Automatic Identification of Monuments in Images using Single-Shot Detectors

[Author name(s) masked for blind review]

[Author information masked for blind review]
[Author contact masked for blind review]

Abstract
Monuments, embodying historical, archaeological, and cultural significance, serve as gateways to unraveling rich histories, particularly for foreigners. To aid monument identification within images, we fine-tuned the lightweight CNN model, MobileNetV2, with SSD for feature extraction and prediction of monument locations and labels. Subsequently, we trained the more resource intensive YOLOv5s model. Our dataset comprised manually collected databases from Kathmandu Valley's three Durbar Squares: Kathmandu, Bhaktapur, and Patan. The SSD reached a maximum mAP@0.5 score of 78.68% for test data, while the YOLOv5s model demonstrated superior performance, with mAP@0.5 scores peaking at 92.77%.

Keywords
CNN, Fine Tuning, MobileNetV2, Monument Detection, SSD, Transfer Learning, YOLOv5s

1. Introduction

Recognizing and preserving historical monuments is crucial for understanding diverse cultural heritages and fostering educational experiences. However, accessing information about these monuments can be challenging for many without specialized knowledge or resources. This research addresses this gap by leveraging deep learning technology to automatically identify historical landmarks in digital images. Detecting monuments and accurately identifying them among similar structures in the background presents challenges in properly bounding each monument and assigning accurate class labels. This task is particularly complex when dealing with monuments which share similar structural details.

The major contribution of this research lies in pioneering the use of single shot object detection models to identify monuments, departing from previous approaches focused on classification models. This research project integrates two different object detection models: the MobileNetV2-SSDLite model for offline inference using smartphone hardware, and the relatively heavier YOLOv5s model for online inference in a mobile application.

Moreover, the research involves collecting and annotating monument images from all Durbar Squares within the Kathmandu Valley of Nepal namely Kathmandu, Bhaktapur and Patan Durbar squares.

2. Related Papers

Recent advancements in landmark detection have primarily focused on classification models, with notable successes reported. Authors of [1] utilized transfer learning with pre-trained Inception V3 to identify 12 Indian monuments in major cities, training on a dataset of around 300 images per monument. The model demonstrated test accuracies of 96-99% for 20 images. A study was conducted by [2] employing comparative study employing Simple Multiclass Model (SMM), Ensemble of Binary Classification (EBC), and Transfer learning algorithm for cultural heritage site classification. The transfer learning approach on MobileNet achieved 98.75% accuracy, outperforming the other two methods by 10%. Using MobileNet V2 reduced dimensionality from 100 MB to 20 MB without sacrificing accuracy. Researchers from [3] developed a monument recognition mobile app using a CNN algorithm based on MobileNet through Transfer learning, recognizing 46 monuments with varied sizes and conditions. The dataset which comprised of 50-100 pictures of each monument was expanded using data augmentation techniques. The model achieved 95% accuracy on the test set.

In [4], SSD, utilizing VGG-16 as its backbone, introduced the Single Shot Multibox Detector. It discretizes feature maps to predict detection windows per cell, offering multiple windows per class and outperforming two-stage methods like Fast R-CNN and Faster R-CNN. In [5], researchers compared MobileNetV2 and MobileNetV1 for object detection using the mobile-friendly SSD variant, SSDLite, on the COCO dataset. MobileNetV2-SSDLite achieved comparable mAP to other SSD variants (SSD300 and SSD512) but with significantly fewer parameters and lower computation cost. It was found to be 20 times more efficient and 10 times smaller than the original SSD while outperforming YOLOv2 on COCO. A study done by [6] compared YOLO and MobileNet-SSD for single-stage object detection, finding both suitable for diverse scenarios. YOLO prioritized accuracy but faced localization challenges, while SSD excelled in speed. However, SSD with MobileNetV2 could offer comparable speed to YOLOv5s with less demanding hardware, with a slight accuracy trade-off.

The mentioned research advancements in monument classification often struggle with localizing and detecting multiple monuments within a single image, necessitating the use of object detection models. In our exploration, no studies were found on identifying Nepalese monuments, and similar monuments clustered closely together. Object detection thus became crucial for accurately classifying and localizing these monuments.

3. Dataset Preparation

3.1 Dataset Collection

Training an object detection model to detect and recognize monuments requires a large amount of image data, which was not readily available. Therefore, data of all the monuments was manually acquired by taking photos and videos during on-site visits. Additionally, relevant images were acquired by scraping websites. Photos of prominent monuments were taken from every possible angle, and videos were recorded for later extraction of frames to obtain images. The image dataset comprises prominent monuments in Kathmandu, Bhaktapur, and Patan Durbar Square located in Kathmandu Valley, Nepal. A total of 18734 images were collected, covering 59 different monuments in Kathmandu, Bhaktapur and Patan Durbar Square combined.

Table 1: Image Collected Count

Durbar Squares	Monument Class Count	Original Dataset Count	Augmented Dataset Count
Kathmandu Durbar Square	15	4853	9787
Bhaktapur Durbar Square	29	9291	18206
Patan Durbar Square	15	4590	9702
Total	59	18734	37695

3.2 Dataset Preprocessing

To facilitate object detection model training, ground truth annotations were created. This involved manually labeling bounding boxes around objects of interest in each image. Prior to this annotation process, images were resized to a standard size of 512 x 512 pixels using bilinear interpolation. A third-party tool, LabelImg, was employed to generate annotation files. These files contained the bounding box coordinates for each object. The format differed depending on the target model: Pascal VOC XML for the MobileNetV2-SSDLite model and a simpler .txt format for the YOLOv5s model.

3.3 Dataset Augmentation

Data augmentation is the process of artificially synthesizing new image samples from existing data. This technique increases the size of the dataset, helps reduce overfitting, improves generalization, and ultimately enhances model performance. Different data augmentation techniques were applied, categorized into photometric and geometric. Photometric techniques modify the image's visual properties, such as adjusting brightness, contrast, and saturation. Additionally, techniques like adding Gaussian noise and applying Gaussian blur were also used. Geometric techniques, on the other hand, modify the image's spatial layout. This involved translation (shifting the image within -40 to 40 pixels) and rotation (rotating the image in either clockwise or counter-clockwise direction within the range of 8 to 16 degrees).

4. Methodology

4.1 System Architecture

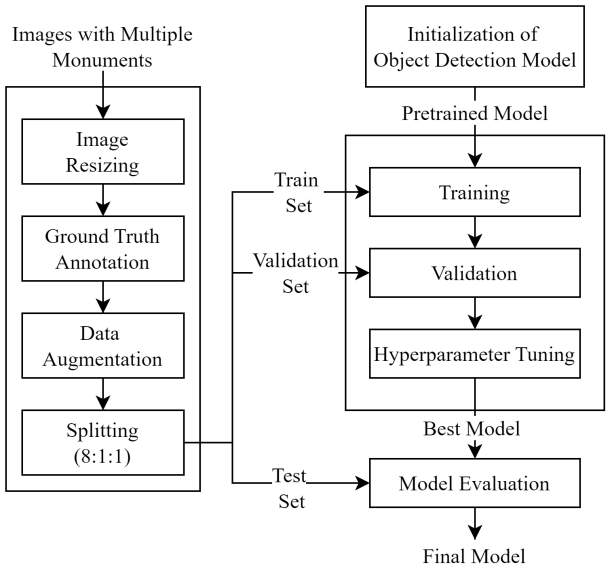

Figure 1: High-Level System Block Diagram

After preparing the dataset, the next step involved selecting a split ratio of 8:1:1. For MobileNetV2-SSDLite, the base feature extractor model, MobileNetV2, was initially acquired as a pretrained model trained on the ImageNet dataset. Before integrating the base model with SSDLite, it underwent fine-tuning using the monument dataset to address monument class detection.

The hyperparameter tuning process involved manually selecting parameter values and training to validate the model's loss and accuracy. Conversely, the YOLOv5s model was obtained and directly trained from Ultralytics.

4.2 MobileNetV2-SSDLite Model

The original Single Shot Multibox Detector (SSD) utilizes VGG-16 as its backbone network. However, due to its large size, it is not suitable for mobile applications. To address this limitation, MobileNetV2, a CNN-based streamlined architecture designed for lightweight deep neural networks in mobile and embedded vision applications, is used as a feature extractor alongside SSD layers. MobileNetV2 employs an inverted residual structure, where residual connections exist between bottleneck layers. Each bottleneck residual block consists of a Depthwise Separable Block with an additional Expansion Layer and a Skip Connection between two ends. This use of the inverted residual module effectively addresses the issue of vanishing gradients, ensuring proper propagation of gradient information across deeper network layers during the backpropagation process, thereby facilitating effective training.

The convolution block begins with a 3x3 depthwise convolution layer, which applies convolution along a single spatial dimension (i.e., channel). This is followed by a 1x1 pointwise convolution layer that combines the output channels of the depthwise convolution to create new features. Together, the depthwise and pointwise convolutions form a 'Depthwise

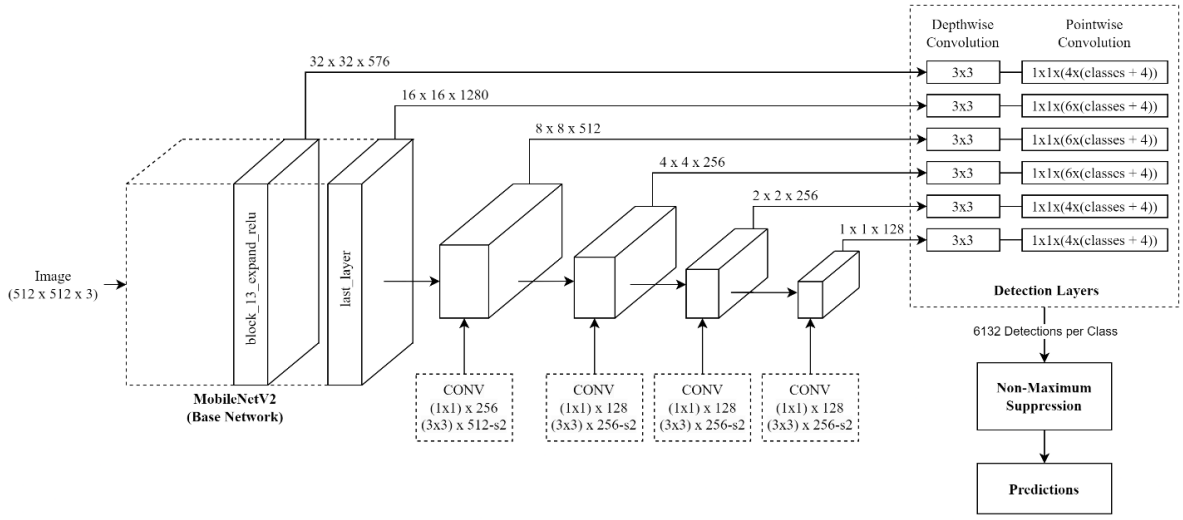

Figure 2: MobileNetV2-SSDLite Model Architecture

Separable' convolution block, which efficiently extracts meaningful information from the input data. The full architecture of MobileNetV2 includes an initial fully Convolution Layer with 32 filters, followed by seventeen Residual Bottleneck Layers. Feature maps are then extracted from intermediate layers of MobileNetV2, and the results of subsequent standard convolution are applied to the last layer of the feature model.

SSDLite utilizes depthwise separable convolution blocks instead of standard convolution blocks, reducing the parameter size. The detection layer produces a total of 6132 detection heads for each class label. The Non-max Suppression algorithm (NMS) is employed to suppress bounding boxes with high degrees of overlap with one another. NMS selects the bounding box with the highest confidence score and suppresses all other bounding boxes with an Intersection over Union (IoU) value lower than a predefined threshold.

4.3 YOLOv5s Model

YOLOv5 offers four variants - small (s), medium (m), large (l), and extra-large (x) - tailored to different memory capacities while retaining core principles from previous versions. Despite the absence of official documentation, YOLOv5's architecture includes familiar components: CSPDarknet53 as the backbone, SPP and PANet in the neck model, and a detection head similar to YOLOv3. Key enhancements feature the Focus layer and CSP in bottleneck layers, streamlining operations without compromising performance.

The Focus layer in YOLOv5 consolidates YOLOv3's initial layers, cutting computational load. The CSPBottleneck layer improves accuracy and speed via convolutional, batch normalization, and SiLU activation sub-blocks, with a cross-stage hierarchy connection for gradient management. Spatial Pyramid Pooling (SPP) boosts YOLOv5's flexibility with input size, aiding feature extraction from diverse image dimensions, especially useful for crowded object detection.

SPP-Fast accelerates pooling operations by exploiting spatial correlation, preserving accuracy while reducing computational costs. PANet facilitates feature fusion across network levels, ensuring comprehensive representation for improved detection. YOLOv5's detection head employs three convolutional layers to predict bounding boxes, objectness scores, and classes, yielding 16128 detections per class for 512x512 input images. Non-Maximum Suppression filters redundant detections for the final output, highlighting YOLOv5's efficacy in real-world object detection.

Comparing parameter sizes, MobileNetV2-SSDLite has 5.8 million parameters, significantly fewer than MobileNetV2-SSD's 14.8 million parameters. Utilizing 'Depthwise Separable' convolution blocks in the detection head halved the parameter size. In contrast, the smallest YOLOv5 version has 7.2 million parameters.

5. Result and Analysis

5.1 Mean Average Precision (mAP) Score

Using accuracy plots for object detection models is considered unreliable due to the complexity of multi-class, multi-label problems. The performance of object detection and localization algorithms was evaluated using Average Precision (AP). Average precision for each class was calculated using the precision-recall curve for the corresponding class, taking 11 points from each curve.

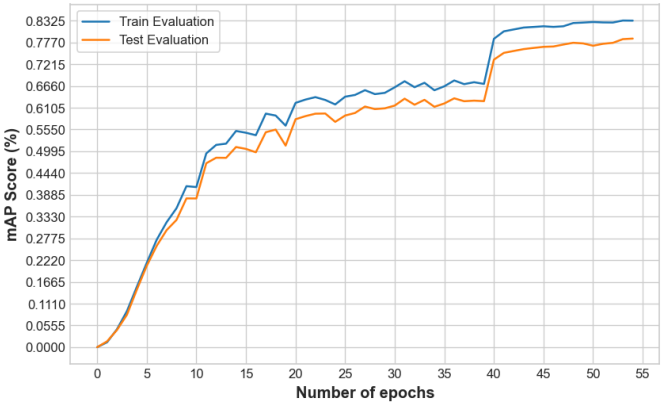

Figure 4: mAP Curve - MobileNetV2-SSDLite Model

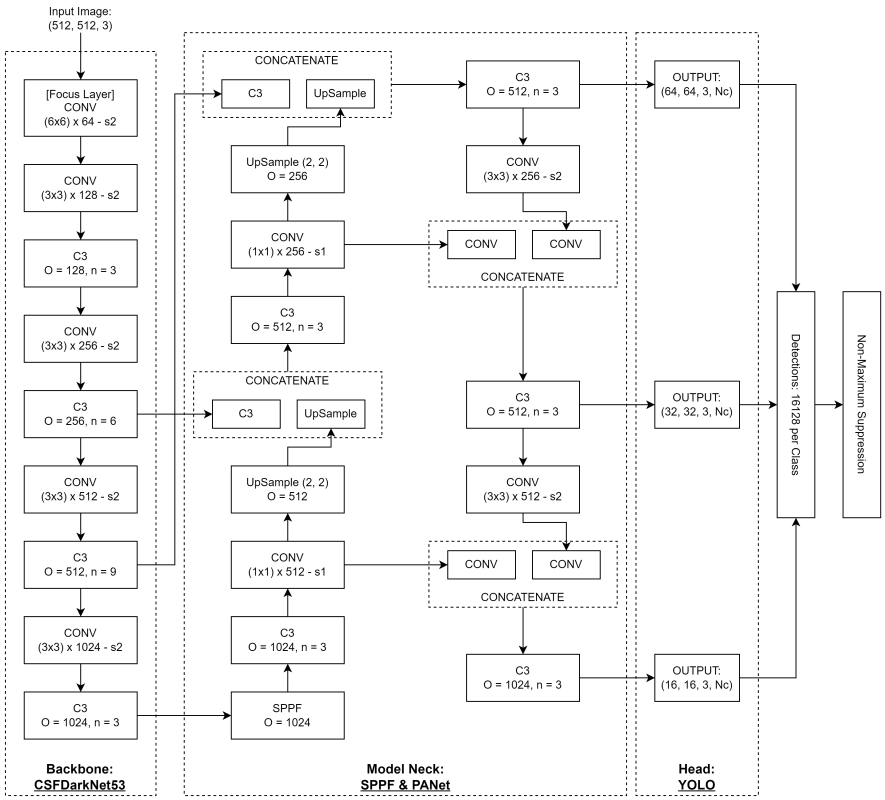

Figure 3: YOLOv5s Model Architecture

The mean average precision score, obtained by averaging the individual average precision scores from all classes, provides the Mean Average Precision (mAP) value. Figure 4 shows graph of mAP score on the test and train dataset after each epoch. It appears that the mAP score increased around the 38-40th epoch, which is dues to adaptive learning rate scheduler which changed from 1×10^{-4} to 6×10^{-5}. The mAP score started at 0 and gradually increased, plateauing at 78.68% by the 55th epoch for the test evaluation while for the train evaluation the mAP score started at 0 and pleateaued at 83.25%.

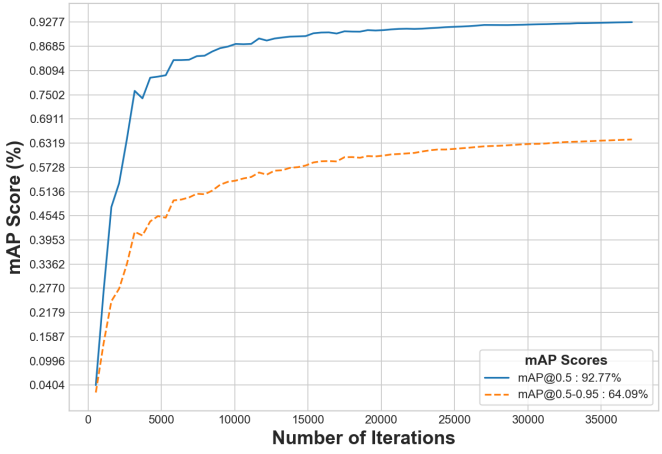

Figure 5: mAP Curve - YOLOv5s Model

The YOLOv5s model produces two mAP scores, as shown in Figure 5. The first score, mAP@0.5, represents the mean average precision at a 50% IoU threshold, peaking at 92.77%. The second, mAP@0.5-0.95, averages the mean average precision obtained between 0.5 to 0.95 at 0.05 step intervals, reaching a maximum of 64.09%. These scores indicate satisfactory performance on the training dataset and potential generalization to unseen input images, as evaluated on a test dataset of online-collected images.

5.2 Precision-Recall Curve

Figure 6 illustrates PR curves for best and worst performing class detections from MobileNetV2-SSDLite inference. "Trailokya Mohan" class achieves the highest precision, while "Bhupatindra Malla Column" shows the lowest. The model effectively localizes and classifies the former but struggles with the latter, possibly due to atypical aspect ratios in bounding boxes. Adjusting anchor box aspect ratios using anchor box detectors, through K-means clustering on bounding box sizes, may enhance precision for the 'Bhupatindra Malla Column' class.

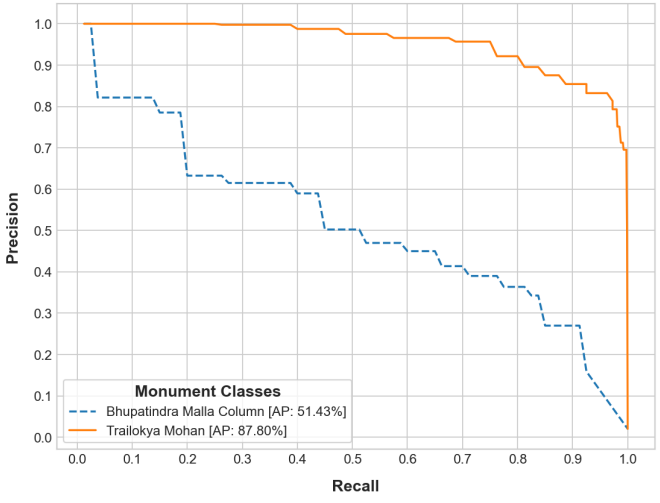

Figure 6: PR Curves for Best and Worst Classes

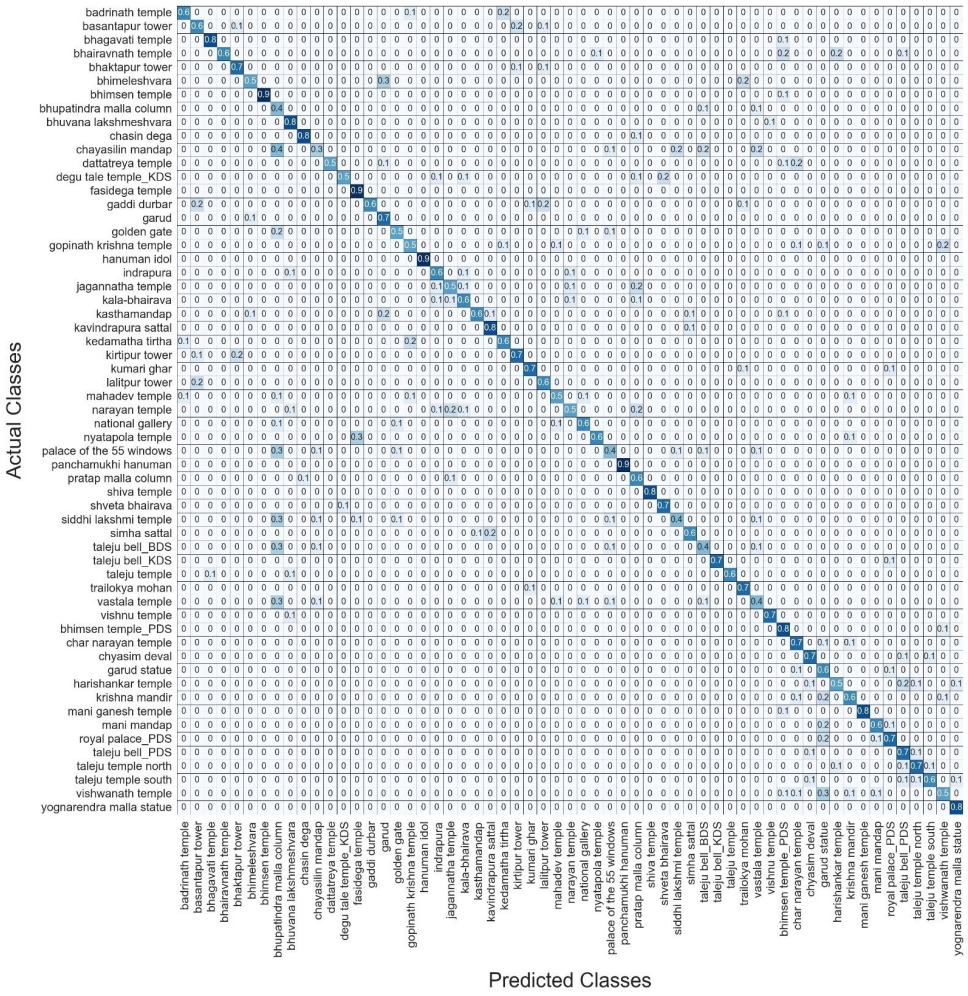

Figure 7: Confusion Matrix - MobileNetV2-SSDLite

Figure 8 reveals that the YOLOv5s performed exceptionally well for the ”Krishna Mandir” class with an average precision of 99.49%, while achieving its lowest performance for the ”Pratap Malla Column” class with an average precision of 74.60%. Possible reasons for the lower AP score for this class include fewer instances in the evaluation dataset or bounding box issues, given the columnar shape of the monument. However, discrepancies in annotating the ”Pratap Malla Column” class, where annotators varied in delineating the bounding boxes, likely contributed to this behavior, leading to errors in the annotation phase and subsequently affecting the model's performance.

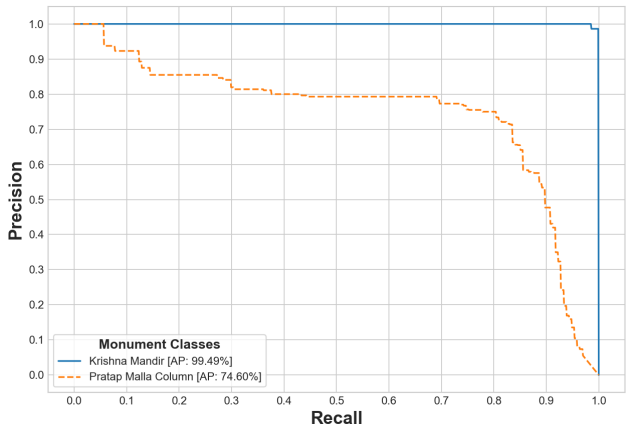

Figure 8: PR Curves for Best and Worst Classes

5.3 Confusion Matrix

Figure 7 shows the confusion matrix from MobileNetV2 SSDLite on the test dataset. The matrix, normalized by column, illustrates the distribution of model predictions for each class. The model achieved higher prediction accuracy for classes such as ”Bhimsen Temple,” ”Fasidega Temple,” ”Hanuman Idol,” and ”Panchamukhi Hanuman,” possibly because these monuments are typically isolated in their surroundings, making them easier to localize and classify accurately.

In contrast, the model showed lower prediction accuracy for the ”Chayasilin Mandap” class and others like ”Bhupatindra Malla Column,” ”Palace of the 55 Windows,” ”Vatsala Temple,” ”Siddhi Lakshmi Temple,” and ”Taleju Bell_BDS.” This could be because these monuments often appear together in a single frame, making bounding box regression challenging, especially when structures overlap or are occluded. ”Chayasilin Mandap” is particularly challenging due to its high level of occlusion. These factors contribute to the model's difficulty in generalizing bounding box regression in such scenarios.

The confusion matrix for the YOLOv5s as shown in Figure 9 highlights its overall strength, with an average prediction accuracy exceeding 80% across all classes. The remaining 20% of predictions typically either represent background classes implicitly or indicate a failure to recognize any class at all. Notably, the ”Garud” class exhibits the lowest accuracy among all predictions, likely attributed to the monument's dark color

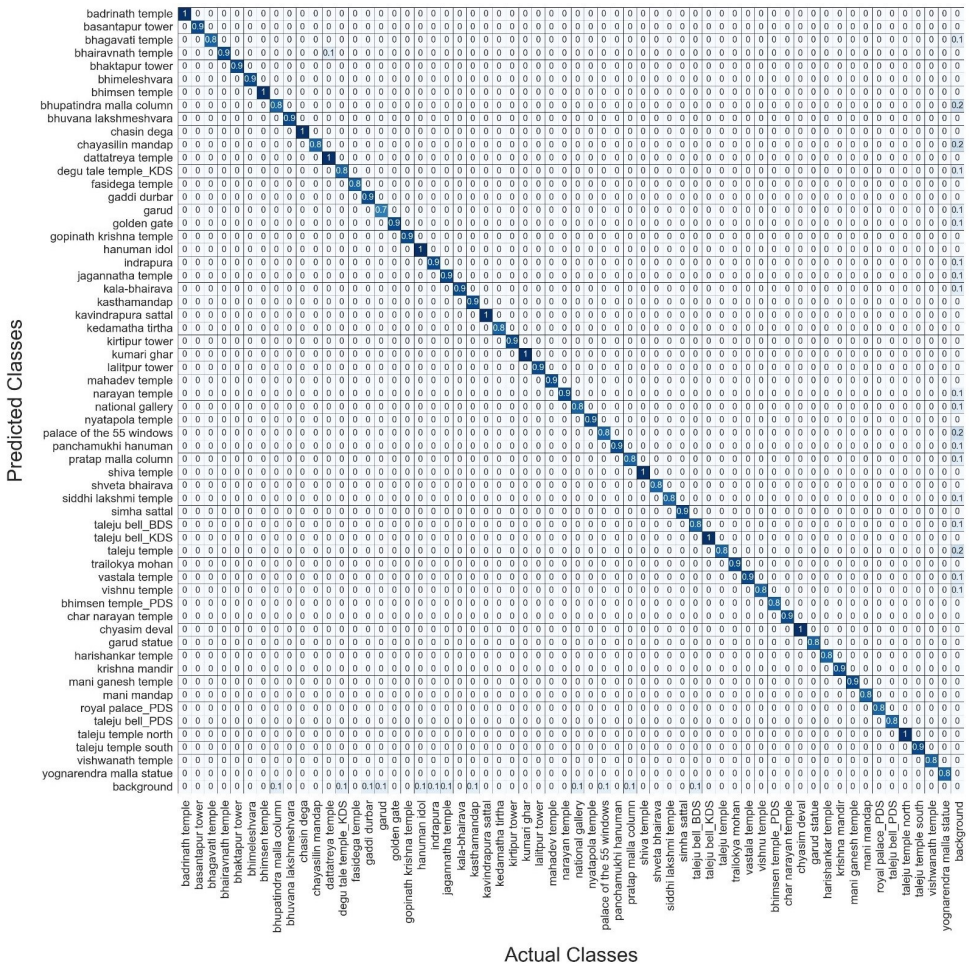

Figure 9: Confusion Matrix - YOLOv5s Model

and shade, leading the model to misclassify dark objects in the frame as "Garud."

The background class, appended to the end of the labels, captures instances where the model fails to classify any positive or negative class. Although not explicitly specified during training, it serves as a representation for cases not belonging to any monument class. Consequently, while the background class score may appear as 0, it plays a crucial role in assisting the classification of other classes.

5.4 Performance Metrics

Table 2: Performance Metrics for Top-10 Highest AP scores

S.N.	Monument Names	F1 Score (%)	AP@0.5 (%)
1	Trailokya Mohan	95.48	87.83
2	Basantapur Tower	91.5	86.78
3	Krishna Mandir	92.65	86.72
4	Badrinath Temple	93.22	86.47
5	Dattatreya Temple	76.43	86.47
6	Gaddi Durbar	86.06	85.69
7	Bhimeleshvara	91.53	85.47
8	Shiva Temple	94.37	85.41
9	Bhairavnath Temple	79.75	84.69
10	Chayasilin Mandap	84.08	84.65

Tables 2 and Table 3 display performance metrics for the top-10 highest average precision scores obtained during the inference test of MobileNetV2-SSDLite and YOLOv5s, respectively. A comparison of the top 10 lists from both models reveals that 'Krishna Mandir' is highly recognized by both due to its unique features compared to other monuments. The offline model identifies 'Trailokya Mohan' as highly precise to locate, followed by 'Basantapur Tower', while the online (YOLO) model finds 'Chyasim Deval' highly precise.

Table 3: Performance Metrics for Top-10 Highest AP Scores

S.N.	Monument Names	F1 Score (%)	AP@0.5 (%)
1	Chyasim Deval	98.79	99.5
2	Krishna Mandir	96.05	99.5
3	Mani Ganesh Temple	95.89	99.4
4	Gopinath Krishna Temple	94.95	98.7
5	Harishankar Temple	94.92	98.7
6	Lalitpur Tower	96.6	98.3
7	Char Narayan Temple	93.46	98
8	Chasin Dega	97.29	97.9
8	Taleju Bell_PDS	92.21	97.9
10	Taleju Temple North	98.12	97.8

This comparison between both models based on average precision for corresponding classes suggests that SSDLite excels in understanding monuments with less occlusion in their surroundings, whereas YOLO outperforms SSDLite in all aspects and exhibits better handling of background noise.

5.5 Inference Results

5.5.1 MobileNetV2-SSDLite Model Inference

Figure 10: Success Case of MobileNetV2-SSDLite Model

MobileNetV2 SSDLite accurately predicts present classes when monuments in the input image are large and unoccluded. Its proficiency is attributed to SSD's better performance with larger objects, allowing for true positive predictions.

Figure 11: Failure Case of MobileNetV2-SSDLite Model

The model's false negative for the 'patan malla column' class resulted from a failure to consider the unique aspect ratio of long columns depicted in the annotated image. Adapting the model's aspect ratios to diverse shapes may address this issue.

Figure 12: Generalization Assessment of MobileNetV2-SSDLite Model

An object detection model's ability to generalize on unseen images is crucial for leveraging its learning capabilities. In this instance, despite the absence of explicit annotation for the 'bhaktapur tower' class, the model successfully localized and classified it in the prediction result with a high confidence score of 86.2%.

5.5.2 YOLOv5s Model Inference

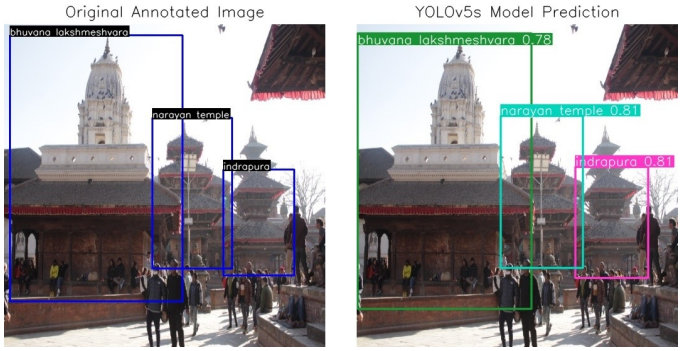

Figure 13: Success Case of YOLOv5s Model

The YOLOv5 model outperforms most of the object detectors in accurately classifying and localizing images, which is due to its robustness and superior learning capability. Therefore, the YOLOv5 model is able to accurately predict objects in the given image.

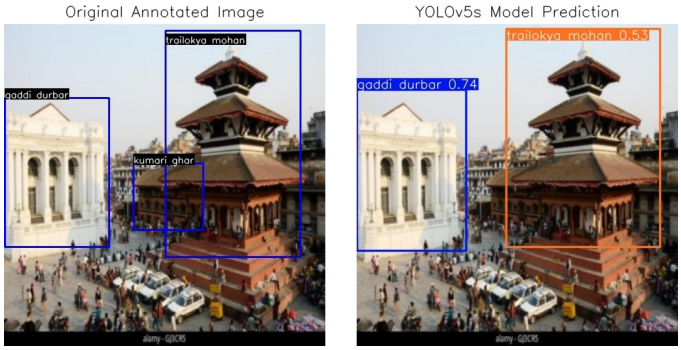

Figure 14: Failure Case of YOLOv5s Model

In the scenario described, the model fails to detect the "Kumari Ghar" monument hidden behind the "Trailokya Mohan" monument. This limitation may stem from insufficient image captures from this particular viewpoint. The photograph was taken from the "Maju Dega" monument, which no longer exists due to an earthquake. Consequently, the model lacks data from this perspective, hindering its ability to locate the "Kumari Ghar" monument.

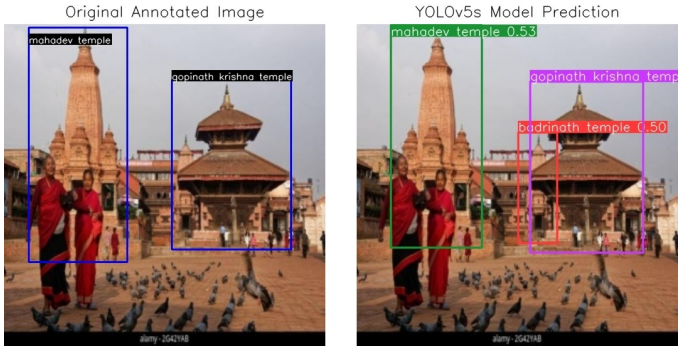

Figure 15: Generalization Assessment of YOLOv5s Model

YOLOv5 demonstrates exceptional learning capabilities, allowing it to discern complex features and patterns in monumental structures. Its remarkable generalization ability is evident in accurately predicting the presence of monuments not originally annotated in the image.

5.6 Inference Comparison between Models

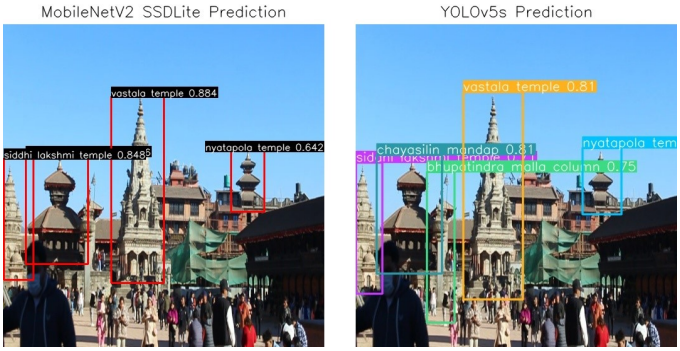

Figure 16: Comparison of Inference Result from Two Models

The side-by-side images display predictions from both the MobileNetV2 SSDLite and YOLOv5s models. YOLOv5s' ability to learn dynamic anchor boxes facilitated accurate detection and localization of the 'patan malla column' and 'bhupatindra malla column', surpassing SSDLite's limited bounding box knowledge, which does not include long column detection.

5.7 User Interface

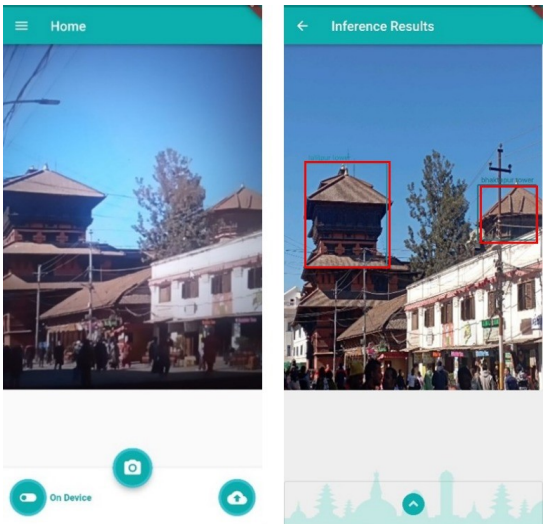

Figure 17: Image Capture and Loading Interface

6. Conclusion

In conclusion, MobileNetV2 SSD Lite and YOLOv5s object detection models were utilized for monument identification in the Kathmandu Valley's Durbar Squares. Integration into a mobile application facilitated both offline and online inference. This research presents an innovative solution for monument detection, scalable to meet diverse user needs.

Despite advances in monument detection, challenges remain in dataset diversity and augmentation techniques. Diverse datasets spanning various lighting conditions and elevations are essential for improving model accuracy and generalization. Innovative augmentation methods, like mosaic augmentations, can further enhance dataset diversity, improving model performance on new data. Additionally, techniques such as federated and continual learning offer promising solutions for addressing these challenges and expanding model capabilities.

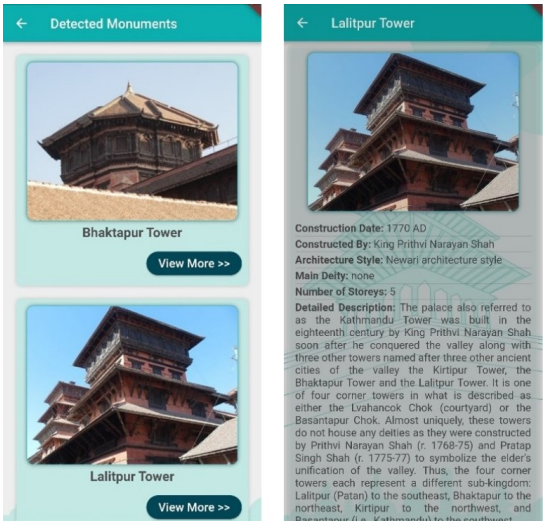

Figure 18: Inference Results and Monument's Detail View

Acknowledgments

We also acknowledge the support and suggestions received from the college department for the selection and hopeful execution of this project. Additionally, we appreciate the contribution of our designer for her valuable assistance in designing our application. Finally, we are sincerely grateful for the opportunity to present our work.

References

[1] Alice Lo Valvo, Domenico Garlisi, Laura Giarr´e, Daniele Croce, Fabrizio Giuliano, and Ilenia Tinnirello. A cultural heritage experience for visually impaired people. *IOP Conference Series: Materials Science and Engineering*, 949(1):012034, nov 2020.

[2] Uday Kulkarni, Meena S M, Sunil V Gurlahosur, and M. Uma. Classification of cultural heritage sites using transfer learning. pages 391–397, 09 2019.

[3] Valerio Palma. Towards deep learning for architecture: A monument recognition mobile app. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W9:551–556, 01 2019.

[4] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing.

[5] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[6] N Sabina, M P Aneesa, and P V Haseena. Object detection using yolo and mobilenet ssd: A comparative study. *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)*, 11(06), June 2022.