HOCHSCHULE
RAVENSBURG-WEINGARTEN
UNIVERSITY
OF APPLIED SCIENCES

Synthetic Data Generation for Enhanced Computer Vision applications: A CAD model and Blender approach

Dhanush Gangadharaiah[a,*]

[a] *Institute for Photonic systems, Ravensburg-Weingarten University, Doggenriedstraße, 88250 Weingarten, Germany*

Abstract

In today's AI-driven era, computer vision, including autonomous driving, robotics, and healthcare, is prevalent. However, acquiring ample data while managing resources and privacy constraints is challenging. This article proposes a solution: synthetic data generation. We use CAD software to craft intricate 3D models, process them in Blender, and evaluate quality using metrics like Structural Similarity and PSNR (Peak Signal to Noise Ratio). Synthetic data achieves up to 90% similarity with real data and an average PSNR of 21dB. Our method offers a streamlined, dependable approach for enhancing computer vision, especially in object detection, addressing data acquisition challenges.

Keywords: Image Processing; Realistic Data Generation; Structural Similarity Index and Peak Signal to Noise Ratio

[*]Corresponding author

Email address: dhanush.sagi@gmail.com (Dhanush Gangadharaiah)

1. Introduction

Data is paramount in today's world, permeating every facet of daily life. Extensive research centers around artificial intelligence technologies, where data's pivotal role cannot be overstated. Deep learning and machine learning models, devoid of data, lack the essential essence. Industries heavily reliant on data encompass finance, healthcare, government, autonomous systems, and robotics. Data types are categorized as Image, Text, Sound, Signal, Physical, Biological, Anomaly, and Multivariate, each serving unique applications. Our primary focus is on Image data, vital in machine learning for computer vision applications, including object detection, facial recognition, and multi-label classification [1].

Numerous datasets are accessible online to facilitate various tasks like object detection. However, it's crucial to acknowledge that not all available datasets are suitable for specific purposes, particularly when the intended application differs significantly from the dataset's inherent characteristics. Therefore, it becomes imperative when the ability to gather data gets challenging because the data of our interest and also for the application required was never recorded or does not exist. For instance, attempting to utilize datasets primarily designed for categorizing flower species to train a deep learning model for predicting car attributes is impractical due to the fundamental dissimilarity between flowers and cars. In such circumstances, it becomes imperative to initiate the data collection process from the ground up, tailored precisely to the target application. In the case mentioned, this involves the meticulous acquisition of car-related data, ensuring that the necessary resources are in place to gather, annotate, and curate the data effectively. This meticulous approach is essential to align the dataset with the specific requirements and nuances of the intended application, thus optimizing the model's performance.

Blender graphics software, due to its versatile capabilities is employed as a pivotal instrument to generate the data automatically without costing time management of resources. This approach is adapted when there is no data available nor it has been captured for the desired application. The synthetic data which reflects realism with the actual data is used for training of machine learning models along with libraries such as fastai [2]. Also, there are applications such as [3,4] Finite Element Method, where simulations with the help of Blender is incorporated.

1.1 Purpose and Focus

It is readily apparent that data is a fundamental component, particularly within the domain of machine learning models, where the outcome's reliability is intrinsically tied to the quantity and quality of the data utilized, alongside the precision of parameter optimization which in turn, results in the attainment of precise outcomes in the context of detection and recognition tasks. Nonetheless, certain applications may find themselves in the predicament of inadequate or entirely absent relevant data, thereby necessitating the initiation of data generation processes from scratch. This endeavor is resource-intensive and time-consuming. The disadvantage of alternative methods, which Blender 3D aims to address, could be the deficiency or nonexistence of essential data. Traditional methods may struggle to provide sufficient and diverse datasets for training models or conducting experiments. This limitation can hinder the progress of data-driven initiatives, impacting the quality and robustness of the results. Blender 3D, in this context, emerges as a valuable solution by effectively resolving the issues associated with data scarcity or absence. It offers a versatile platform for generating synthetic data, filling the gaps left by traditional methods.

The scope of datasets generated through Blender significantly impacts their realism. While the conventional use of 3D cameras for data capture ensures reliability, it also entails substantial resource costs, especially in uncontrollable scenarios. Hence, in this section, we conducted a comparative analysis between data obtained from a 3D camera, such as Azure Kinect, and data generated through Blender. The Kinect sensor, as a development kit [5], serves as a computer add-on, establishing a connection with the Microsoft Azure cloud. It harnesses machine intelligence-based sensors to facilitate computer vision and speech models. The kit encompasses a 12-megapixel RGB lens, a 1-megapixel depth camera for body tracking, a 360-degree seven-microphone array, and a position sensor. The depth sensor, initially introduced in 2018 ISSCC, forms the core component of this sensor, which is widely applied in vision applications and exhibits notable performance [6,7]. To evaluate the quality of generated data, we conduct a comparative analysis between rendered images from Blender and real data captured by the Kinect Azure. Our data acquisition involves recording wrench frames at a 1080p resolution, aligning the intrinsic of Blender's virtual camera with the real-world setup for rendering the model.

In this experimental framework, synthetic data generation is facilitated through the utilization of a Computer-Aided Design (CAD) model, a versatile tool adept at creating, modifying, and visually representing intricate engineering components, commonly referred to as 3D models. The transformation and processing of these models are achieved through the utilization of 3D graphics software, with Blender serving as the focal point within this research due to its open-source nature and the presence of a pre-existing scripting feature that enhances the versatility of Python programming.

The central **hypothesis** underpinning this endeavor revolves around the objective of producing synthetic data that closely mirrors real-world counterparts in terms of realism. To attain this goal, it is postulated that the virtual camera employed within Blender must meticulously focus on the object of interest from a multitude of perspectives, encompassing various lighting conditions for the rendering process. Furthermore, critical 3D object model properties, including but not limited to texture and color, must be addressed. This approach is designed to ensure that the generated synthetic data is indistinguishable from authentic data, contributing to the reliability and efficacy of subsequent machine learning or computer vision applications.

2. Materials and Methods

This section provides a vivid depiction of the conducted experiments, encompassing a range of scenarios and detailing the primary objects employed as integral components of the research investigation.

2.1 Literature

Synthetic data generation encompasses various approaches, and one prominent method involves the utilization of neural networks. Within this domain, several techniques have emerged, such as Variational Auto-Encoder (VAE) which was trained on MNIST dataset produces a continuous and structured latent space which is useful for image construction [8], Generative Adversarial Networks (GAN) with a U-Net based architecture for Biomedical Image Segmentation and conditional adversarial networks for image to image translation [9,10] and Stable Diffusion [11] which is a powerful, open source text to image generation model. It includes XLA (Accelerated Linear Algebra) and mixed precision support that results in state-of-the-art generation speed. Each of these techniques carries distinct significance in the pursuit of producing synthetic data that closely emulates authentic data, thus contributing to the realism and quality of the generated datasets.

The concept of generating data through the utilization of CAD models has previously been explored, as exemplified in existing literature, such as [12] which presents an end-to-end framework. This framework holistically simulates the intricate functioning of devices, concurrently generating data through the incorporation of 3D models. It comprehensively models essential parameters, including sensor-induced noise, material reflectance, and surface geometry. Furthermore, it extends its scope to assess the influence of these data generation processes on the training of neural networks for diverse recognition tasks. In [13], a novel approach to image generation, reliant upon CAD models, is introduced, and it is facilitated through the utilization of Computer Graphics Software (CGS). This software serves as the essential tool for rendering 3D models representing the objects targeted for detection. Among the available CGS options, such as Blender and Unity, Blender emerged as the more effective choice during the course of this study. It is worth noting that, besides surface texture, the image generation process can be entirely automated, encompassing the critical aspect of labeling.

2.2 Computer Aided Design (CAD)

Diverse formats are available for representing 3D models, including STL, OBJ, FBX, and DAE, each tailored to specific applications and industry requirements. Within the context of this experimental study, the utilization of OBJ format models was chosen due to their prevalence in various industrial sectors and their consistent compatibility with a broad spectrum of software environments. This format ensures seamless integration and dependable interpretation by the software systems employed. In numerous domains, including automotive, aerospace, industrial engineering, and dental prosthetics, Computer-Aided Design (CAD) stands as a pivotal and irreplaceable industrial art. Consequently, CAD tools find extensive usage among engineers and designers, especially in the context of component design and development [14]. To exemplify and validate our proposed methodology, we opted to employ 3D models of open-ended and combined wrenches. These models can be readily generated using an array of CAD software applications, such as Autodesk, Solid works, Creo, Catia, and others, underscoring the universality and adaptability of the approach across diverse research and engineering domains.

2.3 Method

The proposed end-to-end methodology predominantly revolves around the incorporation of CAD models within the Blender environment. This process entails detailed modifications and adjustments to the CAD model to align it with the specified lighting conditions. Furthermore, within the Blender viewport, the pre-existing virtual camera is directed to focus on the designated object of interest, in this instance, the open-end and combined wrench. The camera's pivotal role lies in rendering the scenes within the Blender interface. When a camera is active, it triggers the visibility of object and editing setup panels. The selection of the camera lens plays a critical role in determining the portrayal of the 3D scenes as they are transposed into 2D images, influencing the visual outcome of the rendered scenes.

Blender: employs an intrinsic Python interpreter [15], which is active upon its launch and proves essential for both its internal utilities and user interface scripts. This interpreter grants access to Blender's data, classes, and functions through a suite of embedded Python modules. To facilitate the effective execution of scripts that interact with Blender data, it is imperative to import all requisite modules. Transformations are indispensable for data augmentation, including random rotation, scaling, and dimension adjustments. In this study, we specifically utilized rotation and dynamic color modification during execution. Euler rotation mode and the Principled BSDF [16] play key roles in introducing random color variations, consolidating multiple layers into a single node.

The color transformation incorporates a saturation value within the range of (0, 1), wherein values less than 0 yield grayscale images, and values exceeding 1 intensify saturation [17]. Subsequently, the rendering process transpires, converting the 3D scene into a 2D plane. Blender provides rendering engines such as Cycles and EEVEE, each bearing unique significance in the rendering phase. Furthermore, the introduction of noise during data generation assumes a crucial role, serving to assess the deep learning model's performance. Noise not only challenges the model's feature recognition but also enhances the dataset by introducing distinctive characteristics, fostering more robust learning.

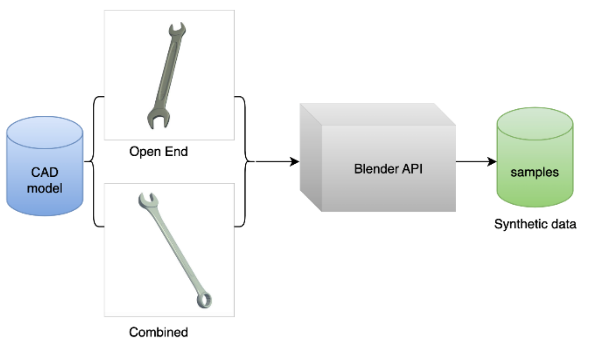

Figure 1: Pipeline for generating Synthetic data

Structural Similarity Index: We employ the Structural Similarity Index (SSIM) to assess the resemblance between the reference image from Kinect and the rendered image from Blender. SSIM quantifies disparities between sample and comparison images by evaluating values for relevant pixels. Our focus centers on human visual perception, which adeptly discerns structural information in scenes to distinguish image details. SSIM dissects three essential components: Luminance, Contrast, and Structure, resulting in a comparison score within the range of (-1, 1). In this scale, a score of -1 signifies dissimilarity between images, while a score of +1 indicates similarity. Therefore, a metric that aligns with this pattern proves more effective for tasks involving comparisons between real and generated images [18,19].

$$SSIM(x,y) = \frac{(2\,\mu_x\,\mu_y+C_1)(2\,\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \qquad (1)$$

where $\boldsymbol{\mu_x}$, $\boldsymbol{\mu_y}$ is the pixel sample mean of x and y. $\boldsymbol{\sigma_x^2}$, $\boldsymbol{\sigma_y^2}$ is the variance of x and y. $\boldsymbol{\sigma_{xy}}$ covariance of x and y. $\boldsymbol{C_1 = (k_1\,L)^2}$, $\boldsymbol{C_2 = (k_2\,L)^2}$ are two variables to stabilise the division with weak denominator. L is the dynamic range of pixel values and $\boldsymbol{k_1 = 0.01}$, $\boldsymbol{k_2 = 0.03}$ a small constant by default.

Peak Signal to Noise Ratio: PSNR, measures the relationship between a signal's maximum power [20] and the power of degrading noise, a factor that can compromise the precision of signal representation. Given the extensive dynamic range of many signals, PSNR is typically represented in a nonlinear manner using the decibel system. It finds frequent application in evaluating the quality of reconstruction, particularly in scenarios involving the compression of images and videos. Significantly, PSNR offers valuable insights into how humans perceive the quality of reconstructed data during the assessment of compression techniques.

$$PSNR = 20\,log_{10}\frac{(MAX_f)}{\sqrt{MSE}} \qquad (2)$$

where MAX_f represents the highest signal level within the original image, and MSE [21] denotes the Mean Squared Error, which quantifies the dissimilarity in pixel values between the real image and the rendered image. In essence, it measures the disparity between the pixel values of these two images. A higher PSNR corresponds to enhanced image quality post-reconstruction. It's worth noting that PSNR becomes undefined when attempting to compute the MSE between nearly identical images, as the MSE value approaches zero, leading to a division by zero.

3. Results

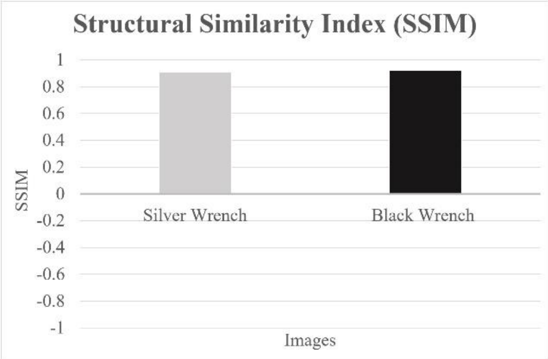

Figure 2: Structural Similarity Index

In [22], a comparative analysis was undertaken to assess the metrics concerning image quality reconstruction. Our examination primarily hinges on the evaluation of Structural Similarity (SSIM) and Peak Signal-to-Noise Ratio (PSNR) values. PSNR is characterized by a range extending to infinity, with higher PSNR values indicative of superior image quality. To guide our analysis, we established a threshold value of 20 dB to delineate the measurement range. In evaluating the reference and rendered images, we observed a PSNR of 22.01 dB for the Silver wrench and 20.82 dB for the Black wrench. Similarly, for SSIM, the resulting images exhibited a 91% similarity to the original image for the Silver wrench and 92% for the Black wrench, respectively. The examination of metrics, encompassing Structural Similarity and Peak Signal-to-Noise Ratio, in the course of the experiments is executed as elaborated here.

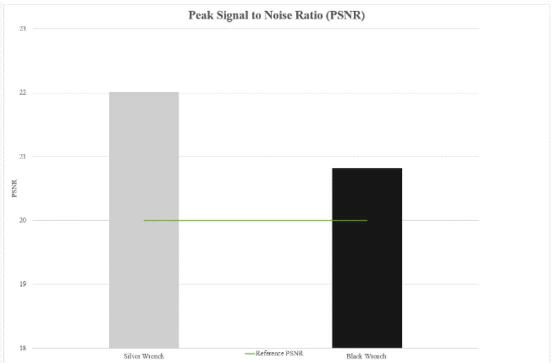

Figure 3: Peak Signal to Noise ratio

Prior to processing synthetic data in Blender, the utilization of a Kinect sensor was integral for capturing authentic wrench images depicting various combinations, as illustrated in the figure below. This approach enabled the acquisition of genuine visual representations that served as the basis for subsequent data manipulation within Blender. The processed data derived from Blender exhibited notable promise, closely resembling the characteristics of real-world data. This resemblance was crucial in ensuring the authenticity and accuracy of the synthetic data generated through the processing pipeline.

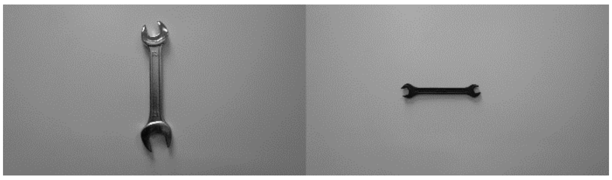

Figure 4: Real data prior to undergoing processing in Blender

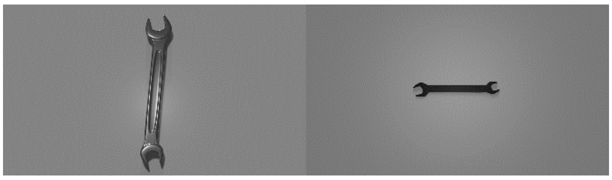

Figure 5: Synthetic data generated using Blender

Comparative analysis between the real and synthetic datasets was conducted using metrics such as Structural Similarity Index and Peak Signal-to-Noise Ratio. These evaluation methods provided valuable insights by quantifying the similarities and differences between the authentic and synthetic data. The widespread application of SSIM and PSNR in this context has solidified their significance as essential tools for assessing data fidelity and reliability across various industries and research domains.

4. Discussion

The primary objective of this experiment is to address the challenges associated with procuring authentic data, often hindered by factors that impose significant time and financial costs. In the context of employing computer graphics software such as Blender, it is crucial to meticulously consider physical attributes, including aspects like lighting conditions and pixel irregularities, which are integral to generating high-fidelity data. These physical property issues become particularly pronounced with the increasing complexity of 3D data, and Blender offers a straightforward solution for managing them.

By diligently accounting for all environmental factors and employing a configuration akin to that of a real-world camera, the resulting images attain a heightened level of realism. This meticulous approach enables us to create images that closely emulate real-world counterparts. Moreover, it is imperative to recognize the presence of biases in artificial data, as they have the potential to introduce errors, particularly in scenarios involving the training of neural networks. This consideration becomes especially critical in fields where privacy is of utmost importance, such as the medical industry, where data integrity must be maintained without compromise. The fundamental drawback of PSNR measure is that it only considers numerical comparisons and ignores all levels of genetic aspects of the human visual system, including the Structural Similarity Index (SSIM).

Data	Advantages	Disadvantages

Real	-Increased accessibility -Information up to date -Actual situations	-Data quantity -Difficult to record frames -Perfect annotation -Privacy
Synthetic	-Data size -Perfectly annotated -Full user controlled -Multispectral -Privacy	-Lightning conditions -Input data is vital -Anomalies can be missing

Table 1: Comparison between Real data and Synthetic data

Utilizing this method for data generation is essential due to Blender's adaptability and user-driven nature. It allows for the manipulation of CAD models across various formats. Its cost-effectiveness renders it suitable for numerous applications and experiments, requiring no substantial capital investment. The incorporation of a sophisticated rendering engine produces high-fidelity images comparable to real data. Blender's complete customizability empowers users to configure diverse scenarios and parameters aligning with the research's specific direction. There are other methods such as GAN and VAE that predominantly uses Neural Network to generate synthetic data. However, there are limitations and also beneficial facts with these two approaches.

In certain scenarios involving the utilization of GANs/ VAEs to generate synthetic data, specific considerations should be considered. GANs may encounter issues such as collapsing when confronted with limited data variety, thereby failing to encompass the entirety of the data distribution. Training these models can pose challenges due to their intricate and volatile characteristics, which heavily rely on meticulous hyperparameter tuning. Moreover, interpreting the acquired representations of the data becomes challenging due to the inherent black box nature of these models.

Method	Positives	Pitfalls
Our Method (Blender)	-Versatility -User Control -Cost-Effective -Realistic render -Customization	-Complexity -Resource intensive
GAN/ VAE	-Learned representatives -Data Augmentation -Unsupervised learning process	-Mode collapse -Challenges during training -Quality control -Interpretability

Table 2: Strengths and weakness between proposed method and other methods

5. Conclusions

It is evident that conventional data collection methods yield higher-quality photos when deployed in real-world scenarios. However, the manual annotation process for each image, tailored to the end user's specific object of interest, can be resource-intensive and time-consuming.

- In such cases, 3D graphics software like Blender emerges as a pragmatic solution to mitigate these challenges, enabling the generation of photorealistic samples.
- These synthetic samples find valuable application in training convolutional neural networks, which heavily rely on high-quality data.
- A notable advantage of employing Blender lies in its accessibility, as it allows for the capture of not only RGB frames but also depth properties of the model, offering a comprehensive dataset.

Creating synthetic datasets using Blender presents a superior approach due to its exceptional capacity for precise user control and extensive customization. This platform effectively addresses the aspects of realism and high-quality image generation through its advanced rendering engines. The ability to annotate ground truth and craft domain-specific data stands out as pivotal advantages when considering Blender for dataset generation. One of the notable strengths of Blender lies in its capability to offer diversified datasets, consequently mitigating the risk of overfitting in machine learning models. Unlike the complexity associated with Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), Blender's efficiency in managing both time and cost becomes evident, particularly when the need arises to gather vast volumes of data. Furthermore, utilizing Blender eliminates concerns surrounding data privacy, making generated data readily available for research purposes. Its versatility and accessibility enhance the suitability of Blender for various scientific and research endeavors. By leveraging Blender's capabilities, researchers and practitioners can benefit from high-quality, diverse datasets while circumventing privacy issues, ultimately contributing to advancements in machine learning and data-driven research.

Wrench Data	SSIM	PSNR
Silver	0.91	22.01
Black	0.92	20.82

Table 3: Evaluation score between Silver and Black wrench

6. Future Scope

The successful demonstration of generating synthetic datasets whilst using Blender tool produced the realistic data throughout the experiment. However, there are potential further steps to be considered to mimic the data as real as possible which is application specific. Some of the factors that helps in tuning this process such as, examine the critical parameters such lightning conditions and input data quality within the environment. This involves conducting experiments to understand the possible variations that impact the data's fidelity and applicability.

Refinement in rendering process is also critical to reduce data omissions and ensure completeness. It involves changing rendering parameters, improve algorithm and also other rendering engines. There is a scope to explore about other data augmentation techniques which plays a pivotal role in enhancing the generalization capability of neural network trained and validated on the data created. Investigation of Transfer learning strategy where the characteristics learned from synthetic data to real world scenario. This involves domain adaptation techniques to bridge the gap between real and fake data. The assessment metrics and evaluation methodologies can be substantially improved to quantify the quality and realism of synthetic data. The application of synthetic dataset is beneficial in the field of Autonomous vehicles such that simulating diverse driving scenarios, Medical Imaging for diagnosis and Robotics where we can create simulated environment for training and testing.

Author Contributions

Conceptualization and methodology, D.G., J.E.; writing—original draft, D.G..; validation and valuation of the results, all authors; writing—review and editing, D.G., J.E., and N.N. The whole project was conducted under the supervision of J.E. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by the Federal Ministry for Economic Affairs and Energy (BMWI).

Informed Consent statement

Not applicable

Data availability statement

Not applicable

Conflicts of Interest

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

[1] Dataset list for machine learning research, https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research. (online: 10th September 2023)

[2] Sieczka R, Pa´nczyk M. Blender as a tool for generating synthetic data. Journal of Computer Sciences Institute. 2020 Sep 30; 16:227-32.

[3] Zdziebko P, Holak K. Synthetic Image Generation Using the Finite Element Method and Blender Graphics Program for Modeling of Vision-Based Measurement Systems. Sensors. 2021 Sep 9;21(18):6046.

[4] Boikov A, Payor V, Savelev R, Kolesnikov A. Synthetic data generation for steel defect detection and classification using deep learning. Symmetry. 2021 Jun 29;13(7):1176.

[5] Azure Kinect Sensor (online: 10th September 2023), https://en.wikipedia.org/wiki/Azure_Kinect

[6] Tölgyessy M, Dekan M, Chovanec L', Hubinský P. Evaluation of the azure Kinect and its comparison to Kinect V1 and Kinect V2. Sensors. 2021 Jan 8;21(2):413.

[7] Albert JA, Owolabi V, Gebel A, Brahms CM, Granacher U, Arnrich B. Evaluation of the pose tracking performance of the azure kinect and kinect v2 for gait analysis in comparison with a gold standard: A pilot study. Sensors. 2020 Sep 8;20(18):5104.

[8] Kingma DP, Welling M. An introduction to variational autoencoders. Foundations and Trends® in Machine Learning. 2019 Nov 27;12(4):307-92.

[9] Isola P, Zhu JY, Zhou T, Efros AA. Image-to-image translation with conditional adversarial networks. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 1125-1134).

[10] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. InMedical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 2015 (pp. 234-241). Springer International Publishing.

[11] Image generation using Stable Diffusion in KerasCV, https://www.tensorflow.org/tutorials/generative/generate_images_with_stable_diffusion (online: 27th September 2023)

[12] B. Planche et al., "DepthSynth: Real-Time Realistic Synthetic Data Generation from CAD Models for 2.5D Recognition," 2017 International Conference on 3D Vision (3DV), Qingdao, China, 2017, pp. 1-10, doi: 10.1109/3DV.2017.00011.

[13] Hodapp, J., Schiemann, M., Arcidiacono, C. S., Reichenbach, M., & Bilous, V. (2020, January). Advances in Automated Generation of Convolutional Neural Networks from Synthetic Data in Industrial Environments. In HICSS (pp. 1-7).

[14] Computer aided design, (online: 10th September 2023) https://en.wikipedia.org/wiki/Computer-aided_design.

[15] Blender Python API, (online: 12th September 2023 https://docs.blender.org/api/current/info_overview.html

[16] Principled BSDF, (online: 22nd September 2023), https://docs.blender.org/manual/en/latest/render/shader_nodes/shader/principled.html

[17] Hue Saturation Value, (online: 22nd September 2023), https://docs.blender.org/manual/en/latest/compositing/types/color/hue_saturation.html

[18] All about Structural Similarity Index (SSIM), (online: 15th September 2023), https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e

[19] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing. 2004 Apr 13;13(4):600-12.

[20] Peak Signal to Noise ratio, (online: 24th September 2023), https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio

[21] Peak Signal-to-Noise Ratio as an Image Quality Metric, https://www.ni.com/de-de/innovations/white-papers/11/peak-signal-to-noise-ratio-as-an-image-quality-metric.html. (online: 24th September 2023)

[22] SARA, Umme; AKTER, Morium; UDDIN, Mohammad Shorif. Image quality assessment through FSIM, SSIM, MSE and PSNR—a comparative study. Journal of Computer and Communications, 2019, 7. Jg., Nr. 3, S. 8-18.