Automation in Interior Space Planning

Subjects: Computer Science, Artificial Intelligence

Contributor: Hanan Tanasra , Tamar Rott Shaham , Tomer Michaeli , Guy Austern , Shany Barath

In interior space planning, the furnishing stage usually entails manual iterative processes, including meeting design objectives, incorporating professional input, and optimizing design performance. Machine learning has the potential to automate and improve interior design processes while maintaining creativity and quality.

interior space planning CGANs image-to-image translation

1. Introduction

To provide a clear representation of a given design plan, comprehensive drawings of the walls, doors, windows, and furniture are required. Designers tend to manually depict their furnishing work using Computer-Aided Design (CAD) software based on their knowledge and experience and in line with their clients' requests [1]. Optimally integrating furniture items within floor plans plays a pivotal role in various stages of the design process. Not only do furniture arrangements serve to unveil and communicate the quality of a given space, but they also accentuate the space's functionality and performance. As such, when incorporating furniture items into floor plans, interior designers meticulously consider a wide array of factors in order to create a space that is both aesthetically pleasing and functional.

To assist with this process, architectural automation has been rapidly improving via the utilization of techniques such as *parametric design* and *generative design* within CAD software. The recent emergence of Machine-Learning (ML) and Artificial Intelligence (AI) techniques has introduced processes that impact a wide range of fields [2]. As these advances become more widely incorporated in architectural designs, it is critical to establish effective assessment methods for measuring their impact on design quality. In turn, it is essential to investigate how these technologies might be utilized to improve design processes while examining their possible challenges and limitations.

2. Automation in Interior Space Planning

2.1. Machine Learning in Architectural Design Processes

The rapidly expanding field of ML is frequently employed in applications such as online browsing, information filtering, and credit card fraud detection. One key strength of ML models is their ability to learn complex rules without being explicitly programmed, thereby achieving human-like performance [2]. Most modern ML models are based on artificial neural networks, which require large training datasets in order to achieve good results [3].

In architecture, ML techniques are used for a range of tasks. For example, in legacy 2D floor plan analyses, vision-based detection models are used to identify architectural objects, such as windows, doors, and furniture in the design [4]. In similar drawings, segmentation models are used for identifying room boundaries and tracing walls [5][6]. Optical Character Recognition (OCR) models help identify function types by reading texts on scanned floor plan images [7]. Owners and real estate agents could use such methods to reduce the need to create 3D models, which require greater time and effort. Additionally, semantic and graph-based analyses of Building Information Models (BIM) offer the ability to recognize a room type based on the objects within its boundaries and on the space features [8][9].

Chaillou [10] demonstrates the potential of CGAN-type ML models for generating space and furniture layouts of the apartment's footprint. DPrix et al. [11] use similar models for generating new exterior designs based on an existing archive of images in their architecture firm. State-of-the-art advances in image synthesis models, such as Midjourney (Midjourney, n.d.) [12], Stable Diffusion [13], and DALL-E2 [14], have also been used to generate architectural images.

The novel method presented in this research study employs the CGAN-based generation method, as proposed by [10], who provided the foundation of this line of research and demonstrated its feasibility. The researchers also presented several ways in which CGANs can be used in interior design, such as room allocation, function assignment, and furnishing. This current study focuses on the interior furnishing of a single functional room, i.e., the bathroom, and provides a dataset and comprehensive workflow that outlines the necessary steps for achieving the research aims. Finally, a set of developed evaluation criteria is presented to enable the measuring of the performance of the suggested method.

2.2. Interior Residential Design Furnishing

Throughout history, scientists and researchers have studied human body movements, with behaviors and patterns being collected as a basis for designing spaces that enhance levels of human comfort [15]. Indeed, object scales and sizes are often defined when positioned next to a human body [16]. Although definitions of terms such as “comfortable” or “qualified” spaces may be subjective, unified guidelines and standards are usually followed when designing spaces for humans.

To reduce the manual work entailed in creating and designing interior spaces, studies suggest employing interactive tools for positioning objects of furniture while responding to the users’ real-time selection of furniture pieces. Doing so could provide users with access to an editable library of furniture pieces [17]. Geman and Geman [18] address the concept of *simulated annealing*, which can be used to create indoor scenes for optimizing ergonomic factors. In a study conducted by [19], the researchers utilize the hierarchy between furniture objects to enable the generation of indoor furniture arrangements. Generating furniture layouts has also been achieved via geometrical analyses of objects, using algorithmic hyper-relations between and within groups of items to assign the positioning and orientation of each group. For example, the relationship between a group of objects, comprising a sofa, table, and television, could change based on the layout of the room [20]. Alternatively, Kim and Lee [21] generated furniture layouts based on a design-style perspective, whereby different types of rooms were created according to the desired style (casual, modern, classic, or natural). Their model’s dataset comprised perspective images of interior spaces in which each architectural style had distinct features, such as materials, forms, patterns, and colors.

2.3. Interior Residential Design Applications

PlanFinder, for instance, is a software plugin for CAD and BIM software that enables designers to furnish spaces with greater ease and speed simply by entering a door point and clicking on the *furnish* button [22]. Finch 3D, an additional online service, leverages AI and graph-based algorithms to optimize building designs and even provides performance feedback, error detection, and optimal solution identification [23]. Finally, the web-based Rayon Design service offers an intuitive online interface that allows users to make changes to their floor plans, delete walls, and add furniture from a suggested set of furniture objects. Users must, however, have a drawn floor plan prior to uploading the design [24].

2.4. Conditional Generative Adversarial Neural Networks Models

Deep neural networks and other recent ML advancements have significantly expanded the scope and accuracy of generative modeling across domains to reflect the diverse and complex nature of real-world data. These highly varied models exhibit a wide range of capabilities and characteristics that depend on the applied algorithms and parameters. These include Generative Adversarial Networks (GANs), which offer a range of solutions based on their ability to learn to generate novel data from a given set of examples [25]. Its architecture consists of two neural networks that are trained in a competing manner: the *generator* strives to create fake yet realistic data samples, while the *discriminator* attempts to distinguish between the real and fake samples. Moreover, the *generator* continuously improves its ability to generate increasingly realistic data through this adversarial process, while the *discriminator* continuously improves its ability to detect real versus fake data [26]. GANs have been used in a wide range of applications, including image generation [27], text-to-image synthesis [28], and video synthesis

[29]. Conditional GANs (CGANs) improve GAN models by consuming a given condition as input, such as the category or class label desired for generation. The CGAN generator is trained to generate images according to the condition injected into it. The CGAN discriminator is trained not only to distinguish between real and fake data but also to consider added conditioning information. CGANs have been found to be useful in image-to-image translation tasks, where they translate an input image from one domain into a corresponding output image from a different domain; they do so by maintaining some elements of the original image as the basis of the newly generated one. This process strives to learn the mapping between input and output pictures so that the generated output image resembles the desired target image as precisely as possible [30].

The CGAN family includes many different models. For example, based on label mappings, the pix2pix model learns to map between pairs of input–output images and then produces a single synthesized image. This model involves training a generator network based on the input image using two losses. The first is a regression loss that helps the generator produce an output image that is similar to a paired ground-truth image. The second is a learned discriminator loss that encourages the generator to produce realistic images [31]. This model has been tested on various architectural datasets, such as labels-to-street and labels-to-facades scenes [30]. An additional CGAN model is SPADE, which is based on a more advanced mechanism for turning a semantic label map into a photorealistic image. The model converts the labels and combines them with the style to generate the desired realistic content [32]. Finally, the BicycleGAN model offers a multimodal image-to-image translation, generating multiple potential images for the same input. This model proposes a latent mapping vector between the generator and the discriminator, encouraging a bijection between the output and the latent space, thereby leading to less ambiguous or diverse results [31].

2.5. CGAN Models in Architecture

Huang and Zheng [33] utilized a variant of pix2pix to segment room boundaries and detect doors and windows. In an additional study, Yang et al. [34] applied certain modifications to the standard CGAN model to create functional floor divisions. Next, they deployed a fully connected, three-tier neural network for each type of functional area to place each piece of furniture within each space. Yet, the researchers do not measure or quantify the quality of such placings. Chaillou [10] introduced the capabilities of CGAN models in generating interior design, floor plan layout, and urban footprint. His groundbreaking work demonstrates the potential of ML to generate architectural designs by filling in “empty” black-and-white architectural drawings. Chaillou [10] proposes using this framework to address the style and organization of different scale floor plans. Yet, the researcher does not provide comprehensive descriptions or details of the employed training processes, GAN models, or the employed settings. In this study, CGAN models such as pix2pix, SPADE, and BicycleGAN were trained to generate furnishing layouts in empty floor plans using the room data that were created in this study. This work presents the training process of the models together with the generated results and proposes various metrics for quantifying the quality of the generated results.

2.6. Available Architectural Datasets

Most Internet databases provide photographed perspective images of rooms that can be gathered by anyone, anywhere, via their mobile devices. Examples of such datasets include the 3D-FRONT Benchmark of synthetic indoor scenes and a large number of rooms [35]. Kaggle, an online community of data scientists and ML professionals, exposes images of homes that are taken on mobile telephones and that appear on the Renthop website, which helps people find home rentals (RentHop|Kaggle) [36][37]. The walls, doors, windows, and room boundaries in the dataset are presented via three online sources: (1) Rent3D (R3D) [36], which displays 222 images of round-shaped layouts, as well as straight ones with nonuniform wall thickness; (2) Raster-to-Vector (R2V), which includes 815 images of rectangular shapes with uniform wall thickness [6][38]; (3) CVC-FP, which includes 500 images in two versions, i.e., original floor plans in black-and-white and color-mapped images [39][40]. The international Ikea furniture chain has an online furniture dataset that uses furniture objects from their product lists and 298 room scene photos to convey the context in which these objects can be used or placed [41]. Each of these available public datasets was created for a specific research topic and does not meet the requirements of this research since this case study requires a flexible dataset, one that includes top-view room layouts that can be modified to include or exclude furnishings as a means for training the models to achieve the defined aims. A customized database, therefore, was created, as well as a tool that can generate various types of architectural datasets.

2.7. Machine Learning Dataset Formats and Available Annotation Tools

Comma-separated values (CSV) are a popular format for conducting text-data loading; a text can contain numbers or strings organized in a tabular format [42]. JavaScript Object Notation (JSON) is another well-known format that is easy to read for humans and can easily be parsed via machines. This format is interchangeable between several computer languages and may hold a variety of data [43]. The Common Objects in Context (COCO) format is a specific JavaScript Object Notation (JSON) structure format. COCO JSON files are commonly used for object detection and segmentation purposes, with files being able to store entire data, and with translators into numerous formats [44]. Some online image annotation platforms provide segmentation polygons and bounding box detection, with a reference label to imported images. One example of such a platform is the CVAT website [45]. Image labeling necessitates a significant amount of time and resources since it is performed manually via annotators. As such, online dataset annotation platforms may hinder the efficiency with which custom data is created. Designers do not typically use these online platforms in their work, and, to the best of the authors' knowledge, no online tagging tool exists that is compatible with architectural CAD programs. Thus, those who best understand and have the most to gain from ML design-automation methods, i.e., designers, are not included in the development processes.

2.8. Evaluation Metrics for CGAN Models

2.8.1. GANs Evaluation Metrics

When dealing with image-to-image models, the following two evaluation metrics are commonly used to measure similarities and differences between two sets of images and to evaluate the quality of the generated images in CGANs models: Fréchet Inception Distance (FID) and Kernel Inception Distance (KID) [46].

FID is a metric for measuring the distance between two multivariate Gaussian distributions, which are calculated from the real and generated image features that are extracted using a pre-trained Inception-v3 network. The FID score is calculated by first computing the mean and covariance of the feature representations for real and generated images discretely. Next, the distance between the two sets of images is calculated based on the statistics of their feature representations in the pre-trained neural network. Lower FID scores mean smaller distances, indicating higher similarities between the real and generated image distributions. However, FID requires a large dataset and can be overly sensitive to the choice of the pre-trained network used to extract the image features. As such, it may fail to capture certain aspects of image quality, such as diversity or coherence [47][48]. KID is also based on feature representations of the images extracted using a pre-trained Inception-v3 network, a statistical method that is used to measure the similarity between two sets of data, such as real and generated images. Both FID and KID measure distances between the distributions of deep features of real and fake images. They differ in how they measure these differences. However, KID can provide accurate results with less data since its predicted value is independent of the sample size [46][49].

2.8.2. Object Detection Metrics

Precision, recall, and intersection over union (IOU) metrics are commonly used for evaluating computer vision tasks, such as object detection, tracking, and segmentation [50][51]. These metrics compare the generated image against its respective ground truth. *Precision* is the metric that measures the proportion of true positive predictions out of the total number of predictions and can be seen as a measure of quality. *Recall*, on the other hand, is the metric that measures the proportion of true positive results out of the total number of actual positive instances and can be seen as a measure of quantity [50]. Finally, the *IOU* calculates the intersection of the predicted and ground-truth bounding boxes divided by the union of the two boxes.

CGAN models are expected to generate new and optimally synthesized images while considering the overall features of the dataset. Although these three detection metrics cannot be used to compare the generated images to overall ground-truth ones, it is important to find an alternative that will enable the evaluation of each generated image against its respective ground truth to evaluate similarities and diversities of the output.

References

1. Sönmez, N.O. A review of the use of examples for automating architectural design tasks. CAD Comput.-Aided Des. 2018, 96, 13–30.
2. Reddy Edla, D.; Lingras, P.; Venkatanareshbabu, K. Advances in Machine Learning and Data Science: Recent Achievements and Research Directives; Springer: Singapore, 2018; Volume 705.
3. Goodfellow, I.; Yoshua, B.; Aaron, C. Deep-learning. In Deep; MIT Press: Cambridge, MA, USA, 2016; Volume 12, Available online: http://www.deeplearningbook.org (accessed on 28 May 2023).
4. Ziran, Z.; Marinai, S. Object Detection in Floor Plan Images. 2021. Available online: https://link.springer.com/book/10.1007/978-3-319-99978-4 (accessed on 28 May 2023).
5. Dodge, S.; Xu, J.; Stenger, B. Parsing floor plan images. In Proceedings of the 15th IAPR International Conference on Machine Vision Applications, MVA 2017, Nagoya, Japan, 8–12 May 2017; pp. 358–361.
6. Zeng, Z.; Li, X.; Yu, Y.K.; Fu, C.W. Deep floor plan recognition using a multi-task network with room-boundary-guided attention. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9095–9103.
7. Ahmed, S.; Liwicki, M.; Weber, M.; Dengel, A. Automatic room detection and room labeling from architectural floor plans. In Proceedings of the 10th IAPR International Workshop on Document Analysis Systems, Gold Coast, QLD, Australia, 27–29 March 2012; pp. 339–343.
8. Bloch, T.; Sacks, R. Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models. Autom. Constr. 2018, 91, 256–272.
9. Buruzs, A.; Šipetić, M.; Blank-Landeshammer, B.; Zucker, G. IFC BIM Model Enrichment with Space Function Information Using Graph Neural Networks. Energies 2022, 15, 2937.
10. Chaillou, S. Space Layouts & GANs. Artificial Intelligence and Architecture. 2020. Available online: https://www.academia.edu/42289131/Space_Layouts_and_GANs_GAN_enabled_Floor_Plan_Generation (accessed on 28 May 2023).
11. DPrix, W.; Schmidbaur, K.; Bolojan, D.; Baseta, E. The Legacy Sketch Machine: From Artificial to Architectural Intelligence. Archit. Des. 2022, 94, 14–21.
12. Midjourney. (n.d.). Available online: https://www.midjourney.com/home/?callbackUrl=%2Fapp%2F (accessed on 18 June 2023).
13. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10674–10685.
14. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents. 2022. Available online: http://arxiv.org/abs/2204.06125 (accessed on 28 May 2023).
15. Panero, J.; Zelnik, M. Human Dimesion & Interior Space; Watson-Guptill: New York, NY, USA, 1979; Available online: www.crownpublishing.com%0Awww.watsonguptill.com (accessed on 28 May 2023).
16. Peter, N.; Ernst, N. Architect’s Data, 3rd ed.; Wiley-Blackwell: Hoboken, NJ, USA, 2002; pp. 1–2.
17. Merrell, P.; Schkufza, E.; Li, Z.; Koltun, V.; Agrawala, M. Interactive Furniture Layout Using Interior Design Guidelines. ACM Trans. Graph. 2011, 30, 1–10.
18. Geman, S.; Geman, D. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Trans. Pattern Anal. Mach. Intell. 1984, PAMI-6, 721–741.

19. Yu, L.F.; Yeung, S.K.; Tang, C.K.; Terzopoulos, D.; Chan, T.F.; Osher, S.J. Make it home: Automatic optimization of furniture arrangement. ACM Trans. Graph. 2011, 30, 1–11.

20. Zhang, S.K.; Xie, W.Y.; Zhang, S.H. Geometry-Based Layout Generation with Hyper-Relations AMONG Objects. Graph. Model. 2021, 116, 101104.

21. Kim, J.; Lee, J.K. Stochastic detection of interior design styles using a deep-learning model for reference images. Appl. Sci. 2020, 10, 7299.

22. PlanFinder. Available online: https://www.planfinder.xyz/ (accessed on 15 May 2023).

23. Finch—Optimizing Architecture. Available online: https://www.finch3d.com/ (accessed on 15 May 2023).

24. Rayon. Collaborative Space Design. Available online: https://www.rayon.design/ (accessed on 15 May 2023).

25. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. Commun. ACM 2014, 63, 139–144.

26. GAN Deep Learning: A Practical Guide. Available online: https://datagen.tech/guides/computer-vision/gan-deep-learning/ (accessed on 15 May 2023).

27. Brock, A.; Donahue, J.; Simonyan, K. Large scale GaN training for high fidelity natural image synthesis. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019; pp. 1–35.

28. Sauer, A.; Karras, T.; Laine, S.; Geiger, A.; Aila, T. StyleGAN-T: Unlocking the Power of GANs for Fast Large-Scale Text-to-Image Synthesis. arXiv 2023, arXiv:arXiv:2301.09515.

29. Skorokhodov, I.; Tulyakov, S.; Elhoseiny, M. StyleGAN-V: A Continuous Video Generator with the Price, Image Quality and Perks of StyleGAN2. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 3616–3626.

30. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 5967–5976.

31. Zhu, J.Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A.A.; Wang, O.; Shechtman, E. Toward multimodal image-to-image translation. Adv. Neural Inf. Process. Syst. 2017, 30, 466–477.

32. Park, T.; Liu, M.Y.; Wang, T.C.; Zhu, J.Y. Semantic image synthesis with spatially-adaptive normalization. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2332–2341.

33. Huang, W.; Zheng, H. Architectural drawings recognition and generation through machine learning. In Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture, ACADIA 2018, Mexico City, Mexico, 18–20 October 2018; Recalibration on Imprecision and Infidelity. pp. 156–165.

34. Yang, B.; Li, L.; Song, C.; Jiang, Z.; Ling, Y. Automatic furniture layout based on functional area division. In Proceedings of the 2019 International Conference on Cyberworlds, CW 2019, Kyoto, Japan, 2–4 October 2019; pp. 109–116.

35. Fu, H.; Cai, B.; Gao, L.; Zhang, L.X.; Wang, J.; Li, C.; Zeng, Q.; Sun, C.; Jia, R.; Zhao, B.; et al. 3D-FRONT: 3D Furnished Rooms with layOuts and semaNTics. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10913–10922.

36. Liu, C.; Schwing, A.; Kundu, K.; Urtasun, R.; Fidler, S. Rent3D: Floor-Plan Priors for Monocular Layout Estimation. Available online: https://www.cs.toronto.edu/~fidler/projects/rent3D.html (accessed on 15 May

2023).

37. Mwiza. RentHop|Kaggle. 2017. Available online: https://www.kaggle.com/code/mwizasimbeye11/renthop/data (accessed on 28 May 2023).
38. Zeng, Z.; Li, X.; Yu, Y.K.; Fu, C. GitHub—Zlzeng/DeepFloorplan. 2019. Available online: https://github.com/zlzeng/DeepFloorplan (accessed on 28 May 2023).
39. de las Heras, L.P.; Ahmed, S.; Liwicki, M.; Valveny, E.; Sánchez, G. Statistical segmentation and structural recognition for floor plan interpretation: Notation invariant structural element recognition. Int. J. Doc. Anal. Recognit. 2014, 17, 221–237.
40. Projects, H.; Datasets, P.P. CVC-FP: Database for structural floor plan analysis Images: International Journal on Document Analysis and Recognition. 2015. Available online: http://dag.cvc.uab.es/resources/floorplans/ (accessed on 28 May 2023).
41. IKEA. IKEA Furniture. 2014. Available online: https://www.kaggle.com/datasets/ahmedkallam/ikea-sa-furniture-web-scraping (accessed on 20 April 2020).
42. Voisey, B.S. Data Loading for CE8. 2019. Available online: https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_data_loading_for_ml_proje (accessed on 28 May 2023).
43. Python JSON—Guide—Machine Learning Plus. 2020. Available online: https://www.machinelearningplus.com/python-json-guide/ (accessed on 28 May 2023).
44. Colleges, G.T.U.A.; Academy, O.; Academy, O.; Academy, O.; Science, A.C.; Technology, I.; Science, A.C. Microsoft COCO. In Proceedings of the ECCV 2014: European Conference on Computer Vision, Zurich, Switzerland, 5–12 September 2014; pp. 740–755.
45. CVAT. Available online: https://www.cvat.ai/ (accessed on 19 September 2022).
46. Betzalel, E.; Penso, C.; Navon, A.; Fetaya, E. A Study on the Evaluation of Generative Models. arXiv 2022, arXiv:arXiv:2206.10935.
47. Brownlee, J. How to Implement the Frechet Inception Distance (FID) for Evaluating GANs. Available online: https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/ (accessed on 15 May 2023).
48. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. Adv. Neural Inf. Process. Syst. 2017, 30, 6627–6638.
49. Binkowski, M.; Sutherland, D.J.; Arbel, M.; Gretton, A. Demystifying MMD GANs. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018—Conference Track Proceedings, Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–36.
50. Padilla, R.; Netto, S.L.; Da Silva, E.A.B. A Survey on Performance Metrics for Object-Detection Algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niteroi, Brazil, 1–3 July 2020; pp. 237–242.
51. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.

Retrieved from https://encyclopedia.pub/entry/history/show/107980