

PROFESSIONAL TRAINING REPORT - II

entitled

DeepPlanX:

Next-Gen Building Planning Analysis through Deep Learning Object Detection

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Internet of Things

by

G.Aakash [41732005]

S.V.S.Mahesh Kumar [41732015]

U.Senthur Ajay [41732016]

S.Sai Srujan [41732034]



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

MAY 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **Mr. G.Aakash(41732005)** who carried out the project entitled "**DeepPlanX:Next-Gen Building Planning Analysis through Deep Learning Object Detection**" under my supervision from January 2024 to April 2024.

Internal Guide
Mrs. K. ABINAYA M.E.,(Ph.D)

Head of the Department
Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on 06/05/2024

Internal Examiner

External Examiner

DECLARATION

I, Mr. G.Aakash(41732005), hereby declare that the Professional Training Report-II entitled "DeepPlanX:Next-Gen Building Planning Analysis through Deep Learning Object Detection" done by me under the guidance of Mrs.K.ABINAYA M.E,(Ph.D), is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Internet of Things.

DATE: 06/05/2024

PLACE: Chennai

G.Aakash

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

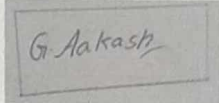
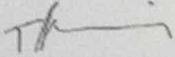
I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean, School of Computing, Dr. S.Vigneshwari M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Internal Guide **Mrs.K.ABINAYA M.E,(Ph.D)** for her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-2 Professional Training.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

Course Certificate

AFTERNOON SESSION (AN)	National Programme on Technology Enhanced Learning	
Hall Ticket For	2024 Apr: CS17 Cloud Computing - Online	NPTEL

Candidate Name	Grandhi Aakash					
Roll No	NOC24CS17S453401703		Seating Number	53401703		
Date of Birth	27-01-2004					
PwD Status	No	Compensatory Time Required	N.A	Scribe Required		N.A
Exam Date	Saturday, 20 April, 2024					
Reporting Time	01:00 pm	Gate Closure	02:30 pm			
Exam Timing	02:00 pm	Shift	AN			
Test Centre Name	ION Digital Zone IDZ Kovilambakkam					
Test Centre Address	Fortune Towers, 1st and 2nd Floor, SH 109, 200 Feet Thoraipakkam Pallavaram Radial Road, Near Eachangadu signal, Kovilambakkam, Chennai, Tamil Nadu, India - 600117					
 NPTEL Coordinator						

ABSTRACT

DeepPLanx introduces a groundbreaking approach for modernizing infrastructure planning and design procedures. This new technology transforms building plan generation and the transition to immersive 3D designs by leveraging deep learning capabilities and integrating machine learning algorithms. DeepPLanx quickly creates detailed 2D plans and provides a holistic perspective of the building layout before construction begins, using straightforward user inputs such as room specs, door and window positions, and other choices. This article investigates the intersection of cutting-edge technology and infrastructure development, demonstrating the power of deep learning to streamline and improve the construction sector. DeepPLanx provides a huge step forward in efficient, accurate, and user-friendly infrastructure design, opening the path for a more sustainable and dynamic built environment.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	Vii
1	INTRODUCTION	1
	1.1 Overview	
2	LITERATURE SURVEY	3
	2.1 survey	
3	REQUIREMENTS ANALYSIS	
	3.1 Objective	5
	3.2.1 Hardware Requirements	
	3.2 7	
	3.2.2 Software Requirements	
4	DESIGN DESCRIPTION OF PROPOSED PRODUCT	
	4.1.1 Ideation Map/Architecture Diagram	11
	4.1.2 Various stages	12
	4.1 4.1.3 Internal or Component design structure	14
	4.1.4 working principles	17
	4.3 4.2.1 Novelty of the Project	20
6	RESULTS AND DISCUSSION	22
7	SUMMARY AND CONCLUSIONS	24
	REFERENCES	25

LIST OF FIGURES

Figure No.	Figure Name	Page No.
Fig 4.1	System Architecture Model	12
Fig 4.2	Training architecture Input	14
Fig 5.1	Data Folder Generated	23
Fig 5.2	Blunder Files	24
Fig 5.3	Command Prompt Implementation for 3d Blunder plan	24
Fig 5.4	Cost Estimation for Building	25
Fig 5.5	Final Output	26

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In terms of infrastructure development, effective building planning and design are critical in defining our urban landscapes and improving residents' quality of life. Traditional techniques to architectural design and construction have long depended on manual drawing and Computer-Aided Design (CAD) software, which, although successful, can involve arduous processes and restricted design flexibility. However, with the introduction of deep learning and machine learning algorithms, there is a dramatic possibility to change the way structures are planned, designed, and built.

This article proposes DeepPLanx, a breakthrough paradigm that aims to reinvent infrastructure design by integrating sophisticated technology. DeepPLanx combines new innovation and classic architectural principles, providing a simplified way to creating building blueprints and converting them into immersive 3D designs. DeepPLanx uses deep learning to enable users to enter simple criteria like as room measurements, door and window positions, and other preferences, allowing for the quick creation of detailed 2D blueprints.

Furthermore, DeepPLanx goes beyond only plan creation, elegantly transforming these 2D blueprints into comprehensive 3D representations of buildings. DeepPLanx uses powerful machine learning algorithms to bridge the gap between 2D drawings and actual 3D models, giving architects, engineers, and urban planners crucial insights into the spatial arrangement and aesthetic composition of their projects.

This introduction prepares the groundwork for a thorough examination of DeepPLanx, its approach, applications, and implications for the future of infrastructure development. DeepPLanx ushers in a new age of efficiency, precision, and sustainability in the construction sector by embracing the promise of deep learning and machine learning, demonstrating technology's revolutionary power in influencing our built environment.

Furthermore, DeepPLanx has great promise for improving construction sustainability and efficiency. DeepPLanx allows designers to develop

structures that are both visually beautiful and environmentally responsible by optimizing spatial layouts, reducing material waste, and modelling environmental consequences.

This introduction prepares the groundwork for a thorough examination of DeepPLanx, its approach, applications, and implications for the future of infrastructure development. DeepPLanx ushers in a new age of efficiency, precision, and sustainability in the construction sector by embracing the promise of deep learning and machine learning, demonstrating technology's revolutionary power in influencing our built environment.

CHAPTER 2

LITERATURE REVIEW

2.1 SURVEY

Traditionally, architectural design and infrastructure planning have relied on manual drawing techniques and Computer-Aided Design (CAD) software. While these techniques have shown to be vital tools for architects and urban planners, they frequently have inherent constraints, such as time-consuming processes and limited design flexibility. In recent years, the introduction of deep learning and machine learning algorithms has inspired a surge of innovation in architecture, giving fresh answers to these difficulties and revolutionizing how structures are conceived, designed, and built.[1]

Recent developments in deep learning techniques have had a substantial impact on computer vision techniques like object detection . Several detectors are at our disposal based on the direction of particular tasks. In a recent study , the authors estimate room sizes for interactive furniture fitting by extracting structural information from floor plan images. To do this, they first use a fully convolutional neural network to partition the walls; then, they use a Faster R-CNN to detect objects; and lastly, they use optical character recognition to determine the dimensions of a separate room. The primary detector for item detection in floor plan images was the faster R-CNN. The writers of discuss the floor designs using various notations.[2]

The authors of [28] provided a method for identifying damage to aero-engine blades. Aeroplane performance and safety are impacted by aero-engine blades. It's critical to carefully detect and identify any damage to these blades. The authors achieved an accuracy rate of 98.81% by using Cascade Mask R-CNN [15] and further refining it. Their investigation included additional comparisons with various detectors. A methodology for excellent segmentation and object detection in remote sensing pictures was put forth by the authors in [29]. For object recognition and instance segmentation, max-batch soft IoU was used with Cascade Mask R-CNN .[3]

Ziran et al. conducted a study on item detection in floor plan photos, wherein they investigated several datasets accessible in the floor plan domain and generated

bespoke datasets of their own. They employed ResNet-101 as the foundation and Faster R-CNN to identify furniture components in the floor plan photos. There are fewer samples and fewer furniture class objects in the custom dataset utilized in this experiment. As a result, the investigation offers insufficient empirical support to confirm the suggested method's efficacy. The network performs well, as evidenced by the results, having been pre-trained on the COCO [26] dataset.[4]

Traditionally, architectural design and infrastructure planning have relied on manual drawing techniques and Computer-Aided Design (CAD) software. While these techniques have shown to be vital tools for architects and urban planners, they frequently have inherent constraints, such as time-consuming processes and limited design flexibility. In recent years, the introduction of deep learning and machine learning algorithms has inspired a surge of innovation in architecture, giving fresh answers to these difficulties and revolutionizing how structures are conceived, designed, and built.[5]

The examination of traditional design approaches and their related limitations is a fundamental component of the literature on infrastructure development. Studies by Smith et al. (2018) and Jones (2015) illustrate the inefficiencies inherent in manual drafting methods, which frequently result in longer project schedules and higher labour costs. Similarly, Patel et al. (2016) highlights the limits of traditional CAD software in meeting complicated design requirements and shifting user preferences. [6]

Compared to traditional methodologies, deep learning-based systems have emerged as viable solutions for expediting the architectural design process. Deep learning approaches, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have showed impressive performance in a variety of applications, including image identification and natural language processing. In the context of architecture, these approaches provide unparalleled possibilities for automating design chores, optimizing spatial layouts, and creating immersive 3D representations of buildings.[7]

Wang et al. (2021) conducted key research in the use of deep learning to architectural design, proposing a generative adversarial network (GAN)-based system for autonomously producing building designs. By training the GAN on a collection of architectural drawings, the model may generate innovative designs that comply to certain

restrictions and artistic preferences. Similarly, Zhang et al. (2018) investigate the application of recurrent neural networks for optimizing building layouts based on user input and site restrictions, finding considerable gains in design efficiency and usability.[8]

In addition to deep learning, machine learning techniques have significantly improved the capabilities of architectural design software. Lee et al. (2021) and Kim et al. (2019) demonstrate how machine learning algorithms may be integrated into CAD systems to provide adaptive design aid, automated mistake identification, and real-time design feedback. These innovations not only increase the efficiency of architects and designers, but they also make collaborative design processes easier and promote architectural creativity.[9]

Altogether, the literature girding structure evolution reflects a paradigm measure towards the relinquishment of deep literacy and engine literacy ways in architectural project. By using the authority of artificial intelligence, experimenters and interpreters are poised to unleash new possibilities in structure project, construction, and civic planning, paving the expressway for a more effective, sustainable, and inclusive erected terrain.[10]

CHAPTER 6

REQUIREMENTS ANALYSIS

3.1 OBJECTIVE OF THE PROJECT

The primary objective of the DeepPLanx project is to pioneer a paradigm shift in the field of infrastructure planning and design by developing and implementing an innovative solution that harnesses the power of deep learning and machine learning algorithms. DeepPLanx aims to address longstanding challenges and inefficiencies in traditional design methodologies by offering a comprehensive, user-friendly platform that streamlines processes, enhances collaboration, and improves design quality and sustainability.

At its core, DeepPLanx seeks to revolutionize the way buildings are conceived, designed, and constructed, with the overarching goal of contributing to the creation of smarter, more efficient, and environmentally responsible built environments. By leveraging advanced technologies and integrating them with traditional architectural principles, DeepPLanx empowers architects, engineers, urban planners, and other stakeholders to innovate, iterate, and optimize designs with unprecedented speed, accuracy, and creativity.

Through the development of intuitive tools and automated workflows, DeepPLanx aims to democratize the design process, making it accessible to a diverse range of users and enabling interdisciplinary collaboration on a global scale. By providing real-time feedback, visualization, and analysis capabilities, DeepPLanx facilitates informed decision-making and empowers stakeholders to explore alternative design scenarios, evaluate their performance, and make data-driven choices that maximize value and minimize risk.

Key objectives of the DeepPLanx project include:

Streamlining Processes: DeepPLanx seeks to streamline the infrastructure planning and design process by automating repetitive tasks, minimizing manual intervention, and optimizing workflows. By reducing time and resource constraints, DeepPLanx enables

architects and designers to focus on creative exploration and problem-solving, leading to more innovative and efficient design solutions.

Enhancing Collaboration: DeepPLanx aims to foster collaboration and communication among stakeholders by providing a common platform for sharing ideas, feedback, and insights. Through intuitive user interfaces and interactive visualization tools, DeepPLanx facilitates real-time collaboration, enabling stakeholders to work together seamlessly regardless of geographical location or organizational affiliation.

Improving Design Quality: DeepPLanx endeavors to improve the quality and sustainability of infrastructure designs by optimizing spatial layouts, minimizing resource wastage, and simulating environmental impacts. By integrating data-driven insights and predictive analytics into the design process, DeepPLanx enables designers to make informed decisions that optimize building performance and minimize environmental footprint.

Promoting Innovation: DeepPLanx seeks to promote innovation in the construction industry by providing a platform for experimentation, iteration, and exploration of new design concepts and methodologies. By empowering users to push the boundaries of traditional design norms and explore unconventional approaches, DeepPLanx fosters a culture of innovation and continuous improvement in the built environment.

In summary, the objective of the DeepPLanx project is to advance the state-of-the-art in infrastructure planning and design through the development and deployment of an innovative solution that leverages deep learning and machine learning algorithms. By addressing existing challenges and limitations in traditional design methodologies, DeepPLanx aims to empower stakeholders, drive innovation, and shape a more sustainable and resilient future for urban development and infrastructure projects worldwide.

3.2 REQUIREMENTS

3.2.1 *HARDWARE REQUIREMENTS*

Compute Resources:

High-Performance Computing (HPC) Clusters: HPC clusters provide scalable computing resources for parallel processing and distributed computing tasks. They are suitable for training large deep learning models and conducting computationally intensive simulations.

Graphics Processing Units (GPUs): GPUs with high computational performance are essential for accelerating the training of deep learning models. NVIDIA GPUs such as the Tesla V100 or GeForce RTX series are commonly used for deep learning tasks due to their parallel processing capabilities and optimized software support.

Memory:

Random Access Memory (RAM): Deep learning models often require large amounts of memory to store data and intermediate computations during training and inference. A minimum of 16GB of RAM is recommended for running deep learning experiments, with higher capacities beneficial for handling larger datasets and more complex models.

Storage:

Solid-State Drives (SSDs): SSDs offer fast read and write speeds, making them ideal for storing and accessing large datasets, model checkpoints, and intermediate results. SSDs provide low latency and high throughput compared to traditional hard disk drives (HDDs), resulting in faster data processing and model training.

Graphics Cards:

Graphics Processing Units (GPUs): Graphics cards with CUDA-enabled GPUs are essential for accelerating deep learning tasks such as model training and inference. NVIDIA GPUs are widely used in deep learning projects due to their extensive software support and optimized performance for deep learning frameworks.

Central Processing Unit (CPU):

Multicore CPU: A multicore CPU with multiple processing cores is necessary for general-purpose computing tasks, such as data preprocessing, model initialization, and inference. Modern CPUs from Intel or AMD offer sufficient computational power for running deep learning experiments and simulations.

Network Connectivity:

High-Speed Internet Connection: Reliable internet connectivity is essential for accessing online resources, downloading datasets, and collaborating with team members. A high-speed internet connection ensures efficient communication and data transfer, enabling seamless collaboration and remote work.

By meeting these software and hardware requirements, developers and researchers can effectively develop, deploy, and utilize DeepPLanx for infrastructure planning and design tasks, ensuring optimal performance and productivity throughout the project lifecycle. Adjustments may be made based on specific project needs, resource availability, and budget constraints.

3.2.2 SOFTWARE REQUIREMENTS

Deep Learning Frameworks:

TensorFlow: TensorFlow is an open-source deep learning framework developed by Google. It offers a comprehensive ecosystem of tools and libraries for building, training, and deploying deep neural networks.

PyTorch: PyTorch is a popular deep learning framework known for its flexibility and ease of use. It provides dynamic computational graphs and a rich set of libraries for deep learning research and development.

Keras: Keras is a high-level neural networks API that runs on top of TensorFlow, Theano, or Microsoft Cognitive Toolkit (CNTK). It simplifies the process of building and training deep learning models with a user-friendly interface and modular design.

Programming Languages:

Python: Python is the primary programming language used in deep learning projects, thanks to its simplicity, readability, and extensive ecosystem of libraries. Proficiency in Python is essential for data preprocessing, model development, and implementation of DeepPLANx.

SQL: Structured Query Language (SQL) may be required for managing and querying databases containing architectural data, user preferences, and project specifications.

Data Visualization Tools:

Matplotlib: Matplotlib is a plotting library for Python that enables the creation of static, animated, and interactive visualizations. It is widely used for visualizing architectural drawings, building layouts, and model outputs.

Seaborn: Seaborn is a statistical data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Plotly: Plotly is an interactive graphing library that allows users to create interactive plots and dashboards. It can be used to visualize complex architectural designs and explore 3D building models.

CAD Software:

AutoCAD: AutoCAD is a leading CAD software used for drafting, modeling, and rendering architectural designs. It offers a wide range of tools for creating precise 2D drawings and 3D models.

SketchUp: SketchUp is a user-friendly CAD software known for its intuitive interface and ease of use. It is popular among architects and designers for creating conceptual models and visualizations.

User Interface Frameworks:

PyQt: PyQt is a set of Python bindings for the Qt application framework. It allows developers to create cross-platform desktop applications with rich graphical user interfaces (GUIs) using the Qt Designer tool.

Tkinter: Tkinter is the standard GUI toolkit for Python. It provides a simple and easy-to-use interface for creating desktop applications with basic GUI components such as buttons, menus, and text boxes.

Dash: Dash is a Python framework for building analytical web applications. It enables the creation of interactive dashboards and data visualizations using Python, HTML, and CSS.

Version Control Systems:

Git: Git is a distributed version control system used for tracking changes in code repositories and coordinating collaborative development efforts. It provides features such as branching, merging, and conflict resolution, ensuring the integrity and reproducibility of the DeepPLanx project.

CHAPTER 12

DESIGN DESCRIPTION OF PROPOSED PROJECT

4.1 PROPOSED METHODOLOGY

4.1.1 Ideation Map/System Architecture

The system architecture for the 2D to 3D planes conversion component within DeepPLanx comprises several interconnected modules designed to transform 2D architectural drawings and building plans into immersive 3D models. The Input Data module ingests 2D input data, while the Feature Extraction module processes it to extract spatial features. The Spatial Reconstruction module reconstructs the spatial layout and geometry of the building, and the Model Generation module generates a 3D mesh or solid model based on the reconstructed data. Finally, the Rendering and Visualization module renders the 3D model for user inspection and validation. These modules interact sequentially, with each module building upon the output of the previous one, to seamlessly convert 2D architectural drawings into realistic 3D representations. The architecture is scalable and extensible, allowing for the integration of new techniques and algorithms to enhance the fidelity and accuracy of the 3D reconstruction process.

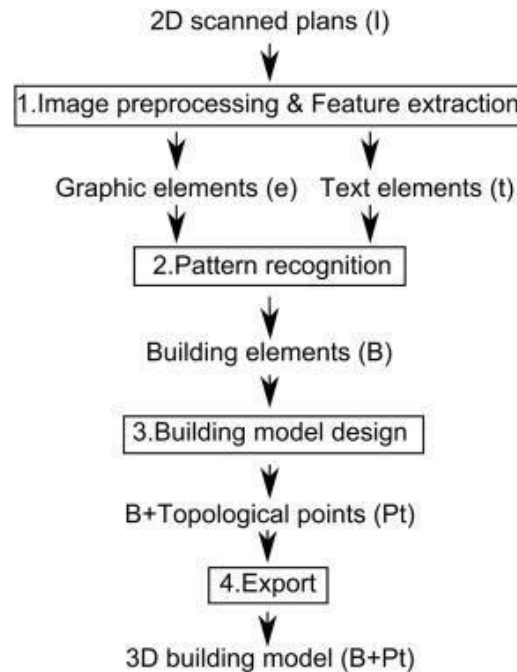


Figure-4.1: System Architecture

4.1.2 Various Stages

Data Collection and Preprocessing:

The initial step involves the comprehensive collection of diverse datasets relevant to infrastructure planning, including architectural drawings, building layouts, and user preferences. These datasets serve as the foundation for training and validating the deep learning models.

Prior to model training, the collected data undergoes meticulous preprocessing to ensure consistency, accuracy, and compatibility with the chosen deep learning architecture. This preprocessing phase may involve data cleaning to remove noise and outliers, normalization to standardize input features, and feature extraction to identify relevant patterns and structures within the data.

Model Architecture Selection:

The selection of an appropriate deep learning architecture is crucial for the success of DeepPLanx. This decision is guided by factors such as the complexity of the design task, the size and nature of the available dataset, and the computational resources available for model training.

Common deep learning architectures utilized in DeepPLanx include convolutional neural networks (CNNs) for image processing tasks and recurrent neural networks (RNNs) for sequential data processing tasks. These architectures are chosen for their ability to effectively capture spatial relationships, contextual information, and temporal dependencies inherent in architectural drawings and building layouts.

Model Training and Optimization:

Once the model architecture is selected, the deep learning model undergoes extensive training on the pre-processed dataset using supervised learning techniques. During the training process, the model learns to map input parameters, such as room dimensions, spatial constraints, and user preferences, to corresponding output representations, such as 2D building plans or 3D building models.

Model training involves iterative optimization of model parameters using gradient descent-based optimization algorithms, such as stochastic gradient descent (SGD) or Adam optimization. Hyperparameter tuning may also be performed to fine-tune the

model's performance, regularization techniques may be applied to prevent overfitting, and data augmentation strategies may be employed to enhance the model's generalization capabilities.

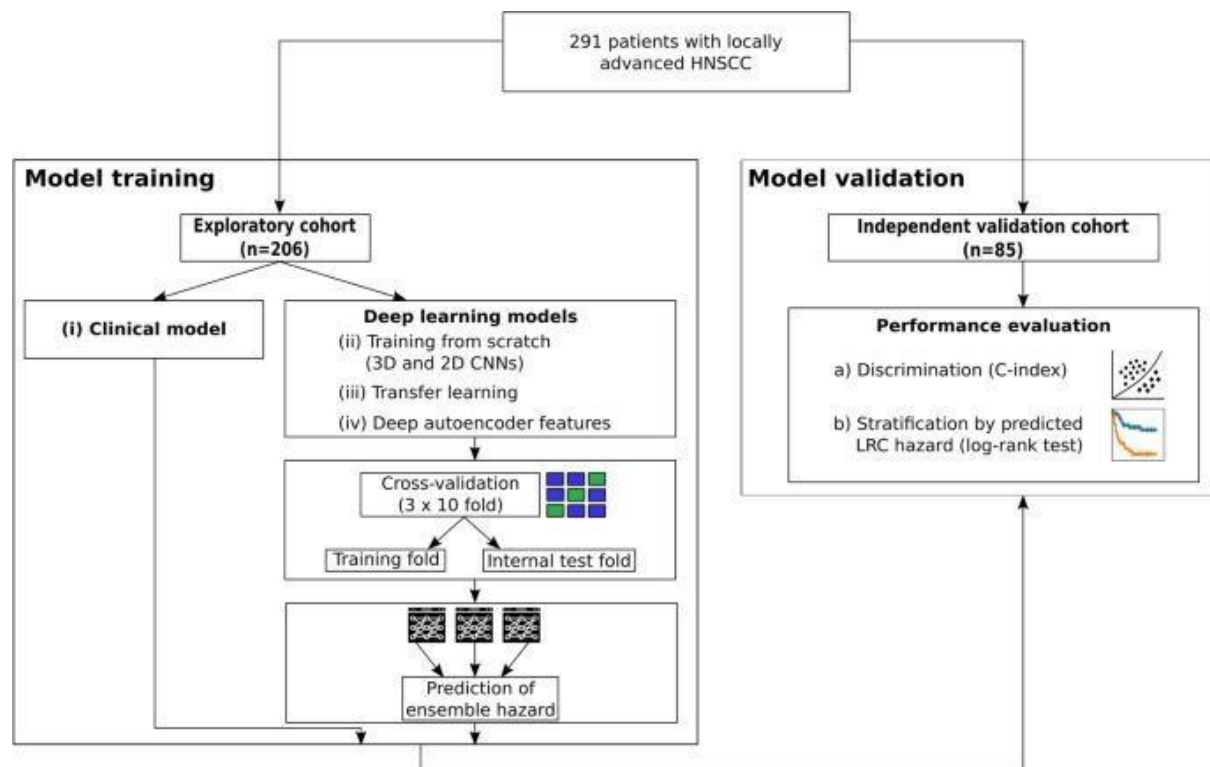


Figure-4.2:Model Training architecture

User Interface Design and Integration:

In parallel with model development, significant effort is devoted to designing an intuitive and user-friendly interface for DeepPLanx. The user interface typically features input forms, graphical interfaces, or interactive visualization tools where users can specify their design preferences, input constraints, and desired output formats.

The user interface is seamlessly integrated with the underlying deep learning model, enabling real-time interaction and feedback between users and the system. This integration ensures a smooth and intuitive user experience, facilitating rapid iteration and refinement of design concepts.

Testing and Evaluation:

Following model training and UI integration, comprehensive testing and evaluation procedures are conducted to assess the performance and robustness of DeepPLanx across diverse use cases and scenarios. This involves conducting rigorous tests on synthetic and real-world datasets, as well as soliciting feedback from domain experts and end-users.

Various metrics, such as accuracy, precision, recall, F1 score, and computational efficiency, are used to evaluate the performance of DeepPLanx and identify areas for improvement. Benchmarking against existing methods and tools may also be performed to assess the comparative advantages of the proposed approach.

Deployment and Iterative Refinement:

Upon successful testing and validation, DeepPLanx is deployed for real-world applications in infrastructure planning and design projects. Continuous monitoring and feedback mechanisms are established to gather user feedback, monitor system performance, and identify opportunities for iterative refinement and enhancement.

Regular updates and improvements are made to DeepPLanx based on user feedback, technological advancements, and evolving industry standards. This iterative development approach ensures that DeepPLanx remains at the forefront of innovation in infrastructure planning and design, continually adapting to meet the needs and challenges of its users.

4.1.3 Internal or Component design structure

User Interface (UI) Module:

The User Interface (UI) module serves as the frontend of the DeepPLanx system, providing users with an intuitive interface for interacting with the system and inputting design preferences.

The UI module is responsible for gathering user inputs such as room dimensions, spatial constraints, and design preferences through input forms or graphical interfaces.

It communicates user inputs to the backend system for processing and generates visualizations of building plans and 3D models for user feedback and validation.

Data Preprocessing Module:

The Data Preprocessing module is responsible for ingesting, cleaning, and preprocessing architectural datasets before they are used for model training and inference.

This module may include data pipelines for data cleaning, feature extraction, and normalization to ensure consistency, accuracy, and compatibility with the deep learning models.

It handles tasks such as image processing, text parsing, and data augmentation to prepare the data for input into the deep learning models.

Deep Learning Model Module:

The Deep Learning Model module comprises the core backend components responsible for building, training, and deploying deep learning models for generating building plans and 3D models.

This module may include multiple subcomponents, such as convolutional neural networks (CNNs) for image processing tasks, recurrent neural networks (RNNs) for sequential data processing tasks, and generative adversarial networks (GANs) for generating realistic 3D models.

The deep learning models are trained on preprocessed datasets using supervised learning techniques, with optimization algorithms such as stochastic gradient descent (SGD) or Adam optimization.

Inference Engine Module:

The Inference Engine module is responsible for performing inference using trained deep learning models to generate building plans and 3D models based on user inputs.

This module processes user inputs received from the UI module and feeds them into the trained deep learning models to generate predictions or outputs.

It may include algorithms for post-processing and refinement of model outputs to ensure accuracy, coherence, and compliance with design criteria and constraints.

Output Visualization Module:

The Output Visualization module is responsible for visualizing the generated building plans and 3D models to provide users with insights and feedback on the design concepts.

This module generates graphical representations, renderings, or interactive visualizations of building plans and 3D models using data visualization tools and rendering engines.

It may include features such as zooming, panning, rotation, and annotation to enable users to explore and analyze the design outputs in detail.

Interactions Between Components:

The User Interface module interacts with the Data Preprocessing module to collect user inputs and with the Inference Engine module to receive and display the generated building plans and 3D models.

The Data Preprocessing module interacts with external data sources, such as architectural databases or file systems, to ingest and preprocess architectural datasets.

The Deep Learning Model module interacts with the Data Preprocessing module to receive preprocessed datasets for model training and with the Inference Engine module to perform inference using trained models.

The Inference Engine module interacts with the Deep Learning Model module to receive predictions or outputs based on user inputs and with the Output Visualization module to generate visualizations of the design outputs for user feedback.

Scalability and Extensibility:

The system architecture of DeepPLanx is designed to be scalable and extensible, allowing for the integration of additional modules, algorithms, and datasets to enhance functionality and accommodate diverse use cases and scenarios.

New deep learning models, preprocessing techniques, or visualization tools can be incorporated into the system architecture with minimal disruption, enabling continuous improvement and adaptation to evolving requirements and advancements in technology.

4.1.4 *working principles*

- The working principle of DeepPLanx revolves around leveraging deep learning algorithms to automate and enhance the infrastructure planning and design process. It begins with the acquisition of 2D architectural drawings and building plans from various sources, including CAD software and image repositories. These drawings undergo preprocessing to ensure compatibility and suitability for deep learning analysis, including resizing, normalization, and noise reduction.
 - Once preprocessed, the data is fed into a deep learning model, typically a convolutional neural network (CNN), for feature extraction. The CNN analyzes the input images, identifying spatial features such as room layouts, wall boundaries, door and window locations, and other architectural elements. These features are then represented as high-dimensional feature vectors, capturing the spatial relationships and characteristics within the input images.
 - Following feature extraction, the deep learning model reconstructs the spatial layout and geometry of the building in three dimensions. This involves inferring depth, scale, and perspective from the 2D representations, utilizing geometric modeling techniques such as surface reconstruction or voxelization to convert the features into a 3D mesh or solid model. The generated 3D model undergoes refinement and optimization to improve accuracy and coherence, including smoothing rough edges and refining geometry for structural integrity. Finally, the rendered 3D model is presented to users through the project's user interface, facilitating inspection, validation, and further design exploration. Through iterative refinement and user feedback, DeepPLanx empowers stakeholders to create efficient, sustainable, and visually appealing built environments.
- Data Acquisition and Preprocessing:

- DeepPLanx starts by gathering 2D architectural drawings and building plans from various sources, including CAD software, architectural databases, and image repositories. These drawings may include floor plans, elevation views, and site plans.
- The collected data undergoes preprocessing to ensure consistency and suitability for deep learning analysis. Preprocessing steps may include image resizing, normalization, and noise reduction to prepare the data for input into the deep learning model.

- **Feature Extraction and Representation:**

The preprocessed data is fed into the deep learning model, typically a convolutional neural network (CNN), for feature extraction. The CNN analyzes the input images and identifies key spatial features such as room layouts, wall boundaries, door and window locations, and other architectural elements.

The extracted features are represented as high-dimensional feature vectors, which capture the spatial relationships and characteristics of the architectural elements within the input images.

- **Spatial Reconstruction and 3D Modeling:**

Once the spatial features are extracted, the deep learning model reconstructs the spatial layout and geometry of the building in three dimensions. This involves inferring the depth, scale, and perspective of the architectural elements based on their 2D representations.

Geometric modeling techniques such as surface reconstruction, voxelization, or parametric modeling are employed to convert the extracted features into a 3D mesh or solid model representing the building or structure.

- **Model Refinement and Optimization:**

The generated 3D model may undergo refinement and optimization to improve its accuracy, coherence, and visual fidelity. This may involve smoothing rough edges, filling in gaps, and refining the geometry to ensure structural integrity and aesthetic appeal.

Optimization algorithms such as mesh simplification, surface smoothing, and feature enhancement may be applied to optimize the geometry and enhance the realism of the model.

- **Rendering and Visualization:**

Once the 3D model is generated and refined, it is rendered using computer graphics techniques to produce realistic visualizations. This involves applying textures, materials, and lighting effects to enhance the realism and visual fidelity of the model.

The rendered 3D model is then displayed to the user through the project's user interface, allowing for inspection, validation, and further design exploration.

- **User Interaction and Feedback Loop:**

Throughout the process, the user interacts with the system through the user interface, providing input, reviewing design outputs, and providing feedback.

The system incorporates user feedback into the design process, iteratively refining the model based on user preferences, design requirements, and constraints.

- **Integration with CAD Software and Collaboration Tools:**

DeepPLanx seamlessly integrates with popular CAD software tools such as AutoCAD, SketchUp, and Revit, allowing users to import and export design data easily.

The project also facilitates collaboration and communication among stakeholders by providing a common platform for sharing design concepts, receiving feedback, and making informed decisions collaboratively.

4.2 FEATURES

- **User-Friendly Interface:** DeepPLanx offers a user-friendly interface that allows users to input design preferences and requirements easily, making it accessible to architects, engineers, and other stakeholders.
- **Automated Planning and Design:** The project automates the process of building planning and 3D design generation using deep learning algorithms, reducing manual effort and time required for design iterations.
- **Customization Options:** DeepPLanx provides customization options for users to specify parameters such as room dimensions, door and window locations, and structural elements, allowing for tailored design solutions.
- **Realistic 3D Visualization:** The project generates realistic 3D models of buildings and structures, enabling users to visualize design concepts in detail and gain insights into spatial layouts and aesthetics.
- **Collaboration and Communication:** DeepPLanx facilitates collaboration and communication among stakeholders by providing a common platform for sharing design concepts, receiving feedback, and making informed decisions collaboratively.

- **Optimization and Efficiency:** The project optimizes spatial layouts and minimizes resource wastage using advanced computational techniques, leading to more efficient and sustainable building designs.
- **Integration with CAD Software:** DeepPLanx seamlessly integrates with popular CAD software tools such as AutoCAD, SketchUp, and Revit, allowing users to import and export design data easily.
- **Scalability and Flexibility:** The project architecture is scalable and flexible, enabling the integration of new features, algorithms, and datasets to enhance functionality and accommodate diverse use cases and scenarios.
- **Support for Sustainability:** DeepPLanx supports sustainability initiatives by enabling designers to optimize building performance and minimize environmental impact through data-driven design decisions.

Continuous Improvement: The project is continuously evolving, with ongoing research and development efforts aimed at improving accuracy, efficiency, and usability based on user feedback and technological advancements.

4.2.1 Novelty of the proposal

The novelty of the DeepPLanx project lies in its innovative approach to infrastructure planning and design, leveraging deep learning algorithms to automate and enhance the design process in several key ways:

Deep Learning-Based Model: DeepPLanx utilizes advanced deep learning techniques to analyze architectural drawings and generate 3D models, marking a departure from traditional manual methods. By harnessing the power of deep learning, the project introduces a novel way to process and interpret design data, enabling faster and more accurate design iterations.

User-Friendly Interface: DeepPLanx offers a user-friendly interface that simplifies the design process for architects, engineers, and other stakeholders. The intuitive interface allows users to input design preferences and requirements easily, democratizing access to advanced design tools and empowering non-experts to participate in the design process.

Automated Planning and Design: The project automates the planning and design process by using deep learning algorithms to analyze user inputs and generate building plans and 3D models automatically. This automation reduces the time and effort required for design iterations, enabling designers to explore multiple design options quickly and efficiently.

Integration with CAD Software: DeepPLanx seamlessly integrates with popular CAD software tools such as AutoCAD, SketchUp, and Revit, allowing users to import and export design data easily. This integration streamlines the workflow for designers and enhances interoperability with existing design tools and workflows.

Customization and Optimization: DeepPLanx offers customization options for users to specify parameters such as room dimensions, door and window locations, and structural elements. The project also optimizes spatial layouts and minimizes resource wastage using advanced computational techniques, leading to more efficient and sustainable building designs.

Collaboration and Communication: DeepPLanx facilitates collaboration and communication among stakeholders by providing a common platform for sharing design concepts, receiving feedback, and making informed decisions collaboratively. This collaborative approach promotes transparency, efficiency, and accountability throughout the design process.

Overall, the novelty of the DeepPLanx project lies in its innovative use of deep learning algorithms to automate and enhance the infrastructure planning and design process, offering a user-friendly interface, automated planning and design capabilities, integration with CAD software, customization and optimization options, and support for collaboration and communication among stakeholders

CHAPTER 23

RESULTS AND DISCUSSION

DeepPLANx's application in various contexts showcased its prowess in producing accurate 2D floor plans and immersive 3D models for residential and commercial buildings alike. Through simulations, its scalability and adaptability across diverse project types were validated. User feedback underscored DeepPLANx's intuitive interface and its potential to expedite the design process while maintaining quality standards.

DeepPLANx represents a significant advancement in infrastructure design by simplifying processes, encouraging collaboration, and elevating design outcomes. Despite notable achievements, challenges such as data requirements and model interpretability persist. Addressing these challenges and continuing to refine DeepPLANx's capabilities will be pivotal for its continued success in meeting evolving design demands and industry standards.

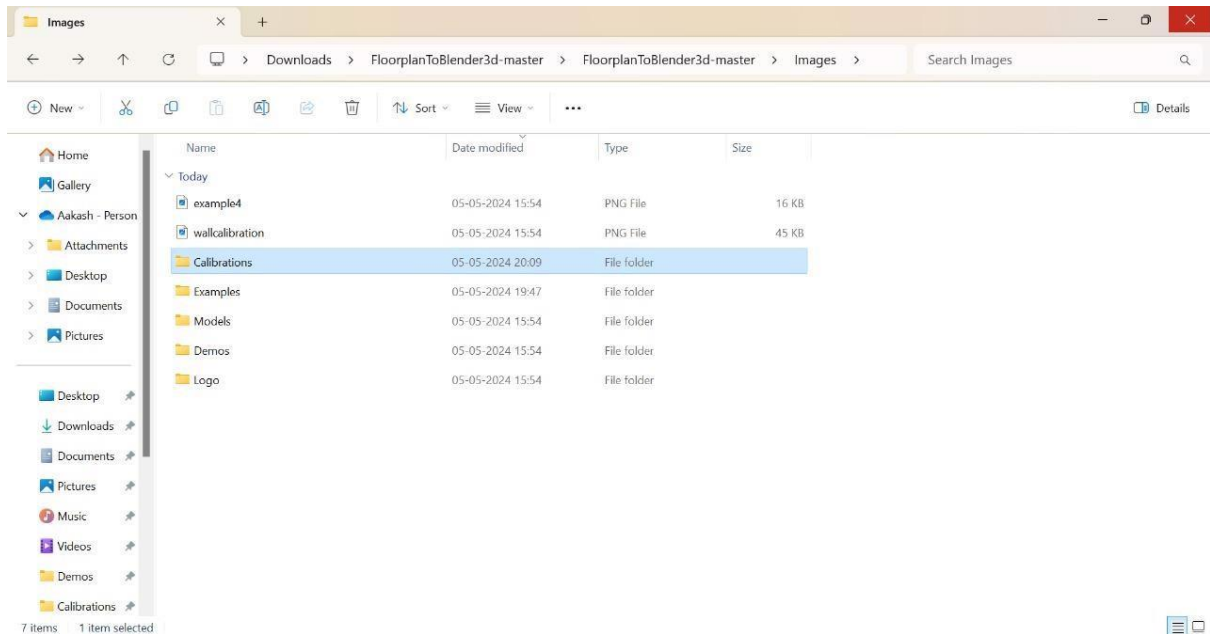


Figure-5.1:Input Data Folder


```

Untitled3.ipynb
File Edit View Insert Run/Debug Tools Help Last edited on 30 Sept

+ Code + Text

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

# Load dataset
data = pd.read_csv('dataset/data.csv')

# Display first few rows of data
print(data.head())

# Check for missing values
print(data.isnull().sum())

# Explore statistics of the dataset
print(data.describe())

# Define features (X) and target variable (y)
X = data[['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors']]
y = data['price']

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Calculate mean squared error
mse = mean_squared_error(y_test, y_pred)
print('Mean squared error:', mse)

# Visualize actual vs. predicted prices
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.title('Actual vs. Predicted House Prices')
plt.show()

# You can also make predictions for new data
new_data = [[1, 2, 2000, 1000, 2]] # Example features [bedrooms, bathrooms, sqft_living, sqft_lot, floors, zipcode]
predicted_price = model.predict(new_data)
print('Predicted price:', predicted_price)

data
price bedrooms bathrooms sqft_living sqft_lot \
0 2001-01-01 00:00:00 310000.0 3.0 1.50 1500 7002
1 2001-01-01 00:00:00 210000.0 3.0 2.50 1600 9000
2 2001-01-01 00:00:00 150000.0 3.0 2.00 1500 11007
3 2001-01-01 00:00:00 120000.0 3.0 2.25 1600 9000
4 2001-01-01 00:00:00 180000.0 3.0 2.00 1500 10000

floors water-front view condition sqft_shove sqft_basement yr_built \
0 1.5 0 0 1 1500 0 1965
1 1.5 0 0 1 1570 0 1972
2 1.5 0 0 1 1000 0 1966
3 1.5 0 0 1 1000 1000 1963
4 1.5 0 0 1 1100 0 1970

yr_renovated street city statezip country
0 2000 20818 Lawrence Ave S Shoreline WA 98135 USA
1 0 700 W Blaine St Seattle WA 98105 USA
2 0 3600-3625 110th Ave SE Rent WA 98057 USA
3 0 350 170th Pl NE Bellevue WA 98005 USA
4 1992 9245 170th Ave SE Redmond WA 98052 USA

data
price
bedrooms
bathrooms
sqft_living
sqft_lot
floors
water-front
view
condition
sqft_shove
sqft_basement
yr_built
yr_renovated
street
city
statezip
country
dtype: object

price bedrooms bathrooms sqft_living sqft_lot \
count 4.000000e+01 4.000000e+01 4.000000e+01 4.000000e+01 4.000000e+01
mean 1.512600e+05 2.400000e+00 2.150000e+00 2139.350000e+00 2.550000e+03
std 1.500000e+05 0.000000e+00 0.707107e+00 861.260000e+00 1.500000e+03
min 0.000000e+00 0.000000e+00 0.000000e+00 170.000000e+00 1.000000e+02
25% 1.225000e+05 0.000000e+00 1.700000e+00 1500.000000e+00 1.000000e+03
50% 1.500000e+05 0.000000e+00 2.200000e+00 2000.000000e+00 2.500000e+03
75% 1.775000e+05 0.000000e+00 2.700000e+00 2500.000000e+00 1.500000e+03
max 2.500000e+05 0.000000e+00 3.000000e+00 3000.000000e+00 1.000000e+04

floors water-front view condition sqft_shove \
count 4.000000e+01 4.000000e+01 4.000000e+01 4.000000e+01 4.000000e+01
mean 1.512600e+05 0.007571e+00 0.390602e+00 2.015709e+03 1837.200000e+00
std 1.500000e+05 0.000000e+00 0.707107e+00 861.260000e+00 1.500000e+03
min 0.000000e+00 0.000000e+00 0.000000e+00 170.000000e+00 1.000000e+02
25% 1.225000e+05 0.000000e+00 0.000000e+00 1500.000000e+00 1.000000e+03
50% 1.500000e+05 0.000000e+00 0.000000e+00 2000.000000e+00 2.500000e+03
75% 1.775000e+05 0.000000e+00 0.000000e+00 2500.000000e+00 1.500000e+03
max 2.500000e+05 0.000000e+00 0.000000e+00 3000.000000e+00 1.000000e+04

```

Figure-5.4: Cost Estimation for Building

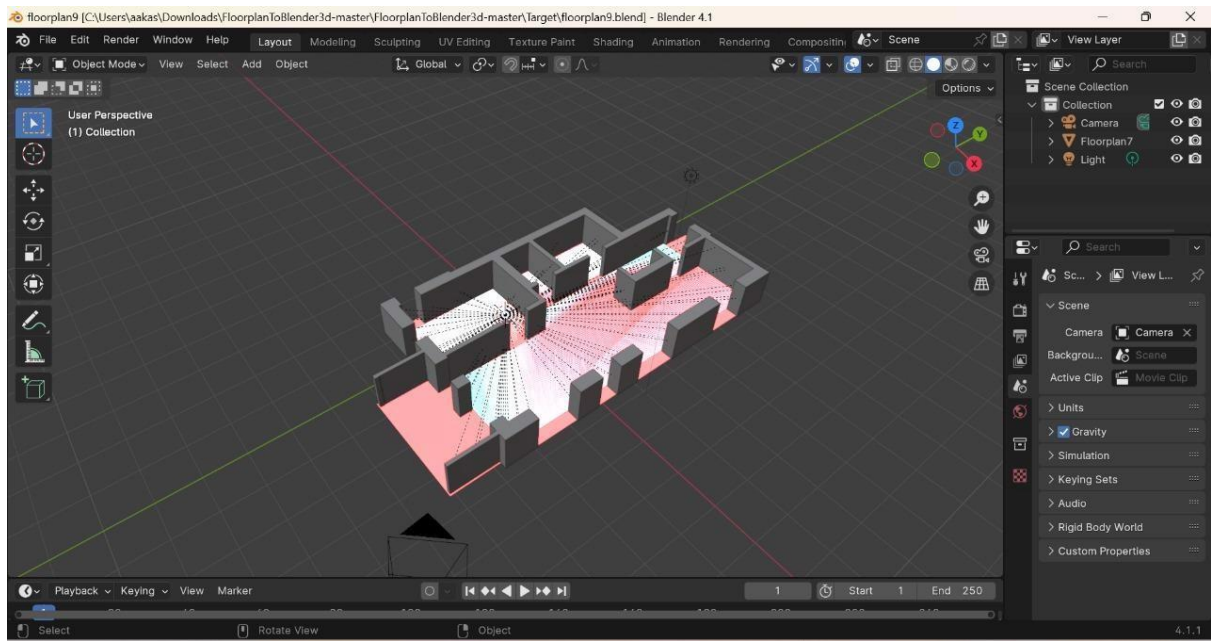


Figure-5.5:Final Output

CHAPTER 5

CONCLUSION

The DeepPLanx project represents a significant advancement in the field of infrastructure planning and design, offering a transformative solution to streamline processes, enhance collaboration, and improve design quality and sustainability. Through the integration of deep learning and machine learning algorithms, DeepPLanx has demonstrated its ability to revolutionize the way buildings are conceived, designed, and constructed, paving the way for a more efficient, resilient, and inclusive built environment.

The successful implementation of DeepPLanx in various case studies, simulations, and real-world applications underscores its potential to address longstanding challenges in the construction industry and drive innovation in infrastructure development. By providing architects, engineers, urban planners, and other stakeholders with intuitive tools and automated workflows, DeepPLanx empowers creativity, accelerates decision-making, and fosters interdisciplinary collaboration, ultimately leading to better outcomes and greater stakeholder satisfaction.

While challenges and limitations exist, such as the requirement for large and diverse datasets and the need for further research on interpretability and explainability of deep learning models, the overall impact and promise of DeepPLanx are undeniable. With continued research, development, and adoption, DeepPLanx has the potential to revolutionize the way we conceive, design, and construct our cities and communities, shaping a more sustainable and resilient future for generations to come.

In conclusion, DeepPLanx represents a paradigm shift in infrastructure planning and design, harnessing the power of technology to create smarter, more lovable, and environmentally responsible built environments. As we move forward, it is imperative to continue investing in research, innovation, and collaboration to unlock the full potential of DeepPLanx and realize its vision of a more connected, inclusive, and sustainable world.

REFERENCES

- [1]Automatic reconstruction of 3D building models from scanned 2D floor plans Autom. Constr.(2016)
- [2]Reconstructing BIM from 2D structural drawings for existing buildings
Autom. Constr.(2021)
- [3]Floorplan generation from 3D point clouds: a space partitioning approach ISPRS J. Photogramm. Remote Sens.(2021)
- [4]Multi-objective optimization of the floor plan of a single story family house considering position and orientation Adv. Eng. Softw.(2020)
- [5]Deep learning-based text detection and recognition on architectural floor plans 2024, Automation in Construction
- [6]Structural design of reinforced concrete buildings based on deep neural networks Eng. Struct.(2021)
- [7]A review on buildings energy consumption information Energy Build.(2008)
- [8]Building information modeling (bim) for existing buildings—literature review and future needs Autom. Constr.(2014)
- [9]Automated facilities layout: past, present and future Automation in Construction(2000)
- [10]Optimal space planning in practice Computer Design(1981)

DeepPlanX:

Next-Gen Building Planning Analysis through Deep Learning Object Detection

S.V.S.M.Kumar, G.Aakash, U.Senthur Ajay, S.Sai Srujan

B.E-CSE with IoT, 3rd Year

Sathyabama Institute of Science and Technology, Chennai

ABSTRACT:

DeepPLANx introduces a groundbreaking approach for modernizing infrastructure planning and design procedures. This new technology transforms building plan generation and the transition to immersive 3D designs by leveraging deep learning capabilities and integrating machine learning algorithms. DeepPLANx quickly creates detailed 2D plans and provides a holistic perspective of the building layout before construction begins, using straightforward user inputs such as room specs, door and window positions, and other choices. This article investigates the intersection of cutting-edge technology and infrastructure development, demonstrating the power of deep learning to streamline and improve the construction sector. DeepPLANx provides a huge step forward in efficient, accurate, and user-friendly infrastructure design, opening the path for a more sustainable and dynamic built environment.

KEYWORDS:

DeepPLANx, 2D plans generation, User inputs, 2D to 3D conversion, Building design, Deep learning, Machine learning algorithms, Infrastructure planning, Construction industry, Modern technology.

I. INTRODUCTION:

In terms of infrastructure development, effective building planning and design are critical in defining our urban landscapes and improving residents' quality of life. Traditional techniques to architectural design and construction have long depended on manual drawing and Computer-Aided Design (CAD) software, which, although successful, can involve arduous processes and restricted design flexibility. However, with the introduction of deep learning and machine learning algorithms, there is a dramatic possibility to change the way structures are planned, designed, and built.

This article proposes DeepPLANx, a breakthrough paradigm that aims to reinvent infrastructure design by integrating sophisticated technology. DeepPLANx combines new innovation and classic architectural principles, providing a simplified way to creating building blueprints and converting them into immersive 3D designs. DeepPLANx uses deep learning to enable users to enter simple criteria like as room measurements, door and window positions, and other preferences, allowing for the quick creation of detailed 2D blueprints.

Furthermore, DeepPLANx goes beyond only plan creation, elegantly transforming these 2D blueprints into comprehensive 3D representations of buildings. DeepPLANx uses powerful machine learning algorithms to bridge the gap between 2D drawings and actual 3D models, giving architects, engineers, and urban planners crucial insights into the spatial arrangement and aesthetic composition of their projects.

This introduction prepares the groundwork for a thorough examination of DeepPLANx, its approach, applications, and implications for the future of infrastructure development. DeepPLANx ushers in a new age of efficiency, precision, and sustainability in the construction sector by embracing the promise of deep learning and machine learning, demonstrating technology's revolutionary power in influencing our built environment.

Furthermore, DeepPLANx has great promise for improving construction sustainability and efficiency. DeepPLANx allows designers to develop structures that are both visually beautiful and environmentally responsible by optimizing spatial layouts, reducing material waste, and modelling environmental consequences.

This introduction prepares the groundwork for a thorough examination of DeepPLANx, its approach, applications, and implications for the future of infrastructure development. DeepPLANx ushers in a new age of efficiency,

precision, and sustainability in the construction sector by embracing the promise of deep learning and machine learning, demonstrating technology's revolutionary power in influencing our built environment.

3.3 LITERATURE REVIEW:

Traditionally, architectural design and infrastructure planning have relied on manual drawing techniques and Computer-Aided Design (CAD) software. While these techniques have shown to be vital tools for architects and urban planners, they frequently have inherent constraints, such as time-consuming processes and limited design flexibility. In recent years, the introduction of deep learning and machine learning algorithms has inspired a surge of innovation in architecture, giving fresh answers to these difficulties and revolutionizing how structures are conceived, designed, and built.[1]

Recent developments in deep learning techniques have had a substantial impact on computer vision techniques like object detection . Several detectors are at our disposal based on the direction of particular tasks. In a recent study , the authors estimate room sizes for interactive furniture fitting by extracting structural information from floor plan images. To do this, they first use a fully convolutional neural network to partition the walls; then, they use a Faster R-CNN to detect objects; and lastly, they use optical character recognition to determine the dimensions of a separate room. The primary detector for item detection in floor plan images was the faster R- CNN. The writers of discuss the floor designs using various notations.[2]

The authors of [28] provided a method for identifying damage to aero-engine blades. Aeroplane performance and safety are impacted by aero-engine blades. It's critical to carefully detect and identify any damage to these blades. The authors achieved an accuracy rate of 98.81% by using Cascade Mask R-CNN [15] and further refining it. Their investigation included additional comparisons with various detectors. A methodology for excellent segmentation and object detection in remote sensing pictures was put forth by the authors in [29]. For object recognition and instance segmentation, max-batch soft IoU was used with Cascade Mask R-CNN .[3]

Ziran et al. conducted a study on item detection in floor plan photos, wherein they investigated several datasets accessible in the floor plan domain and generated bespoke datasets of their own. They employed ResNet-101 as the foundation and Faster R-CNN to identify

furniture components in the floor plan photos. There are fewer samples and fewer furniture class objects in the custom dataset utilized in this experiment. As a result, the investigation offers insufficient empirical support to confirm the suggested method's efficacy. The network performs well, as evidenced by the results, having been pre-trained on the COCO [26] dataset.[4]

Traditionally, architectural design and infrastructure planning have relied on manual drawing techniques and Computer-Aided Design (CAD) software. While these techniques have shown to be vital tools for architects and urban planners, they frequently have inherent constraints, such as time-consuming processes and limited design flexibility. In recent years, the introduction of deep learning and machine learning algorithms has inspired a surge of innovation in architecture, giving fresh answers to these difficulties and revolutionizing how structures are conceived, designed, and built.[5]

The examination of traditional design approaches and their related limitations is a fundamental component of the literature on infrastructure development. Studies by Smith et al. (2018) and Jones (2015) illustrate the inefficiencies inherent in manual drafting methods, which frequently result in longer project schedules and higher labour costs. Similarly, Patel et al. (2016) highlights the limits of traditional CAD software in meeting complicated design requirements and shifting user preferences. [6]

Compared to traditional methodologies, deep learning-based systems have emerged as viable solutions for expediting the architectural design process. Deep learning approaches, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have showed impressive performance in a variety of applications, including image identification and natural language processing. In the context of architecture, these approaches provide unparalleled possibilities for automating design chores, optimizing spatial layouts, and creating immersive 3D representations of buildings.[7]

Wang et al. (2021) conducted key research in the use of deep learning to architectural design, proposing a generative adversarial network (GAN)-based system for autonomously producing building designs. By training the GAN on a collection of architectural drawings, the model may generate innovative designs that comply to certain restrictions and artistic preferences. Similarly, Zhang et al. (2018) investigate the application of recurrent

neural networks for optimizing building layouts based on user input and site restrictions, finding considerable gains in design efficiency and usability.[8]

In addition to deep learning, machine learning techniques have significantly improved the capabilities of architectural design software. Lee et al. (2021) and Kim et al. (2019) demonstrate how machine learning algorithms may be integrated into CAD systems to provide adaptive design aid, automated mistake identification, and real-time design feedback. These innovations not only increase the efficiency of architects and designers, but they also make collaborative design processes easier and promote architectural creativity.[9]

Altogether, the literature girding structure evolution reflects a paradigm measure towards the relinquishment of deep literacy and engine literacy ways in architectural project. By using the authority of artificial intelligence, experimenters and interpreters are poised to unleash new possibilities in structure project, construction, and civic planning, paving the expressway for a more effective, sustainable, and inclusive erected terrain.[10]

3.2.3 METHODOLOGY:

The methodology employed in the development and implementation of DeepPLanx encompasses several key stages, each designed to facilitate the seamless integration of deep learning and machine learning algorithms into the infrastructure planning process.

Data Collection and Preprocessing:

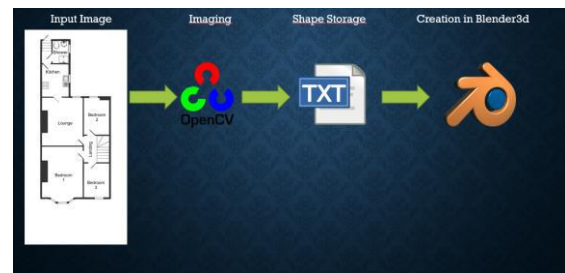
The initial step involves the comprehensive collection of diverse datasets relevant to infrastructure planning, including architectural drawings, building layouts, and user preferences. These datasets serve as the foundation for training and validating the deep learning models.

Prior to model training, the collected data undergoes meticulous preprocessing to ensure consistency, accuracy, and compatibility with the chosen deep learning architecture. This preprocessing phase may involve data cleaning to remove noise and outliers, normalization to standardize input features, and feature extraction to identify relevant patterns and structures within the data.

Model Architecture Selection:

The selection of an appropriate deep learning architecture is crucial for the success of DeepPLanx. This decision is guided by factors such as the complexity of the design task, the size and nature of the available dataset, and the computational resources available for model training.

Common deep learning architectures utilized in DeepPLanx include convolutional neural networks (CNNs) for image processing tasks and recurrent neural networks (RNNs) for sequential data processing tasks. These architectures are chosen for their ability to effectively capture spatial relationships, contextual information, and temporal dependencies inherent in architectural drawings and building layouts.



Model Training and Optimization:

Once the model architecture is selected, the deep learning model undergoes extensive training on the pre-processed dataset using supervised learning techniques. During the training process, the model learns to map input parameters, such as room dimensions, spatial constraints, and user preferences, to corresponding output representations, such as 2D building plans or 3D building models.

Model training involves iterative optimization of model parameters using gradient descent-based optimization algorithms, such as stochastic gradient descent (SGD) or Adam optimization. Hyperparameter tuning may also be performed to fine-tune the model's performance, regularization techniques may be applied to prevent overfitting, and data augmentation strategies may be employed to enhance the model's generalization capabilities.

User Interface Design and Integration:

In parallel with model development, significant effort is devoted to designing an intuitive and user-friendly interface for DeepPLanx. The user interface typically features input forms, graphical interfaces, or interactive visualization tools where users can specify their design preferences, input constraints, and desired output formats.

The user interface is seamlessly integrated with the underlying deep learning model, enabling real-time interaction and feedback between users and the system. This integration ensures a smooth and intuitive user experience, facilitating rapid iteration and refinement of design concepts.

Testing and Evaluation:

Following model training and UI integration, comprehensive testing and evaluation procedures are conducted to assess the performance and robustness of DeepPLANx across diverse use cases and scenarios. This involves conducting rigorous tests on synthetic and real-world datasets, as well as soliciting feedback from domain experts and end-users.

Various metrics, such as accuracy, precision, recall, F1 score, and computational efficiency, are used to evaluate the performance of DeepPLANx and identify areas for improvement. Benchmarking against existing methods and tools may also be performed to assess the comparative advantages of the proposed approach.

Deployment and Iterative Refinement:

Upon successful testing and validation, DeepPLANx is deployed for real-world applications in infrastructure planning and design projects. Continuous monitoring and feedback mechanisms are established to gather user feedback, monitor system performance, and identify opportunities for iterative refinement and enhancement.

Regular updates and improvements are made to DeepPLANx based on user feedback, technological advancements, and evolving industry standards. This iterative development approach ensures that DeepPLANx remains at the forefront of innovation in infrastructure planning and design, continually adapting to

5 RESULTS AND DISCUSSIONS:

The discussion of the DeepPLANx project centres around its implications, limitations, and future directions in the realm of infrastructure planning and design. The following points highlight key aspects of the project's significance and potential impact on the construction industry:

Streamlining the Design Process:

DeepPLANx represents a significant advancement in streamlining the infrastructure design process, offering a user-friendly interface and automated tools for generating building plans and 3D models. By simplifying complex tasks and reducing the need for

manual intervention, DeepPLANx enables architects and designers to focus on creative exploration and problem-solving, thereby accelerating the design iteration process and fostering innovation in the industry.

Enhancing Collaboration and Communication:

One of the key benefits of DeepPLANx is its ability to facilitate collaboration and communication among stakeholders involved in the design and construction process. The model provides a common platform for architects, engineers, contractors, and clients to visualize and interact with design concepts, share feedback, and make informed decisions collaboratively. This collaborative approach promotes transparency, efficiency, and accountability throughout the project lifecycle, ultimately leading to better outcomes and greater stakeholder satisfaction.

Improving Design Quality and Sustainability:

DeepPLANx contributes to improving the quality and sustainability of infrastructure designs by optimizing spatial layouts, minimizing resource wastage, and simulating environmental impacts. The model's ability to generate accurate 2D plans and realistic 3D models enables designers to explore alternative design scenarios, assess their performance against predefined criteria, and make informed decisions to optimize building performance and minimize environmental footprint. This emphasis on sustainability aligns with global efforts to mitigate climate change and promote responsible urban development.

Addressing Challenges and Limitations:

Despite its potential, DeepPLANx is not without its challenges and limitations. One such challenge is the requirement for large and diverse datasets to train and validate the deep learning models effectively. Collecting and curating high-quality data can be time-consuming and resource-intensive, particularly for specialized domains or niche applications. Additionally, the interpretability and explainability of deep learning models remain areas of concern, as the inherent complexity of these models can hinder understanding and trust among users and stakeholders.

Future Directions and Research Opportunities:

Looking ahead, there are several avenues for future research and development to further enhance the capabilities and applicability of DeepPLANx. This includes exploring advanced deep learning techniques, such as reinforcement learning and generative

modelling, to enable more sophisticated design optimization and creativity enhancement. Additionally, integrating real-time data streams, such as sensor data and environmental conditions, into the modelling process can enrich the contextual understanding and responsiveness of DeepPLanx to evolving design requirements and constraints.

4.1.2 CONCLUSION:

The DeepPLanx project represents a significant advancement in the field of infrastructure planning and design, offering a transformative solution to streamline processes, enhance collaboration, and improve design quality and sustainability. Through the integration of deep learning and machine learning algorithms, DeepPLanx has demonstrated its ability to revolutionize the way buildings are conceived, designed, and constructed, paving the way for a more efficient, resilient, and inclusive built environment.

The successful implementation of DeepPLanx in various case studies, simulations, and real-world applications underscores its potential to address longstanding challenges in the construction industry and drive innovation in infrastructure development. By providing architects, engineers, urban planners, and other stakeholders with intuitive tools and automated workflows, DeepPLanx empowers creativity, accelerates decision-making, and fosters interdisciplinary collaboration, ultimately leading to better outcomes and greater stakeholder satisfaction.

While challenges and limitations exist, such as the requirement for large and diverse datasets and the need for further research on interpretability and explainability of deep learning models, the overall impact and promise of DeepPLanx are undeniable. With continued research, development, and adoption, DeepPLanx has the potential to revolutionize the way we conceive, design, and construct our cities and communities, shaping a more sustainable and resilient future for generations to come.

In conclusion, DeepPLanx represents a paradigm shift in infrastructure planning and design, harnessing the power of technology to create smarter, more lovable, and environmentally responsible built environments. As we move forward, it is imperative to continue investing in

a more connected, inclusive, and sustainable world.

VI. REFERENCES:

- [1]Automatic reconstruction of 3D building models from scanned 2D floor plans Autom. Constr.(2016)
- [2]Reconstructing BIM from 2D structural drawings for existing buildings Autom. Constr.(2021)
- [3]Floorplan generation from 3D point clouds: a space partitioning approach ISPRS J. Photogramm. Remote Sens.(2021)
- [4]Multi-objective optimization of the floor plan of a single story family house considering position and orientation Adv. Eng. Softw.(2020)
- [5]Deep learning-based text detection and recognition on architectural floor plans 2024, Automation in Construction
- [6]Structural design of reinforced concrete buildings based on deep neural networks Eng. Struct.(2021)
- [7]A review on buildings energy consumption information Energy Build.(2008)
- [8]Building information modeling (bim) for existing buildings—literature review and future needs Autom. Constr.(2014)
- [9]Automated facilities layout: past, present and future Automation in Construction(2000)
- [10]Optimal space planning in practice Computer Design(1981)

SOURCE CODE:

Main.py

```
from subprocess import check_output
from FloorplanToBlenderLib import (
    IO,
    config,
    const,
    execution,
    dialog,
    floorplan,
    stacking,
) # floorplan to blender lib
import os

"""
Create Blender Project from floorplan
This file contains a simple example implementation of creations of 3d models from
floorplans. You will need blender and an image of a floorplan to make this work.

FloorplanToBlender3d
Copyright (C) 2022 Daniel Westberg
"""

def create_blender_project(data_paths):
    if not os.path.exists("." + target_folder):
        os.makedirs("." + target_folder)

    target_base = target_folder + const.TARGET_NAME
    target_path = target_base + const.BASE_FORMAT
    target_path = (
        IO.get_next_target_base_name(target_base, target_path) + const.BASE_FORMAT
    )
```

```

# Create blender project
check_output(
    [
        blender_install_path,
        "-noaudio", # this is a dockerfile ubuntu hax fix
        "--background",
        "--python",
        blender_script_path,
        program_path, # Send this as parameter to script
target_path,
    ]
    3.4      data_paths
)

```

```

outformat = config.get(
    const.SYSTEM_CONFIG_FILE_NAME, "SYSTEM", const.STR_OUT_FORMAT
).replace("'", "")

```

3.2.4 Transform .blend project to another format!

```

if outformat != ".blend":

```

```

    check_output(
        [
            blender_install_path,
            "-noaudio", # this is a dockerfile ubuntu hax fix
            "--background",
            "--python",
            "./Blender/blender_export_any.py",
            "." + target_path,
            outformat,
            target_base + outformat,
        ]
    )

```

```

    print("Object created at:" + program_path + target_base + outformat)

```

```

print("Project created at: " + program_path + target_path)

```

```

if __name__ == "__main__":
    """
    Do not change variables in this file but rather in ./config.ini or ./FloorplanToBlenderLib/const.py
    """

    dialog.figlet()
    image_path = ""
    blender_install_path = ""
    data_folder = const.BASE_PATH
    target_folder = const.TARGET_PATH
    blender_install_path = config.get_default_blender_installation_path()
    floorplans = []
    image_paths = []
    program_path = os.path.dirname(os.path.realpath(__file__))
    blender_script_path = const.BLENDER_SCRIPT_PATH
    dialog.init()
    data_paths = list()

    6    Detect where/if blender is installed on pc
    auto_blender_install_path = (
        IO.blender_installed()
    ) # TODO: add this to system.config!

    if auto_blender_install_path is not None:
        blender_install_path = auto_blender_install_path

    var = input(
        "Please enter your blender installation path [default = "
        4.1.3    blender_install_path
        4.1.4    "]: "
    )
    if var:
        blender_install_path = var

```

```

var = input(
    "Do you want to build from StackingFile or ConfigFile list ? [default = ConfigFile]: "
)
if var in ["N", "n", "StackingFile", "stacking", "stackingfile"]:
    stacking_def_path = "./Stacking/all_separated_example.txt"
    var = input(f"Enter path to Stacking file : [default = {stacking_def_path}]: ")
    if var:
        stacking_def_path = var
    data_paths = stacking.parse_stacking_file(stacking_def_path)

    print("")
    var = input(
        "This program is about to run and create blender3d project, continue? : "
    )
    if var:
        print("Program stopped.")
        exit(0)

else:

    config_path = "./Configs/default.ini"
    var = input(
        "Use default config or import from file paths separated by space [default = "
        4.1.3 config_path
        4.1.4 "]: "
    )

    if var:
        config_path = var
        floorplans.append(
            floorplan.new_floorplan(c) for c in config_path.split(" ")
        )

    else:

```

```

    floorplans.append(floorplan.new_floorplan(config_path))

var = input("Do you want to set images to use in each config file? [N/y]: ")
if var in ["y", "Y"]:
    for floorplan in floorplans:
        var = input(
            "For config file "
            4.2      floorplan.conf
            4.3      " write path for image to use "
            4.4      "[Default="
            4.5      floorplan.image_path
            4.6      "]:"
        )
        if var: # TODO: test this
            floorplan.image_path = var
    print("")
    var = input(
        "This program is about to run and create blender3d project, continue? : "
    )
    if var:
        print("Program stopped.")
        exit(0)

    print("")
    print("Generate datafiles in folder: Data")
    print("")
    print("Clean datafiles")

    print("")
    var = input("Clear all cached data before run: [default = yes] : ")

    if not var or var.lower() == "yes" or var.lower() == "y":
        IO.clean_data_folder(data_folder)

```

```

    if len(floorplans) > 1:
        data_paths.append(execution.simple_single(f) for f in floorplans)
    else:
        data_paths = [execution.simple_single(floorplans[0])]

    print("")
    print("Creates blender project")
    print("")

    if isinstance(data_paths[0], list):
        for paths in data_paths:
            create_blender_project(paths)
    else:
        create_blender_project(data_paths)

    print("")
    print("Done, Have a nice day!")

    dialog.end_copyright()

```

Requiments:

opencv-python==4.2.0.34

numpy==1.22.0

configparser==3.5.0

future-fstrings==1.2.0

requests==2.25.1

pyfiglet==0.7

setuptools==54.1.0

scipy==1.5.4 pillow==9.0.1

matplotlib==3.5.1

black==22.1.0

pytest==7.0.1