International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Classification and Identification of Ancient Indian Temple Pillars Era Using Deep Learning

Dr.Gurudeva S Hiremath*[1], Dr.Shrinivasa Naik C.L.[2], Dr.Narendra Kumar S[3], Mr.Kiran Ankalakoti[4], Dr.John P Veigas[5]

Submitted:11/09/2024 **Revised:** 12/10/2024 **Accepted:** 22/10/2024

Abstract: The historic Indian temples have endured for several millennia and are linked to notable dynasties in power. The pillars of style, architecture, sculpture, and techniques in this environment each display amazing marvels of their own. Currently, classification and identification of pillars time-line depend on human visual talents, even if many of the characteristics related to this activity are small and difficult to understand. Archaeologists have several complex obstacles throughout this process because there are no dependable digital methods for the scientific classification and identification of pillars time-line. Recently, a number of machine learning algorithms have been presented to automate the segmentation and recognition of pillars. Notably, the field of archaeology has benefited greatly from the application of deep learning. The current study utilizes advanced deep learning models that employ a transfer learning approach to perform classification and identification of pillars time-line. The own prepared dataset has 1030 pillar images categorized according to the time-line based on the Potika type/architecture level. These categories includes Pillar time-lines as 4th to 8th BC, 5th to 9th BC, 9th to 12th to 13th BC , 11th to 13th to 14th BC, 15th to 16th BC and 16th BC Onwards. In this current research, we use cutting-edge transfer learning deep learning models for pillar timeline classification and identification. This research employs five deep learning convolutional neural network (CNN) models, namely DenseNet121, VGG16, InceptionV3, MobileNetV2, and Xception, to diagnose and classify pillars time-line. The average pillar timeline identification efficiency was 92.00%, 90.33%, 89.17%, 87.83%, and 85.17% for all five models respectively. On the ancient temple pillar time-line dataset, DenseNet121 beat VGG16, InceptionV3, MobileNetV2, and Xception CNN models in classification accuracy and computational efficiency.

Keywords: *Pillar Architecture, Transfer learning, Auto-Segmentation, Classification, Convolutional Neural Networks, Identification, Deep Learning, Pillar Time-line.*

1. Introduction

Archaeology studies life as a whole. Life is a continuous equilibrium where the past and present converge [1]. Archaeology has the capacity to furnish the historical context of existence within a particular area and era. Archaeology has the ability to uncover various characteristics of a civilization, such as its culture, level of development, religious beliefs, environment, and more[2]. Indian archaeology has been studied by a diverse group of archaeologists from the 19th century to the present day. In the early 18th century, Western European travelers were the first to develop an interest in the archaeology of the Indian subcontinent. Alexander Cunningham had a significant impact on Indian archaeology overall as the very first the artistic director of the Archaeological Survey of India. The pillars are the most important parts of a temple's structure in their original Sanskrit words. During the building processes of these temples, which took place several millennia ago, there was no software, calculators, or spreadsheets available. However, the pillars (Stambha) are an important part of Hindu temple design. The finished goods that these temples show off are amazing and are called “Architectural Wonders of Construction Engineering” [3].These pillars are still an inspiration to many people today, who are doing study on how to build pillars in modern temples.

1 CANARA Engineering College, Mangaluru-574219 , VTU, Belagavi-590018, INDIA.ORCID ID : 0000-0001-5968-2841
2 U.B.D.T College of Engineering, Davanagere-577004, VTU, Belagavi-590018, INDIA.ORCID ID : 0000-0001-9019-1733
3 J.N.N College of Engineering, Shivamoga-577204, VTU, Belagavi-590018, INDIA.ORCID ID : 0000-0001-6466-279X
4CANARA Engineering College, Mangaluru-574219 , VTU, Belagavi-590018, INDIA.
5A.J Institute of Engineering & Technology, Mangaluru-575006, VTU, Belagavi-590018, INDIA.
** Corresponding Author Email: devguruap4u@gmail.com*

The physical techniques utilized for pillar time-line classification and identification necessitate the involvement of specialists such as archaeologists. Occasionally, these methods are susceptible to error. In recent years, artificial intelligence has proliferated in the form of deep neural networks, and applications that rely on it have become ubiquitous. Artificial intelligence-driven deep learning is a burgeoning the field of academia that has been implemented successfully in a multitude of applications, such as digital image processing(DIP), natural language processing (NLP), speech processing and computer vision(CV) among others. Deep learning algorithms have demonstrated comparable performance and surpassed human specialists in image analysis and object recognition within various sectors, including agriculture, industry, and automotive. Deep learning, or deep supervised learning, is a machine learning technique that utilizes multiple layers to extract highly complicated data from a network. These features are subsequently utilized for tasks such as pattern recognition, classification and identification. The advancement of deep learning methodologies has necessitated the availability of robust computational resources.

This study examines the significance of Pillars Time-Line (Stambha) within the context of Hindu temple architecture. It proposes a more accurate approach to the classification and identification of ancient pillar time-lines using AI-based deep learning techniques. The findings of this research can assist archaeologists in formulating guidelines for the reconstruction of these remarkable temple architectural elements.

The next components of this work are structured as outlined. Section II's literature analysis offers a thorough assessment of recognition, classification, and identification procedures. Section III provides a concise overview of the data and methodology employed in the study. The details of the experimental configurations are addressed in Section IV. The findings of the experiment are given and analyzed in Section V. Section VI closes by providing the ultimate analysis and definitive conclusions.

1. Literature Survey

A comprehensive assessment of the literature was conducted to get knowledge on the latest advancements in recognition, classification & identification techniques within the primary research area of archaeology.

1.1. A study investigating several recognition techniques :

In this paper the authors introduced a new method called the Advance Recognition Algorithm (ARA) to recognize letter characters and determine the time period/era of Kannada inscriptions on stones from the Ganga and Hoysala periods. The letters from the Hoysala and Ganga periods are kept in a template. The character is recognized by form analysis and comparison of distinguishing traits. The approach of analyzing the mean value and sum of absolute difference values is used to match characters. The experimental results reveal a pretty good accuracy in detecting stone inscriptions from both the Ganga and Hoysala periods. It has shown improved time efficiency compared to earlier techniques [4].

This paper presents a novel methodology that employs Genetic Programming Evolved Spatial Descriptor for the identification of Indian tourist sites through the application of linear Support Vector Machine (SVM). The system comprises three main phases: 1. preprocessing, 2. genetic programming evolution, and 3. Categorization. The Preprocessing phase transforms images into a format appropriate for genetic programming system by utilizing Generalized Co-Occurrence Matrix. The second phase produces the most effective spatial descriptor depending on the fitness of the program. Fitness is determined by Support Vector Machine (SVM) calculations. Once the software is generated as output, it can be used for categorization. The system is developed using MATLAB and attains a high level of accuracy. The researchers attained a precision rate of 92.75% [5].

In this paper authors have suggested a real-time application that utilizes monument recognition to provide tourists with an interactive experience. This involves overlaying an instructive video, text, and image onto the live view of the monument. The primary goal of the system is to accurately recognize monuments with little misclassification and offer relevant information about them. Monument Recognition involves extracting features using the SURF (Speed Up Robust Features) algorithm and classifying them using SVM (Support Vector Machine). The experiment utilized a dataset consisting of 120 photographs capturing four different monuments. Specifically, the real-time execution of the experiment focused on the Shaniwarwada monument, while the remaining three monuments were analyzed using samples obtained from alternative sources. Out of the 15 pictures of Shaniwarwada, a total of 13 were successfully classified during the on-site testing process. Based on empirical evidence, the proposed approach demonstrates an average classification accuracy of 90% on training photographs, and 86.66% on test images [6].

In this paper authors suggests a system for identifying biological pollutants present on the surfaces of cultural heritage monuments. Identifying the specific biological pollutants is essential for safeguarding monuments against their detrimental effects. The suggested recognition system takes a sequence of photos depicting cultural heritage monuments as its input. The collection includes photos captured in the visible and near-infrared spectral ranges. When preprocessing a series, all photos are transformed to a chosen perspective and the background is eliminated. A feature vector is constructed using various formal vegetation indicators. A pre-trained classifier utilizing the SVM method with an RBF kernel is employed to identify the type of biological pollutants [7].

In this paper the authors have suggested creating an automated approach to recognize the structural features of ancient temples such as Pillars & Vimana for archaeological investigation. This suggested methodology utilizes GP-Genetic Programming to develop a spatial descriptor and categorize the structural aspects of temples explored by archaeologists using the Linear Discriminant Analysis (LDA) method. The proposed methodology comprises three main phases: 1. pre-processing, 2. genetic programming evolution, and 3. recognition. The Generalised Co-Occurrence Matrix is employed during the pre-processing phase to transform images into a format amenable to analysis by genetic programming systems. The next stage creates an advanced spatial descriptor program that is currently the most effective, with fitness as its foundation. The Fitness is determined using LDA. Upon receiving the previous step output, it might be utilized for recognition purposes. The empirical findings demonstrate significant accuracy in identifying both vimana (gopura) and the pillars found in different temples. The suggested approach will attain accuracy of 98.8% in identifying pillars and accuracy of 98.4% in recognizing vimanas[8].

In this paper the authors focuses especially on Character recognition and information extraction by using different preprocessing methods, such as scaling, grayscale conversion, brightness and contrast improvement, smoothing, noise reduction, morphological procedures, and thresholding. Here conducted a study using stone inscription photographs taken from the Tanjore Brihadeeswar Temple, which dates to the eleventh century, during the reign of Raja Raja Chola, in order to thoroughly evaluate these methods. Character recognition and information extraction are then applied to the processed outputs, with an emphasis on contrasting the results of different pre-processing techniques, such as grayscale conversion and binarization. This work attempts to elucidate optimal pre-processing procedures to enhance the legibility and preservation of ancient Indian script images inscribed on diverse stone substrates [9].

In this paper the approach suggested uses SAS Planet-obtained satellite imagery. It improves the identification of archaeological structures in densely vegetated areas by employing supervised machine learning alongside two colour spaces (RGB and HSL) and filters (Canny, Sobel, and Laplacian). The method demonstrates an average performance of no less than 93% in accuracy, precision, recall, and F1 score. Consequently, in contrast to traditional manual or semi-automatic structure detection methods, our proposal is an excellent option for the expedited identification of archaeological sites. This study seeks to elucidate optimal pre-processing techniques to enhance the clarity and preservation of ancient Indian script images inscribed on diverse stone substrates [10].

In this paper the main focus is on the recognition and preservation of historical texts by demonstrating the potential of contemporary deep learning techniques in archaeological research. Several significant discoveries and scientific advancements are the result of our investigation. This analysis examines the performance of YOLOv8 and Roboflow 3.0 in Palmyrene character segmentation, emphasising the strengths and weaknesses of each algorithm in this specific scenario. The segmentation models are trained and assessed using the dataset. We provide bespoke visualization tools for anticipated segmentation masks and use comparable assessment criteria to quantitatively evaluate the segmentation outcomes, guaranteeing the accuracy and reproducibility of our conclusions. This study establishes a benchmark for future research and advances the field of semi-automatic reading of Palmyrene inscriptions [11].

In this paper main focus on end-to-end cross-scene recognition model that incorporates the self-attention Transformer encoder module (TEM), the normalized generator module (NGM), and the structural-semantic extractor (SSE). Tests demonstrate that our suggested NGM can resolve the task's multi-scene compatibility issue by learning a transfer mapping relationship from numerous scenes to standard scenes. SSE lowers the spatial dimension of sample characteristics and eliminates unnecessary background pixels. TEM recognizes the global structural aspects of characters after calculating the attention between local patches to create a global relationship matrix. Our model outperforms other state-of-the-art in terms of recognition accuracy in the cross-scene ancient character datasets [12].

In this paper main focus on identifying and recognizing buildings, together with their attributes, we require a model or technology. Technology would be required to assist them, though, as recognizing and identifying them is a challenge in and of itself. A subset of artificial intelligence technology centered on pattern recognition and image processing, the Convolutional Neural Network model is the technology or model that will be employed in this study. There are various steps in this procedure. The initial stage utilises Gaussian Blur, SuCK, and CLAHE methods, which are advantageous for image recognition and enhancement. The subsequent step involves the extraction of features from constructing image properties. The third procedure, involving the retrieval of building images based on their attributes, will be underpinned by the findings from image processing. Gaussian Blur of 88.96% is the greatest accuracy in the feature extraction process retrieval using the DenseNet 121 model with the starting procedure. 88.46%, 88.3 and 87.8% uses the SuCK technique, and 87.7% and 87.6% using CLAHE [13].

This paper examines the ancient Tamil script and presents a novel algorithmic approach for the reorganisation of characters and the contextual interpretation of temple inscriptions. This approach addresses challenges posed by noisy images, script variations, and the understanding of historical context through the integration of advanced preprocessing techniques, deep learning models, and contextual analysis. We assembled a collection of 100 high-resolution photographs of temple inscriptions from various eras and locations. The preprocessing stage enhances the quality of inscription images through the application of algorithms for adaptive binarization, noise reduction, contrast enhancement, and orientation correction. Convolutional neural networks utilising transfer learning are employed in the character recognition phase, while the multi-head attention mechanism of Vision Transformers (ViT) enhances this process. The character segmentation algorithm utilised was the Stroke Width Transform. Transfer learning was employed for adjustment. The results demonstrate the effectiveness of the proposed methodology. The accuracy metrics for character recognition include a precision of 97.25%, a recall of 95.05%, and an F1-score of 95.17%. The model achieved a recognition rate of 98.92% for significant phrases related to historical events, deities, and rulers. Furthermore, it demonstrated a 95% identification rate for historical dates and a 94% recognition rate for context-specific terms [14].

This paper focusses on the following steps that constitute the technique: (1) Conducting LiDAR surveys and data collection; (2) Processing LiDAR data to generate the Digital Elevation Model (DEM); (3) Enhancing outcomes through the development of derived models based on Visual Techniques (VTs) to facilitate the identification of potential looting features; (4) Implementing pattern recognition based on geomorphology to semi-automatically extract potential looting features; (5) Validating results through (a) field verification of multiple looted tombs using GNSS in Real The developed method demonstrates satisfactory predictive capability, achieving a success rate between 85% and 95% due to the advantageous penetration capability of LiDAR [15].

In this paper focuses on using a deep learning model as a classifier and feature extractor to recognize 33 classes of fundamental characters found in Devanagari ancient manuscripts. For the experimental study, a dataset with 5484 characters was employed. Numerous tests demonstrate that CNN outperforms other cutting-edge methods in terms of feature extraction accuracy. The model presented in this research has been used to recognize Devanagari ancient characters with a recognition accuracy of 93.73% [16].

1.2. A study investigating several classification and identification techniques :

This paper demonstrates the application of Content-Based Image Retrieval (CBIR) methods for the automated classification of archaeological structures. It employs visual cues such as shape and texture to analyse the art style and retrieve similar images from a reference database. Morphological operators extract shape-based data, while the grey level co-occurrence matrix (GLCM) is employed to extract texture features. A comprehensive feature set is developed to identify similar photographs. Experiments were conducted on a dataset comprising 500 photographs organised into five categories. The results of the proposed method are compared with those of the Canny and Sobel methods. The empirical findings of this study indicate that the average retrieval rate of the proposed method is 78% [17].

This paper presents a system that employs a spatial descriptor developed through Genetic Programming to classify Indian sites frequented by visitors, utilising a linear Support Vector Machine (SVM). The framework is organised into three main stages: preprocessing, the evolution of genetic programming, and categorisation. The preprocessing phase converts images into a suitable format for analysis by a genetic programming system that employs a Generalised Co-Occurrence Matrix. The second phase generates the most efficient spatial descriptor based on the program's fitness. Fitness is assessed through calculations using Support Vector Machines (SVM). Upon generation of the software output, it becomes applicable for categorisation purposes. The system has been developed with MATLAB, achieving a notable level of accuracy. The study indicates that the Indian monument database system attained an average accuracy of 97%, demonstrating impressive precision and recall [18].

This paper introduces a method that employs retrieved information to categorise Indian monuments frequented by visitors through the use of a linear Support Vector Machine (SVM). The proposed system was divided into three main phases: preprocessing, feature vector generation, and classification. The features originate from techniques such as Local Binary Pattern, Histogram, Co-occurrence Matrix, and Canny Edge Detection. Following the development of the feature vector, classification was performed using Linear SVM. A comprehensive database comprising 10 renowned Indian monuments has been established, featuring 50 photographs for each monument. The system has been developed utilising MATLAB and achieves remarkable precision. The system in question was additionally assessed using other commonly utilised benchmark databases [19].

This research paper proposes two novel ways for classifying Indian monuments according to their distinctive architectural styles. While the historical significance of numerous Indian monuments is well-documented, the details of their architectural styles are not as comprehensively recorded. Numerous Indian architectural styles sometimes display shared traits, complicating classification efforts. The researchers propose two techniques for classifying monuments according to their styles: Radon Barcodes and Convolutional Neural Networks. The initial approach is efficient and exhibits lower memory consumption; however, the alternative method attains an accuracy of 82%, above the 76% accuracy of the former. The Indian dataset attained a classification accuracy of 82.31%, a precision of 0.81, a recall of 0.82, and an F1-score of 0.82 [20].

In this paper authors suggest a monument classification scheme that employed a hierarchical structure of collaborating artificial neural networks (ANNs). Every artificial neural network (ANN) was trained to execute a distinct logical component of the comprehensive task of classifying monuments. Two distinct datasets were created from a pool of 125 classified monuments. The initial group, comprising 80 individuals, was solely utilized for training objectives. The second set, comprising 45 members, was kept unseen and utilized to determine the point at which optimal generalization had been achieved. Based on the ultimate artificial neural network (ANN) models, it was seen that none of the 45 previously unseen monuments were misclassified. Moreover, measures were taken to increase the level of "strength of conviction" associated with each of these classes [21].

This research paper presents a mechanism for image comparison based on the spectral characteristics of the monument being studied. This method has shown the capability to classify the main categories of organisms found on monuments and assess the degree of biological deposit on surfaces throughout time. This methodology was utilised in the examination of ancient petroglyphs in the Republic of Khakassia, Tomsk petroglyphs, and many historical locations in St. Petersburg [22].

In this paper the authors have devised techniques for the automated detection of cracks in historical monuments and the identification of damaged prominent facial features, with the aim of facilitating digital restoration. Out of a total of 40 photographs, it was seen that 50 places exhibited signs of damage, while the remaining 50 spots remained unaffected. A total of 50 regions that were not affected were successfully classified, while 47 out of 50 damaged regions were appropriately categorized. The source region was accurately identified in 49 out of 50 regions. The performance measures for Damaged and Undamaged, specifically recall and precision, are reported as 0.94, 1.00, and 1.00, respectively [23].

In this paper the authors focuses on the application of computer vision techniques in the field of archaeology, specifically in the classification of medieval coins. The study adopts an independent approach to this classification process. The present study aims to provide instruction on the coin classification method pertaining to a collection of Dutch early-medieval coins. The collection comprises a total of 2,270 unique coin faces, encompassing 692 distinct coin types. The method accurately classifies around 78% of the coins in the test set. Misclassifications often arise due to the presence of heavily soiled or unidentified coins [24].

According to the review of the literature, numerous methodologies, including Advance Recognition Algorithm (ARA), Genetic Programming Evolved Spatial Descriptor(GPESD) Algorithm, SURF (Speed Up Robust Features) algorithm, SVM (Support Vector Machine) method with an RBF kernel, Content-Based Image Retrieval (CBIR), Linear Discriminant Analysis (LDA) method, Convolutional Neural Networks(CNNs), Artificial Neural Networks (ANNs), are used for recognition, classification & identification of Indian monuments. The review of the literature also showed that deep learning algorithms fared well in the recognition, classification & identification of Indian monuments, outperforming other techniques and their success is mostly due to a modification in the neural network architecture that gradually adds more layers.

To the best of knowledge gain through the literature review, there hasn't been much research on the creation of deep learning models for the classification & identification of ancient pillars time-lines (Era) based on Potika type/architecture level for images of temple pillars. This research work proposes five deep learning convolutional neural network (DCNN) models, namely DenseNet121, VGG16, InceptionV3, MobileNetV2, and Xception models for classifying and identifying ancient pillars time-line, aiming to achieve higher accuracy.

2. Data Set and Methodology

2.1. Temple Pillar Time-Line(Era) Dataset

When working with customized data, it is very important to make sure you have the right information. Using a high-quality dataset is crucial for achieving the highest level of accuracy [25].The images for dataset are captured from different historical ancient temples in and around the places of Karnataka state such as: temples in shivamogga district, temples in Chikamagaluru district, some temples of Hampi from vijayanagar district, some temples of Badami & Pattdkallu from Bagalkote district& some temples in Mysore district. The following are the temples names from which pillar images were obtained from several temples such as : Amrutheshwara Temple from Amruthapura, Yoga Narasimha Temple from Baggavalli, Aghoreshwara Temple from Ikkeri, Parvathi-Brahmeshwara Temple from Kudli, Sri Malikarjuna Temple from Hire-Nalluru, Hucharaya Swamy Temple from Shikaripura, 1st Cave & 3rd Cave Temple from Badami, Choleswara & Nageswara Temple from Beguru, Papanatha Temple from Pattdkallu.

We have enhanced our dataset for AIML methods by include a variety of images obtained from the Archaeological Survey of India portal and the Trip-Advisor travel portal. Images are captured in diverse postures for the purpose of training and assessment. Images are captured of each class at various times during the day, such as morning, afternoon, and evening, from the front, far away, left-side 45-degree, and right-side 45-degree angles. To achieve diversity, the positioning of subjects in front of temples, whether sitting or standing, is considered. Figure 1 displays a few representative images of six distinct classifications of pillar time-lines according to potika architecture. The dataset is only accessible for research on the Kaggle and JNNCE college websites [26, 27].

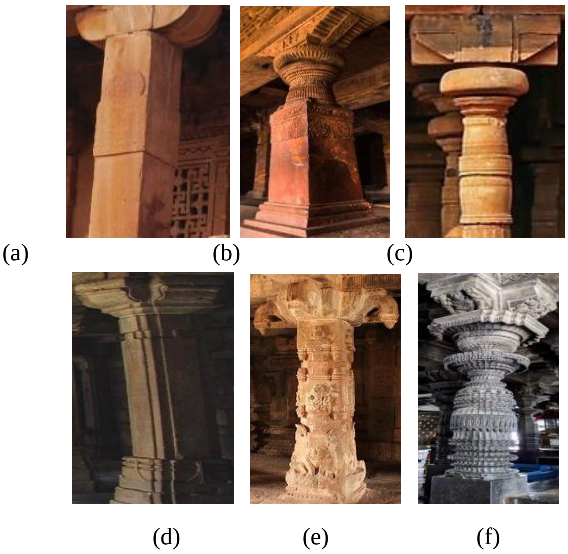


Fig. 1. Sample Images of different time-lines of Pillars based on potika architecture levels. (a) Rutha-Potika (4th-6th BC) , (b) Tharanga-Potika(5th-9th BC), (c) Aadhaar-Potika (9th - 13th BC), (d) Mustibhanda-Potika(11th - 14th BC) (e) Pushpa-Potika (15th - 16th BC) (f) Chitra-Potika (16th BC Onwards).

The camera used for capturing images is a Samsung model, namely the SM-G615F, with an aperture size of f/1.7. Archaeology professionals have confirmed the various architectural designs of historical pillars. Specialists and archaeologists classify Ancient Indian Pillars into six different eras based on the pillars' potika construction levels, as depicted in figure 1.Each image in the dataset is labeled by an expert to indicate different classes based on potika architecture type/level, ranging from 4th to 8th BC, 5th to 9th BC, 9th to 12th to 13th BC, 11th to 13th to 14th BC, 15th to 16th BC, and 16th BC Onwards as shown in Table 1. Table 1 presents the description of ancient temple Pillar Time-Line dataset.

TABLE 1. Ancient Temple Pillar Time-Line (Era) Dataset Description.

Sl.No.	*Pillar Time-Line(Era)*	*Potika architecture levels*	*Total Number of Samples*
1.	4th to 8th BC	Rutha Potika	226
2.	5th to 9th BC	Tharanga Potika	152
3.	9th to 12th to 13th BC	Aadhaar Potika	153
4.	11th to 13th to 14th BC	Mustibhanda Potika	167
5.	15th to 16th BC	Pushpa Potika	124
6.	16th BC Onwards	Chitra Potika	208
Total images in Dataset →			**1030**

Understanding the timelines is crucial for reconstructing temples. The probability of identifying time-lines can be done by studying the old temple pillars and potika constructions. Currently, research on the timelines of ancient temple pillars is frequently needed by archaeologists and research academics to extract archaeological information through digitization.

2.2. Proposed Methodology

This current study has selected five unique and established deep convolutional neural network (DCNN) architectures extensively used in computer vision. The architectures presented are DenseNet121, VGG16, InceptionV3, MobileNetV2, and Xception. These models known for their exceptional performance on large datasets like Image-Net and CIFAR-100. The models have shown outstanding performance while using less computational power.

A deep convolutional neural network (DCNN) extracts features from modified unprocessed images received as input. The extracted characteristics are utilized to train the classifier segment of the DCNN. A Convolutional Neural Network (CNN) consists of essential components such as the input layer, convolutional layer, Rectified Linear Unit (ReLU) layer, max-pooling layer, and fully connected layer, commonly referred to as the output layer. Transfer learning is employed to train the suggested deep convolutional neural network (DCNN) models. The weights are computed via a pre-trained ImageNet database [28].

The Image-Net dataset comprises over 1.2 million images and covers 1000 classes. The ancient temple pillar time-line dataset utilized in this study consists of 1030 original images that were labeled by archeologists without any augmentation. This dataset has six distinct categories, as illustrated in Table 1.The DenseNet121, VGG16, InceptionV3, MobileNetV2, and Xception deep convolutional neural network (DCNN) models are initialised with pre-trained ImageNet weights across all layers and trained until convergence. The network is optimised via the stochastic gradient descent (SGD) method to minimise the loss function. CNN models are assessed by examining their performance on the test dataset and computing cross-entropy loss. All model implementations originate from the open-source deep learning framework Keras. The predictive accuracy of the individual deep convolutional neural network (DCNN) model is assessed by partitioning the dataset of 1,030 photos from the Ancient Pillar timeline dataset into a training set (80% - 825 images) and a testing set (20% - 205 images). The validation set consists of 103 photos, representing 10% of the training dataset.

The proposed method comprises ten unique phases. Figure 2 displays a detailed overview of the conducted research. The efficacy of classifying using pre-trained DenseNet121, VGG16, InceptionV3, MobileNetV2, and Xception convolutional neural network (CNN) models is assessed by using Equations (1) and (2). Algorithm 1 showcases the DCCN classification algorithm developed through the utilization of transfer learning.

Classification efficiency (%) =

$$\frac{\text{Sample images that have been correctly classified}}{\text{The total number of sample images available for testing}} \times 100 \quad (1)$$

Average classification efficiency (%) =

$$\frac{\text{Sum of sample images classified correctly}}{\text{Total sample images}} \times 100 \quad (2)$$

Algorithm 1: Classification of Pillar Time-Line (Era) based on Potika Architectural Features using Deep Convolutional Neural Network (DCNN).

Aim: Algorithm takes input images from the dataset consists of unprocessed raw pillars images that have not been processed, and the algorithm uses these images to retrieve features. In order to classify the images into respective time-line, their essential features are evaluated using both base layers and top layers. Then the algorithm produces the classification results using the condensed confusion matrix.

Parameters: DCNN learning rate is represented by η; batch size is by β; epochs are by ε; Image-Net weights are by w; each iteration step by ξ; and the number of training instances in each iteration is by n. P(xi=k): the likelihood that the input xi will be connected to the anticipated kth class, where k is the classes index.

Input: Pillars' raw images as input.

Output: Classification results w.r.t time-line.

Step 1: Read Images from Pillar Time-Line Dataset.

Step 2: Generate two sets of the Pillar Time-Line Dataset: a training set of 80% and a testing set of 20%. Select samples randomly before creating the labeled datasets for training, validation, and testing.

Step 3: Assemble the CNN models' input, hidden, and output layers in the appropriate order.(DenseNet121, VGG16, InceptionV3, MobileNetV2 and Xception).

Step 4: Construct the convolutional, pooling, flattening, dense, and dropout layers in sequence for the CNN models in the upper layers.(DenseNet121, VGG16, InceptionV3, MobileNetV2 and Xception).

Step 5: Enter the parameters ε, β, η and upload the pre-trained CNN and assign network weights (w1, w2,....wn). Set the network parameters and choose weights in this step of the process.

Step 6: Train the CNN and determine the starting weights and at the end of training calculated weights should be updated & saved in the database.

for ξ =1 to ε do

Pick a mini-batch (size: β) at random from X_{train}

Utilising expression Eq. (3), do forward propagation and compute loss (E).

$$E(W) = -\frac{1}{n}\sum_{xi=1}^{n}\sum_{k=1}^{k}\left[\, y_{ik}\log P(x_i = k) + (1 - y_{ik})\log\left(1 - P(x_i = k)\right)\right] \quad (3)$$

Step 7: Expression Eq. (4) is used for back-propagation and weight adjustments with SGD.

$$W = W_{k-1} - \eta(\partial E(W)/\ \partial W) \quad (4)$$

With Adam Optimizer, update weights.

End

Step 8: Calculate the CNN models' X_{test} Maximum Classification Accuracy using the test dataset based on condensed confusion metrics.

Step 9: If X_{test} is Maximum go to step 10, else go back to step 5.

Step 10: Display Classification Results.

3. Experimental Research Testimonials

The recommended approaches for all five DCNN model architectures are implemented in Keras 1.0 using Tensorflow[29]. The investigations were carried out on an Ubuntu Linux server powered by an Intel(R) Core(TM) i7-4600U CPU with 16 GB of memory and a frequency of 2.69 GHz.

3.1. DCNN Classification Models Accuracy and Loss in Training and Validation

3.1.1. DenseNet121 :

DenseNet121 requires a 224-by-224-pixel RGB image as input. DenseNet121 consists of almost 8 million layers. Dense Blocks are subsets of feature maps that have the same dimensions but different filter counts.

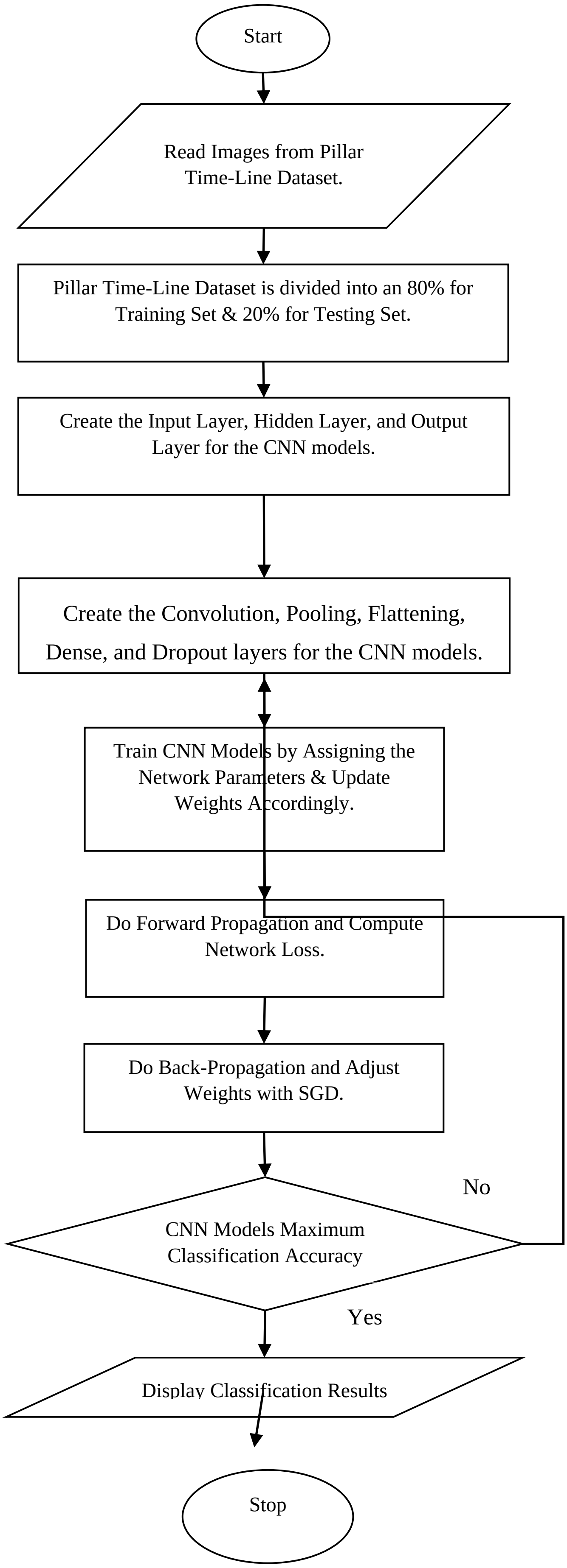


Fig. 2 **Schematic overview of the proposed methodology.**

Batch normalization is utilized for down-sampling in transition layers located between the blocks. The experiment utilized a batch size of 32, 40 epochs, and a learning rate of 0.0001 to train the model. The desertion rate remains at its default setting of 0.2 percent. The network is trained using the Adam Optimizer.

3.1.2. MobileNetV2:

MobileNet-v2 is a CNN of 53 layers. The highest achievable resolution for uploading images is 224 pixels by 224 pixels. The MobileNetV2 design utilizes an inverted residual structure instead of employing extended representations at the input and output of the residual block, as is commonly done in standard residual models. MobileNetV2 employs efficient depth-wise convolutions to selectively process features in the intermediate expansion layer. In this research, the network is trained using the Adam optimizer for 40 epochs, with a batch size of 32.

3.1.3. VGG 16:

VGG16 requires an input of a 224x224 pixel RGB image. The neural network consists of 16 layers, which include a softmax classifier, three fully connected layers, max-pooling for dimensionality reduction, and 13 convolutional layers. The research specifies that the network's maximum epoch count is 40 and the learning rate is 0.0001. The network's maximum batch size is 32. The Adam Optimizer is used to train the network.

3.1.4. InceptionV3:

InceptionV3 consists of 484 layers, which are composed of 11 inception modules. The input image has a resolution of 299 by 299 pixels. Each module contains convolution filters, pooling layers, and ReLu activation functions. InceptionV3 use convolutions to reduce the number of learning parameters while maintaining network performance. InceptionV3 suggested compressing new features. The images in this research have been resized to 224X224 pixels. The network is trained using the Adam optimizer for a maximum of 40 epochs and a batch size of 32.

3.1.5. Xception :

Xception is a unique category of convolutional neural networks. The pretrained network Xception can classify a mouse, a keyboard, various animals, and more into 1000 different item categories. The network supports photos with a resolution of 299x299. The images in this research have been resized to 224X224 pixels. The Xception design relies on 36 convolutional layers for feature extraction. The data follows the input flow initially, then the middle flow (repeated eight times), and lastly the exit flow.

Remember that batch normalization is applied after each layer of convolution and separable convolution, even if it is not shown in the picture. The Separable Convolution layer uses a depth multiplier of 1, avoiding depth expansion.

3.2. Comparative analysis of the training and validation characteristics of DCNN models

Table 2 displays comparisons of training and validation performances for all five models. From Table 2, the DenseNet121 model achieved the highest training efficiency of 0.9652, while the MobileNetV2, VGG16, InceptionV3, and Xception models achieved training efficiencies of 0.9405, 0.9353, 0.9174, and 0.9005, respectively. For validation efficiency, the DenseNet121 model achieved the highest score of 0.9484, followed by the MobileNetV2, VGG16, InceptionV3, and Xception models with scores of 0.9219, 0.9112, 0.8944, and 0.8891, respectively. The DenseNet121 model has demonstrated exceptional efficiency on both the training and validation datasets across various training methods, achieving the best training time.

Table 2 : Comparison of training and validation performance(computational efficiency) results of various DCNN models over 40 epochs for the pillars time-line dataset.

Sl. No	*DCNN Model*	*Image size (Pixels)*	*Training accuracy*	*Training Loss*	*Validation accuracy*	*Validation Loss*	*Training time (hh:mm:ss)*
1	***DenseNet121***	224 × 224	0.9652	0.0497	0.9484	0.0699	01:08:38
2	***MobileNetV2***	224 × 224	0.9405	0.0745	0.9219	0.0964	00:55:58
3	***VGG16***	224 × 224	0.9353	0.0796	0.9112	0.1071	00:52:59
4	***InceptionV3***	224 × 224	0.9174	0.0972	0.8944	0.1239	00:50:48
5	***Xception***	224 × 224	0.9005	0.1179	0.8891	0.1292	00:49:18

Table 3: Performance metrics of all 5 models for the classification of pillar time-lines (Era).

Sl. No	*DCNN Models/ Pillar Time-Line (Era)*	*Evaluation parameters*														
		DenseNet121			*MobileNetV2*			*VGG16*			*InceptionV3*			*Xception*		
		Precision (%)	*Recall (%)*	*F1 score (%)*	*Precision (%)*	*Recall (%)*	*F1 score (%)*	*Precision (%)*	*Recall (%)*	*F1 score (%)*	*Precision (%)*	*Recall (%)*	*F1 score (%)*	*Precision (%)*	*Recall (%)*	*F1 score (%)*
1	*4th_8th_BC*	0.90	0.91	0.89	0.95	0.95	0.92	0.87	0.91	0.89	0.79	0.91	0.87	0.83	0.86	0.80
2	*5th_9th_BC*	0.89	0.86	0.84	0.86	0.86	0.85	0.88	0.86	0.90	0.93	0.71	0.90	0.77	0.71	0.87
3	*9th_12th_13th_BC*	1.00	0.95	0.92	0.89	0.95	0.92	0.89	0.91	0.89	0.80	0.95	0.79	0.86	0.91	0.87
4	*11th_13th_14th_BC*	0.87	0.92	0.89	0.85	0.88	0.84	0.88	0.92	0.88	0.87	0.92	0.85	0.93	0.88	0.82
5	*15th_16th_BC*	0.84	0.93	0.91	0.85	0.87	0.82	0.91	0.93	0.91	0.80	0.87	0.83	0.85	0.93	0.86
6	*16th_BC_onwards*	0.90	0.91	0.90	0.89	0.91	0.88	0.84	0.82	0.78	0.81	0.91	0.79	0.79	0.82	0.79
Weighted Average (%)		**90.00**	**92.00**	**89.16**	**88.16**	**90.3**	**87.16**	**87.83**	**89.17**	**87.5**	**83.33**	**87.83**	**83.83**	**83.83**	**85.17**	**83.50**

Similarly from Table 2, the most efficient training loss values are 0.0497, 0.0745, 0.0796, 0.0972, and 0.1179 for the DenseNet121, MobileNetV2, VGG16, InceptionV3, and Xception models, respectively. The most efficient validation loss values are 0.0699, 0.0964, 0.1071, 0.1239, and 0.1292 for the same models. The DenseNet121 model has demonstrated better loss efficiency on both the training and validation datasets across various training methodologies.

4. Results and Discussion

4.1. Performance analysis parameters

When the model accurately predicts the positive class, it is called a True Positive (TP). A True Negative (TN) model is one that accurately forecasts the absence of the positive class. When the model forecasts the positive class inaccurately, it produces a False Positive (FP).The metrics utilized to evaluate classification performance include precision, recall, and f1-score.

$$Precision = \frac{TP}{(TP + FP)} \quad (5)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (6)$$

$$f1 - score = \frac{(Precision * Recall)}{(Precision + Recall)} \quad (7)$$

4.2. Results of DCNN Models for Pillars Timeline Classification & Identification

Evaluating the performance of DenseNet121, MobileNetV2, VGG16, InceptionV3, and Xception models using a test image dataset, and the confusion matrices are plotted for the test results of classification models for pillars timelines as shown in Figures 3, 4, 5, 6 and 7.

As indicated in Table 3, the confusion matrices are used to calculate the performance measures for each of the five models, including precision, recall, and F1-score.

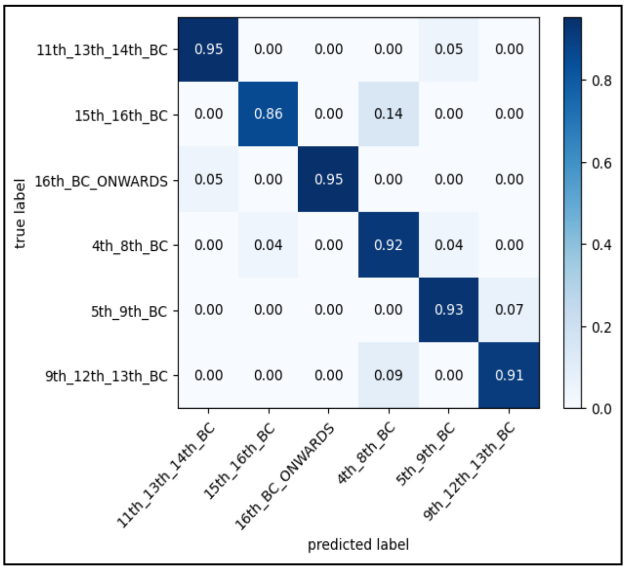


Figure 3: Confusion matrix generated by the DenseNet121 model for classifying pillar time-lines.

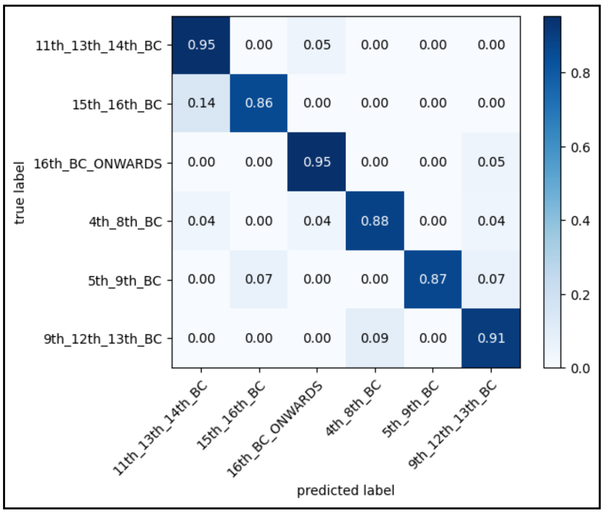


Figure 4: Confusion matrix generated by the MobileNetV2 model for classifying pillar time-lines.

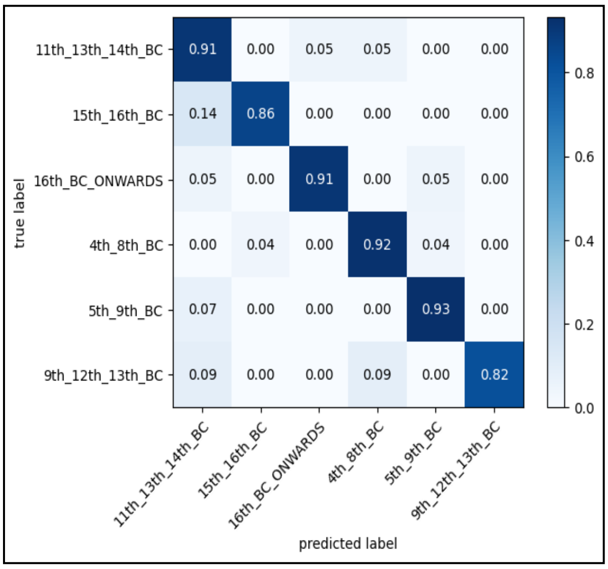


Figure 5: Confusion matrix generated by the VGG16 model for classifying pillar time-lines.

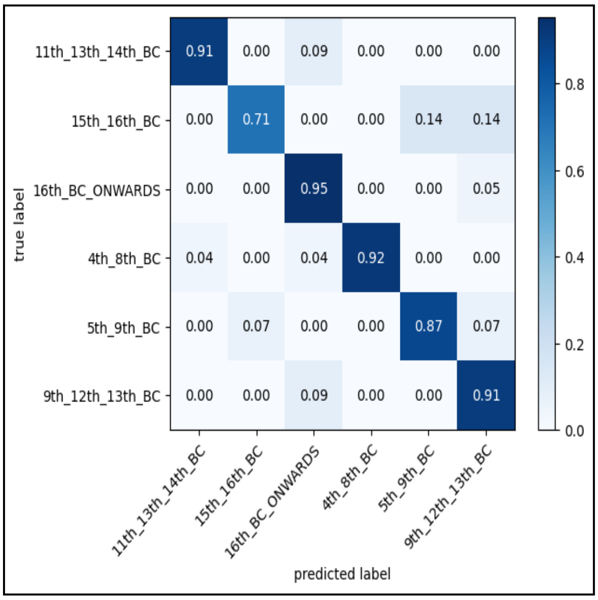


Figure 6: Confusion matrix generated by the InceptionV3 model for classifying pillar time-lines.

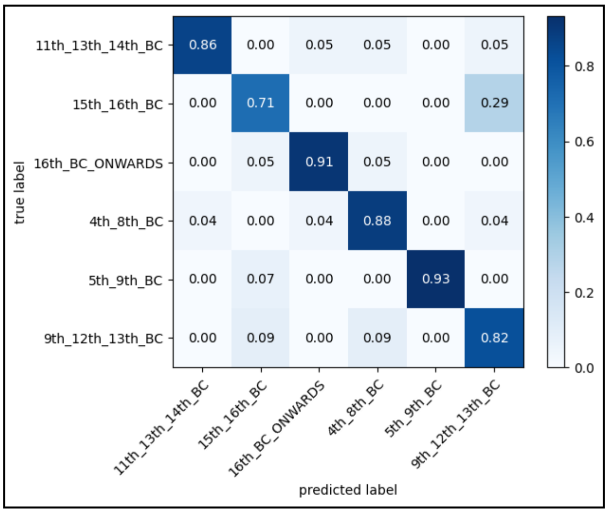
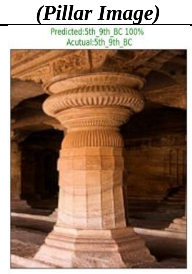
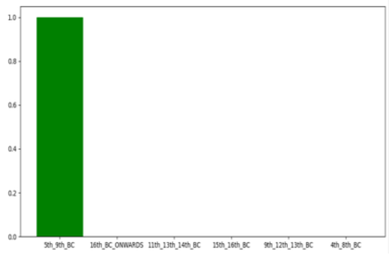
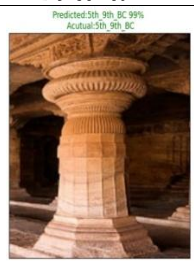
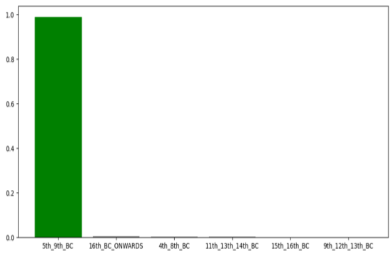
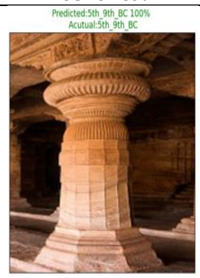
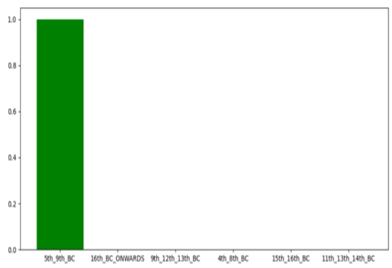
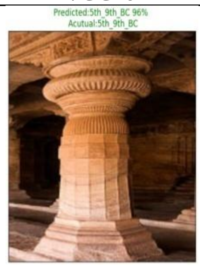
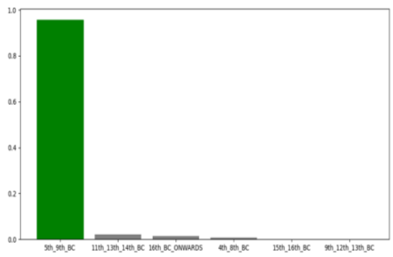
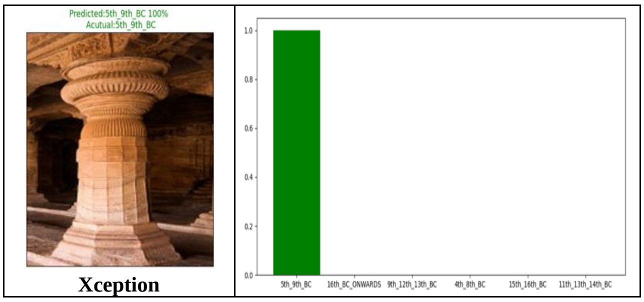


Figure 7: Confusion matrix generated by the Xception model for classifying pillar time-lines.

Table 4: Classification Accuracy of pillar Time-Line (Era) using the all five DCNN models.

Input to Models (Pillar Image)	***Output form the Models (Pillar Time-line(Era))***
 **DenseNet121**	
 **MobileNetV2**	
 **VGG16**	
 **InceptionV3**	



Xception

Table 4 shows the classification accuracy of pillar Time-Line (Era) using the all five DCNN models. Table 5 presents a summary and comparison of the test findings obtained from Tables 3 and 4. Table 5 shows that the DenseNet121 model outperforms the MobileNetV2, VGG16, InceptionV3, and Xception models in predicting the pillar timeline from temple field pillar images. It achieved the highest average pillar timeline classification efficiency of 92.00%. The test results indicate that the MobileNetV2, VGG16, InceptionV3, and Xception models performed similarly, achieving average pillar timeline classification accuracies of 90.33%, 89.17%, 87.83%, and 85.17% respectively.

Table 5: Comparison of the Classification Accuracy of the results for pillar time-line (Era) classification derived from all five models.

Sl. No	***Pillar Time-Line***	***Models Classification Accuracy (%)***				
		DenseNet121	***MobileNetV2***	***VGG16***	***InceptionV3***	***Xception***
1	***4th_8th_BC***	0.91	0.95	0.91	0.91	0.86
2	***5th_9th_BC***	0.86	0.86	0.86	0.71	0.71
3	***9th_12th_13th_BC***	0.95	0.95	0.91	0.95	0.91
4	***11th_13th_14th_BC***	0.92	0.88	0.92	0.92	0.88
5	***15th_16th_BC***	0.93	0.87	0.93	0.87	0.93
6	***16th_BC_onwards***	0.91	0.91	0.82	0.91	0.82
Average Accuracy (%)		**92.00 %**	**90.33 %**	**89.17 %**	**87.83 %**	**85.17 %**

5. Conclusions and Future Scope

State-of-the-art deep learning models utilizing a transfer learning strategy are employed in this research to classify pillars time-line (era) based on potika architecture at different time periods. Deep learning models can assist archaeologists in classifying the chronological timeline (era) of pillars in ancient Indian temple, at the same time minimizing computational processing needs. The experimental results demonstrate that the DenseNet121 model's average classification efficiency in classifying pillar timelines reached 92.00%, the highest ever. The test findings show that the MobileNetV2, VGG16, InceptionV3, and Xception models had comparable performance, with average classification accuracies of 90.33%, 89.17%, 87.83%, and 85.17% respectively in classifying pillar timelines. The DenseNet121 model surpasses remaining all 4 models by achieving the highest computational efficiency of Training -96.52% & Validation – 94.84% with very minimum losses. On the ancient temple pillar time-line dataset, DenseNet121 surpasses VGG16, InceptionV3, MobileNetV2, and Xception DCNN models in classification accuracy and computational efficiency. Trained models can significantly decrease the effort for archaeologists and novices, while enhancing the precision and efficiency in developing norms and standards for repairing intricate temple architectural structures.

The research progress was hindered by the constraints of the existing computational resources. Optimizing the hyper-parameters and scaling up the dimensions of the input images are likely to enhance the performance of the five models. One can utilize more commonly renowned models such as Google-Net, Nas-Net-Large, and Inception-ResNet-V2. Additionally, the existing dataset of pillar images can be broadened for identification by integrating drone shots and videos.

Acknowledgements

We express our gratitude for the assistance and collaboration of Dr. C. S. Vasudevan and Dr. S. Y. Somashekhar, Professors in the Department of Archaeology and Ancient History at Kannada University, Hampi, India. Both individuals have aided us in gathering datasets, annotating them, and offering crucial archaeological expertise for the ongoing study. We express our gratitude to the Archaeological Survey of India for permitting us to capture photographs of temple pillars, facilitating our visits to historic temples, and offering assistance from their personnel during our trips.

Author contributions

Dr. Gurudeva S Hiremath : Conceptualization, Methodology, Software, Field study**,** Data curation, Writing-Original draft preparation, Validation. **Dr. Shrinivasa Naik C.L. & Dr. Narendra Kumar S :** Visualization, Investigation, Writing-Reviewing and Editing. **Mr. Kiran A & Dr. John P Veigas** : Reviewing and Editing and proof reading.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Free, J. P., & Vos, H. F. (1992). *Archaeology and Bible history*. Zondervan.

[2] Ceram, C. W. Gods, Graves & Scholars. Victor Gollancz & Sidgwick & Jackson, London, 1954.

[3] Kumar, M. (2018). Pillars (Stambh)-The Supportive Elements of Hindu Temples. *International journal of current research*, *10*(06), 70223-70228.

[4] Mohana, H. S., & Rajithkumar, B. K. (2014, July). Era identification and recognition of Ganga and Hoysala phase Kannada stone inscriptions characters using advance recognition algorithm. *In 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT) (pp. 659-665).* IEEE.

[5] Bhatt, M. S., & Patalia, T. P. (2015, November). Genetic programming evolved spatial descriptor for Indian monuments classification. In *2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)* (pp. 131-136). IEEE.

[6] Jethale, A., Nath, N. V., Arawkar, T., Bajpeyi, A., & Nirwal, D. (2018). Monument Informatica: A Tour based Guide system using Real Time Monument Recognition. *Research Journal of Engineering and Technology*, *9*(4), 373-379.

[7] Grishkin, V., Kovshov, A., Schigorec, S., Vlasov, D., Zhabko, A., & Iakushkin, O. (2015, October). A system for the recognition of biofouling on the surface of the monuments of cultural heritage. In *2015 International Conference" Stability and Control Processes" in Memory of VI Zubov (SCP)* (pp. 630-633). IEEE.

[8] Gurudev S. Hiremath, Narendra Kumar S, Shrinivasa Naika C. L.(2023, March).A Study on Ancient Temple Structural Elements Recognition Using Genetic Algorithm, *In International Journal of Engineering and Manufacturing (IJEM)*, Vol.13, No.3, pp. 34-47.

[9] Jayanthi, J., & Maheswari, P. U. (2024). Comparative study: enhancing legibility of ancient Indian script images from diverse stone background structures using 34 different pre-processing methods. *Heritage Science*, *12*(1), 63.

[10] Fuentes-Carbajal, J. A., Carrasco-Ochoa, J. A., Martínez-Trinidad, J. F., & Flores-López, J. A. (2023). Machine learning and image-processing-based method for the detection of archaeological structures in areas with large amounts of vegetation using satellite images. *Applied Sciences*, *13*(11), 6663.

[11] Hamplová, A., Lyavdansky, A., Novák, T., Svojše, O., Franc, D., & Veselý, A. (2024). Instance Segmentation of Characters Recognized in Palmyrene Aramaic Inscriptions. *CMES-Computer Modeling in Engineering & Sciences*, *140*(3).

[12] Gao, J., Giunchiglia, F., Zhao, T., Li, C., & Xu, H. A Cross-Scenes Ancient Character Recognition Model Based on Normalized Generator and Transformer Encoder in Computational Archaeology. *Available at SSRN 4741242*.

[13] Rangkuti, A. H., Athala, V. H., Indallah, F. H., Tanuar, E., & Kerta, J. M. (2023). Optimization of Historic Buildings Recognition: CNN Model and Supported by Pre-processing Methods. *JOIV: International Journal on Informatics Visualization*, *7*(4), 2230-2239.

[14] Bhuvaneswari, S., & Kathiravan, K. (2024). Enhancing epigraphy: a deep learning approach to recognize and analyze Tamil ancient inscriptions. *Neural Computing and Applications*, *36*(31), 19839-19861.

[15] Danese, M., Gioia, D., Vitale, V., Abate, N., Amodio, A. M., Lasaponara, R., & Masini, N. (2022). Pattern recognition approach and liDAR for the analysis and mapping of archaeological looting: application to an etruscan site. *Remote Sensing*, *14*(7), 1587.

[16] Narang, S. R., Kumar, M., & Jindal, M. K. (2021). DeepNetDevanagari: a deep learning model for Devanagari ancient character recognition. *Multimedia Tools and Applications*, *80*, 20671-20686.

[17] Desai, P., Pujari, J., Ayachit, N. H., & Prasad, V. K. (2013, August). Classification of archaeological monuments for different art forms with an application to CBIR. In *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1108-1112). IEEE.

[18] Bhatt, M. S., & Patalia, T. P. (2015, November). Genetic programming evolved spatial descriptor for Indian monuments classification. In *2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)* (pp. 131-136). IEEE.

[19] Bhatt, M. S., & Patalia, T. P. (2017). Indian monuments classification using support vector machine. *International Journal of Electrical and Computer Engineering*, *7*(4), 1952.

[20] Sharma, S., Aggarwal, P., Bhattacharyya, A. N., & Indu, S. (2018). Classification of Indian monuments into architectural styles. In *Computer Vision, Pattern Recognition, Image Processing, and Graphics: 6th National Conference, NCVPRIPG 2017, Mandi, India, December 16-19, 2017, Revised Selected Papers 6* (pp. 540-549). Springer Singapore.

[21] Agarwal, A., & Saxena, V. (2010). Content Based Multimodal Retrieval for Databases of Indian Monuments. In *Contemporary Computing: Third International Conference, IC3 2010, Noida, India, August 9-11, 2010. Proceedings, Part I 3* (pp. 446-455). Springer Berlin Heidelberg.

[22] Grishkin, V. M., Shigorets, S. B., Vlasov, D. Y., Miklashevich, E. A., Zhabko, A. P., Kovshov, A. M., & Vlasov, A. D. (2016). Development of stone monuments monitoring system using computer technology. In *Biogenic—Abiogenic Interactions in Natural and Anthropogenic Systems* (pp. 415-421). Springer International Publishing.

[23] Padalkar, M. G., & Joshi, M. V. (2017). Automatic Detection and Inpainting of Defaced Regions and Cracks in Heritage Monuments. *Digital Hampi: Preserving Indian Cultural Heritage*, 225-244.

[24] Maaten, L. V. D., Boon, P., Lange, G., Paijmans, H., & Postma, E. (2007). Computer vision and machine learning for archaeology.

[25] Rocca, B. (2019). Handling imbalanced datasets in machine learning. *Towards data science*.

[26] https://www.kaggle.com/datasets/devguruap4u/ancient-temple-pillar-images-dataset.

[27] https://jnnce.ac.in/TempleDataSets/GURUDEV%20%20Description_of_KU-UBDTCE_JNNCE_Temple_Pillar_Database.pdf

[28] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90.

[29] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[30] F. Chollet et al., “Keras,” GitHub, 20156. [Online]. Available: https:// github.com/keras-team/keras.