Automatic Generation of Images Using Unreal Engine for Supervised Learning

Caio Brasil, Ryan Reis, Ailton Oliveira, Carnot Braun, Lucas Damasceno, Ilan Correa and Aldebaro Klautau

***Abstract*— Many applications of machine learning (ML) require a large amount of labeled data to be used in practical deployments. Collecting data with labels can be a laborious and time-consuming task. A sensible alternative that has been widely adopted recently is to use synthetic data, generated by simulations, and automatically create labels within this process. This paper uses this approach to generate realistic datasets for training ML models to be used in computer vision. The proposed methodology is based on 3D virtual environments created by the Unreal Engine, and geometric relations to properly position the bounding boxes corresponding to each object of interest. To validate the methodology, a dataset of 3000 labeled images was generated in 2.5 minutes.**

***Keywords*— Computer vision, synthetic data, virtual world, supervised learning.**

I. Introduction

Due to the advances of convolutional neural networks (CNNs) and related algorithms, computational vision has gained popularity and has been an important asset for machine learning applications, such as image classification and object detection. Though, the versatility and accuracy of these networks are proportional to the quality and diversity of the dataset used to train the model. However the process of collecting and labeling large datasets can be expensive and time-consuming [1]. To overcome this challenge, generating synthetic datasets is a common approach to speed up the tasks of verification, simulation or proof of concept [2] in many research areas.

There is a diversity of methods and platforms to help generating and labeling datasets. For instance, the heuristic applied in the Synscapes dataset [3] is based on a procedural engine to create a unique image in each interaction. However, this engine only allows the generation in a street scenario. Another example is UnrealCV [4], which inspects the environment's features and provides rich ground truth along the images in pre-generated scenarios. However, the UnrealCV doesn't handle with labeling tasks.

In this context, the contribution of this work is to present a general approach to generate synthetic datasets of images automatically labeled. The approach is based on a trigonometric algorithms, with the possibility of using customized scenarios and allowing a 360º field of view of target object, removing the manual work in the steps of taking and labeling images to compose the dataset. Thus, complex methodologies which use visual data [5] can be positively influenced by the automatic data generation and labeling process.

The environment presented in this paper is based on the integration of open-source tools, such as Airsim and Unreal Engine, to automate the generation of images from a 3D environment, as well as the process of placing the *bounding box* (BB) and labels. In this way, it simplifies the generation of large volumes of data, which can accelerate the final application's research and development process. Another contribution of this paper is to present a benchmark of reliability of the labeled data generated. For that, the YOLOV7 with the tiny architecture [6], was employed with the synthetic dataset generated to perform object detection for two classes: motorcycle and car. The codes and the dataset developed in this paper are available at GitHub[1].

II. Methodology For data generation

Unreal Engine is a 3D computer graphics game engine developed by Epic Games that was used to obtain realistic 3D environments. To simulate the sensors and fluid movement of the camera to extract data from the scenario and ensure flexibility in getting images from different angles, it was used Airsim, a cross-platform simulator developed by Microsoft with an emphasis on artificial intelligence and vehicular automation research. The methodology described in this work can be applied in a drone to automate the generation of real objects dataset's, thanks to Airsim support for software-in-the-loop and hardware-in-loop simulation with flight controllers, such as PX4, allowing the transposition of the implementation in software to the drone's hardware. It is important to emphasize that the software was designed to work in any Unreal environment, but for the experiments described in this article, it was used the Mountain Landscape environment, which has realistic roads, mountains, and lighting effects.

To set the camera's position, we have considered that the object under analysis is situated at the center of the lower face of two co-located rectangular prisms, of which one is slightly greater than the other. This strategic placement allows the camera to be flexibly positioned within the region defined by the difference in volume of these two solids, thereby enabling precise control over the distance between the camera and the object of analysis.

Besides the camera's position, it is also needed to set its orientation, which allows to focus on the object. The pitch

C. Brasil, R. Reis, A. Oliveira, C. Braun, L. Damasceno, I. Correa and A. Klautau are with LASSE UFPA, Belém-Pará, Brasil; E-mails: [caio.brasil, ryan.oliveira, ailton.pinto, carnot.filho, lucas.damasceno.silva] @itec.ufpa.br, [ilan, aldebaro] @ufpa.br. The authors thank RNP/MCTI for the financial assistance to the project Brasil 6G (01245.010604/2020-14).

[1]https://github.com/lasseufpa/yolo-data-generation-SBRT23

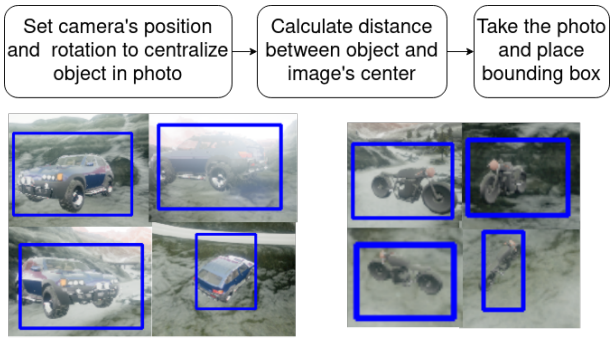

Fig. 1: Steps to obtain dataset's process with result's images.

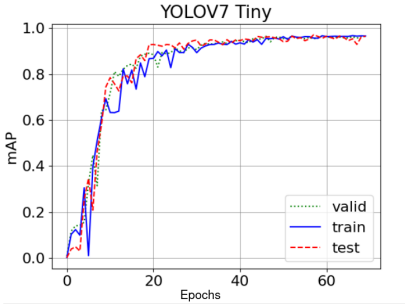

Fig. 2: mAP using the synthetic data with YOLOV7 tiny.

and yaw rotations are calculated as,

$$pitch = \arctan\left(\frac{Z_{rel}}{\sqrt{X_{rel}^2+Y_{rel}^2}}\right) + \alpha \quad (1)$$

$$yaw = \arctan\left(\frac{Y_{rel}}{X_{rel}}\right) + \beta \quad (2)$$

where X_{rel}, Y_{rel} and Z_{rel} is the relative distance between the camera and the object. α and β are random factors used to add diversity to object's position within the image. In this paper, α has a uniform distribution between -0.1 and 0.1 radian, and β a uniform distribution between -0.6 and 0.6 radian.

After calculating the rotation, it is necessary to analyze the relative position between the object and the camera to determine whether the camera is positioned in front or behind the object. If the camera is positioned in front, an additional π radian is added to the yaw, so the camera can be directed to the object. With this procedure to set the position and rotation of the camera, the pitch and yaw values are adjusted to allow the variation in the camera's rotation within a range that ensures that the object is framed in the image. This methodology promotes the diversification of the images generated by framing different angles of the object.

The following step involves calculating the position of the BB on the image. Initially, it is assumed that the BB's are centered on the images. Next, it is analyzed the influence of the distance between the object and the camera on the width and height of the BB's. The formula acquired was:

$$\mathbf{Bx} = f_1\sqrt{2D^2(1-\cos(yaw)} \quad (3)$$

$$\mathbf{By} = f_2\sqrt{2D^2(1-\cos(pitch)} \quad (4)$$

where **Bx** and **By** are the list of BB vertices axis to frame the object, and D is the distance between the object and the camera. The parameters f_1 and f_2 are used to convert the distance between the object position and the images's center into number of pixels. Figure 1 resumes the process to get the pictures with the BBs and show some images generated.

III. Evaluation

To validate the generated dataset, it was used the YOLOV7 Tiny because of its speed and accuracy in comparison of other models. It was used a batch size of 7 and 75 epochs. The training was executed in a computer with the eight core Intel Core i7-7700 CPU@3.60GHz, 32GB of RAM and a GeForce RTX 2070. The dataset consists of 3000 images obtained with the methodology presented in this paper, wherein 1500 focused on the car and 1500 on the motorcycle. To train the model, the dataset was divided in the proportion: 70% for training, 20% for validation, and 10% for testing.

To evaluate the learning, the mAP metric was used, which can provide a comprehensive measure of the model performance over the labeling process. This metric analyses whether the process could place the BBs in such a way that allows the model to converge about all the classes analyzed.

Figure 2 shows the network's performance over the epochs for the training, validation and testing datasets with the YOLOV7 tiny, wherein it was achieved a mAP higher than 96% for those datasets. The high value of mAP indicates the competence of the methodology in placing the BBs properly, allowing the network's generalization.

IV. Conclusion

This work presented a methodology to automatically generate images using free 3D environments where any object of interest can be simulated. The objects in the generated images are automatically identified with bounding boxes. This method could generate 3000 samples in 2.5 minutes, indicating its efficiency and effectiveness. With these features, the studies with supervised learning models can become less time-consuming and more efficient when removing the difficulty of obtaining a dataset with a high number of diverse examples. Some further investigations consist of analyzing the viability of this methodology over application of computer vision in real life scenarios.

References

[1] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, 2023.

[2] G. Albuquerque, T. Lowe, and M. Magnor, "Synthetic generation of high-dimensional datasets," *IEEE transactions on visualization and computer graphics*, vol. 17, no. 12, pp. 2317–2324, 2011.

[3] M. Wrenninge and J. Unger, "Synscapes: A photorealistic synthetic dataset for street scene parsing," 2018.

[4] W. Qiu and A. Yuille, "Unrealcv: Connecting computer vision to unreal engine," 2016.

[5] I. Correa, A. Oliveira, B. Du, C. Nahum, D. Kobuchi, F. Bastos, H. Ohzeki, J. Borges, M. Mehta, P. Batista *et al.*, "Simultaneous beam selection and users scheduling evaluation in a virtual world with reinforcement learning," 2022.

[6] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.