Proceedings of the IEEE International Conference on Robotics and Automation (2012)
30. Chen, X., Golovinskiy, A., Funkhouser, T.: A benchmark for 3D mesh segmentation. In: ACM SIGGRAPH 2009 papers on - SIGGRAPH 2009 (2009)
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition, September 2014
32. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**, 1345–1359 (2009)