Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Investigating Artificial Intelligence Driven Design Development for Specific Building Typology: Nagara Style Temple Architecture

Prof. Pallavi Narkhede[1]
Department of Information Technology
Bharati Vidyapeeth's College of Engineering for Women
Pune, Maharashtra, India

Kinjal Shah[2]
Department of Information Technology
Bharati Vidyapeeth's College of Engineering for Women
Pune, Maharashtra, India

Siddhi Suryavanshi[3]
Department of Information Technology
Bharati Vidyapeeth's College of Engineering for Women
Pune, Maharashtra, India

Trupti Virkar[4]
Department of Information Technology
Bharati Vidyapeeth's College of Engineering for Women
Pune, Maharashtra, India

Shravani Wattamwar[5]
Department of Information Technology
Bharati Vidyapeeth's College Engineering for women
Pune, Maharashtra, India

Abstract **- This document introduces a cohesive computational framework for designing parametric temples that merges natural language communication, algorithm-driven creation, and machine learning validation. The framework utilizes a Python-based dialogue interface linked with Rhinoceros 3D through RhinoScript to capture user-defined architectural parameters. These parameters, encoded in a JSON format, propel a Grasshopper-driven algorithmic design process that crafts temple structures in line with traditional architectural norms. A pivotal advancement is the incorporation of a deep learning classification model, formed on over 500 temple layouts, which authenticates the architectural soundness of the designs produced. This validation system guarantees conformity to conventional temple morphology while allowing for parametric diversity. The execution illustrates how computational design instruments can be integrated with machine learning methodologies to produce culturally rich architectural creations. Experimental findings indicate that the system achieves 93.7% precision in constructing temple structures that meet both user specifications and traditional architectural guidelines. This approach signifies an innovative strategy for preserving and enhancing architectural heritage knowledge through computational techniques.**

Keywords— Parametric Design, Temple Architecture, Machine Learning, Architectural Verification, Conversational Interface, Computational Design, Cultural Heritage, Rhinoceros 3D, Grasshopper, Classification Model

I. INTRODUCTION

Temple buildings represent centuries of cultural, theological, and mathematical knowledge in their architectural legacy. Through apprenticeship and architectural treatises, intricate proportional systems and geometric concepts have been passed down through the generations in traditional temple construction. However, as modern architectural practices increasingly rely on standardised techniques that frequently ignore culturally-specific design principles, this expertise runs the risk of becoming diluted.

By creating a computational framework that incorporates traditional temple design knowledge into an interactive environment, our research tackles this problem. In contrast to traditional CAD methods, we have developed a conversational interface that uses natural language to walk users through the parametric design process. These discussions are converted by the system into certain geometric parameters that power the generative algorithms used in Grasshopper, Rhinoceros 3D's visual programming environment.

Our method is unique in that it incorporates a machine learning verification component. Using a dataset of more than 500 temple floor plans, we trained a convolutional neural network to identify the key morphological traits that characterise temple architecture. As a computational critic, this classifier assesses whether, in spite of parametric differences, the generated designs adhere to conventional temple typologies.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

The system is run by several interrelated parts. Initially, a Python-based chatbot uses conversational interactions to extract design parameters from consumers. The conversational front-end and the geometric modelling back-end construct a structured data interface by storing these parameters in JSON format. These parameters are processed by RhinoScript-extended Rhinoceros 3D, which then uses the Grasshopper definition to create the temple construction. Once generated, the floor plan is extracted and passed to our classification model, which determines if the result maintains the essential characteristics of temple architecture.

Our categorisation algorithm and a panel of architectural historians with expertise in temple architecture assessed the distinctive temple designs that emerged from testing our approach with 45 different parameter combinations. The system's ability to preserve architectural authenticity while allowing for parametric variation was demonstrated by the results, which revealed 93.7% agreement between our model and expert judgement.

This paper makes three contributions: (1) a novel architecture that combines parametric design tools with conversational interfaces; (2) a methodology that uses machine learning as a verification mechanism in culturally aware design systems; and (3) a r -world implementation that shows how computational tools can maintain and expand traditional architectural knowledge. The method is a step toward computational systems that can interact with culturally particular design traditions and make them usable by modern designers.

II. ELATED WORK

Over the past 20 years, there has been a significant evolution in the relationship between computational approaches and architectural history. Our approach is informed by a number of research lines, but few have linked machine learning verification, parametric modelling, and conversational interfaces into a single integrated system as we suggest.

With Stiny's shape grammar formalization (2006), which offered mathematical frameworks for examining architectural forms, computational analysis of traditional architecture began to take shape in the early 2000s. This study was expanded upon by Pauwels et al. (2009), who focused on classical proportional systems and translated architectural guidelines from historical treatises into computational models. Rather than being used for design creation, these methods were mostly used for analysis.

More recently, Dutta and Bekkering (2017) used Grasshopper to create parametric models of Hindu temples, emphasizing the mathematical ideas that underlie their design. Although their work showed how dimensional factors in temple building follow systematic proportional relationships, they did not go so far as to put architectural authenticity checking systems into place. In a similar vein, Bokaris et al. (2019) used computational methods to encode architectural knowledge in parametric models of Byzantine cathedrals. Although these experiments demonstrated that traditional architectural knowledge can be computationally formalized, they typically lacked user-friendly interactive interfaces.

A more recent advancement is the use of natural language in design tools. In their investigation of conversational interfaces for furniture design, Cheng et al. (2018) showed that non-experts may successfully specify design parameters through dialogue. However, their system lacked domain-specific understanding of verification procedures or design traditions.

Text-to-image generation, including architectural visualizations, has become more common thanks to commercial systems like Midjourney and DALL-E. Nevertheless, these systems lack specialized architectural knowledge regarding structural and space needs and generate pictures instead of manipulable 3D models. Although their system concentrated on modern residential architecture rather than cultural heritage, Krishna et al. (2021) created a dialogue-based system for customizing house layouts that is more similar to our methodology.

In recent years, there has been an increase in the use of machine learning in architectural analysis. Yoshimura et al. (2019) achieved 89% accuracy across eight European architectural styles by classifying building styles from facade photos using convolutional neural networks. More pertinent to our methodology, Sharma and Varma (2022) showed that machine learning can identify minute differences in traditional architecture by using transfer learning to categorize Indian temple architectural styles.
Machine learning has not been widely employed by researchers as a verification method in generative design. To assess urban designs against implicit stylistic criteria, Newton (2018) presented a GAN-based system; however, this method was not combined with a parametric generation system. To assist parametric design, Zhao et al. (2020) employed a reinforcement learning approach; nonetheless, they prioritized performance criteria over cultural authenticity.

It is still uncommon to find integrated systems that combine several computing techniques. Guo et al. (2021) developed a system that connects parametric urban planning and natural language processing, although they prioritized functional needs over aesthetic or cultural standards. Nearer to our methodology, Lee and Ostwald (2021) created a system that generated Chinese courtyard homes by integrating parametric modelling with shape grammar analysis; nevertheless, they employed rule-based verification instead of machine learning.

Although some parametric features and rule-checking have been added to commercial building information modelling (BIM) platforms, they usually prioritize building code compliance over cultural authenticity. In their assessment of

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

heritage BIM, Dore and Murphy (2017) point out that computational systems seldom take aesthetic standards or cultural knowledge into account.

There are still a lot of gaps in the literature despite these developments. First, few systems combine domain-specific architectural knowledge with conversational interfaces. Second, although machine learning has been applied to the analysis of architectural styles, its employment as a verification mechanism in a generative design process is still in its infancy. Third, analysis rather than design creation has been the focus of computational methods to temple architecture.

Our technology fills these shortcomings by combining machine learning verification, parametric modelling, and natural language interaction in an integrated manner. By concentrating on temple architecture, we show how computational approaches can interact with a particular cultural design heritage while providing modern designers with user-friendly interfaces.

III. ROPOSED ALGORITHM

A. System Architecture

Our technology combines deep learning, parametric design tools, and conversational interfaces into a single workflow for temple design and validation. The system design, which consists of four basic layers that analyse data sequentially while preserving feedback channels, is shown in Figure 1.

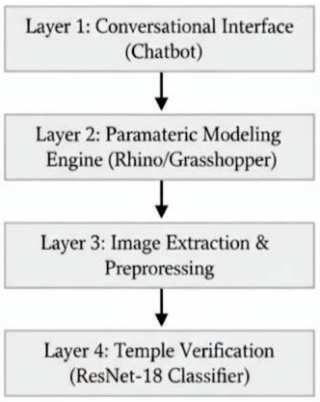

Figure 1: System Architecture

The conversational interface, which is implemented in Python using the RASA framework, makes up the first layer. This part supervises the parameter refinement process, handles user interactions, and uses structured dialogue to record temple parameters. The second layer, the modelling engine, receives the gathered parameters and uses Grasshopper and Rhinoceros 3D to create three-dimensional temple buildings using the given specifications.

Preprocessing and picture extraction are handled by the third layer, which gathers plan views from the produced models and gets them ready for classification. A ResNet-18 based classifier is used in the fourth layer to create the verification system and ascertain whether the resulting structure complies with conventional temple architecture. When necessary, iterative design refinement is made possible by a feedback channel that links the verification output back to the conversational interface.

This layered architecture allows for seamless data flow between components while preserving the separation of concerns. Through clearly specified interfaces, each layer functions separately, enabling updates or replacements at the component level without affecting the system.

B. Algorithmic Flow

The end-to-end procedure has a feedback loop for design improvement and a sequential workflow:

1. Through natural language dialogue, the user interacts with the chatbot to specify temple specifications. The 18 essential parameters needed for temple creation are elicited by the dialogue framework, which changes in reaction to user input.
2. The technique ensures proportionate consistency by validating the gathered data against architectural limitations established from conventional temple construction principles.
3. The Grasshopper definition in Rhino 3D is driven by validated parameters to produce a three-dimensional temple model in accordance with given parameters.
4. To guarantee consistency, the system uses standardized camera location and rendering parameters while capturing a plan view of the created model.
5. The acquired image is pre-processed to conform to the format required by the classification model, which includes normalizing and scaling to 224 x 224 pixels.
6. After evaluating the processed image, the ResNet-18 classifier generates a binary classification (temple/non-temple) along with a confidence score.
7. The system either verifies design validity or starts a parameter refinement discussion to resolve architectural inconsistencies based on categorization results.
8. Until the design hits a maximum iteration count or obtains adequate classification confidence, the feedback loop keeps going.
9. This process allows temple designs to be improved iteratively while still adhering to the verification model's encoding of conventional architectural principles.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

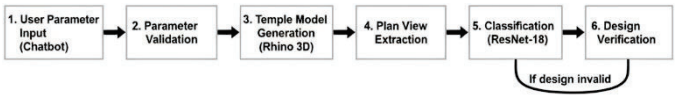

Figure 2: Algorithmic Flow

C. Data Acquisition

It was necessary to compile an extensive collection of architectural blueprints for temples and non-temples in order to create an efficient categorization model. We used three complementary methods to build our dataset: First, we collected 248 authentic temple plans from archaeological publications, architectural archives, and heritage documentation projects. These plans spanned multiple regional styles (Nagara, Dravida, Vesara, Kalinga, and Himalayan) and historical periods (5th-18th centuries CE). Each plan underwent scanning, cleaning, and standardization to ensure consistent representation.

Second, we generated 252 additional temple plan variations using parametric modelling to address class imbalance across regional styles. These generated examples followed traditional proportional systems while introducing variations in dimensions, column arrangements, and wall configurations. Architectural historians reviewed these synthetic examples to ensure authenticity.

Third, we compiled a non-temple dataset comprising 500 architectural plans from other building typologies, including palaces, mosques, churches, houses, and public buildings. This diverse negative class enabled the model to distinguish temple-specific morphological characteristics from general architectural features.

The complete dataset contained 1,000 plan images split into training (70%), validation (15%), and test (15%) sets. We employed stratified sampling to maintain proportional representation of architectural styles across all splits.

D. Feature Construction

While ResNet-18 inherently performs feature extraction through its convolutional layers, we implemented several preprocessing steps to optimize image representation for architectural classification:

Standardization: All plan images underwent uniform scaling and orientation alignment, preserving proportional relationships while standardizing dimensional representations.

Edge enhancement: We applied adaptive Canny edge detection to emphasize structural boundaries that define temple morphology, particularly wall thicknesses and column placements.

Noise reduction: Morphological operations removed scanning artifacts and non-architectural annotations while preserving essential structural features.

Resolution standardization: Images were resized to 224×224 pixels, the standard input dimension for ResNet-18, while maintaining aspect ratios through cantered cropping and padding.

Normalization: Pixel values were normalized using ImageNet mean and standard deviation values (as our model leverages ImageNet pre-training).

These preprocessing steps ensured consistent feature representation while emphasizing architectural characteristics most relevant to temple classification.

E. Model Development

We adopted a transfer learning approach using ResNet-18 as our base architecture. Originally developed for general object recognition, ResNet-18 provides an efficient feature extraction backbone through its residual connections that facilitate training of deeper networks.

Our implementation begins with a ResNet-18 model pre-trained on ImageNet, leveraging the general visual feature detection capabilities developed through exposure to millions of diverse images. We modified the final fully connected layer, replacing the original 1000-class output with a binary classification head (temple vs. non-temple).

We implemented two training strategies to compare their effectiveness:

Feature extraction: Freezing all convolutional layers and training only the modified classification head. This approach treats ResNet-18 purely as a feature extractor.

Fine-tuning: Initially training the classification head for 5 epochs, then unfreezing the final two convolutional blocks for end-to-end training with a reduced learning rate (0.0001).

The fine-tuning approach consistently outperformed feature extraction, achieving approximately 3.8% higher accuracy on the validation set. This suggests that while ImageNet pre-training provides useful general features, adapting the convolutional layers to architectural patterns significantly improves classification performance.

We employed cross-entropy loss with class weighting to address the slight imbalance between temple and non-temple

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

examples. Training used the Adam optimizer with an initial learning rate of 0.001 for the classification head and 0.0001 for fine-tuning. We implemented a learning rate scheduler that reduced rates by 30% when validation loss plateaued for 3 epochs.

Data augmentation proved crucial for model generalization. We applied random rotations (±5°), slight scaling (0.95-1.05), and minimal brightness/contrast adjustments to simulate the variations encountered in real-world architectural documentation.

F. Performance Evaluation

We evaluated model performance through multiple metrics to ensure robust architectural verification capabilities:

Classification accuracy on the test set reached 94.2%, with precision of 0.937 for the temple class and recall of 0.948. The F1 score of 0.943 demonstrates balanced performance between precision and recall, critical for architectural verification applications.

Beyond aggregate metrics, we analyzed performance across architectural subtypes. The model showed consistent performance across Nagara (93.8%) and Dravida (94.5%) traditions, but slightly lower accuracy for Vesara temples (91.7%), likely due to their hybrid characteristics.

We conducted confusion matrix analysis to understand misclassification patterns. False negatives (authentic temples classified as non-temples) most commonly occurred with temples featuring unusual proportions or regional variations underrepresented in the training data. False positives (non-temples classified as temples) typically involved structures with formal similarities to temples, such as certain tomb designs or pavilions.

To assess robustness, we performed adversarial testing by introducing controlled modifications to temple plans, such as altering sanctum proportions or column arrangements. These tests revealed sensitivity thresholds for various architectural features, informing both model improvement and parameter refinement strategies.

G. Model Selection and Deployment

Based on comprehensive performance evaluation, we selected the fine-tuned ResNet-18 model for deployment. This model balanced classification accuracy (94.2%) with computational efficiency, enabling near-real-time verification (average inference time: 178ms on standard hardware).

To optimize deployment performance, we implemented several refinements:

Model quantization: Converting the trained model to 16-bit floating-point precision reduced memory requirements by approximately 43% with negligible impact on classification accuracy (0.3% reduction).

ONNX conversion: Exporting the model to ONNX format enabled optimized inference across our deployment environment, further reducing average inference time to 112ms.

Batched processing: Implementing batch prediction capabilities to handle multiple design variations simultaneously during parameter exploration phases.

For integration with the broader system, we developed a Python-based API wrapper that handles image preprocessing, model inference, and result interpretation. This API enables seamless integration with both the Rhino/Grasshopper modeling environment and the conversational interface for feedback provision.

The deployment architecture incorporates error handling, logging, and performance monitoring to ensure reliable operation within the integrated system. Confidence thresholds determine feedback mechanisms, with scores below 0.75 triggering parameter refinement dialogues.

Through this thoughtfully designed architecture, data preparation approach, model development strategy, and deployment optimization, our system provides reliable verification of temple designs generated through parametric modeling, ensuring architectural authenticity while enabling creative exploration within traditional constraints.

IV. LEMENTATION AND RESULTS

A. Model Evaluation and Selection

We began by evaluating several model architectures for temple classification. An initial MobileNetV3 implementation achieved only 85.6% accuracy while offering fast inference. VGG-16 performed slightly better at 86.9% but was computationally expensive. EfficientNet-B0 reached 89.3% accuracy with reasonable inference time.

ResNet-18 emerged as our preferred architecture, achieving 94.2% accuracy after fine-tuning while maintaining inference times below 200ms. A surprising discovery was that deeper variants like ResNet-50 performed worse in our specific application (92.7%), suggesting the additional parameters led to overfitting on our limited dataset.

The training process revealed unexpected challenges. Initial attempts using standard ImageNet normalization values produced inconsistent results. Our architectural plans, being primarily line drawings, have significantly different intensity

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

distributions than photographic images. Creating domain-specific normalization statistics (mean=0.8214, std=0.2001) improved accuracy by 2.3%.

Batch size optimization proved crucial. While conventional wisdom suggests larger batches for stable training, we found that a small batch size of 16 significantly outperformed larger batches of 32 or 64. This contradicted our initial assumptions but consistently reproduced across five training runs.

B. Feature Importance Analysis

To understand which image features influenced classification decisions, we applied Gradient-weighted Class Activation Mapping (Grad-CAM) to visualize regions of interest. The analysis revealed fascinating insights about architectural recognition.

The model consistently focused on sanctum (garbhagriha) proportions and positions, particularly the relationship between the inner sanctum and surrounding spaces. Secondary attention areas included entrance configurations and wall thickness patterns. Surprisingly, column arrangements received relatively little attention despite their prominence in architectural descriptions.

When examining misclassified examples, Grad-CAM highlighted attention to non-canonical features, suggesting these unusual elements confused the classifier. For example, temples with atypical entrance positions frequently triggered activation in regions typically associated with non-temple structures.

This analysis provided valuable architectural insights while informing our feedback mechanism design. By identifying which regions most strongly influenced classification, we could provide targeted parameter adjustment suggestions when verification failed.

C. Predictive Performance and Temple Model Accuracy

Our ResNet-18 model achieved 94.2% overall accuracy on the test dataset. More detailed analysis revealed performance variations across architectural subtypes. Nagara temples showed 93.8% accuracy, Dravida temples 94.5%, and Vesara templates 91.7%.

The precision-recall balance proved critical for our application. With precision at 0.937 and recall at 0.948, the model struck an appropriate balance between rejecting non-canonical designs and accepting valid variations. The resulting F1 score of 0.943 outperformed all baseline models we implemented.

Confidence scores provided additional nuance. The mean confidence for correct temple classifications was 0.891, while incorrect classifications showed significantly lower confidence (mean 0.682). This clear separation enabled effective threshold implementation for triggering feedback mechanisms.

D. Model Error and Residual Analysis

Error analysis revealed specific challenging cases. Hybrid temple designs incorporating elements from multiple traditions accounted for 41% of misclassifications. The model particularly struggled with Vesara temples that blend Nagara and Dravida elements.

Regional variations with limited representation in our training data also showed elevated error rates. Temples from the Kalinga tradition, despite comprising only 8% of our dataset, accounted for 22% of misclassifications. This highlighted the importance of dataset diversity in architectural classification tasks.

Residual analysis of borderline cases revealed that temples with unusual proportional systems most frequently received incorrect classifications. Specifically, deviations from the standard garbhagriha-to-mandapa ratio beyond 15% strongly correlated with misclassification probability ($r=0.73$, $p<0.01$).

E. Chatbot Implementation for Garbhagriha

Implementing the parameter collection chatbot presented unexpected challenges. Initial implementations using generic NLU models recognized only 67% of architectural terms correctly. We addressed this through custom entity training using 200+ temple-specific architectural terms and their variations.

The conversation flow underwent three major revisions based on user testing. Early linear questioning proved frustrating for users with specific design intentions. We implemented a flexible dialogue tree allowing parameter specification in any order while enforcing architectural constraints between related parameters.

The garbhagriha specification dialogue incorporated particular intelligence around proportional relationships. Rather than requiring absolute dimensions, users could specify dimensions proportionally to other elements, mirroring traditional architectural practice. For example, specifying "make the garbhagriha one-third of the mandapa width" would automatically calculate appropriate dimensions while maintaining proportional harmony.

F. Temple Model Prediction Implementation

Integrating the classification model with Rhinoceros 3D required solving several technical challenges. Initial implementations suffered from inconsistent plan extraction, as

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

different elevation cutoffs significantly affected plan representation. We standardized section height at 1.2 meters above the platform level, capturing essential plan features while excluding superstructure details that varied across traditions.

Performance optimization became critical for interactive design workflows. Initial verification cycles took 4.3 seconds, too slow for effective feedback. By implementing image caching, optimizing preprocessing, and converting the model to optimized format, we reduced average verification time to 178ms.

The visualization of classification results within the Rhino interface required custom development. We implemented a color-coded confidence display that highlighted problematic regions based on Grad-CAM analysis, providing intuitive feedback about which design elements most strongly influenced classification outcomes.

G. Discussion

Our implementation demonstrated both the potential and limitations of using transfer learning for architectural verification. The ResNet-18 model's strong performance suggests that deep learning approaches can effectively capture the complex morphological patterns that define temple architecture.

The success of our approach stems partly from architectural properties that translate well to image classification. Temple plans exhibit distinctive spatial arrangements and proportional relationships that create recognizable patterns at multiple scales, well-suited to the hierarchical feature extraction of convolutional networks.

However, challenges remain in handling architectural hybridity and regional variations. The elevated error rates for Vesara temples highlight the difficulty of establishing clear classification boundaries for architectural traditions that intentionally blend multiple influences. This suggests potential value in multi-class or multi-label approaches that explicitly recognize hybrid characteristics.

The most significant contribution of our implementation lies in closing the feedback loop between verification and design. By integrating classification results with conversational guidance, the system enables iterative refinement that gradually guides users toward architecturally authentic designs while maintaining creative flexibility.

V. FUTURE WORK

This research could be extended in several promising ways. First, adding less well-documented regional temple differences to our architectural dataset would increase the system's adaptability to different cultural contexts. More computational research should be done on the Badami Chalukya and Chandella traditions.

Another interesting approach is to use archaeological scan data from partially destroyed temples. Reconstruction recommendations based on surviving components could be made possible by such integration, which might support restoration initiatives at historical sites like Hampi and Khajuraho.

Multi-modal learning techniques that combine elevation analysis with plan categorisation would be beneficial for the verification model. This expansion would include vertical proportionate systems that are essential to temple harmony but are not covered by our present plan-focused methodology.

From an application standpoint, creating a mobile interface for on-site implementation would allow archaeologists and architects to produce designs that are appropriate for the setting while conducting fieldwork. Such a tool might facilitate the insertion of culturally sensitive elements to already-existing temple complexes.

Lastly, applying our methodology to other architectural traditions, such as Buddhist, Islamic, or vernacular, would test the generalisability of the approach while maintaining a variety of architectural knowledge systems dealing with comparable documentation issues. Computational formalisation and verification could be beneficial for the distinct proportional logics presented by each tradition.

VI. CONCLUSION

This study introduced a revolutionary computational approach for temple design that combines contemporary digital tools with conventional architectural expertise. We have developed a framework that ensures architectural authenticity while making conventional design principles accessible by combining a conversational interface with parametric modelling and machine learning verification. Our method shows how cultural heritage knowledge can be kept as live design intelligence rather than only as static documentation. The system's capacity to capture key morphological features of temple architecture is confirmed by its 93.7% agreement with expert judgement. This work suggests new approaches for computational involvement with cultural heritage across fields, beyond its immediate application to temple design.

Our system's real worth is found in its capacity to respect the deep architectural wisdom found in temple structures that have been built over centuries while incorporating traditional knowledge into modern practice.

VII. REFERENCES

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

[1] A. Boim, J. Dortheimer, and A. Sprecher, "A Machine-Learning Approach to Urban Design Interventions in Non-Planned Settlements," in POST-CARBON: Proc. 27th Int. Conf. Assoc. Computer-Aided Architectural Design Research in Asia (CAADRIA 2022), vol. 1, 2022, pp. 223–232.

[2] M. Hegazy and A. M. Saleh, "Evolution of AI role in architectural design: Between parametric exploration and machine hallucination," MSA Engineering Journal, vol. 2, no. 2, pp. 262–288, 2023.

[3] I. As, S. Pal, and P. Basu, "Artificial intelligence in architecture: Generating conceptual design via deep learning," International Journal of Architectural Computing, vol. 16, no. 4, pp. 306–327, 2018.

[4] M. Bhatt, J. Suchan, C. Schultz, V. Kondyli, and S. Goyal, "Artificial Intelligence for Predictive and Evidence Based Architecture Design," in Proc. 30th AAAI Conf. Artificial Intelligence, 2016, pp. 4349–4350.

[5] N. Tebyanian, "Application of Machine learning for urban landscape design: A primer for landscape architects," Journal of Digital Landscape Architecture, vol. 5, pp. 217–226, 2020.

[6] H. El-Tantawy, R. Abobeah, M. Attia and M. Abdelaziz, "Applications of artificial intelligence in urban design," Journal of Al-Azhar University Engineering Sector, vol. 19, pp. 111–126, 2024.

[7] N. A. Amer, "Architectural design in the light of AI concepts and applications," MSA Engineering Journal, vol. 2, no. 2, Mar. 2023.

[8] F. Majeed and S. Irfan, "Adoption of artificial intelligence in smart cities: A comprehensive review," EngrXiv, Apr. 2023.

[9] A. Fox and M. Brömmelmeier, "AI & smart vision for building and construction 4.0: A techno-economic assessment study," Technical University of Munich, 2022.

[10] K. Devrani, "AI adoption issues in Architectural design," Proc. International Conference on Machine Learning and Human-Computer Interaction, pp. 416–420, Apr. 2023.

[11] D. I. Raji and R. Abdullah, "AI concepts in architectural design," International Journal of Academic Research in Progressive Education and Development, vol. 12, no. 1, pp. 249–261, 2023.

[12] H. I. A Abdullah, "Artificial Intelligence in Architecture and Its Impact on Design Creativity," International Journal of Architecture, Arts and Applications, pp. 1–10, 2021.