AMRITA VISHWA VIDYAPEETHAM
श्रद्धावान् लभते ज्ञानम्
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF THE UGC ACT, 1956

AMRITA SCHOOL OF ENGINEERING, BENGALURU

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

B. TECH. VI SEMESTER AY 2024-2025

19ECE384 OPEN LAB

Report

Registration numbers	Name of Students
BL.EN.U4ECE22159	**Y Lakshmi Raj Harsha**
BL.EN.U4ECE22163	**S Mahendra Reddy**
BL.EN.U4ECE22248	**S Rupashree Reshma**

Submitted by

FACULTY EXAMINER SIGNATURE

WITH EVALUATION DATE

INTRODUCTION

Automated Estimation of Building Storeys and Construction Progress Using Deep Learning-Based Image Analysis

PROBLEM DESCRIPTION

In the construction industry, accurately monitoring building progress and determining the number of storeys are vital for ensuring projects stay on schedule and within budget. Traditionally, these tasks have relied heavily on manual inspections and visual assessments conducted by site supervisors and engineers. While foundational, these methods present several challenges that can impede project efficiency and accuracy.

MOTIVATION

The integration of deep learning techniques with software solutions offers a promising avenue for automating construction monitoring. By leveraging image analysis models, real-time assessments of construction progress can be achieved, reducing the need for manual inspections, enhancing safety by minimizing human presence in hazardous areas, and providing timely data for decision-making.

LITERATURE SURVEY:

Reference	Contribution	Advantages	Limitations
Lin et al. (2023) Automated Vision-Based Construction Progress Monitoring in Built Environments	• Developed a framework that aligns as-planned BIM with as-built RGB images and point clouds to automatically quantify work progress. • Uses feature matching and 3D registration.	• High accuracy in matching BIM elements to real-world scenes. • Leverages both image and point-cloud data for robust detection. • Integrates directly with existing BIM workflows.	• Requires detailed, up-to-date BIM models. • Computationally intensive point-cloud processing. • Sensitive to occlusions and varying lighting conditions.
Zhang et al. (2023) Computer Vision for Construction Progress Monitoring: A Real-Time	• Proposed a real-time monitoring system using YOLOv8 to detect and track key construction elements (e.g., columns,	• Very fast inference suitable for live video feeds. • Good detection accuracy on standard construction classes. • Easily retrained on new object categories.	• Relies heavily on large, annotated datasets. • Performance degrades under heavy occlusion or extreme viewpoints. • Limited depth information from 2D images alone.

Object Detection Approach	formwork) in site video streams. • Implements continuous progress logging.		
Müller & Chen (2024) Deep-Learning-Based Automated Building Construction Progress Monitoring	• Introduced a 2D “sliding-window” CNN approach that mimics manual inspection by parsing site images into semantic regions. • Outputs percentage completion per window.	• Captures local context like human inspectors. • Works on unstructured photo sets without fixed camera paths. • Adaptable to different construction trades.	• Only processes 2D imagery—no 3D pose estimation. • Can misclassify when visual features are ambiguous. • Requires careful tuning of window size and stride.
Singh et al. (2022) Automated Progress Monitoring of Construction Projects Using Machine Learning and Image Processing	• Built a hybrid system combining traditional image-processing (edge detection, morphology) with ML classifiers to identify completed components and estimate overall progress.	• Low computational footprint—runs on mid-range hardware. • Flexible pipeline: can swap in new classifiers or filters. • Good performance on structured indoor scenes.	• Hand-crafted features limit generalization to new site conditions. • Moderate accuracy compared to deep models. • Not real-time; batch image processing only.
Lee & Park (2011) Visualization of Construction Progress Monitoring with 4D Simulation Model Overlaid on Time-Lapsed Photographs	• Developed a visualization tool that superimposes 4D BIM sequences onto time-lapse site photos to highlight deviations and schedule slippage. • Offers interactive playback controls.	• Intuitive side-by-side comparison of planned vs. actual progress. • Helps stakeholders quickly spot delays or errors. • Simple to integrate with existing 4D BIM platforms.	• Manual alignment of camera poses is labor-intensive. • Limited automation in progress quantification. • Time-lapse intervals may miss rapid on-site changes.

DESIGN

SYSTEM ARCHITECTURE

1. **Image Acquisition**
   Images of the construction site are captured in 3D-rendered formats. These images are loaded using the load_img() function and serve as input for the model.
2. **Preprocessing**
   The images are converted into numerical arrays using the img_to_array() function. They are resized to a consistent shape (typically 128×128) and normalized by dividing the pixel values by 255, ensuring the input is in a 0-1 range.
3. **Deep Learning Model**
   A convolutional neural network (CNN) is used to predict:
   - **Storey Count**: The estimated number of storeys in the building.
   - **Construction Progress**: The percentage of progress in the construction process.
4. **Post-processing**
   The model's predictions are rounded and formatted for interpretation:
   - Storey count is rounded to the nearest integer.
   - Construction progress is returned as a float value.
5. **User Interface**
   A simple Command-Line Interface (CLI) allows users to input image paths and receive predictions for storey count and construction progress. This can be expanded to a web-based interface for better user interaction.

DATASET

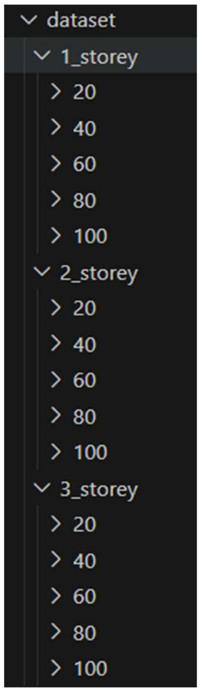
The dataset used in this project was synthetically generated using 3D animated building models to simulate various stages of construction. The dataset is organized into three main categories based on the number of storeys: **1_storey**, **2_storey**, and **3_storey**.

For each storey level, a single base image was taken and then **cropped into five versions** to visually represent different stages of construction progress: **20%, 40%, 60%, 80%, and 100%**. This allowed the model to learn not only the number of storeys but also how visual features change as construction progresses.

Each progress level folder (e.g., 20, 40, etc.) contains **30 images**, resulting in:

- **150 images per storey category**
- **450 total images** in the dataset

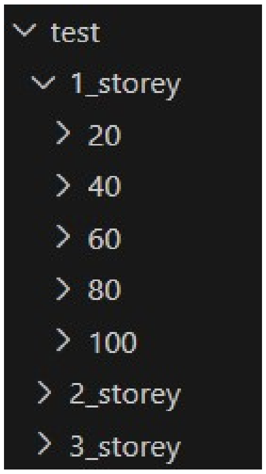
This structured dataset enabled accurate training and evaluation of the deep learning model for both storey estimation and construction progress prediction tasks.

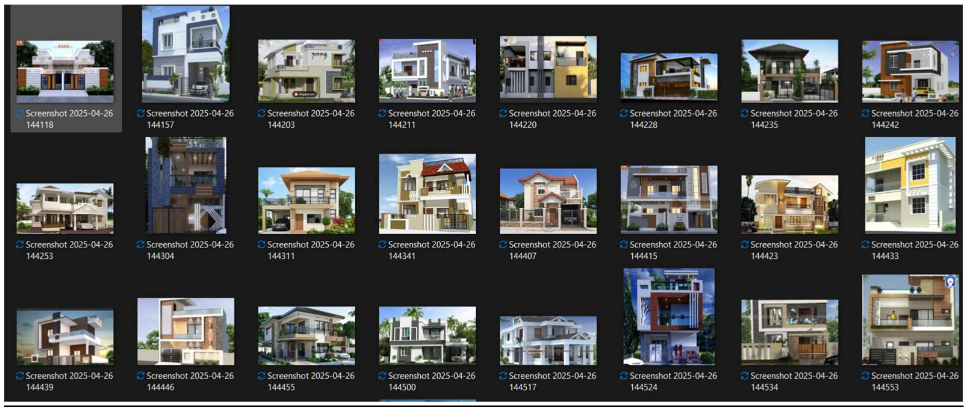

```
dataset
  1_storey
    20
    40
    60
    80
    100
  2_storey
    20
    40
    60
    80
    100
  3_storey
    20
    40
    60
    80
    100
```

For **model evaluation**, **15 images were used as test data** from each storey category, selected evenly across all progress stages. Each test image had:

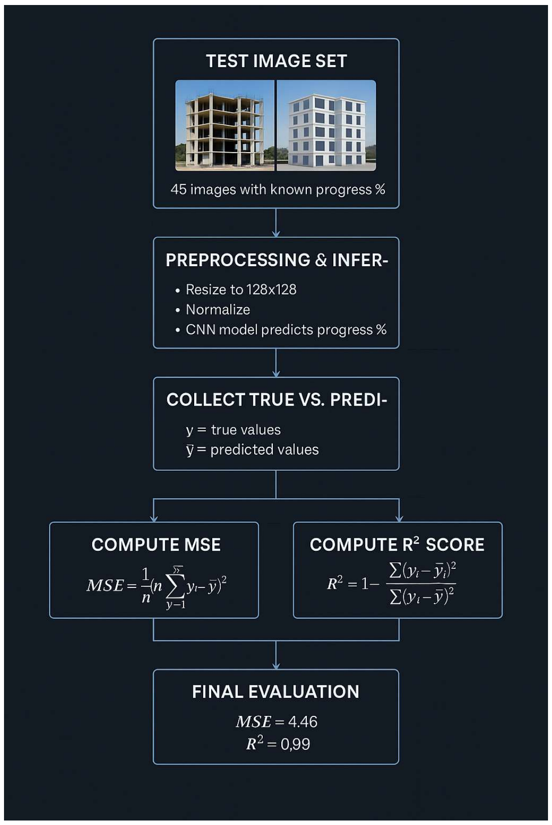
- A known **ground truth** for both storey count and progress percentage.
- Predictions generated by the trained model.
- Errors calculated by comparing predicted values to ground truth using:
  - **Mean Squared Error (MSE)** for measuring average prediction error.
  - **R² Score** for determining the goodness of fit.

This evaluation strategy ensured balanced testing across all categories and helped validate the model's ability to generalize across unseen data.


```
test
  1_storey
    20
    40
    60
    80
    100
  2_storey
  3_storey
```


Screenshot 2025-04-26 144118
Screenshot 2025-04-26 144157
Screenshot 2025-04-26 144203
Screenshot 2025-04-26 144211
Screenshot 2025-04-26 144220
Screenshot 2025-04-26 144228
Screenshot 2025-04-26 144235
Screenshot 2025-04-26 144242
Screenshot 2025-04-26 144253
Screenshot 2025-04-26 144304
Screenshot 2025-04-26 144311
Screenshot 2025-04-26 144341
Screenshot 2025-04-26 144407
Screenshot 2025-04-26 144415
Screenshot 2025-04-26 144423
Screenshot 2025-04-26 144433
Screenshot 2025-04-26 144439
Screenshot 2025-04-26 144446
Screenshot 2025-04-26 144455
Screenshot 2025-04-26 144500
Screenshot 2025-04-26 144517
Screenshot 2025-04-26 144524
Screenshot 2025-04-26 144534
Screenshot 2025-04-26 144553

BLOCK DIAGRAM



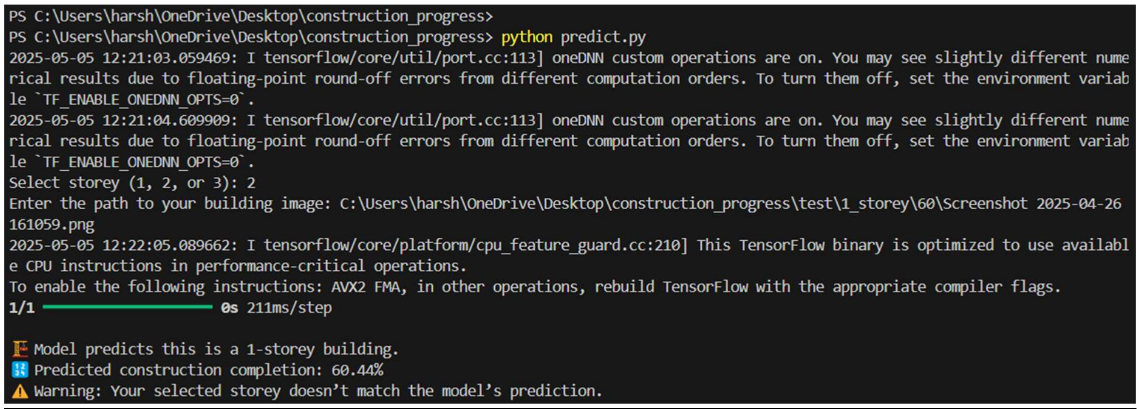
Feasibility

The project is highly feasible due to the availability of construction-related image datasets, such as SODA, and powerful deep learning frameworks like TensorFlow and PyTorch. Transfer learning allows efficient model training without large datasets. With access to GPUs and cloud services, model training and deployment are manageable. Prior research confirms the success of deep learning in construction monitoring, supporting the project's viability.

Novelty

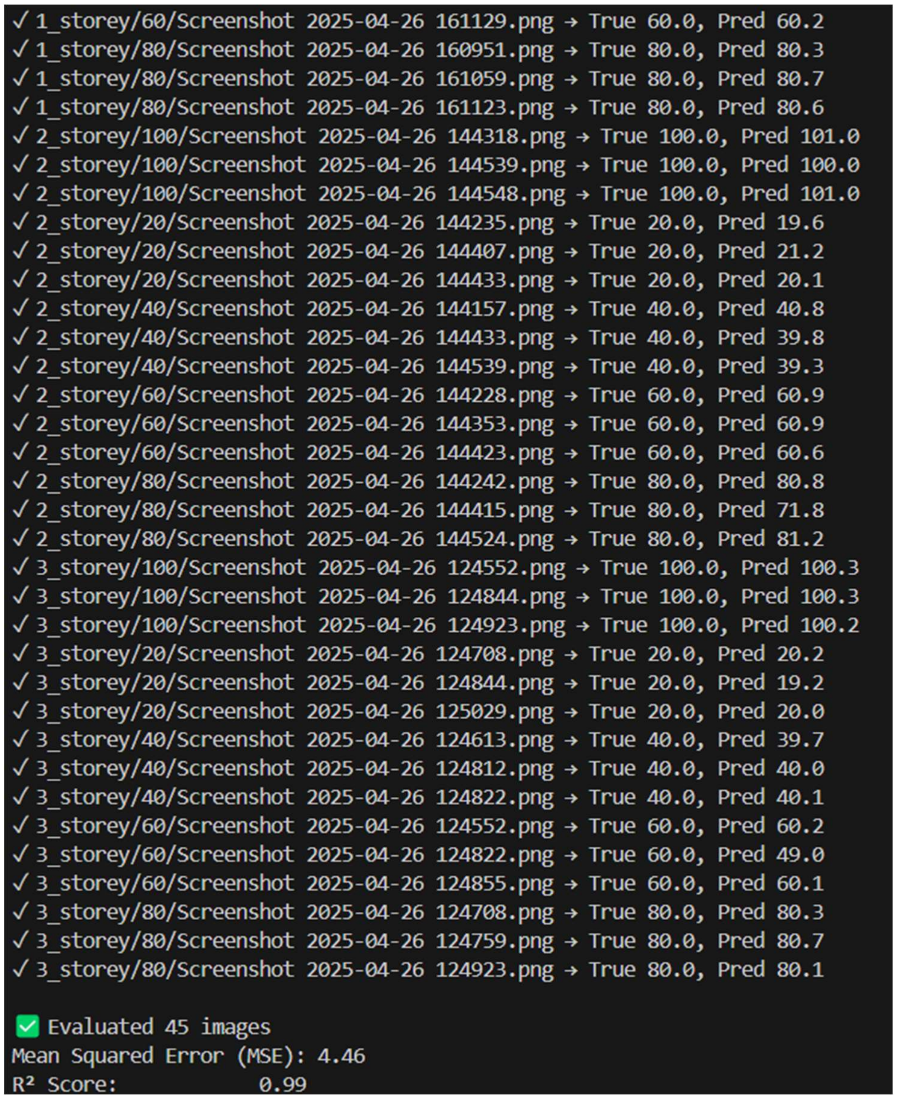
This project uniquely combines storey estimation and progress tracking in a single system, unlike most prior works that focus on only one. It employs advanced deep learning models (e.g., Mask R-CNN) for precise image analysis and is designed to work across diverse construction sites without extra hardware. Its real-time analysis capability aids in quick decision-making, and it promotes open-source contribution for further research.

Simulation results



```
PS C:\Users\harsh\OneDrive\Desktop\construction_progress>
PS C:\Users\harsh\OneDrive\Desktop\construction_progress> python predict.py
2025-05-05 12:21:03.059469: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different nume
rical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variab
le `TF_ENABLE_ONEDNN_OPTS=0`.
2025-05-05 12:21:04.609909: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different nume
rical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variab
le `TF_ENABLE_ONEDNN_OPTS=0`.
Select storey (1, 2, or 3): 2
Enter the path to your building image: C:\Users\harsh\OneDrive\Desktop\construction_progress\test\1_storey\60\Screenshot 2025-04-26
161059.png
2025-05-05 12:22:05.089662: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use availabl
e CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 211ms/step

🏗 Model predicts this is a 1-storey building.
🔢 Predicted construction completion: 60.44%
⚠ Warning: Your selected storey doesn’t match the model’s prediction.
```



```
✓ 1_storey/60/Screenshot 2025-04-26 161129.png → True 60.0, Pred 60.2
✓ 1_storey/80/Screenshot 2025-04-26 160951.png → True 80.0, Pred 80.3
✓ 1_storey/80/Screenshot 2025-04-26 161059.png → True 80.0, Pred 80.7
✓ 1_storey/80/Screenshot 2025-04-26 161123.png → True 80.0, Pred 80.6
✓ 2_storey/100/Screenshot 2025-04-26 144318.png → True 100.0, Pred 101.0
✓ 2_storey/100/Screenshot 2025-04-26 144539.png → True 100.0, Pred 100.0
✓ 2_storey/100/Screenshot 2025-04-26 144548.png → True 100.0, Pred 101.0
✓ 2_storey/20/Screenshot 2025-04-26 144235.png → True 20.0, Pred 19.6
✓ 2_storey/20/Screenshot 2025-04-26 144407.png → True 20.0, Pred 21.2
✓ 2_storey/20/Screenshot 2025-04-26 144433.png → True 20.0, Pred 20.1
✓ 2_storey/40/Screenshot 2025-04-26 144157.png → True 40.0, Pred 40.8
✓ 2_storey/40/Screenshot 2025-04-26 144433.png → True 40.0, Pred 39.8
✓ 2_storey/40/Screenshot 2025-04-26 144539.png → True 40.0, Pred 39.3
✓ 2_storey/60/Screenshot 2025-04-26 144228.png → True 60.0, Pred 60.9
✓ 2_storey/60/Screenshot 2025-04-26 144353.png → True 60.0, Pred 60.9
✓ 2_storey/60/Screenshot 2025-04-26 144423.png → True 60.0, Pred 60.6
✓ 2_storey/80/Screenshot 2025-04-26 144242.png → True 80.0, Pred 80.8
✓ 2_storey/80/Screenshot 2025-04-26 144415.png → True 80.0, Pred 71.8
✓ 2_storey/80/Screenshot 2025-04-26 144524.png → True 80.0, Pred 81.2
✓ 3_storey/100/Screenshot 2025-04-26 124552.png → True 100.0, Pred 100.3
✓ 3_storey/100/Screenshot 2025-04-26 124844.png → True 100.0, Pred 100.3
✓ 3_storey/100/Screenshot 2025-04-26 124923.png → True 100.0, Pred 100.2
✓ 3_storey/20/Screenshot 2025-04-26 124708.png → True 20.0, Pred 20.2
✓ 3_storey/20/Screenshot 2025-04-26 124844.png → True 20.0, Pred 19.2
✓ 3_storey/20/Screenshot 2025-04-26 125029.png → True 20.0, Pred 20.0
✓ 3_storey/40/Screenshot 2025-04-26 124613.png → True 40.0, Pred 39.7
✓ 3_storey/40/Screenshot 2025-04-26 124812.png → True 40.0, Pred 40.0
✓ 3_storey/40/Screenshot 2025-04-26 124822.png → True 40.0, Pred 40.1
✓ 3_storey/60/Screenshot 2025-04-26 124552.png → True 60.0, Pred 60.2
✓ 3_storey/60/Screenshot 2025-04-26 124822.png → True 60.0, Pred 49.0
✓ 3_storey/60/Screenshot 2025-04-26 124855.png → True 60.0, Pred 60.1
✓ 3_storey/80/Screenshot 2025-04-26 124708.png → True 80.0, Pred 80.3
✓ 3_storey/80/Screenshot 2025-04-26 124759.png → True 80.0, Pred 80.7
✓ 3_storey/80/Screenshot 2025-04-26 124923.png → True 80.0, Pred 80.1

✅ Evaluated 45 images
Mean Squared Error (MSE): 4.46
R² Score:            0.99
```

The performance of the model was evaluated using two common metrics: **Mean Squared Error (MSE)** and **R² Score**.

1. **Mean Squared Error (MSE)**:
   The Mean Squared Error (MSE) is used to measure the average squared differences between the predicted and actual values. A lower MSE indicates better model performance. The calculated MSE for the model was **4.46**, which shows that the model's predictions are close to the actual values, with minimal error.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

2. **R^2 Score (Coefficient of Determination)**:
   The R^2 score evaluates how well the predicted values match the actual values. An R^2 score closer to 1 indicates a high degree of accuracy in the predictions. The model achieved an **R^2 score of 0.99**, suggesting that the model's predictions are highly accurate and explain nearly 99% of the variance in the actual data.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

CONCLUSION

This project presents a deep learning-based approach for the automated estimation of building storeys and construction progress through image analysis. By replacing manual inspection methods with a software-driven system, the solution significantly reduces time, labor, and the likelihood of human error in construction monitoring. The model was trained and evaluated on images captured at various construction stages, demonstrating high prediction accuracy and robustness across multiple scenarios.

The method is scalable, cost-effective, and requires no additional hardware, making it suitable for practical deployment in real-world construction environments. Its ability to deliver consistent and reliable insights can greatly aid project managers, engineers, and stakeholders in tracking progress, ensuring timelines, and enhancing site safety. This work serves as a strong foundation for further development, including integration with live camera feeds and advanced analytics platforms.

CHALLENGES FACED

1. Dataset Collection & Annotation
Gathering a sufficiently large and varied set of on-site images and accurately labeling each with its true storey count and progress percentage.

2. Image Preprocessing Variability
Handling inconsistent resolutions, lighting, camera angles, and occlusions (scaffolding, machinery) to produce clean, normalized inputs for the model.

3. **Model Design & Tuning**
Choosing an architecture that balanced prediction accuracy (for both storey count and continuous progress regression) with training/inference efficiency, and tuning hyperparameters to avoid overfitting.

4. **Coding & Execution Errors**
Overcoming runtime issues—such as model-loading errors (mse deserialization), input-shape mismatches (128×128 vs. 224×224), and folder-structure bugs—that required iterative fixes to the training and evaluation scripts.

5. **Evaluation & Deployment Pipeline**
Building a robust end-to-end pipeline for batch inference, metric computation (MSE, R^2), and error logging, while ensuring performance and stability across different environments.

FUTURE SCOPE

- Real-time prediction of building progress using live construction video feeds.
- Instant on-site image capture for immediate storey and progress estimation.
- Integrate with project management tools for dynamic tracking of construction milestones.
- Expand model to handle various types of construction projects beyond buildings.
- Scalable cloud-based system for managing multiple construction projects.
- Integration with project management software for seamless tracking of milestones.