

# A Machine Learning-Based Framework for Real-Time 3D Reconstruction and Space Utilization in Built Environments

Samarth Patel<sup>1</sup>, Abhishek Mukhopadhyay<sup>1</sup>, Priyavrat Sharma<sup>1</sup> and Pradipta Biswas<sup>1</sup>

<sup>1</sup> Indian Institute of Science, Bangalore, India

## Abstract

The process of 3D reconstruction involves transforming 2D images or data into a three-dimensional representation of an object, model, or environment. While supervised 3D reconstruction has made significant strides using deep neural networks, it is often time-consuming due to extensive image stitching and the requirement for specialized imaging sensors such as 360-degree or depth cameras. This paper introduces a machine learning-based 3D reconstruction framework aimed at making informed decisions regarding space utilization and asset management within any built environment. The proposed system comprises three key components: (I) object detection on 2D frames to identify target objects, (II) calculation of their pose using image processing techniques, and (III) utilization of an artificial neural network to map real and virtual environments. The evaluation using YOLOv7 demonstrated an accuracy of F1 score of 0.70 in detecting objects of interest. Pose estimation analysis indicated that the proposed algorithm could estimate object orientation with an error rate of  $8.03^\circ$ . The mapping algorithm exhibited high-quality performance, achieving a correlation coefficient of  $R^2 = 0.97$ . Ultimately, all this information is transmitted and visualized in the reconstructed virtual model, enabling remote monitoring and simulation.

## Keywords

Intelligent Reconstruction, Digital Twin, Object Detection, Mapping, Convolutional Neural Network (CNN), Virtual Environment

## 1. Introduction

The process of reconstructing a real-world scenario in three dimensions (3D) entails creating a 3D model from 2D images, point clouds, silhouettes, and similar data sources [1]. The process aims to generate a virtual representation applicable in visualization, animation, simulation, and analysis across fields like computer vision, robotics, and virtual reality. In the realm of computer vision research, significant attention has been given to 3D reconstruction, with a focus on areas such as structure from motion [2] or Multiview stereo [3]. These methods rely on multiple images to establish accurate correspondences or ensure comprehensive coverage, but they can be time-consuming due to the extensive image stitching and the requirement for specialized imaging sensors like 360-degree or depth cameras. Reconstructing built environments, such as car or ship interiors, is comparatively more straightforward, given the availability of standard Computer-aided design (CAD). Previous works [4, 5, 6] have developed virtual models by leveraging architectural drawings and integrating 3D furniture models. However, this process is time-consuming, involving modeling, physics simulation, and rendering capabilities of commercial game engines. The advent of deep neural networks has facilitated the incorporation of real-world objects into virtual spaces by learning from large datasets. CAD models of real-world objects

---

<sup>1</sup>Joint Proceedings of the ACM IUI Workshops 2024, March 18-21, 2024, Greenville, South Carolina, USA

✉ samarthpatel@iisc.ac.in (S. Patel); abhishekmukh@iisc.ac.in (A. Mukhopadhyay); priyavrats@iisc.ac.in (P. Sharma); pradipta@iisc.ac.in (P. Biswas)

ORCID 0009-0006-0705-3465 (S. Patel); 0000-0002-4341-0523 (A. Mukhopadhyay); 0009-0002-5063-141X (P. Sharma); 0000-0003-3054-6699 (P. Biswas)



© 2024 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).



CEUR Workshop Proceedings (CEUR-WS.org)

assist in learning the mapping from images to virtual models, streamlining the process. Once a virtual reality (VR) model is established, it can be connected to real-time imaging and environmental sensors, enabling the estimation of room occupancy and power consumption within built environments [5]. In this context, we propose a digital twin (DT) of a built environment through an interactive and immersive VR experience. A digital twin represents an object or system virtually throughout its lifecycle, updated with real-time data and employing simulation, machine learning, and reasoning to facilitate decision-making [7]. This paper aims to develop an efficient VR-based DT for built environments in real-time, expediting the existing 3D reconstruction process. This system allows standard virtual walkthroughs and provides real-time room occupancy estimates, energy assessments, and the potential for expansion into asset tracking and maintenance. The detection and localization of objects in the real world involved deploying a pre-trained YOLOv7 model, which underwent transfer learning on a custom dataset. Image processing techniques were employed to estimate the pose of real-world objects and map them into a virtual environment. An extensive comparison study among machine learning models was conducted to determine an accurate mapping technique. Mapping two-dimensional coordinates onto the virtual camera feed establishes a connection between the real and virtual worlds, enabling the real-time simulation of object movements in physical space. The contributions of this work are as follows.

- Proposed a new way of reconstructing real-world space to virtual environment in real-time.
- Validated the mapping between real and virtual environment by comparing several machine learning models.

The paper is organized as follows. Section 2 explains literature review on different methods for object detection systems. Section 3 explains the training and evaluation of object detection models, comparison studies between different machine learning models used for mapping and pose calculation technique in detail. Experiments and results are discussed in detail in Section 4, followed by discussion and conclusion in Section 5 and 6, respectively.

## 2. Related Work

In this section, the previous works on mapping, pose estimation and applications of object detection on digital twin are summarized in detail.

### 2.1. Digital Twin

Research on automated construction of digital twins and the inclusion of secondary objects has employed a blend of simple image processing techniques and sophisticated deep learning. Commercial software such as EdgeWise, Pointfuse, Point Cab, and Leica Cyclone Model leverage shape detection and fitting algorithms, with academic literature also using machine learning for secondary object detection [8-11]. Traditional computer vision techniques and CNN-based methods, such as the DeepLab architecture, have also been used to classify object classes and generate walls, cable trays, and ventilation ducts [12-15]. A noteworthy study utilized laser scanning, object detection, and OCR to enrich a digital twin with secondary objects and semantic information [16]. Despite their potential, deep learning approaches often require extensive labelled training data, a challenge that synthetic data generation techniques may alleviate [6, 17-19].

Perhaps, the most closely related work to ours is by Zhou et al. [20], who used computer vision to update a BIM-based digital twin of a building in real-time. Their approach incorporated YOLOv5 for object detection and utilized the Total3DUnderstanding method [21] for estimating object pose. This work claimed to achieve successful object capture, including desks, chairs, plants, and computer monitors, by transforming object coordinates from the image to the physical world and then to the digital twin. However, the paper did not provide a report on the accuracy of the object detection models, which is a crucial component of the pipeline. Instead, their focus

was primarily on orientation correction. In contrast, our approach targets more significant object categories and includes a comprehensive report on the accuracy of individual components: object detection, mapping algorithm, and pose estimation. This individual accuracy analysis allows for a better understanding of any accuracy limitations in the deployment process.

## **2.2. Mapping Between Real and Virtual Space**

The coordinates of the different objects that were detected has to be mapped in the virtual 3D space of unity. Sun et al. [22] focused on mapping virtual space onto real space using planar mapping, exploring the methods and algorithms employed to achieve accurate and efficient mapping. Ren et al. [23] employed spatial affine transformation to map virtual objects onto 2D images, aiming to integrate virtual objects into real-world scenes seamlessly. Huang et al. [24] proposed an algorithm specifically designed for mapping real space to a virtual globe space, addressing the need for robust and efficient mapping techniques applicable to diverse real-world environments. Schwarz et al. [25] discussed the usage of LIDAR systems by participants in a DARPA-sponsored event to generate a digital view of the surrounding terrain in autonomous vehicles, emphasizing the crucial role of LIDAR technology in facilitating accurate perception and navigation in autonomous systems. Mandli Communications [26] employed LIDAR sensors to generate point clouds and subsequently create digital terrain models, showcasing the practical application of LIDAR for terrain modelling purposes.

## **2.3. Pose Estimation**

The classical approach for estimating pose traditionally treated it as a nonlinear least-squares problem, employing nonlinear optimization algorithms for solving it [27-29]. Patil et al. [30] investigated and compared various vision-based, hybrid, and deep learning-based approaches for pose estimation from monocular vision. Similarly, Lan et al. [31] provided an overview of different deep learning-based techniques for human pose estimation. Another notable contribution by Collet et al. [32] is the MOPED (Multiple Object Pose Estimation and Detection) framework, designed to deliver robust performance in complex scenes and low latency for real-time applications. Vikstenen et al. [33] developed a system that leverages algorithmic multi-cue integration (AMC) and temporal multi-cue integration (TMC) to increase the pose estimation performance.

## **2.4. Deep Learning in Digital Twin**

Deep learning is a subset of a larger family of ML approaches; it can take data and automatically perform tasks like Classification, regression, clustering and pattern recognition. Deep learning is very effective in the detection of complicated structures and things [34]. In another research, Ogunseiju et al. [35] investigated the effectiveness of a variety of deep CNNs for recognizing construction worker activities from images of signals from time-series data using large datasets for training and testing DL algorithms. Deep learning has proved to be very good in object detection [36], and the use of DT involving humans has been proposed by a lot of researchers in the construction activity to prevent causalities and improve ergonomics of workers, Botton et al. [37] explored the use introducing a temporal dimension in the 3D simulation of Construction activities.

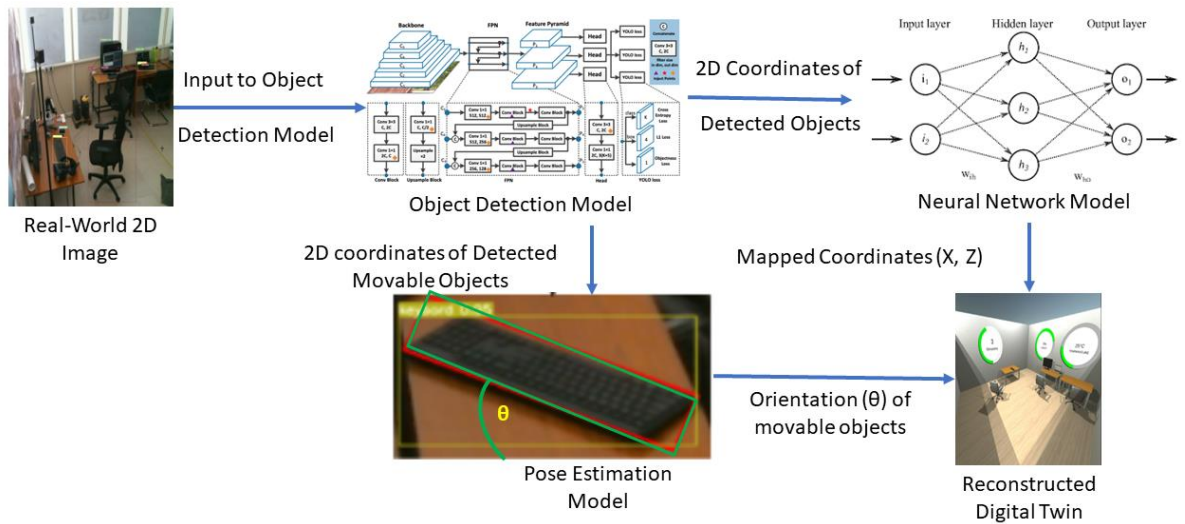
Summarizing the literature survey, this paper aims to address several key research gaps in the field of digital twin construction. Certain limitations are inherent in past work, including the high cost associated with depth cameras and the inability to capture specular and transparent objects. In addition, the manual intervention required to verify detected objects in the point clouds results in a time-consuming process [17]. Furthermore, while deep learning techniques have been applied to detect building elements, the literature lacks evidence of their utilization for identifying room furniture and other entities, which assists in understanding space utilization. Moreover, there has been limited exploration in the architecture, engineering, and construction

(AEC) domain concerning the mapping of secondary objects. Zhou et al. [20] utilize a 3D estimation network for extracting the pose of each object and a camera-BIM location transformation algorithm for mapping the coordinates of the detected objects in BIM. In contrast, our approach utilizes image processing steps and the minimal area rectangle method for pose estimation, and a Machine learning-based algorithm for mapping. By addressing these research gaps, this paper enhances the efficiency, accuracy, and comprehensiveness of digital twin creation and updating in real-time using computer vision techniques.

### 3. Proposed Approach

The proposed 3D reconstruction process begins with the detection of objects in the real-world environment. While numerous pre-trained models are available for this purpose, our study relies on the findings of previous work [38, 39] which demonstrate that YOLO performs better in terms of the tradeoff between latency and accuracy. YOLOv7 was chosen over various YOLO model variants following a comparative analysis. The detected objects are further classified into three categories: movable, partially movable, and immovable. Movable objects encompass persons and chairs, while the partially movable are keyboard, laptop, TV/monitor, and mouse while remaining objects are deemed immovable. This distinction is crucial to maintain real-time accuracy and resource efficiency, as movable, and partially movable objects require frequent updates due to their volatile positions, while immovable objects, remaining static most of the time, necessitate fewer updates. Accordingly, the movable objects undergo more frequent updates, occurring every frame, while partially movable objects are updated every 5 minutes. The immovable objects are updated less frequently, with a refresh rate of every 24 hours. The classification of the objects into groups helps the proposed system to reconstruct real-world in real time.

We also compare linear regression with degrees 1, 2, and 3, support vector regression (SVR) with radial basis function (RBF) and polynomial kernels [40], and a vanilla neural network to map between real-world and virtual objects. The pose estimation of objects is performed using a combination of different image processing techniques. The position and orientation of the objects are communicated to the game engine using socket programming for real-time interaction and synchronization. Figure 1 represents the working of the proposed system.



**Figure 1:** Flow diagram of the proposed system

#### 3.1. Object Detection

In this Section, a detailed analysis of the dataset preparation strategy is discussed followed by comparison study between object detection models in study.

**Dataset Preparation:** The dataset encompasses various day-to-day office utilities, including chair, keyboard, laptop, TV/monitor, refrigerators, desk, mouse, person, and couch. The dataset used in this study was derived from two sources. The first source involved filtering the MS COCO dataset [41] to extract the above-mentioned class labels. The second source was a crowd-sourced dataset collected for this research. The images obtained from the user-generated dataset were annotated with nine class labels. To annotate the images, the Computer Vision Annotation Tool (CVAT) was employed, allowing for manual annotation through visual inspection. The dataset was then divided into train, validation, and test subsets, with each class label having specified entries in each subset, as shown in Table 1. The annotations were compiled into an XML file format using the CVAT tool. Subsequently, the XML file format was converted to the YOLO file format, which was utilized for training the object detection model.

**Comparison between Object Detection Models:** We compared YOLO models (v4 to v7) to choose the best model. By using transfer learning, we leveraged the pretrained weights of these models on a large-scale dataset and fine-tuned the model on bespoke dataset, resulting in improved object detection capabilities within our digital twin environment. Model training was conducted on an NVIDIA 3090 Ti GPU, and performance was assessed using three evaluation metrics, including mean Intersection Over Union (mIOU), precision, and F1 score.

### 3.2. Mapping Techniques

In the next step, the detected objects were used to map corresponding objects in a virtual environment created using Unity 3D. The mapping focused on three out of six degrees of freedom (DOFs), specifically translation along the X and Y-axis, and rotation around the yaw. This selection of three DOFs was specific to this problem, given that all objects are situated within a 2D plane, such as the floor or tabletop. We mapped the center of base from 2D frame to appropriate plane in virtual environment. The extracted center coordinates were utilized as inputs for various regression algorithms, including linear regression with degrees 1, 2, and 3, support vector regression (SVR) with radial basis function (RBF) and polynomial kernels [40], and a vanilla ANN.

**Table 1**  
**Instances of class label in train, validation, and test data**

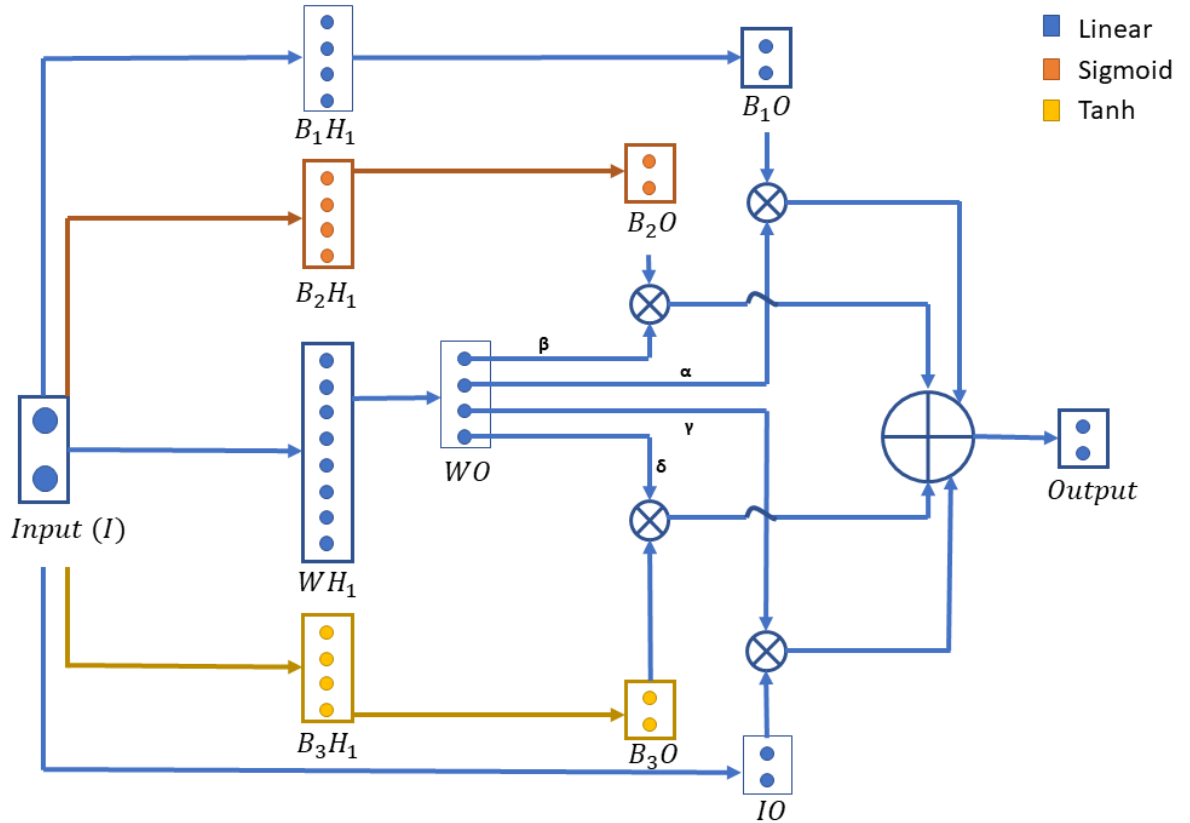
Class	Train	Validation	Test
Chair	12322	7729	1813
Keyboard	1002	585	175
Laptop	1520	1092	231
TV/Monitor	2128	1384	479
Refrigerator	824	629	126
Desk	590	96	43
Mouse	876	464	125
Person	81153	53753	10995
Couch	1741	1192	261

Linear regression with degree 1 fits a straight-line relationship between the features and the target. When degree 2 is used, quadratic terms are introduced to capture curved patterns. Additionally, when degree 3 is employed, cubic terms are added, enabling the model to fit S-shaped patterns and capture more complex relationships. SVR with RBF kernel is a powerful non-linear data regression approach with the use of a Gaussian-like function, it modifies the input space to capture complex connections and trends similarly polynomial relationships can be captured using SVR with a polynomial kernel it converts data into a higher-dimensional space and uses polynomial functions to assess similarity. The architecture of the ANN consists of four Multi-Layer Perceptron (MLP) blocks, with three of them responsible for the mapping process in

conjunction with the input (Figure 2). We propose a hybrid structure by using a weighted summation of the output of these MLP blocks. The structure for the first three blocks is 2-4-2 (input layer-hidden layer-output layer) and for the fourth block (weight block) it is 2-8-4 (input layer-hidden layer-output layer). The activation used in the first three blocks are linear, sigmoid, and tanh respectively and for the weight block linear activation is used. The model was trained using Adam optimizer and mean square error as loss function. The fourth block incorporates weights for all four outputs, scaling the output based on the specific degree of non-linearity. In the feature fusion part, we undertake feature fusion as described in Equation 1.

$$O(x, y) = \alpha \times B_1(x, y) + \beta \times B_2(x, y) + \delta \times B_3(x, y) + \gamma \times I(x, y), \quad (1)$$

where,  $\alpha, \beta, \delta, \gamma$  denotes the weight parameters,  $B_1, B_2$ , and  $B_3$  indicates three different blocks which are connected in parallel and  $O(x, y)$  is the weighted sum of the parallel outputs and input. Each algorithm was trained using a dataset consisting of frame coordinates and corresponding virtual world coordinates. Frame coordinates are generated based on the bounding box location on the screen generated by object detection model. Correspondingly, in virtual environment, we placed objects in same location and generated the virtual world coordinates. After the training process, the algorithms could predict the coordinates of objects within the virtual environment based on the object detection model generated coordinates. The predicted coordinates were then transmitted to virtual environment using socket programming. This communication facilitated the dynamic mapping of the detected objects within the virtual environment, resulting in an immersive and interactive user experience.



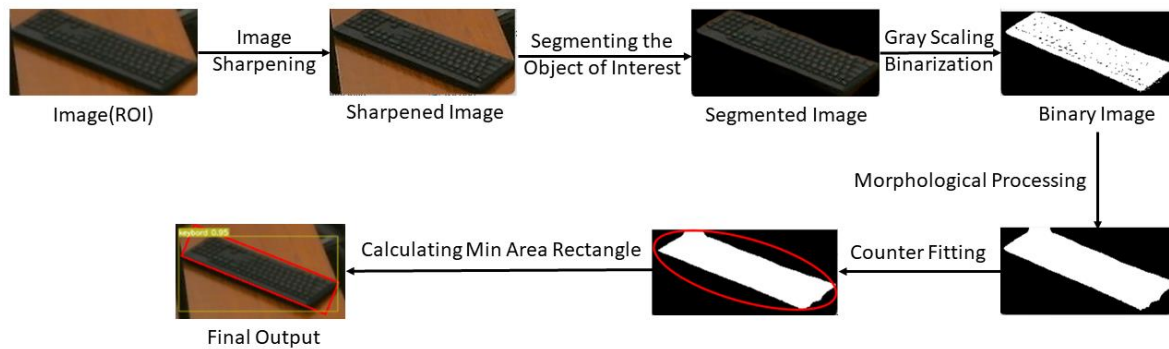
**Figure 2:** The proposed neural network architecture for mapping objects between real and virtual world. Here  $B_1, B_2$ , and  $B_3$  indicates three MLP blocks whereas  $W$  indicate weight block to derive weight outputs.

### 3.3. Pose Correction

A series of image processing steps were followed for estimating the pose of the object in real-world space in the process of digital twin reconstruction. First, the region of interest (ROI) was obtained from the coordinates generated by the object detection model. Subsequently, image sharpening was applied to enhance the visibility of the objects in the images. Next, segmentation was performed to isolate the required objects from the background. The resulting segmented images were then converted into a binary grayscale format, simplifying the subsequent image analysis process. To further refine the binary images, a morphological process was applied to fill in small patches present in the binary images. Following this, contour fitting was performed on the binary images to accurately obtain the shapes and boundaries of the objects. Finally, the minimum area rectangle method was utilized to determine the precise positions and orientations of the objects in the images. This comprehensive approach facilitated the acquisition of more accurate representations of the objects' poses in the digital twin environment. By implementing these image processing steps, the objects were successfully detected and mapped in the digital twin, enabling the creation of an accurate representation of the physical world. The proposed pose detection algorithm is described in Algorithm 1 and their respected process output is shown in Figure 3.

### 3.4. Placing in 3D Virtual Environment

The virtual environment reconstruction involves the transmission of pose and location data for target objects from computer vision algorithms to a virtual environment. Prior to receiving the data stream, a virtual representation of the real-world space is constructed using a game engine, incorporating precise real-world measurements. The virtual environment also captures and represents the movement of individual people within the virtual environment. This is achieved by mapping people's movement in each frame and depicting their actions, such as sitting at specific workstations or engaging in movement. Walking animations are employed to visualize the movement of people.



**Figure 3:** Image processing steps involve pose estimation.

---

Algorithm 1 Image processing steps involve pose estimation.

---

Input: Frames:  $1920 \times 1080$  from video

1 Get ROIs from object detection model.

2 for iterate across detections (ROIs):

    2.1 Sharpen cropped patches.

    2.2 Remove backgrounds (by adaptive segment masks).

    2.3 Convert to grayscale and binarize.

    2.4 preprocessed\_image  $\leftarrow$  Apply morphological transforms to remove outlier pixels.

    2.5 all\_contours  $\leftarrow$  ContourFit (preprocessed\_image).

---



---

```

2.6 Initialize empty all_areas [].
2.7 for contour in all_contours:
    all_areas ← contour.
end for.
2.8 MaxAreaIdx ← argmax(all_areas).
2.9 angle ← min_area_rectangle (all_contours [MaxAreaIdx]).
end for
end

```

---

## 4. Experimental Evaluation

In this section, we have described the results of individual components of the proposed system in detail. outputs.

### 4.1. Object Detection Accuracy

The accuracy of the object detection models was evaluated using various metrics. These metrics were employed to measure the model's performance in identifying and localizing objects in the provided test data. Table 2 illustrates the performance of the models concerning the test data. YOLOv5 showcased the highest IoU score, while YOLOv7 exhibited consistent performance across classes, boasting the highest F1 score among the models. Moreover, YOLOv7 demonstrated a processing speed of 30.23 frames per second.

**Table 2**  
**Comparing Performance between Object Detection Models**

Models	Average IoU	Precision	F1 Score
YOLOv4	0.28	0.73	0.41
YOLOv5	0.59	0.63	0.63
YOLOv6	0.52	0.56	0.56
YOLOv7	0.56	0.72	0.70

### 4.2. Mapping Accuracy

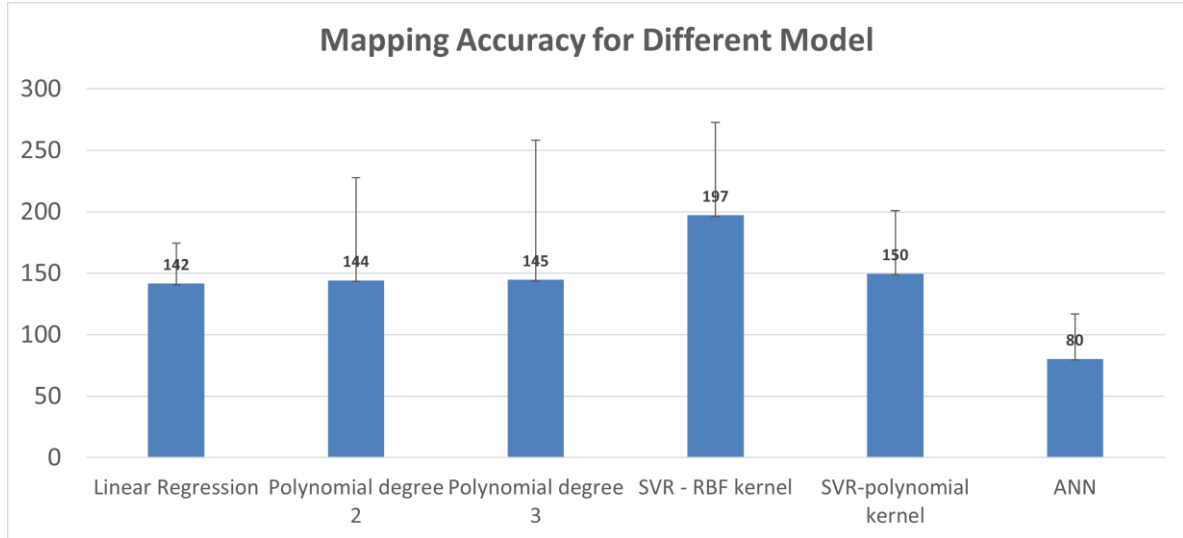
The accuracy of the mapping algorithms is measured using the Euclidean distance between the ground truth and predicted virtual-world positions. Figure 4 summarizes the comparison between the machine learning models as discussed in Section 3.2, deployed for mapping. It may be noted that neural network model achieved highest accuracy with the error of 80 centimeters. Figure 5 shows the correlation graph with coefficient of determination or  $R^2 = 0.97$  between actual distance measured in virtual environment and predicted by the proposed neural network model.

### 4.3. Pose Accuracy

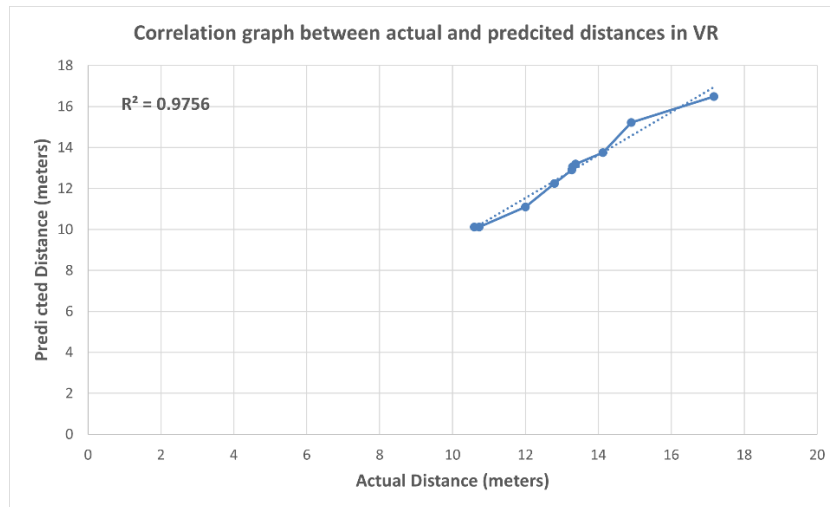
We reported pose accuracy by analyzing the actual orientation of the object and corresponding orientation measured by the algorithm. We marked a cartesian coordinate system on a table by marking the angles. Then we placed the keyboard and TV on the table. We noted their actual angle with reference line and the orientation generated by the algorithm. Figure 6 shows the correlation between actual and predicted orientation. It may be noted that the



coefficient of determination was observed as  $R^2 = 0.99$ , while average error was found to be  $8.03^\circ$ . The mapping between real-world objects and corresponding virtual objects is exemplified in Figure 7, providing a concrete illustration. In this scenario, the focus was only on the yaw angle regarding the positioning of movable objects—keyboards, laptops, TVs/Monitors, and desks—on a level 2D plane. Considering the nature of these objects placed on a flat plane, any rotation around the pitch or roll angles in a 3D scene was deemed unnecessary or not applicable for the specific context being addressed.



**Figure 4:** Comparison of mapping accuracy between different models.

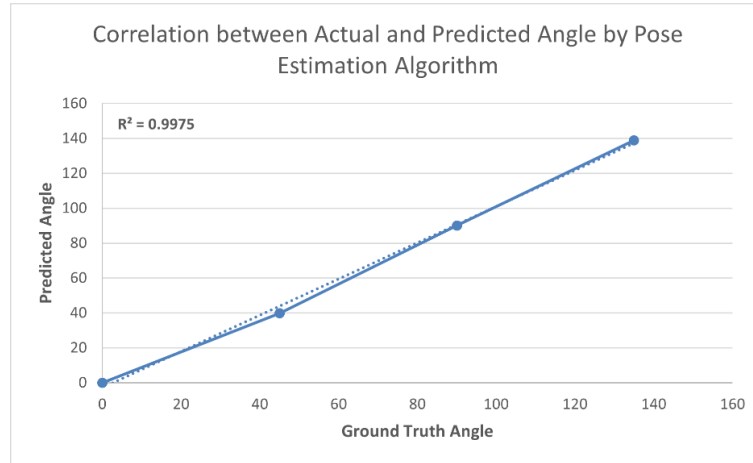


**Figure 5:** Scatter plot of distances measured in actual and predicted by neural network model in virtual environment.

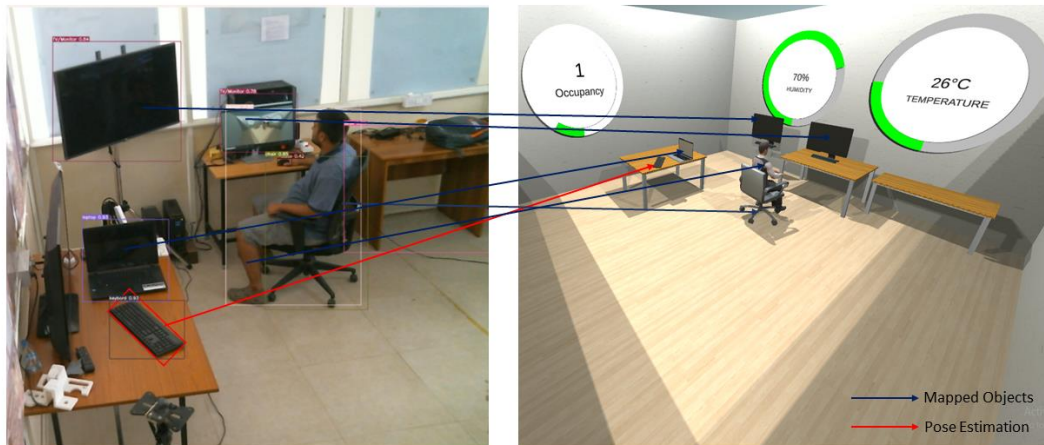
## 5. Discussion

This paper presents a data-driven approach for reconstructing a digital twin of an office space environment. The previous work [5] proposed a VR digital twin of physical space for measuring room occupancy and social distancing measurement. However, the main problem with this approach lies in replicating physical space. It is a time-consuming process as it requires modeling, physics simulation, and rendering capabilities of Unity 3D. On the other hand, this proposed system offers a different and beneficial approach to constructing VR digital twins compared to conventional methods. By incorporating machine learning techniques, the development of digital

twins becomes more scalable and efficient. Machine learning enables accurate object detection, orientation estimation, and 2D-to-3D mapping, resulting in high-quality virtual workspaces. This approach addresses challenges associated with asset placement, enhances space planning and visualization, and promotes energy efficiency and sustainability within workspaces. In this section, we provide a summary of all the key points discussed in Section 1.



**Figure 6:** Correlation graph between actual angle and predicted angle estimated by pose estimation algorithm.



**Figure 7:** Scatter plot of distances measured in actual and predicted by neural network model in virtual environment.

**Reconstruction Techniques:** There are various techniques available to 3D reconstruct a space from 2D images. Cutting-edge automated image orientation techniques, such as Structure from Motion, and dense image matching methods like Multiple View Stereo, which are widely utilized for deriving 3D information from 2D images, can yield 3D outcomes, such as point clouds or meshes, exhibiting diverse levels of geometric accuracy and visual fidelity [42]. This technique requires a lot of images from various directions. It takes ample time to reconstruct an object for a given instance of time, making it harder to reconstruct a real-time scenario of the real world. This paper uses a novel data-driven approach, which utilizes only one 2D image of the real world and reconstructs it in virtual reality. This reconstruction happens in two stages. The first stage is detecting objects from the real world using 2D images and calculating their pose using various image processing steps, and the second stage is mapping the objects from the real world to the virtual world. We utilized the YOLOv7 model as our object detection model to detect objects from the real world. The model performed well with an accuracy of F1 score of 0.70. This is beneficial for reconstruction purposes as it can detect small objects with higher accuracy, e.g., a mouse with a mAP of 0.81. The proposed pose detection algorithm showed its efficiency with error rate of

8.03°. This higher accuracy of pose estimation can be helpful in detecting orientation in the real world to reflect in the corresponding virtual environment. Thus, the proposed system will be impact full in creating a digital twin. A supplementary video (<https://youtu.be/advtkAQ02Nk>) shows the working of the proposed system as well as how accurately real world is depicted in the virtual environment.

**Mapping Techniques:** There is a plethora of mapping techniques available for reconstructing a real space. One widely used technique is COLMAP [2], which is an end-to-end image-based 3D reconstruction pipeline. It employs Multi-View Stereo (MVS) to compute depth and/or normal information for every pixel in an image, using the output of Structure-from-Motion (SfM) [2-3]. By fusing the depth and normal maps of multiple images in 3D, a dense point cloud of the scene is generated. However, this technique requires many images from different viewpoints and high visual overlaps, making it slower and more time-consuming when creating a representation of real-world scenarios at a specific moment in time. In this paper, after comparing various classical machine learning techniques, it was determined that neural networks were the most suitable for the task as it is designed to handle the different degree of non-linearity for different points. The neural networks achieved an impressive average error of only 80 cm when mapping objects within the virtual world. The error was found to be 43.66% lower compared to the second-best model, which was linear regression. The proposed system holds potential for application on workshop floors, facilitating remote monitoring, asset tracking, and various other functions.

This proposed approach employs machine learning techniques to reconstruct digital twins of physical spaces efficiently, streamlining the process and enhancing accuracy in object detection, orientation estimation, and mapping. Addressing limitations in standard 3D construction, such as missing CAD/BIM data of objects, extensive object volumes, or wide capture areas, the approach simplifies real-time reconstruction using a single 2D image. This streamlined process demonstrates promising potential for swift and precise translations from the real world to the virtual realm, contrasting with time-consuming traditional methods.

## 6. Conclusions

This study endeavors to devise a cost-effective solution for constructing a virtual model of a built environment. The suggested system serves as a VR-based digital replica of an office, integrating real-time monitoring of human occupancy and asset utilization. The system's accuracy hinges on the performance of the object detection model, with a reported high level of precision, i.e., an F1 score of 0.70. The pose estimation algorithm significantly corrects the movement of movable objects, such as keyboards, monitors, and desks, exhibiting a high correlation ( $R^2 = 0.99$ ). The proposed neural network model successfully maps objects from the 2D image plane to a 3D plane in a virtual environment, demonstrating a correlation of  $R^2 = 0.97$ . Real-time mapping of human positions and precise estimation of asset poses offer numerous advantages, enabling the floor management team to conduct thorough remote walkthroughs and gain insights into room occupancy and office asset utilization. This information empowers informed decisions on sustainable space usage and asset management. However, challenges are encountered in mapping objects from the 2D plane to the 3D plane. Future work will focus on implementing 3D object detection, promising accurate positioning and orientation of real-world objects. The proposed framework underwent testing in various office spaces, with all data transmitted to the digital twin of the real-world space (Please refer to <https://youtu.be/advtkAQ02Nk>).

## References

- [1] 3D Reconstruction. Retrieved December 10, 2023, URL: <https://paperswithcode.com/task/3d-reconstruction>.
- [2] J. L. Schonberger, and J. M. Frahm. 2016. Structure-from-motion revisited. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

- [3] J. L. Schönberger, E. Zheng, J. M. Frahm, and M. Pollefeys. 2016. Pixelwise view selection for unstructured multi-view stereo. In European Conference on Computer Vision (ECCV), 2016.
- [4] A. Mukhopadhyay, G. S. R. Reddy, S. Ghosh, L. R. D. Murthy and P. Biswas. 2021. Validating Social Distancing through Deep Learning and VR-Based Digital Twins. In 27th ACM Symposium on Virtual Reality Software and Technology VRST '21, December 08-10, 2021, Osaka, Japan. ACM, New York, NY, USA, 4 Pages. <https://doi.org/10.1145/3489849.3489959>
- [5] A. Mukhopadhyay, G. S. R. Reddy, K. S. Saluja, S. Ghosh, A. Peña-Rios, G. Gopal, and P. Biswas. 2022. Virtual-reality-based digital twin of office spaces with social distance measurement feature. *Virtual Reality & Intelligent Hardware* 4, 1 (2022), 55–75.
- [6] A. Mukhopadhyay, G. S. R. Reddy, I. Mukherjee, G. Gopal, A. Peña-Rios, and P. Biswas. 2021. Generating Synthetic Data for Deep Learning using VR Digital Twin. In Proceedings of the 2021 5th International Conference on Cloud and Big Data Computing (ICCBDC '21). Association for Computing Machinery, New York, NY, USA, 52–56.
- [7] IBM. What is a digital twin? Retrieved December 10, 2023, URL: <https://www.ibm.com/topics/what-is-a-digital-twin>.
- [8] U. Krispel, H. L. Evers, M. Tamke, R. Viehauser, and D. W. Fellner, “Automatic texture and orthophoto generation from registered panoramic views,” *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 5, p. 131, 2015.
- [9] L. Díaz-Vilariño, H. González-Jorge, J. Martínez-Sánchez, and H. Lorenzo, “Automatic lidar-based lighting inventory in buildings,” *Measurement*, vol. 73, pp. 544–550, 2015.
- [10] T. Czerniawski, M. Nahangi, C. Haas, and S. Walbridge, “Pipe spool recognition in cluttered point clouds using a curvature-based shape descriptor,” *Automation in Construction*, vol. 71, pp. 346–358, 2016.
- [11] P. Kima, J. Chenb, and Y. K. Choa, “Building element recognition with thermal-mapped point clouds,” in 34th International Symposium on Automation and Robotics in Construction (ISARC 2017), 2017.
- [12] I. Anagnostopoulos, V. Patrăucean, I. Brilakis, and P. Vela, “Detection of walls, floors, and ceilings in point cloud data,” in *Construction Research Congress 2016*, 2016, pp. 2302–2311.
- [13] J. Han, M. Rong, H. Jiang, H. Liu, and S. Shen, “Vectorized indoor surface reconstruction from 3d point cloud with multistep 2d optimization,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 177, pp. 57–74, 2021.
- [14] J. Guo, Q. Wang, and J.-H. Park, “Geometric quality inspection of prefabricated mep modules with 3d laser scanning,” *Automation in Construction*, vol. 111, p. 103053, 2020.
- [15] T. Czerniawski and F. Leite, “Automated segmentation of rgb-d images into a comprehensive set of building components using deep learning,” *Advanced Engineering Informatics*, vol. 45, p. 101131, 2020.
- [16] Y. Pan, A. Braun, I. Brilakis, and A. Borrmann, “Enriching geometric digital twins of buildings with small objects by fusing laser scanning and ai-based image recognition,” *Automation in Construction*, vol. 140, p. 104375, 2022.
- [17] V. Drobnyi, Z. Hu, Y. Fathy, and I. Brilakis, “Construction and maintenance of building geometric digital twins: State of the art review,” *Sensors*, vol. 23, no. 9, p. 4382, 2023.
- [18] S. I. Nikolenko, “Synthetic-to-real domain adaptation and refinement,” in *Synthetic data for deep learning*. Springer, 2021, pp. 235–268.
- [19] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Bochoon, and S. Birchfield, “Training deep networks with synthetic data: Bridging the reality gap by domain randomization,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 969–977.
- [20] X. Zhou, K. Sun, J. Wang, J. Zhao, C. Feng, Y. Yang, and W. Zhou, “Computer vision enabled building digital twin using building information model,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 3, pp. 2684–2692, 2023.
- [21] Y. Nie, X. Han, S. Guo, Y. Zheng, J. Chang, and J. J. Zhang, “Total3dunderstanding: Joint layout, object pose and mesh reconstruction for indoor scenes from a single image,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 55–64.

- [22] Q. Sun, L.-Y. Wei, and A. Kaufman, "Mapping virtual and physical reality," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–12, 2016.
- [23] F. Ren and X. Wu, "Outdoor augmented reality spatial information representation," *Appl. Math*, vol. 7, no. 2L, pp. 505–509, 2013.
- [24] W. Huang and J. Chen, "A multi-scale vr navigation method for vr globes," *International journal of digital earth*, vol. 12, no. 2, pp. 228–249, 2019.
- [25] B. Schwarz, "Mapping the world in 3d," *Nature Photonics*, vol. 4, no. 7, pp. 429–430, 2010.
- [26] Feb 2021. [Online]. Available: <https://www.mandli.com/maverick-by-mandli-communications/>
- [27] G. H. Rosenfield, "The problem of exterior orientation in photogrammetry," *Photogrammetric Engineering*, vol. 25, no. 4, 1959.
- [28] E. Thompson, "The projective theory of relative orientation," *Photogrammetria*, vol. 23, no. 2, pp. 67–75, 1968.
- [29] R. Haralick and L. Shapiro, "Computer and robot vision, ch. 14," 1993.
- [30] A. V. Patil and P. Rabha, "A survey on joint object detection and pose estimation using monocular vision," *arXiv preprint arXiv:1811.10216*, 2018.
- [31] G. Lan, Y. Wu, F. Hu, and Q. Hao, "Vision-based human pose estimation via deep learning: A survey," *IEEE Transactions on Human-Machine Systems*, 2022.
- [32] A. Collet, M. Martinez, and S. S. Srinivasa, "The moped framework: Object recognition and pose estimation for manipulation," *The international journal of robotics research*, vol. 30, no. 10, pp. 1284–1306, 2011.
- [33] F. Viksten, R. Soderberg, K. Nordberg, and C. Perwass, "Increasing pose estimation performance using multi-cue integration," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. IEEE, 2006*, pp. 3760–3767.
- [34] J. Lee, M. Azamfar, J. Singh, and S. Siahpour, "Integration of digital twin and deep learning in cyber-physical systems: towards smart manufacturing," *IET Collaborative Intelligent Manufacturing*, vol. 2, no. 1, pp. 34–36, 2020.
- [35] O. R. Ogunseiju, J. Olayiwola, A. A. Akanmu, and C. Nnaji, "Recognition of workers' actions from time-series signal images using deep convolutional neural network," *Smart and Sustainable Built Environment*, vol. 11, no. 4, pp. 812–831, 2022.
- [36] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [37] C. Botton, "Supporting constructability analysis meetings with immersive virtual reality-based collaborative bim 4d simulation," *Automation in Construction*, vol. 96, pp. 1–15, 2018.
- [38] A. Mukhopadhyay, I. Mukherjee, and P. Biswas, "Comparing cnns for non-conventional traffic participants," in *Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications: Adjunct Proceedings, 2019*, pp. 171–175.
- [39] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez. 2020, On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. *Remote Sensing* 13, no. 1, 89.
- [40] A. J. Smola, and B. Schölkopf. 2004. A tutorial on support vector regression. *Statistics and computing*, 14, 199–222.
- [41] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V* 13. Springer, 2014, pp. 740–755.
- [42] E. K. Stathopoulou, M. Welpner, and F. Remondino, "Open-source image-based 3d reconstruction pipelines: Review, comparison and evaluation," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-2/W17*, pp. 331–338, 2019.