IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 4, August 2026, pp. 3326∼3341
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i4.pp3326-3341

A transfer learning-based approach for automatic monument detection

Santosh Giri[1], Jebish Purbey[1], Sunil Adhikari[1], Paarit Pokharel[1], Babu R. Dawadi[1], Bipun Man Pati[2], Atsushi Ito[3], Sushant Chalise[1], Sanjivan Satyal[1]

[1]Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Lalitpur, Nepal
[2]Advanced College of Engineering and Management, Tribhuvan University, Kathmandu, Nepal
[3]Chuo University, Hachioji, Japan

Article Info

Article history:

Received Nov 20, 2024
Revised Jun 19, 2026
Accepted Jul 9, 2026

Keywords:

Architectural heritage
Artificial intelligence
Convolutional neural network
Deep learning
MobileNet
Monument recognition
Transfer learning

ABSTRACT

Architectural heritage connects us to the cultural achievements of past civilizations. Patan Durbar Square in Nepal is home to many such structures, yet identifying them remains a challenge for tourists. This paper presents an automated monument recognition system built on the backbone of convolutional neural networks (CNNs). A dataset of 1832 images of 9 important monuments from Patan was created, and build a detection system with MobileNetV2, a light-weight CNN, to detect monuments in Patan Durbar Square with a near-perfect F1 score of 98.94%. The approach utilizes transfer learning to adapt the model to local architectural styles. A detailed ablation study is performed to determine the optimal network design and augmentation strategies. Class-wise performance is further analyzed to verify robustness against visual occlusion and similarity. Finally, the model is deployed as a mobile application using Flutter and the FastAPI framework. This work demonstrates the viability of lightweight CNNs for real-time cultural heritage preservation.

Corresponding Author:

Santosh Giri
Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering
Tribhuvan University
Lalitpur, Nepal
Email: santoshgiri@pcampus.edu.np

1. INTRODUCTION

Cultural heritage stands as a tangible testament to human history and artistic evolution. The Kathmandu Valley serves as a prime example, hosting seven UNESCO World Heritage sites within a small radius [1]. This high density of historical sites makes it a major hub for cultural tourism in South Asia. Recent data from the Nepal Tourism Board highlights a strong post-pandemic recovery in visitor numbers [2], [3]. However, the sheer density of monuments often overwhelms visitors who lack deep historical knowledge. Most tourists face significant challenges in identifying specific shrines and temples among the clustered architecture. Professional local guides can be expensive or unavailable during peak seasons. Furthermore, static guidebooks often fail to provide instant visual identification from a specific viewpoint.

This gap creates a need for digital tools that can offer real-time information without human intervention. A mobile-based solution can democratize access to historical knowledge for all visitors. To address this challenge, an end-to-end system is proposed that utilizes deep learning to detect and recognize

monuments in real-time. The system is developed by creating a diverse dataset of 1,832 images that captures nine distinct monuments under varying lighting conditions and angles. A lightweight convolutional neural network (CNN) model is trained on this data, utilizing transfer learning to achieve high accuracy while maintaining the low latency required for mobile devices.

This system allows users to simply point their camera at a structure to receive immediate historical context. An earlier version of this work was presented at the International Symposium on Computing and Networking Workshops (CANDARW) [4]. This manuscript extends that work by providing deeper analysis, expanded methodological justification, and additional discussion on robustness and generalization.

2. LITERATURE REVIEW

Image classification is a foundational problem of computer vision, which has as its main objective to label an image based on its visuals. The emergence of deep learning and, to a greater extent, CNNs [5], [6] has greatly helped in vision tasks, especially for applications in object detection, medical imaging, and self-driving. CNNs are a bracket of deep learning architectures that automatically and adaptively learn spatial features of input from low-level edges to high-level patterns. CNNs can identify patterns directly from raw pixel data, significantly improving the efficiency and accuracy of image recognition tasks [7]. The foundational works in [8]–[10] made substantial contributions to progress in computer vision and image classification, particularly for employing very deep neural networks. Building upon this, Saini *et al.* [11] pioneered the use of CNN techniques for autonomous recognition of Indian cultural heritage sites, opening fresh possibilities for research in monument detection and recognition tasks. Transfer learning enables adaptation of pre-trained models to new tasks with limited data. By leveraging features learned on large datasets like ImageNet [12], transfer learning reduces computational requirements and training time while improving generalization [13], [14]. This approach proves particularly valuable for monument recognition, where training data may be limited.

Recent advances in monument recognition have leveraged various CNN architectures and deep learning techniques. Paul *et al.* [15] conducted an extensive analysis of machine learning and deep learning approaches for Indian monument classification, demonstrating the effectiveness of transfer learning and CNN-based methods. Hesham *et al.* [16] compared deep learning versus traditional machine learning approaches for monument recognition, establishing the superiority of CNN-based techniques. Shamov and Shelest [17] applied CNNs specifically for the recognition of tower lighthouses, demonstrating the adaptability of deep learning techniques across various domains. Jaiswal *et al.* [18] studied identifying monuments through aerial images using you only look once version 5 (YOLOv5) [19] and visual geometry group 16 (VGG-16) [8] as the base model. Kalliatakis and Triantafyllidis [20] explored utilization of graph-based visual saliency for monument recognition from images. Termritthikun and Muneesawang [21] developed Naresuan University-lightweight network (NU-LiteNet), a novel lightweight CNN inspired by SqueezeNet [22], achieving competitive accuracy measures on monument datasets. Using top-1 and top-5 accuracy measures, NU-LiteNet achieved greater accuracy, even surpassing GoogLeNet [23] for the Singapore and Paris datasets [24].

Razali *et al.* [25] developed a landmark recognition model for smart tourism applications using lightweight deep learning combined with linear discriminant analysis, achieving 100% classification accuracy on the Universiti Malaysia Sabah-scene (UMS-Scene) dataset and 94.26% on the Scene-15 dataset using the EfficientNet-CNN architecture. Additionally, the integration of data augmentation techniques, including random flip, rotation, translation, zoom, contrast, hue, brightness, and saturation adjustments, has proven effective in improving robustness [26]. Wang *et al.* [27] applied an improved Inception-V3 model for ancient architecture recognition, incorporating dropout layers and transfer learning to address overfitting in small monument datasets. Mobile deployment of monument recognition systems has gained prominence, with researchers focusing on developing lightweight CNN architectures suitable for real-time heritage site identification on mobile devices, enhancing accessibility for tourism and cultural preservation initiatives [28], [29].

3. MATERIALS AND METHODS

This section outlines the theoretical framework, data collection, and methodology for the monument recognition application. The methodology for monument recognition is subdivided into several steps. Each step is described in the following subsections.

3.1. Dataset creation

To train the model, the dataset was compiled by clicking pictures of cultural heritages around the Patan Durbar Square area. The dataset includes a total of 1,832 images belonging to 9 different classes as shown in Table 1. Figure 1 shows a batch of the dataset. The images were clicked with 4 different mobile devices with varying dimensions to ensure that a particular device doesn't affect the result. However, for training, the images were resized to 224×224 during the pre-processing steps. In the preprocessing phase, both data augmentation and normalization were applied. Initially, each image was rotated $270°$ counter-clockwise during the dataset loading phase to correct their orientation. This adjustment was necessary because the images, which were captured in portrait mode (i.e., width $<$ height), were interpreted as landscape mode by default during loading, causing an unwanted $90°$ counter-clockwise rotation. To address this, the images were rotated back to their correct orientation. The dataset was split into $70 : 15 : 15$ ratios for training, validation, and testing respectively. For data augmentation, random rotation (0.2, 0.4 and 0.7 radians), random brightening (40%), random contrasting (40% and 60%), and random horizontal and vertical flipping were applied. This process created new images by rotating, adjusting intensity, and flipping the originals, which increased the dataset size and introduced variability. This variability helps the model adapt to different angles and perspectives of the monuments. Normalization was achieved using MobileNetV2 [30] preprocessing, which scales the pixel values within the range $[-1, 1]$. The data augmentation is only activated during the training phase by default.

Table 1. Image distribution for each class for each phase

Monument	Train set	Validation set	Test set	Total
Bhimsen Temple	191	40	42	273
Char Narayan Temple	156	33	35	224
Dhungey Dhara	114	24	26	164
Garuda Pillar	70	15	15	100
Harishankara Temple	140	30	30	200
Krishna Mandir	172	36	38	246
Narayan Temple	169	36	37	242
Octagonal Chyasing Deval	130	27	29	186
Vishwanath Temple	137	29	31	197
Total	1,279	270	283	1,832

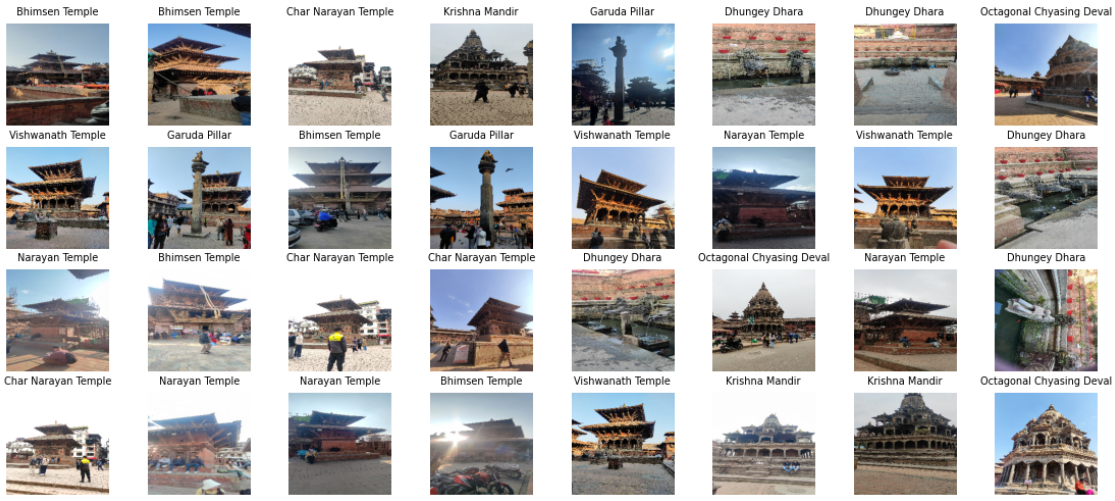

Figure 1. A sample batch of the dataset after every image has been normalized and resized to 224×224

3.2. Monument class selection and dataset representativeness

Nine monument classes were selected from Patan Durbar Square. Bhimsen Temple, Char Narayan Temple, Dhungey Dhara, Garuda Pillar, Harishankara Temple, Krishna Mandir, Narayan Temple, Octagonal Chyasing Deval, and Vishwanath Temple. The selection was guided by the need to balance dataset representativeness with the practical constraints of data collection and annotation.

3.2.1. Selection rationale

To identify appropriate monuments for the task, the help of a local tourist guide was taken. The selected heritage sites have significant cultural and historical value. The goal was to include monuments that are culturally important but also widely recognized by visitors. Patan Durbar Square is one of the most important historical sites in Nepal that reflects the artistic nuances of the Malla era. It is a designated UNESCO World Heritage site that sits at the heart of Lalitpur city. Temples, shrines, and palaces embodying Newari craftsmanship and traditions impart cultural life to every corner of the square. The selection of monuments captures this theme and the beauty of Patan Durbar Square. Input features were also considered in the selection of monuments. The selection consists of cultural heritages that are a blend of distinctive and similar architectures. Some structures are multi-tiered pagodas typical of traditional Newari Temples, while still having vivid differences among them. Krishna Mandir, which is a stone construction and has a very different facade, is completely different from the Newari-traditional temples. Each of these monuments also has pillars, water spouts, and shrines, which introduces morphological variety. Such a dataset was created to improve the generalization of the model. Finally, monuments that are widely popular among tourists were considered. The selection of monuments is among the most popular hotspots within Patan Durbar Square. People frequently visit them and photograph them, especially the Krishna Mandir. As a result, the model developed has a very high degree of practicality. All the monuments were captured under varying light conditions, across days, to ensure robust model training.

3.2.2. Representativeness

The selection of monuments covers a wide range of religious, cultural, and historical contexts present in Patan Durbar Square. Patan Durbar is home to more than 100 monuments, and thus, our dataset is not an exhaustive collection of monuments within this area. However, the selection reflects the key architectural and cultural characteristics of the site. Furthermore, they are 9 of the most visited and referred sites within the area. As such, they can be considered a good representation of the artistic landscape of Patan Durbar Square. Yet, the dataset is only representative of a single heritage zone, while Kathmandu Valley itself contains two more Durbar areas with rich cultural contexts. Future work should extend beyond Patan to other sites within the Valley, and hopefully over the country, and improve the generalization capability of models. However, for the current scope of the work, these nine classes provide an adequate base for analyzing the capability of automated monument recognition in the context of edge devices.

3.3. Model development

The architecture of the model is illustrated in Figure 2. In this figure, MobilenetV2 is used as the backbone for the model. Transfer learning is employed to design the classification network based on the MobileNetV2 backbone, which outputs a feature map of size $7 \times 7 \times 1,280$. MobileNetV2 is a lightweight and efficient CNN designed for low-resource vision applications. It introduces several key enhancements, most notably the use of inverted residuals and linear bottlenecks. It utilizes depth-wise separable convolutions that notably reduce the number of calculations required for performing convolutions, resulting in a lightweight, low-parameter-count network. Inverted residuals enable the network to preserve feature maps through nonlinearities, which improves the representational capacity of the model.

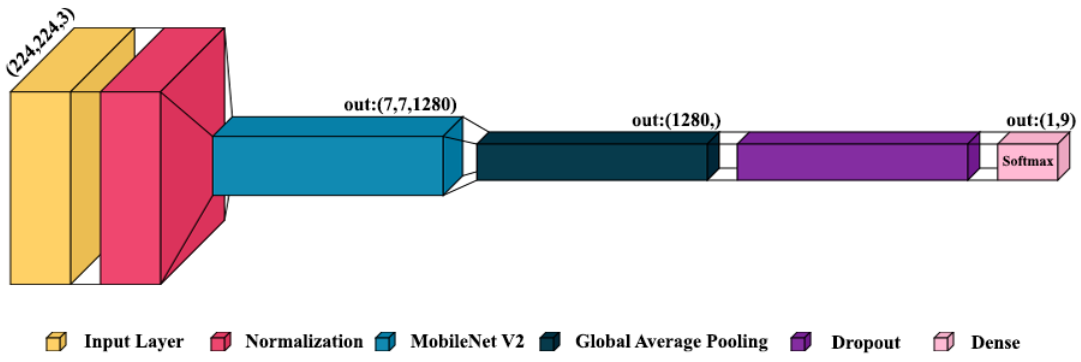

Figure 2. Architectural diagram of the model for monument recognition constructed using Visualkeras [31]

Depthwise separable convolution is an approach in computer vision that reduces both computational cost and model size without compromising performance. It splits the convolution operation into two distinct steps: depthwise convolution accompanied by pointwise convolution. A separate filter is applied to each input channel individually during depthwise convolution. Normally, with n input channels, a standard convolution uses a filter with depth n, but in depthwise convolution, n filters each with a depth of 1 are used, producing an output with a depth of 1 per filter. This operation is succeeded by a pointwise convolution, where 1×1 filters of depth n are applied to the depthwise convolution's output. The number of these 1×1 filters determines the final output depth. Depthwise separable convolution significantly reduces the number of computations, leading to faster inference times.

For an input feature map X with height H, width W, and depth D ($H \times W \times D$), depthwise convolution applies a separate kernel individually to each channel. Let K_d represent the depthwise filter with dimensions $k_h \times k_w \times 1$, where k_h is the height and k_w is the width of the filter. For each input channel d (where d ranges from 1 to D), the depthwise convolution output Y_d can be defined as in (1).

$$Y_d(i,j) = \sum_{n=0}^{k_w-1} \sum_{m=0}^{k_h-1} K_d(m,n) \cdot X_d(i+m, j+n) \tag{1}$$

Where, i and j are spatial indices and, K_d is the depthwise filter applied to the d-the channel of the input.

After performing depthwise convolution, the pointwise convolution kernel is applied to the output across different channels. The pointwise convolution uses a 1×1 filter K_p with dimensions $1 \times 1 \times D \times M$, where M is the desired number of output channels. The output Z_m for the m-th output channel can be expressed as in (2).

$$Z_m(i,j) = \sum_{d=1}^{D} K_p(d,m) \cdot Y_d(i,j) \tag{2}$$

Where $K_p(d,m)$ is the pointwise filter coefficient connecting the d-th input channel to the m-th output channel, and $Y_d(i,j)$ is the output of depthwise convolution for channel d.

The MobileNetV2 backbone is succeeded by a global average pooling (GAP) layer and an immediate dropout layer. GAP reduces $7 \times 7 \times 1,280$ feature maps to a $1,280$-dimensional vector by averaging the values in each 7×7 feature map. This reduces the dimension and computational complexity. A dropout layer of probability p, is then employed to mitigate overfitting by setting a fraction of the neurons' output to zero during training with probability p. This combination ensures that the model generalizes well and maintains high performance on novel data.

Next, a fully connected block is introduced to work as the classifier for the model. The architecture of the fully connected block is varied to determine the optimal network design. The fully connected network (FCN) architectures under consideration are: i) simple FCN: a single-layer FCN with 9 neurons; ii) two-layer FCN: a double-layer FCN, having 18 neurons followed by 9 neurons; and iii) three-layer FCN: a FCN with three layers, consisting of 36, 18, and 9 neurons in the first, second, and final layers, respectively. Additionally, each architecture is tested with and without dropout to evaluate its impact on performance. The detailed results of these experiments, including the impact of dropout, are discussed in the results section.

3.4. Model choice justification

MobileNetV2 [25] is a lightweight CNN model that is specifically built for edge-case usages. In this context, MobileNetV2 was chosen because of its high accuracy and efficiency in public benchmarks. This is important for the system as it aims for real-world deployment in a resource-constrained environment.

3.4.1. Rationale for MobileNetV2

MobileNetV2 uses depthwise separable convolutions for convolution operations. This reduces parameters and floating-point operations while still maintaining high performance. As a result, the model is comparatively very small in weight and is a solid option for mobile deployment. Since practical usability is a long-term goal of this research, model efficiency was weighted alongside accuracy in the design decisions. MobileNetV2 also performs very well in transfer learning. Since the dataset of 1,832 images is very modest compared to standard model training datasets, transfer learning was a necessity. Transfer learning has also proven to be a much better option than training from scratch, even in cases of huge dataset availability.

By using MobileNetV2, the weights of the model that were already trained on 1,000 classes of ImageNet were leveraged [24]. This allowed for faster convergence and high accuracy right off the bat during training. As a result, the model achieved 99% accuracy in the experiments.

3.4.2. Consideration of alternative architectures

Other architectures that are widely used for detection tasks were also considered. EfficientNet [32] models offer very strong performance, above that of MobileNet. However, they are heavier and thus require higher computation resources than MobileNet. This decreases their practicality and makes them less viable for edge-case uses. YOLO-based models [19] are one of the most widely used models in computer vision. They offer the highest performance in complex scenarios and sub-second detection, making them useful for detection within videos too. However, since monument classification is a much simpler task where a monument dominates the image, adding a full detection pipeline complicates the system and introduces the risk of overfitting. As MobileNetV2 was providing similar performance with a simpler pipeline, it was selected for the model.

3.4.3. Design trade-off perspective

The model selection reflects a deliberate trade-off that prioritizes practicality and generalization over higher numbers. A lightweight model reduces the risk of overfitting and is easier to deploy in low-resource devices such as smartphones. These are practical advantages that speed up the model development as well as the deployment time. Still, other lightweight models with similar usability are recognized and discussed in subsection 3.4.2. However, within the scope of this work, MobileNetV2 stands out as a solid option for the backbone of the recognition pipeline.

3.5. Training and evaluation

The functioning of all models was assessed on the monument recognition dataset using accuracy and loss as performance metrics. The total training epochs for each model was 80 epochs, 50 epochs of classification-head training with a learning rate of 0.001, followed by 30 additional epochs of fine-tuning with the learning rate decreased to 0.0001. During fine-tuning, the last 29 layers of the base model were unfrozen. This training process was repeated for 3 turns for each model to account for variability introduced by stochastic factors such as random initialization, data shuffling, and batch processing. The accuracy metrics for the last 5 epochs are also calculated to assess the model's stability during the final phase of fine-tuning. The most robust model is identified with a focus on minimizing overfitting. After selecting the best model, a hyperparameter search is conducted to maximize performance after training. The hyperparameters under consideration are listed in the Table 2. The base model i.e., MobilenetV2 consists of a total of 154 layers including the input layer. The main convolutional block with residual connection consists of 9 layers. The hyperparameter space for the number of layers to unfreeze is chosen in such a way that each step-increase unfreezes a convolutional block in the base model. Random search is used to find the best hyperparameter combinations for both initial training and fine-tuning. Early stopping with a patience value of 5 is also employed during the search, with maximum trials set to 75 for initial training and 40 for fine-tuning. The best hyperparameter combination is then used to train the model and evaluate its performance on the test set. The performance metrics used are recall, precision, and F1-score as defined in [33].

Table 2. Hyperparameters search space

Phase	Hyperparameter	Value
Initial training	Learning rate	0.01 to 0.0001 (Sampling = log)
	Dropout rate	0.2, 0.3, 0.4, 0.5 and 0.6
	No. of epochs	20, 30, and 40
	Batch size	0.2, 16, 32, and 64
	Regularization factor (L2)	0.1 to 0.001 (Sampling = log)
	Optimizer	Adam, root mean square propagation (RMSprop) and Stochastic gradient descent (SGD) (with momentum)
Fine tuning	Learning rate	0.001 to 0.000001 (Sampling = log)
	No. of layers to unfreeze	29, 38, 47, 56, and 65
	No. of epochs	30, 40, and 50

3.6. Data augmentation rationale

Data augmentation helps improve the robustness of deep learning models when the dataset is small. Since the dataset is relatively small, data augmentation was used in the training pipeline. This effectively increases the diversity within the same sample and reduces the risk of overfitting.

3.6.1. Selected augmentation strategies

The augmentation pipeline consisted of several different image transformation operations. These include rotation, brightness, contrast adjustments, and horizontal flipping. These transformations were chosen as they reflect the most commonly occurring scenarios where people click images. Rotation represents the slight viewpoint changes that are caused when handheld cameras get tilted while clicking images. This is also one of the most common forms of aberrations in images. Casual photographers rarely maintain a perfectly aligned camera, and these slight angular variations are common in clicked images. Having datasets that account for these variations helps the model become robust when facing such edge cases in practical usage.

Brightness and contrast adjustments are incorporated to account for images clicked under different lighting conditions. Depending upon the time of day and weather, images captured of monuments can differ significantly. While the original dataset itself contains images clicked during different times of the day, augmentation using brightness and contrast modifications complements the training further. This allows the model to learn features that are less dependent on specific lighting conditions. Horizontal flipping is used to increase the visual diversity of the samples. Many monuments exhibit approximate bilateral symmetry and maintain identity even when mirrored. This also simulates cases where images are not flipped after capturing with a front-facing camera.

3.6.2. Parameter selection considerations

The augmentation parameters are kept within a moderate range that simulates the real-life aberration scenarios. This also prevents generating unrealistic samples, which only consumes training time but doesn't provide any gains. The intention with augmentation was to complement our already diverse dataset with plausible photographic variations. Extreme transformation can lead to degraded model learning, as it can distort the monument completely and shift the model's weight towards recognizing these distorted samples.

3.6.3. Considered but excluded techniques

Other widely used augmentation techniques, such as random cropping and noise injection. However, they were not included in the augmentation pipeline, as they are not indicative of practical photographic variations. Random cropping has the risk of removing the key architectural feature within the image, and may lead to label ambiguity. Similarly, injecting artificial noise could lead to the model trying to learn the features of the noise instead of the monuments. As the initial dataset already contains realistic variations of noise and cropping, the models trained on the dataset are well-equipped to handle such aberrations in practical use.

3.6.4. Role in generalization

The augmentation strategy was designed to introduce realism and balance diversity. By exposing the model to the controlled variations of the same samples, it is encouraged to learn invariant features. This results in improved generalization and robust performance across conditions.

3.7. Development

After evaluating the best model, it is deployed as a smartphone application for real-time monument recognition. Fast application programming interface (FastAPI) is used to develop the backend server, which handles model inference requests. FastAPI allows asynchronous operations for seamless communication between the smartphone application and the backend server. For the mobile application, the Flutter framework is used, which allows us to build a cross-platform app from a single codebase. To manage hypertext transfer protocol (HTTP) requests and interactions with the FastAPI backend, Dio is employed, a powerful HTTP client for Flutter.

4. RESULTS AND DISCUSSION

This section presents a comprehensive analysis of the model training and evaluation outcomes, including ablation studies to validate architectural choices and the effectiveness of regularization techniques.

Different hyperparameter configurations are systematically evaluated to identify the optimal settings for final model training, ensuring robust performance across multiple validation metrics. Additionally, the model's generalization capabilities are assessed through rigorous testing on an independent test set, examining both classification accuracy and error patterns to validate the practical utility of the monument recognition framework.

4.1. Ablation study

Five distinct models were experimented with to examine how different network designs and dropout rates impact performance. Table 3 summarizes the findings, showing average accuracy and average loss metrics for both training and validation phases for 3 total training runs. The single fully connected architecture (9 neurons) achieved the highest training and validation accuracy at 99.77% and 96.42% respectively.

When complexity was added with two fully connected layers (18 and 9 neurons) without dropout, a significant drop in validation accuracy and an increase in validation loss were observed. This indicates that more complex models without dropout might overfit the training data. The three-layer model attained an accuracy of 99.40% for training but performed poorly for the validation dataset (92.84%) compared to simpler architectures, showing signs of overfitting. Incorporating dropout (0.2) into the two-layer and three-layer models reduced training accuracy but kept validation accuracy consistent. This shows that dropout helps in reducing overfitting by improving generalization, even though it affects training accuracy. The most complex configuration with three fully connected layers and dropout (0.2) provided the second highest validation accuracy at 94.44% and improved performance during fine-tuning.

It should be noted that the training accuracy shown for dropout models will be lesser than their actual training accuracy as dropout will reduce the accuracy during training and not during evaluation. The average accuracy metrics of the last 5 epochs show that the single FC architecture and $36 - 18 - 9$ dropout had the most stable training during the last phases. Given the comparable performance between the two models, Occam's Razor [34] principle is used to select the simpler model. Occam's Razor principle suggests that among competing hypotheses that explain the data equally well, the simplest one should be selected. The single fully connected layer model thus emerged as the most robust configuration, balancing high accuracy and minimal overfitting with the simplest architecture design.

Table 3. Average training results for all five fully connected architectures for 3 turns of training

Model	Initial-training (50 epochs)				Fine-tuning (30 epochs)				Average of last 5 epochs	
	Training		Validation		Training		Validation		Training	Validation
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss	Accuracy	Accuracy
9	0.9523	0.1461	0.8543	0.4272	0.9977	0.0069	0.9642	0.0874	0.9971	0.9514
18-9	0.9510	0.1433	0.8556	0.4185	0.9974	0.0122	0.9111	0.2858	0.9968	0.9188
18-9 dropout	0.8027	0.5303	0.8222	0.5423	0.9322	0.1827	0.9222	0.2186	0.9298	0.9081
36-18	0.9453	0.1541	0.8469	0.4247	0.9940	0.0159	0.9284	0.2184	0.9944	0.9163
36-18-9 dropout	0.8548	0.4267	0.8469	0.4987	0.9487	0.1485	0.9444	0.2111	0.9484	0.9321

4.2. Hyperparameter search and training

The hyperparameters listed in Table 2 were searched for using random search. The best values for each hyperparameter are listed in Table 4. The training of the single-layer (9 neurons) model was conducted for a total of 80 epochs.

The hyperparameters used were the ones that gave the best validation accuracy during the hyperparameter search. For the fine-tuning, the last 47 layers of the base model were unfrozen. The training curves of the complete process are shown in Figure 3. The pink dotted line in this figure indicates the start of fine-tuning with the last 47 layers of MobileNetV2 unfrozen. The final training accuracy of the model is 100.00% with a loss of 0.0345.

The model demonstrated exceptional performance on the validation set as well, with a loss of 0.0470 and a near-perfect accuracy of 99.63%. The model's validation accuracy at the end of the 30th epoch, i.e., before fine-tuning was 82.22%. These findings show that the model's performance and generalization improved considerably during the fine-tuning phase. The negligible difference between the accuracy metrics also indicates that the model is not overfitting.

Table 4. Hyperparameters best values

Phase	Hyperparameter	Best value
Initial training	Learning rate	0.001
	Dropout rate	0.2
	No. of epochs	30
	Batch size	32
	Regularization factor (L2)	0.006285
	Optimizer	Adam
Fine tuning	Learning rate	0.000193
	No. of layers to unfreeze	47
	No. of epochs	50

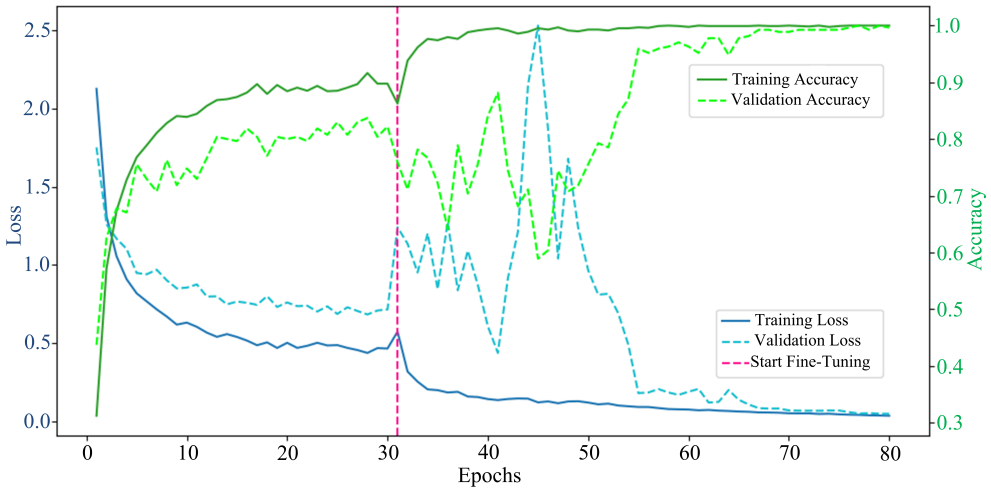

Figure 3. Training curves for the best model

4.3. Test set evaluation

A final evaluation of the model was conducted on the test set consisting of 283 images across the 9 classes with distribution as in Table 1. The model achieved a splendid accuracy of 98.94% with minimal loss of 0.0810. The performance across the classes was splendid as perfect F1-score, recall, and precision scores were achieved in 4 classes. The performance metrics of individual classes along with the average are listed in Table 5. Out of 283 images, only 3 instances of images were misclassified and had metrics less than 1, as shown in Figure 4. An instance of the Garuda Pillar, the Dhungey Dhara, and the Harishankara Temple were misclassified. The performance metrics of misclassified classes are greater than 0.96 with the exception being the recall of Garuda Pillar (0.9333) and precision of Vishwanath Temple (0.9394). The macro average recall (0.9846), precision (0.9904), and F1-score (0.9872) are all above 0.98. The weighted averages of performance metrics are also greater than 0.98. This demonstrates that the model performs consistently across the classes.

Table 5. Performance metrics of all the classes for the test set

Class	Precision	Recall	F1-score	Size
Bhimsen Temple	1.0000	1.0000	1.0000	42
Char Narayan Temple	1.0000	1.0000	1.0000	35
Dhungey Dhara	1.0000	0.9615	0.9804	26
Garuda Pillar	1.0000	0.9333	0.9655	15
Harishankara Temple	1.0000	0.9667	0.9831	30
Krishna Mandir	0.9744	1.0000	0.9870	38
Narayan Temple	1.0000	1.0000	1.0000	37
Octagonal Deval	1.0000	1.0000	1.0000	29
Vishwanath Temple	0.9394	1.0000	0.9688	31
Accuracy		0.9894	283	
Macro average	0.9904	0.9846	0.9872	283
Weighted average	0.9899	0.9894	0.9894	283

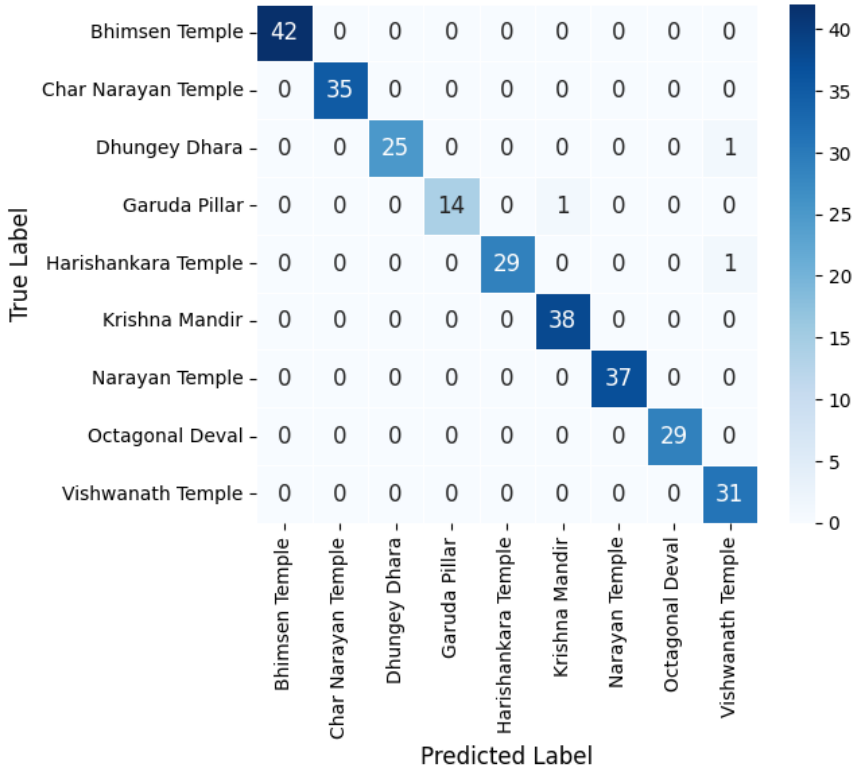

Figure 4. Confusion matrix highlighting true and predicted labels between monument classes

4.4. Class-wise error and robustness analysis

In order to understand the model behavior, a class-wise analysis is performed. In cultural heritage recognition tasks, different monuments present different levels of visual complexity. Thus, examining per-class performance helps identify the strengths and limitations of the approach better, compared to an overall number.

4.4.1. Class-wise performance variability

The results indicate high precision and recall across most monument classes. This suggests that the model is effectively capturing the visual features of monuments. Still, monuments that have clearly identifiable structures and unique silhouettes exhibit higher recognition reliability. The Krishna Mandir has a very distinct facade and structure, and shows a near perfect reconition rate across different training runs. Yet, a slight performance variability is observed among a few classes. This arises due to several factors. Many monuments share similar architectural styles. Harishankara Temple and Vishwanath Temple are almost identical, which even confuses the locals of Patan. This is also translated in the model's performance, as the model sometimes classifies Harishakara Temple incorrectly. This happens when both images are captured from similar viewpoints. The background clutter and occlusion also affect recognition. Patan Durbar Square is an active hotspot. Many of the samples within the dataset also reflect this. Because of this, many monuments are often partially occluded by people or surrounding structures. This makes it difficult for models to concentrate on features that are representative of monuments. Moreover, an edge case in our analysis is when two monuments are present within the same image. The Garuda Pillar is surrounded by many of the other monuments within our dataset. It is also not possible to capture the Garuda Pillar without other monuments, as that doesn't depict the real-life scenario. As such, the model sometimes misclassified the Garuda Pillar as the other monument present in the image.

4.4.2. Misclassifications patterns

The analysis of misclassified samples shows that errors typically occur between monuments that share visual similarity. Some of these have been discussed in depth in subsection 4.4.1. This highlights that while the model is learning meaningful visual representations of monuments, they still struggle when discriminative

features between classes are very subtle or partially occluded. However, these misclassifications are not random and reflective of mistakes that even local makes.

4.4.3. Robustness considerations

Despite these challenges, the model shows robust performance as the overall misclassification rate is negligibly small. This robust performance can be attributed to transfer learning and the data augmentation pipeline. The model also maintains its performance across varying lighting conditions, highlighting the importance of careful dataset curation. Nevertheless, it is acknowledged that the robustness could be further improved through larger datasets. This is especially true for cases where monuments share visual similarities. Incorporating more images with controlled variation in occlusion and crowd presence may help the model better handle complex real-world scenarios.

4.4.4. Practical implications

The observed performance indicates that the proposed approach is suitable as a supportive tool for monument identification. It can be acceptable in many heritage-assistance applications, particularly when predictions are accompanied by confidence scores or contextual information. Nevertheless, there remains room for future refinement.

4.5. Overfitting and generalization considerations

Overfitting is a common concern in deep learning applications involving relatively small datasets. With 1,832 images distributed across nine monument classes, there is an inherent risk that a model may memorize training samples rather than learn generalizable visual patterns. Therefore, several design choices in this study were guided by the goal of promoting generalization.

4.5.1. Architectural simplicity and Occam's Razor

One notable finding of the work is that a single fully connected layer on top of the pretrained backbone achieved better performance than deeper classifier heads. This shows that simple models can generalize better when the training data is limited. Adding more parameters increases the representational power of the model, but also introduces the problem of overfitting. A case was also found where both simpler and complex model architectures performed comparatively during the initial model selection phase. To solve this selection dilemma, Occam's Razon principle was utilized [34]. Occam's Razor is a problem-solving principle that states the simplest explanation with the fewest assumptions is usually the best or most likely to be correct. In machine learning terms, Occam's Razor is understood as a principle favoring simpler models over complex ones when both achieves comparative predictive performance. This is because simpler models usually generalize better to new data and are less prone to overfitting noise.

4.5.2. Role of transfer learning

Transfer learning is now a standard approach in computer vision tasks. In any computer vision task, detecting a certain structure involves understanding the points, edges, vertices, and design elements that form that structure. Training a model from scratch will lead to model fitting to the training dataset. This is because the model is limited to a smaller dataset. In reality, many structures share similar design elements and geometry despite being completely different in overall form. By utilizing weights of a model already trained on a huge dataset, the model can be trained to concentrate only on distinctive elements required for classifying the monuments. This leads to higher performance, generalization, and robustness.

4.5.3. Effect of regularization and augmentation

Regularization is used to combat the issue of overfitting. Dropout is used to ensure that every parameter in the model contributes to the final prediction. This leads to a better generalization. During dropout, the model also effectively becomes much simpler and reduces overfitting risks. Data augmentation is used to improve the robustness of the model. As discussed in subsection 3.6, data augmentation helps the model to learn invariant features. This also reduces overfitting and improves generalization. Together, these strategies help reduce the gap between training and testing performance.

4.5.4. Generalization scope

While the model shows exceptional performance within the Patand Durbar Square dataset, it cannot be assumed that this will translate well to other heritage sites without evaluating on samples of those sites. Different monuments have different architectural styles, textures, and contexts. These vary across regions. These affect the recognition performance. Computer vision models usually don't generalize to unseen data without further training. Thus, the present results only presents a strong evidence of performance within the studied heritage zone.

4.5.5. Practical perspective

The current generalization capability of the model is suitable for localized heritage assistance and guided-tour support systems, focused on the monuments of Patan Durbar Square. Despite dataset constraints, the model provides very robust performance across conditions. This makes the edge deployment of the model all the more practical.

4.6. Future work

In future work, the dataset can be expanded to include a broader range of monument types in different heritage sites of the Kathmandu Valley to better capture visual diversity and improve the robustness of the model. Further research can be done to explore multi-monument detection in complex scenes where multiple heritage structures appear within the same view. Hence, a video-based object detection formulation for the multi-monument detection task might be an alternative future approach.

5. CONCLUSION

A comprehensive deep-learning solution was developed for a monument recognition smartphone application trained on the dataset. Pre-trained weights from MobileNetV2 were used to create a specialized fully linked architecture. Following thorough testing with several architectures, a single fully connected layer model was determined to perform the best. The model attained an excellent test accuracy of 98.94% and showed robust performance with macro-average recall of 0.9846, precision of 0.9904 , and F1-score value of 0.9872. These metrics illustrate the model's excellent generalization skills as well as its ability to distinguish between similar architectural structures. The mobile application was created using Flutter with the backend powered by FastAPI. The application is intended to be user-friendly and efficient, allowing users to quickly and accurately identify monuments using their mobile devices. Although the model performed admirably, it could be enhanced in the future by using a bigger dataset with a greater variety of classes to represent all the monuments in the Kathmandu Valley. Furthermore, exploring attention mechanisms and multi-modal datasets for recognizing monuments could also be a path to enhance the model. Real-time training could be used on the application side to constantly enhance the model's capabilities. This study demonstrated that transfer learning and data augmentation achieve near-perfect accuracy (98.94%) for automated monument recognition in Patan Durbar Square. The framework successfully integrates MobileNetV2 with deep learning for practical heritage site identification, establishing a foundation for cultural preservation and tourism initiatives across South Asia. Future work should focus on expanding the dataset to encompass all Kathmandu Valley monuments and investigating advanced techniques such as attention mechanisms and on-device transfer learning to enhance scalability and real-time capabilities.

ACKNOWLEDGMENTS

We would like to acknowledge the Department of Archaeology, Nepal, for its coordination during the data collection.

FUNDING INFORMATION

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. No funding was raised or provided for the conduct of this study. The authors confirm that this work was completed without external financial support.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Santosh Giri	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
Jebish Purbey		✓	✓	✓	✓				✓	✓	✓		✓	
Sunil Adhikari			✓	✓		✓	✓	✓	✓					
Paarit Pokharel			✓	✓		✓	✓	✓	✓					
Babu R. Dawadi	✓									✓		✓		
Bipun Man Pati		✓								✓		✓		
Atsushi Ito		✓								✓		✓		
Sushant Chalise			✓	✓		✓		✓	✓					
Sanjivan Satyal		✓								✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

Not applicable. This study did not involve human participants, human data, or any personally identifiable information. All data used were either publicly available, fully anonymized, or derived from non-human sources, and therefore no informed consent was required from individuals.

ETHICAL APPROVAL

Not applicable. This research did not involve human subjects, human biological materials, or experimental procedures on animals. The work was conducted solely on computational models, publicly available datasets, or non-sensitive data that did not require intervention with living organisms. Therefore, ethical approval from an institutional review board or animal ethics committee was not necessary for this study.

DATA AVAILABILITY

The data that support the findings of this study are available on request from the corresponding author, [SG].

REFERENCES

[1] UNESCO World Heritage Convention, "Kathmandu Valley," whc.unesco.org. Accessed: Jul. 29, 2024. [Online]. Available: https://whc.unesco.org/en/list/121

[2] The Kathmandu Post, "Nepal's tourism potential," kathmandupost.com. Accessed: Jul. 29, 2024. [Online]. Available: https://kathmandupost.com/editorial/2024/01/02/nepal-s-tourism-potential

[3] Government of Nepal, Ministry of Culture, Tourism and Civil Aviation, "Nepal tourism statistics 2024," tourism.gov.np. Accessed: Jul. 29, 2024. [Online]. Available: https://www.tourism.gov.np/content/83/nepal-tourism-statistics-2024

[4] S. Giri, J. Purbey, S. Adhikari, A. Ito, B. Dawadi, S. Chalise, and S. Satyal, "Automatic monument detection: leveraging deep learning for cultural heritage applications," in 2025 *Thirteenth International Symposium on Computing and Networking Workshops (CANDARW)*, 2025, pp. 69–75.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[7] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: a comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017, doi: 10.1162/NECO_a_00990.
[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.
[11] A. Saini, T. Gupta, R. Kumar, A. K. Gupta, M. Panwar, and A. Mittal, "Image based Indian monument recognition using convoluted neural networks," in *2017 International Conference on Big Data, IoT and Data Science*, 2017, pp. 138–142, doi: 10.1109/BID.2017.8336587.
[12] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. F. Fei, "ImageNet: a large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.
[13] X. Yang, L. Jiao, and Q. Pan, "Transfer adaptation learning for target recognition in SAR images: a survey," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, pp. 13577–13601, 2024, doi: 10.1109/JSTARS.2024.3434448.
[14] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 512–519, doi: 10.1109/CVPRW.2014.131.
[15] A. J. Paul, S. Ghose, K. Aggarwal, N. Nethaji, S. Pal, and A. D. Purkayastha, "Machine learning advances aiding recognition and classification of indian monuments and landmarks," in *2021 IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering*, 2021, doi: 10.1109/UPCON52273.2021.9667619.
[16] S. Hesham, R. Khaled, D. Yasser, S. Refaat, N. Shorim, and F. H. Ismail, "Monuments recognition using deep learning vs machine learning," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference*, 2021, pp. 258–263, doi: 10.1109/CCWC51732.2021.9376029.
[17] I. A. Shamov and P. S. Shelest, "Application of the convolutional neural network to design an algorithm for recognition of tower lighthouses," in *2017 24th Saint Petersburg International Conference on Integrated Navigation Systems (ICINS)*, 2017, pp. 1-2, doi: 10.23919/ICINS.2017.7995568.
[18] G. K. Jaiswal, R. Chaudhary, Mohit, and Srishty, "Identification of monuments from aerial images using deep learning techniques," *International Journal of Engineering Applied Sciences and Technology*, vol. 8, no. 2, pp. 123–129, 2023, doi: 10.33564/ijeast.2023.v08i02.017.
[19] G. Jocher *et al.*, "Ultralytics/yolov5: v6.2 - YOLOv5 classification models, Apple M1, reproducibility, ClearML and Deci.ai integrations," *Zenodo*, 2022, doi: 10.5281/zenodo.7002879.
[20] G. E. Kalliatakis and G. A. Triantafyllidis, "Image based monument recognition using graph based visual saliency," *Electronic Letters on Computer Vision and Image Analysis*, vol. 12, no. 2, pp. 88–97, 2013, doi: 10.5565/rev/elcvia.524.
[21] C. Termritthikun and P. Muneesawang, "NU-LiteNet: mobile landmark recognition using convolutional neural networks," *ECTI Transactions on Computer and Information Technology*, vol. 13, no. 1, pp. 21–28, 2019, doi: 10.37936/ECTI-CIT.2019131.165074.
[22] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and ¡0.5MB model size," 2016, *arXiv:1602.07360*.
[23] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
[24] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: improving particular object retrieval in large scale image databases," in *26th IEEE Conference on Computer Vision and Pattern Recognition*, 2008, doi: 10.1109/CVPR.2008.4587635.
[25] M. N. Razali, E. O. N. Tony, A. A. A. Ibrahim, R. Hanapi, and Z. Iswandono, "Landmark recognition model for smart tourism using lightweight deep learning and linear discriminant analysis," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 2, pp. 198–213, 2023, doi: 10.14569/IJACSA.2023.0140225.
[26] K. Alomar, H. I. Aysel, and X. Cai, "Data augmentation in classification and segmentation: a survey and new strategies," *Journal of Imaging*, vol. 9, no. 2, 2023, doi: 10.3390/jimaging9020046.
[27] X. Wang *et al.*, "A recognition method of ancient architectures based on the improved Inception v3 model," *Symmetry*, vol. 14, no. 12, 2022, doi: 10.3390/sym14122679.
[28] M. S. Hasan, M. S. Uddin, K. J. Hasan, M. Rahman, and M. Ali, "Deep learning for cultural heritage: a mobile app for monument recognition using convolutional neural networks," *International Journal of Interactive Mobile Technologies*, vol. 19, no. 3, pp. 22–40, 2025, doi: 10.3991/ijim.v19i03.50935.
[29] S. Gada, V. Mehta, K. Kanchan, C. Jain, and P. Raut, "Monument recognition using deep neural networks," in *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2018, pp. 1-6, doi: 10.1109/ICCIC.2017.8524224.
[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
[31] P. Gavrikov, "Visualkeras," github.com. Accessed: Jul. 29, 2024. [Online]. Available: https://github.com/paulgavrikov/visualkeras
[32] M. Tan and Q. V. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019.
[33] A. C. Müller and S. Guido, *Introduction to machine learning with Python: a guide for data scientists*. Sebastopol, United States: O'Reilly Media, Inc., 2016.
[34] G. I. Webb, "Occam's razor," in *Encyclopedia of Machine Learning*, Boston, United States: Springer, 2011, doi: 10.1007/978-0-387-30164-8_609.

BIOGRAPHIES OF AUTHORS

Santosh Giri is an assistant professor of Computer Engineering at the Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, where he also leads the artificial intelligence and machine learning cluster's research works. He has completed his diploma in Computer Engineering from the Institute of Engineering, Tribhuvan University and bachelor's and master's degrees in Computer Engineering from NCIT, Pokhara University, respectively. He has received many research funding supports from UGC-Nepal. His research interests focus on applying AI algorithms, particularly computer vision and transfer learning, to low-resource learning problems. He is dedicated to developing intelligent systems that address complex, domain-specific challenges. He has published several research papers in national and international journals. He can be contacted at email: santoshgiri@pcampus.edu.np.

Jebish Purbey received the bachelor's degree in Electronics and Communication Engineering from Pulchowk Campus, Institute of Engineering, Tribhuvan University, Patan, Nepal. He is currently a research assistant with the Department of Electronics and Computer Engineering at Pulchowk Campus, where he is involved in projects at the intersection of artificial intelligence and language technologies. He also works as an AI Engineer at Dogma Group and teaches undergraduate courses at his alma mater. Additionally, he serves as the machine learning-agent lead at the Cohere Labs Community. His research interests include natural language processing, large language models, multilingual, multimodal, and cross-lingual learning, evaluation methodologies in NLP, and the design of agentic systems. He can be contacted at email: 074bex412.jebish@pcampus.edu.np.

Sunil Adhikari has completed his bachelor's degree in Electronics, Information, and Communication Engineering from the Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University. He is currently pursuing a master's degree in Data Science at Charles Sturt University, Australia. His research interests include computer vision, deep learning, and AI in cybersecurity and data handling.. He can be contacted at email: 076bei043.sunil@pcampus.edu.np.

Paarit Pokharel recently completed his bachelor's degree in Electronics, Information, and Communication Engineering from the Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University. His research interests include image processing, computer vision, and deep learning techniques for computer vision tasks. He can be contacted at email: 076bei021.paarit@pcampus.edu.np.

Dr. Babu R. Dawadi earned his Ph.D. degree in Computer Engineering from the Institute of Engineering (IoE), Tribhuvan University in 2021 AD with a scholarship from NTNU, Norway. He holds an M.Sc. in Information and Communication Engineering (2008), a bachelor's degree in Computer Engineering (2003), and a master's degree in Public Administration (2013). He served as assistant director at the Nepal Telecommunications Authority from 2009 to 2012 and has been a full-time faculty member at the Department of Electronics and Computer Engineering, IoE, Pulchowk Campus since 2012. He has contributed to many national ICT policy formulation including AI and cyber security policy. Within IoE, he has taken on several academic and administrative roles, including deputy head of department, chief technical officer for the computer-based entrance examinations, board member of the entrance and admission board, deputy chief of examination control division, and program coordinator of different M.Sc. programs. He has received many research funding supports from EnPe-NORAD, Nepal Academy of Science and Technology (NAST), US National Science Foundation, UGC-Nepal, and TU Research Directorate. He is the co-chairman of the Laboratory for ICT Research and Development (LICT) and director of its Network, Cyber Security, and Digital Forensics Wing. Currently, he serves as an assistant dean (Administration) and chief of Examination Control Division at IoE, Tribhuvan University. He can be contacted at email: baburd@ioe.edu.np.

Dr. Bipun Man Pati received the bachelor's degree in Information Technology from Nepal College of Information Technology, Nepal, in 2011, and the M.Eng. and D.Eng. degrees in Telecommunication Engineering from Asian Institute of Technology, Thailand, in 2016 and 2020, respectively. His master's degree was sponsored by the Asian Development Bank Japan Scholarship Program. He was a C2F Postdoctoral Fellow with Chulalongkorn University from September 2021 to November 2022. He was a senior research associate with Glodal, Inc., Yokohama, Japan, from February 2021 to July 2022. He is currently a senior assistant professor and the chief of the AI Research Center, Advanced College of Engineering and Management, Tribhuvan University, Kathmandu, Nepal. His research interests include wireless communications, signal processing, and machine learning. He can be contacted at email: bemaanpati@gmail.com.

Dr. Atsushi Ito received B.S. and M.S. degrees from Nagoya University in 1981 and 1983, respectively. He also received a Ph.D. degree from Hiroshima City University in 2007. From 1983 to 2014, he was with KDDI Corporation's Research and Development Laboratories. From 2014 to 2020, he was a professor at the Graduate School of Engineering of Utsunomiya University. Now he has been a professor at Chuo University since 2020. During 1991-1992, he was a visiting scholar at the Center for the Study of Language and Information (CSLI) of Stanford University. His current research interests include ad hoc networking, user interface design, AI applications for the hammering test, and ICT-based agriculture. He can be contacted at email: atc.00s@g.chuo.

Sushant Chalise is currently pursuing a master in Data Science and Analytics at Pulchowk Campus, Institute of Engineering, Tribhuvan University, as well as working as a system engineer at the Center for Information and Technology, Pulchowk Campus, where he focuses on designing and implementing practical digital solutions to support academic and administrative operations. His research interests include marker-based vision systems, system interoperability, and the practical application of intelligent systems to address real-world challenges. He also serves as the chair of IEEE Young Professionals Nepal, fostering technical collaboration and professional growth among early-career engineers. He can be contacted at email: sushant.chalise@pcampus.edu.np.

Sanjivan Satyal is an assistant professor at the Institute of Engineering, Pulchowk Campus. He has completed his bachelor's degree in Electronics and Communication Engineering from the Kathmandu Engineering College, Institute of Engineering, Tribhuvan University, and his master's degree in Information and Communication Engineering from Pulchowk Campus, Institute of Engineering, Tribhuvan University. With over 15 years of experience in teaching and research-oriented projects, his academic and professional interests span deep learning, signal processing, audio processing, the internet of things (IoT), and communication systems. He can be contacted at email: sanziwan.satyal@pcampus.edu.np.