Automatic Identification of Monuments in Images using Single-Shot Detectors

Submitted by:

Prajesh Shrestha	**THA076BCT028**
Rishav Subedi	**THA076BCT036**
Santosh Pandey	**THA076BCT041**
Ujjwal Paudel	**THA076BCT048**

Supervised by:

Er. Dinesh Baniya Kshatri

Department of Electronics and Computer Engineering
IOE, Thapathali Campus

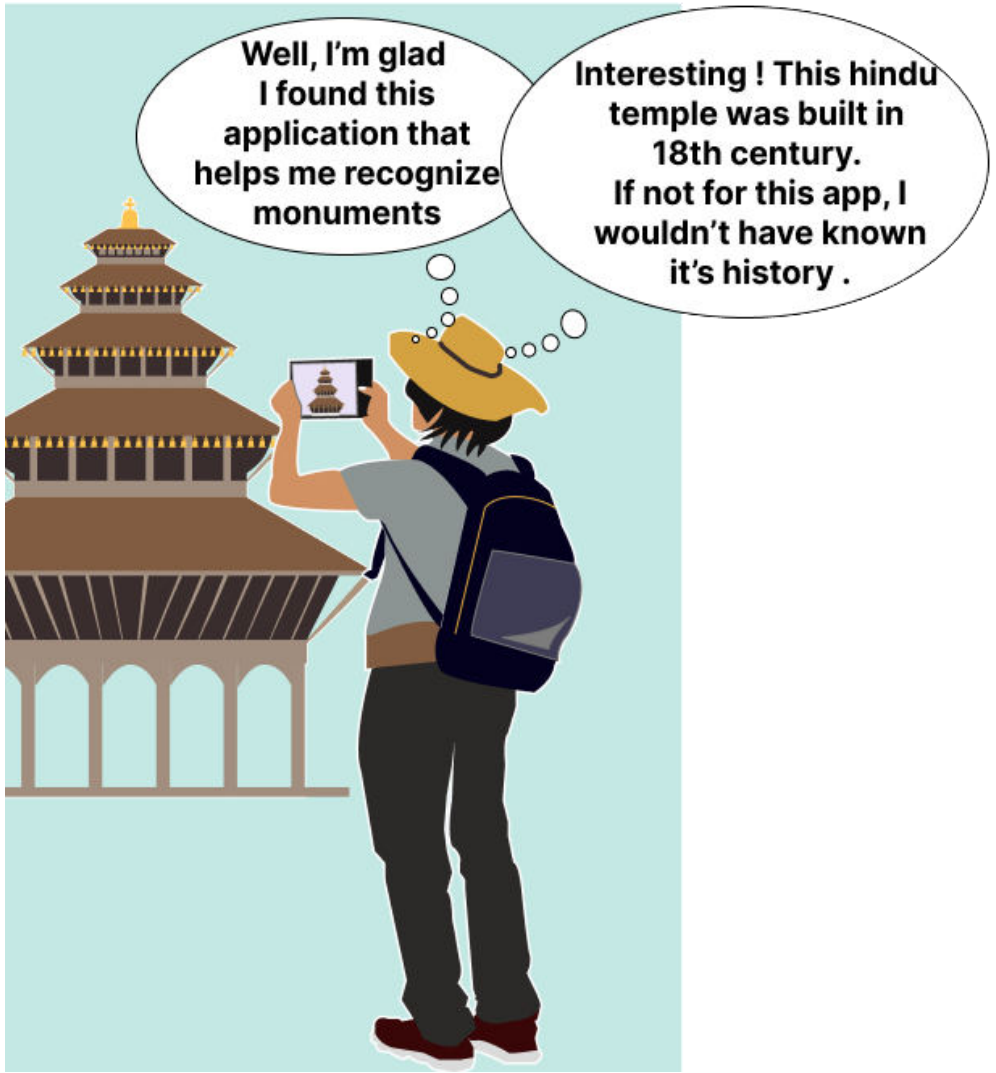
March, 2023

Presentation Outline

- Motivation
- Introduction
- Problem Statement & Objectives
- Scope of Project
- Project Applications
- Methodology
- Results
- Discussion of Results
- Future Enhancements
- Conclusion
- References

Motivation


I cannot seem to locate this temple.I dont even know its name.


Well, I'm glad I found this application that helps me recognize monuments
Interesting ! This hindu temple was built in 18th century. If not for this app, I wouldn't have known it's history .

Introduction

- System expects as input an image with monuments and provides their historical details
- Monument localization and recognition happens via neural networks requiring a single-pass over the input image
- Single-shot detectors receive training using resource intensive platforms
- Lightweight object detection model allows inferencing on mobile platforms
- Internet access allows inferencing on a server that runs a more complex model

Problem Statement & Objectives

Problem Statement

- No specialized tool to detect and identify historical monument
- Manual identification is time-consuming & may lead to inaccuracies
- Lack of available information sources and difficult to access

Objectives:

- To implement MobileNetV2 SSDLite and YOLOv5s object detection models to identify prominent monuments of the three Durbar squares within Kathmandu valley.
- To integrate the trained models into a mobile application that uses MobileNetV2 SSDLite for offline on-device inference and YOLOv5s for online inference facilitated by a REST API.

Scope of Project

Project Capabilities:

- Detects multiple monuments using single-shot object detectors
- Operates with or without internet connection
- Supports cross-platform development using Flutter
- Provides details of detected monuments extracted from database

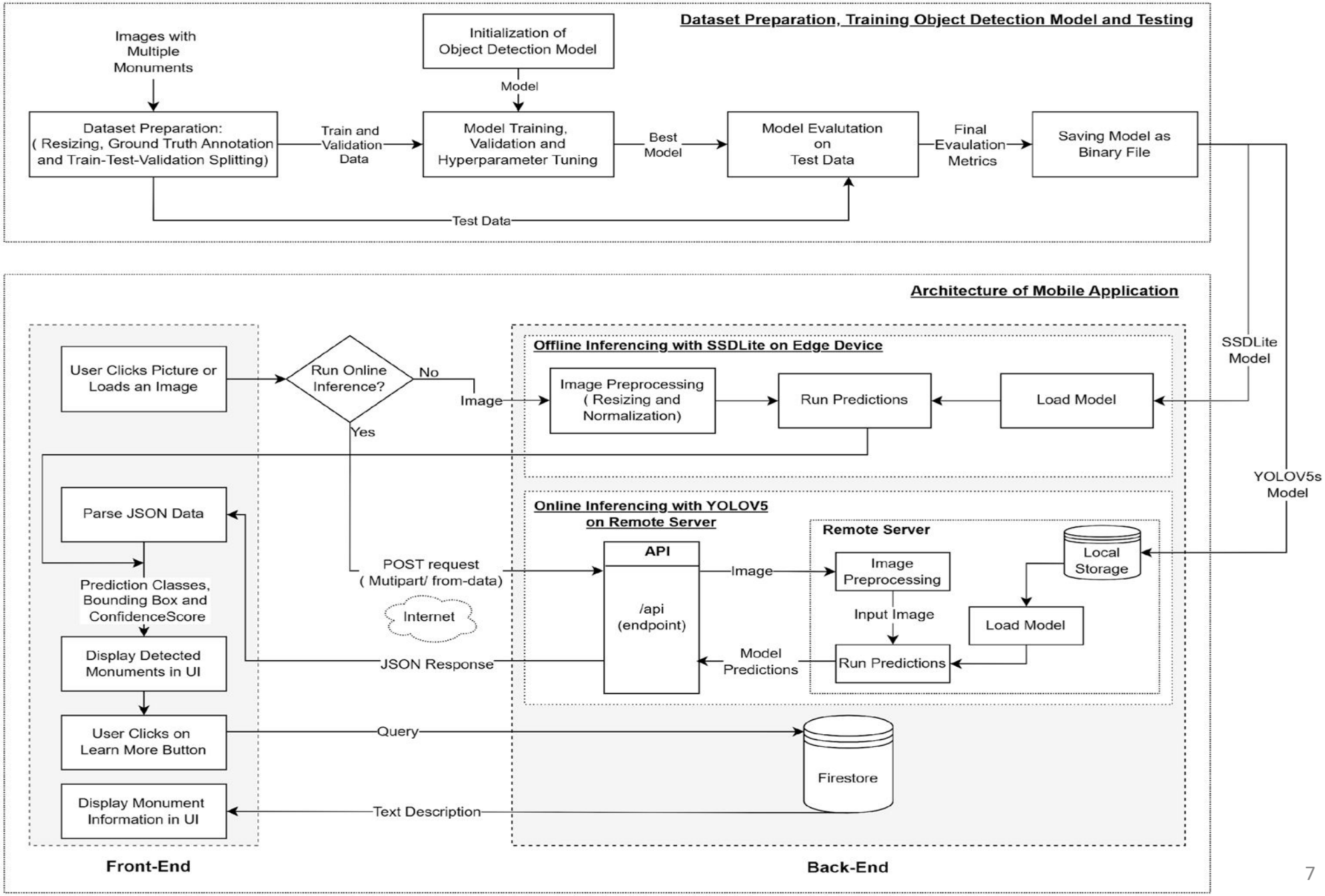
Project Limitations:

- Does not facilitate monument detection on live-feed or video
- May not detect some monuments due to lighting or image quality
- Historical details may be limited to publicly available information
- Information need to be updated manually when new details acquired

Project Applications

- Visual Monument Recognition
  - Detect monuments in images
- Visual Search Engine
  - Content based image retrieval
- Digital Mapping
  - Street view and panoramic site images
- Monument Database Management
  - Monument preservation and documentation
- Virtual Tour of Historical Sites
  - Interactive virtual environment

Methodology
(System Block Diagram)



Methodology
(Hardware Requirements - Training)

Specification	Detail
Architecture	Pascal
CUDA Cores	3584
Tensor Cores	64
Memory Size	16 GB
Base Clock	1.3 GHz
Boost Clock	1.48 GHz
Supported APIs	CUDA, DirectCompute, OpenCL, Vulkan
Operating System Support	Windows, Linux

Methodology
(Software Requirements)

Framework & Library	Usage
Tensorflow (2.11.0)	- Dataset encoding and decoding in TFRecords format - Building model architecture using Functional API - Model compilation and training
LabelImg (1.7.0)	- Image Annotation in PASCAL VOC XML format
Python Image Library (PIL) (9.4.0)	- Image Resizing - Image Data Augmentation (Photometric & Geometric)
OpenCV2 (4.7.0.68)	- Drawing bounding box on model predictions
Pytorch (1.13.1)	- Training YOLO model
Django (4.1.7)	- REST API
Futter (3.7.6-0.0.pre.1)	- Mobile App Development
Firebase	- Backend of Mobile Application - Firebase cloud storage - Firebase auth - Firestore databse

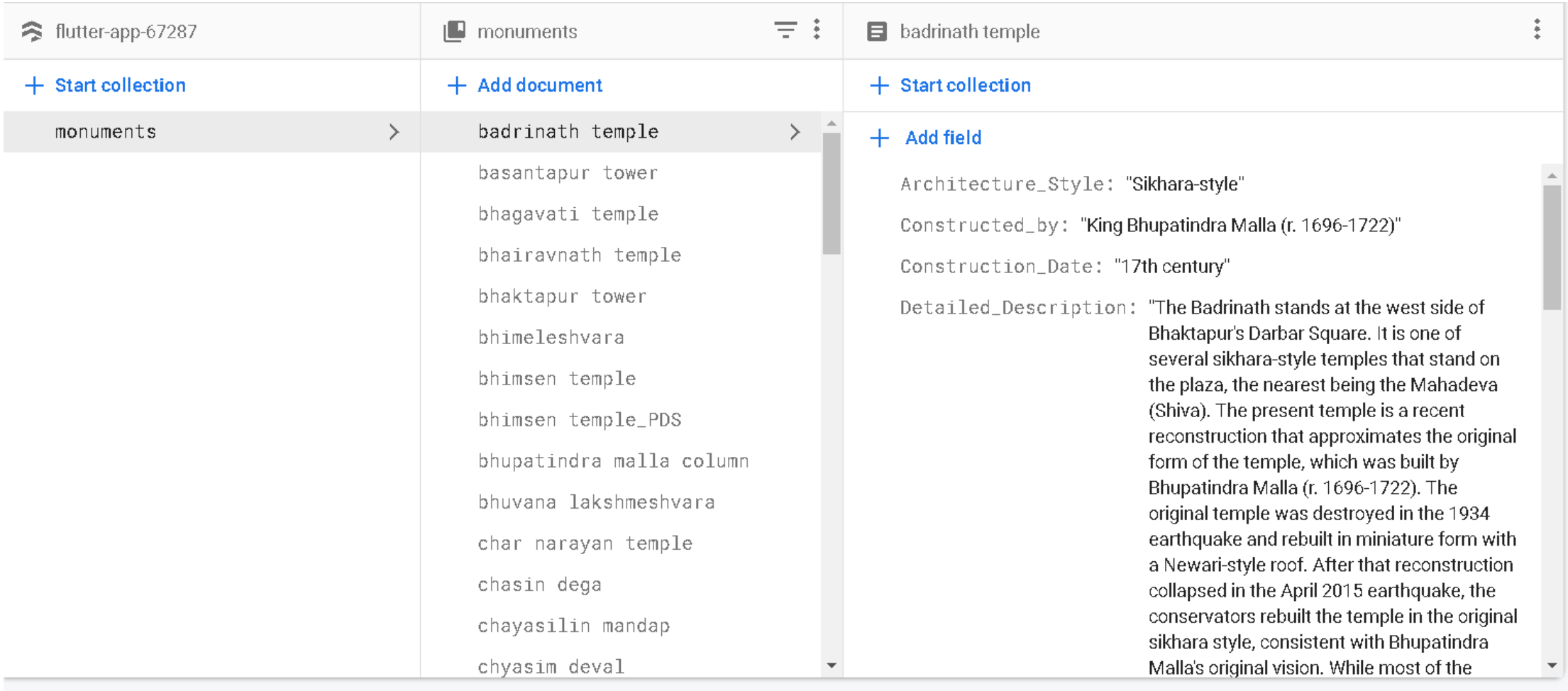
Methodology
(Dataset Collection and Preparation)

- Data Collection
  - On-site visits to collect images
  - Scraped websites to collect relevant images
  - A total of 11,146 images were collected :-
    - Kathmandu Durbar Square: 5,026
    - Bhaktapur Durbar Square: 1,634
    - Patan Durbar Square: 2,977
    - Online Scrapped: 1,509
  - Synthetically increased dataset size to 24,849 images

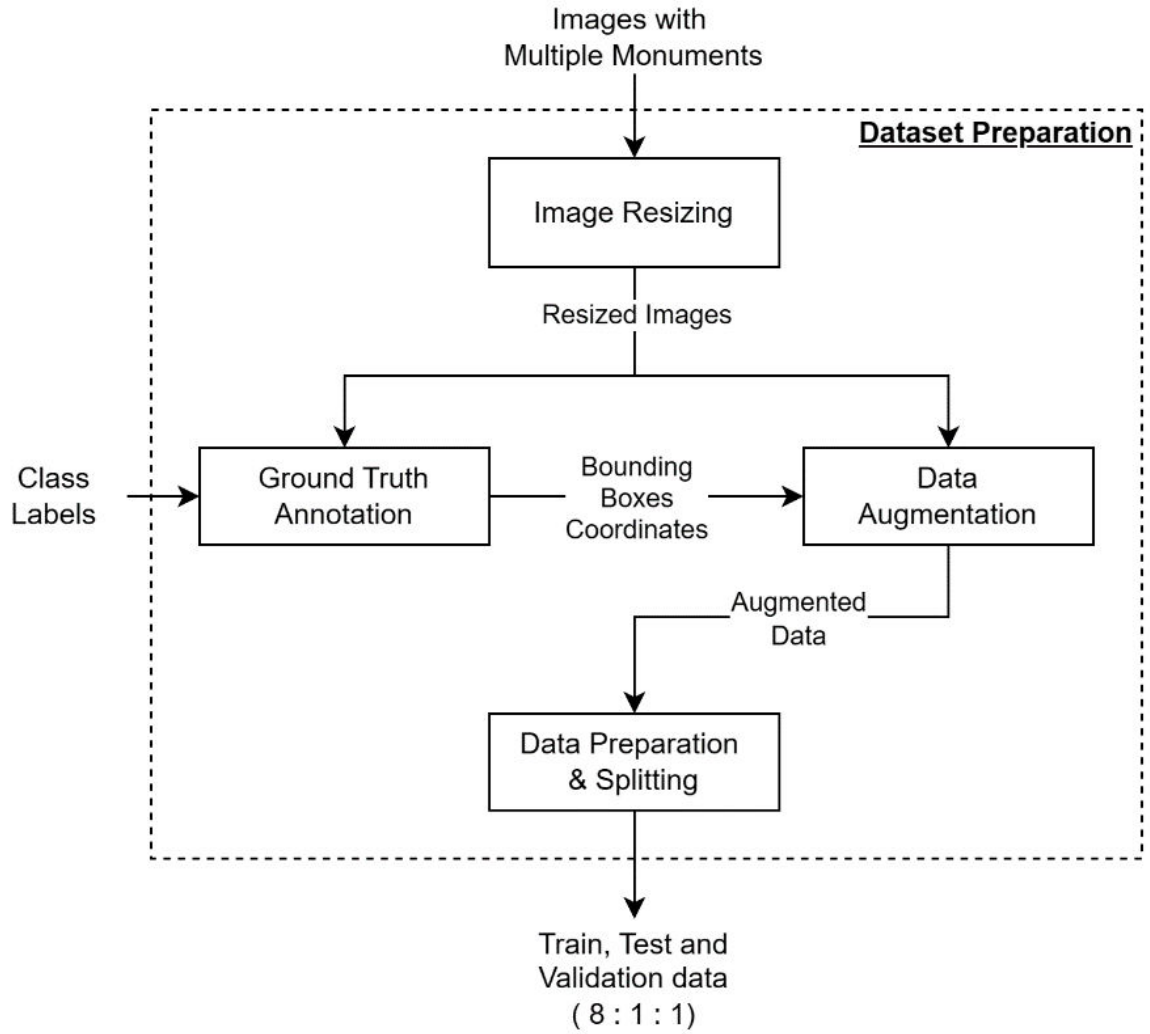
Methodology
(Information Collection and Database Integration)

- Monument Information was collected from various sources and stored in the database.
- Mobile Application Uses Firebase as database solution (Firebase Auth, Firestore Database, Firebase Cloud Storage)
- Firebase integration procedure
  - Set up Firebase Project
  - Register App to the project
  - Download and add Configuration Files
  - Add dependencies for the Firebase services used in app
  - Use the services by calling the required methods

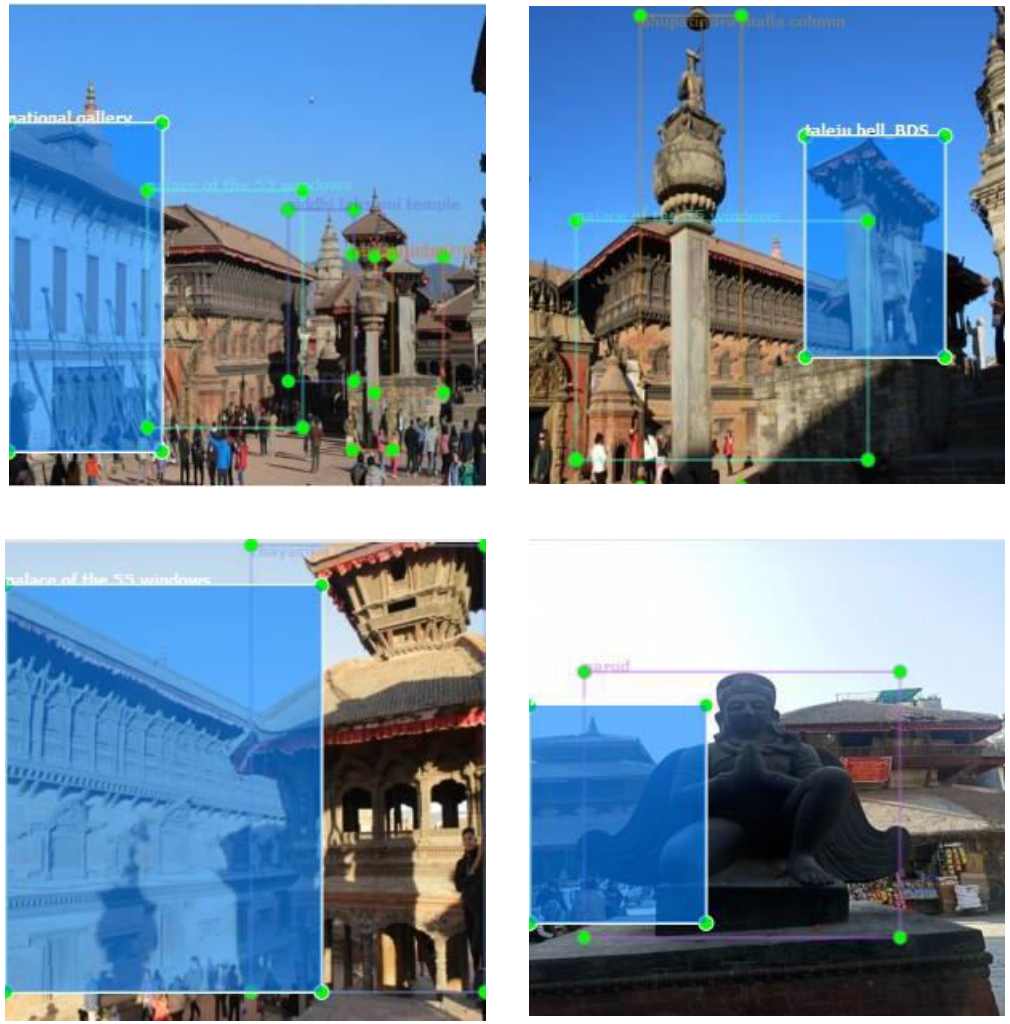
Methodology
(Database Integration)



Methodology
(Dataset Preparation)



Methodology
(Ground Truth Annotation)



Ground Truth Annotation

Generated PASCAL VOC XML Format

```
<annotation>
<folder>  …foder_path…</folder>
<filename> …image_name… </filename>
<path> …image_path… </path>
<source>
<database>Unknown</database>
</source>
<size>
<width>512</width>
<height>512</height>
<depth>3</depth>
</size>
<segmented>0</segmented>
<object>
                    <name>   …obj2 …</name>
                    <pose>Unspecified</pose>
                    <truncated>1</truncated>
                    <difficult>0</difficult>
        <bndbox>

                    <xmin>1</xmin>
                    <ymin>41</ymin>
                    <xmax>99</xmax>
                    <ymax>278</ymax>
        </bndbox>
</object>
<annotation>
```

YOLO Darknet TXT

```
<object-class_1> <x_center> <y_center> <width> <height>
<object-class_2> <x_center> <y_center> <width> <height>


0   0.511   0.405   0.184   0.366
5   0.127   0.545   0.193   0.303
```

Methodology
(Augmented Data)

Brightness & Contrast

Saturation

Gaussian Noise

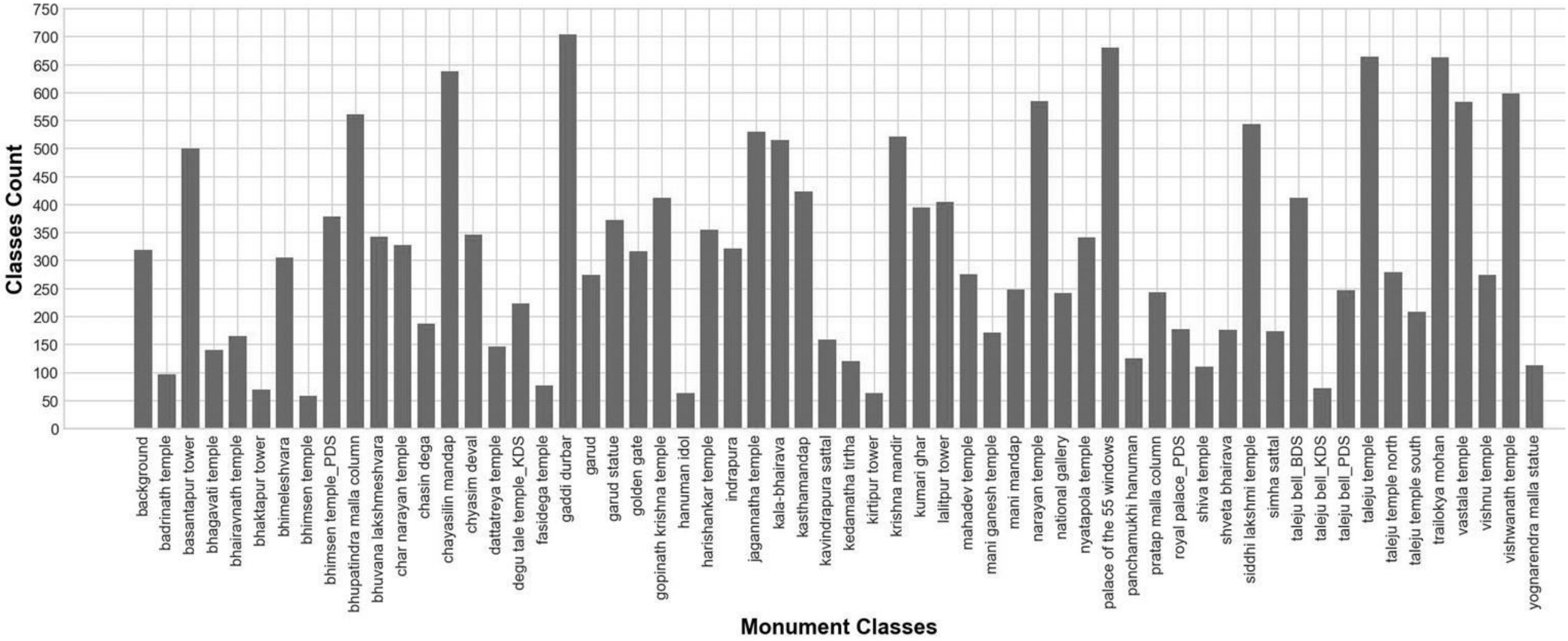
Translation

Gaussian Blur

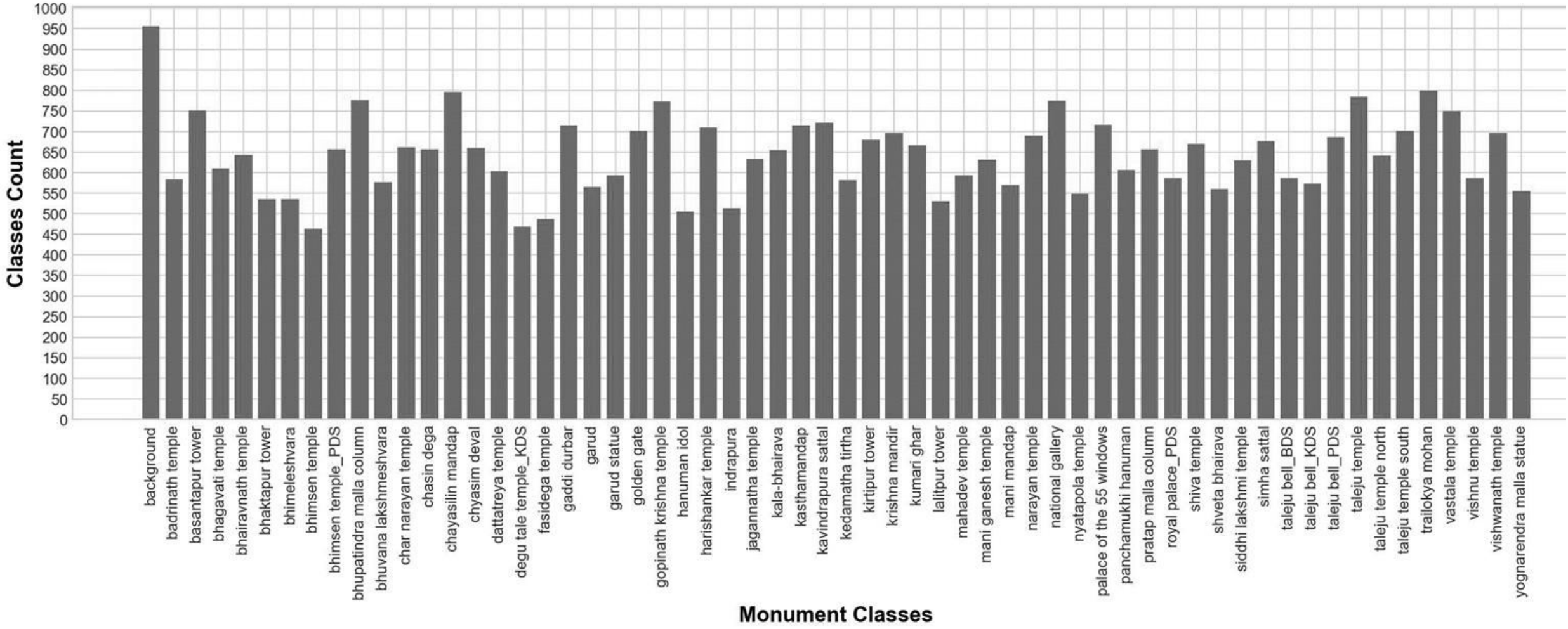
Rotation

Original and Augmented Images Side-by-side

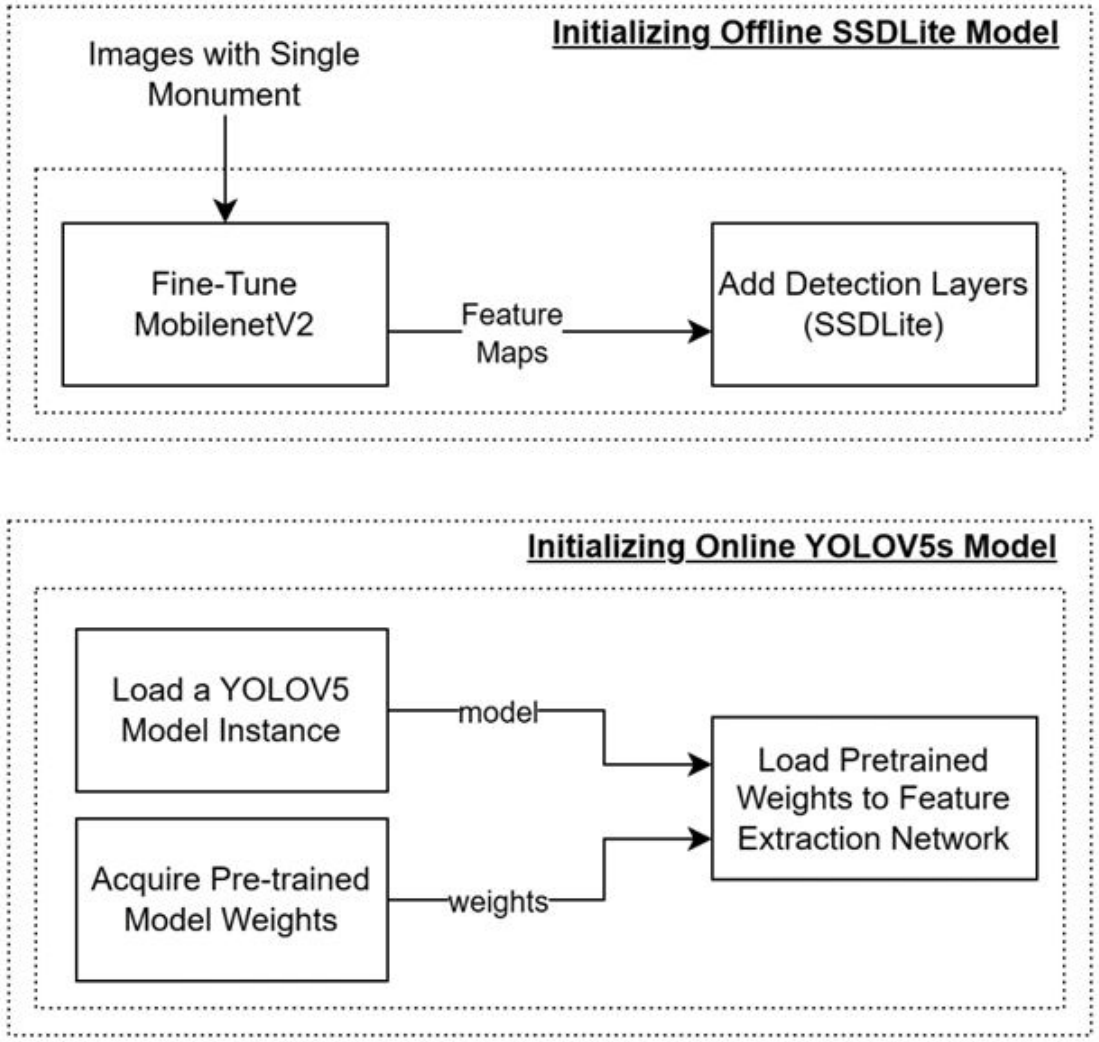
Methodology
(Class Objects Instances in Original Dataset)



Methodology
(Class Objects Instances in Augmented Dataset)

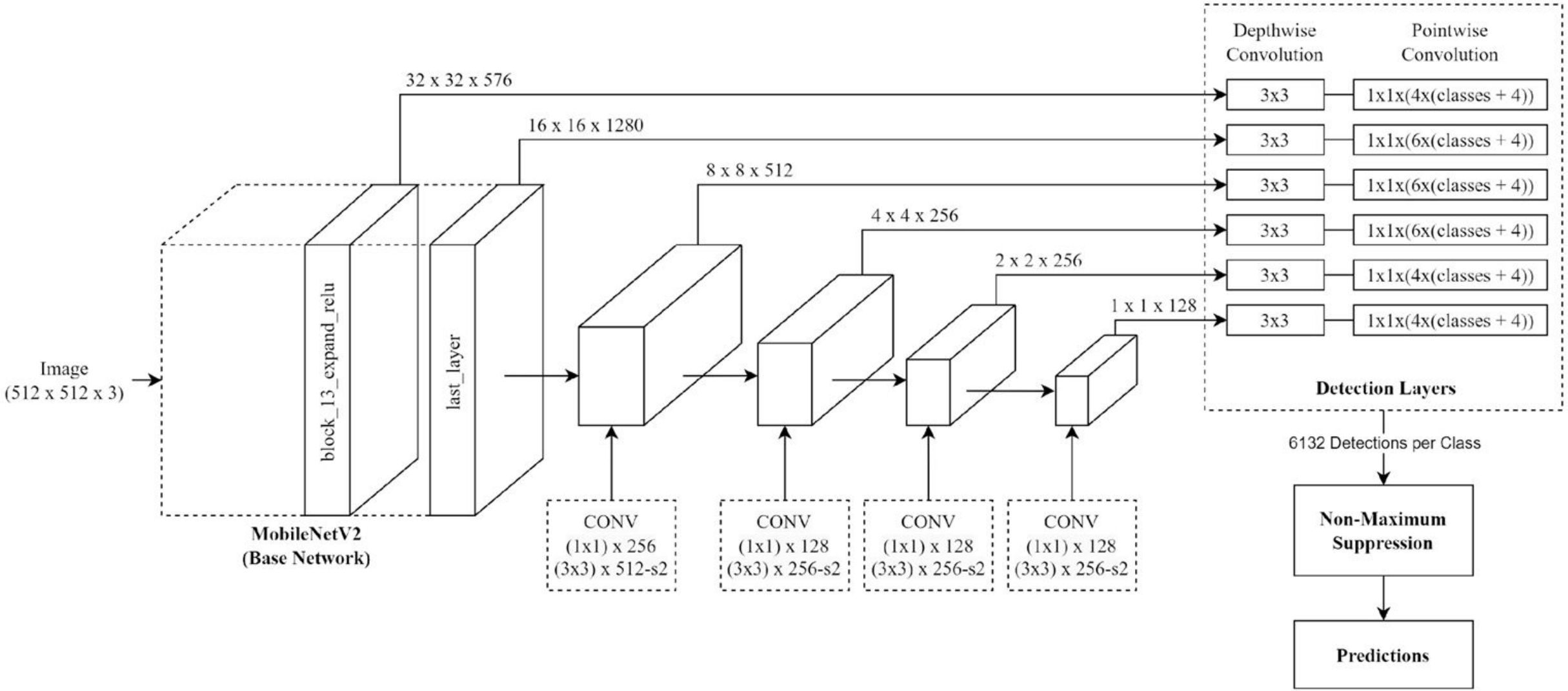


Methodology
(Initialization of Object Detection Models)

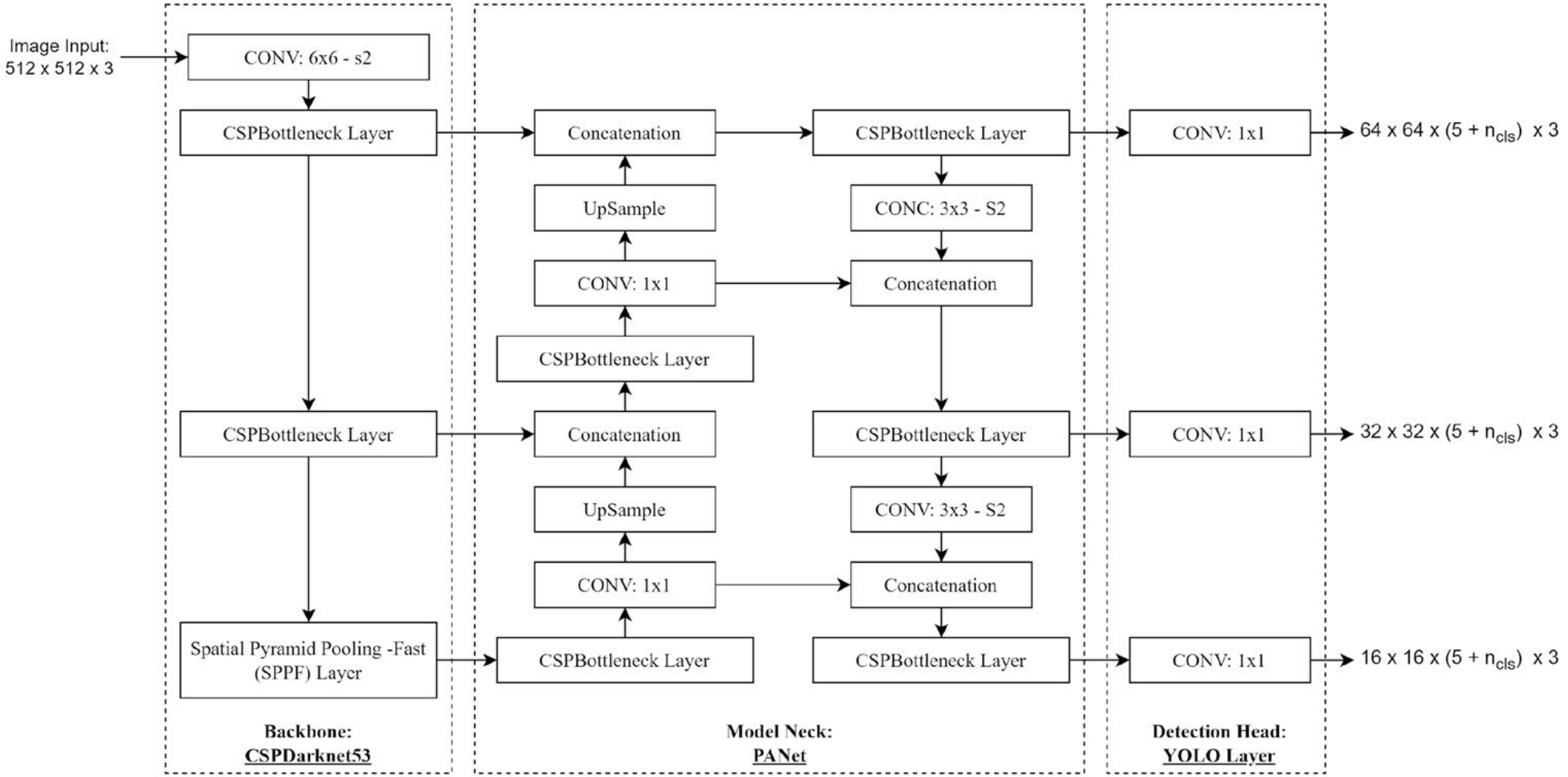


- Fine tuning MobilenetV2 base network
  - MobilenetV2 was previously trained on ImageNet
  - ImageNet comprises of 1000 classes none of which contain anything related to monuments
  - Then, it was fine-tuned on single monument images
  - First 70 layers were frozen
  - Fine-tuning performed to make the extracted features relevant to monuments

Methodology
(Architecture of MobileNetV2 SSDLite Model)



Methodology
(Architecture of YOLOv5s Model)



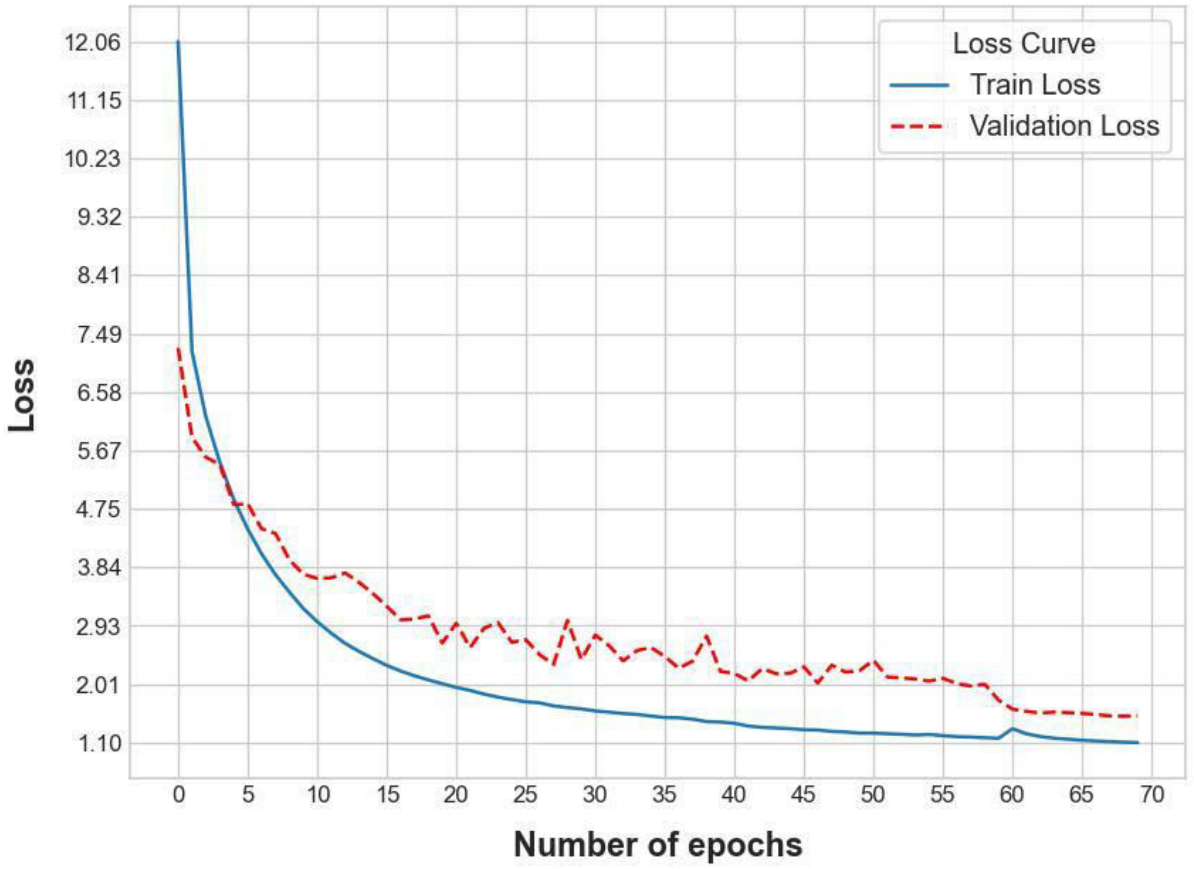
Methodology
(Model Complexity Comparison)

S.N.	Model Name	Parameters (Million)
1	MobileNetV1 SSD	6.8
2	MobileNetV2 SSD	14.8
3	MobileNetV2 SSDLite	5.8
4	VGG16 SSD	23.93
5	YOLOv5n	1.9
6	YOLOv5s	7.2
7	YOLOv5m	21.2
8	YOLOv5l	46.5
9	YOLOv5x	86.7

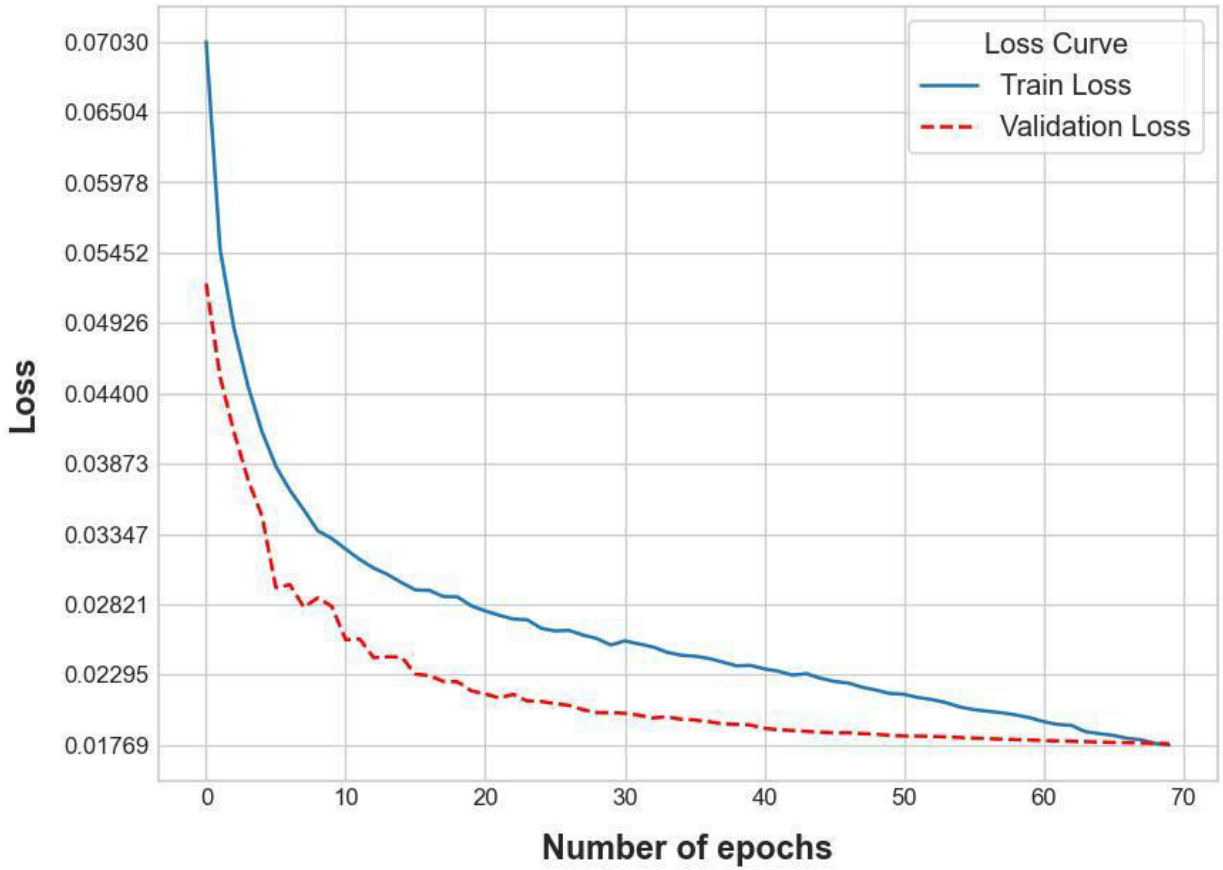
Methodology
(Hyperparameter Settings)

- MobileNetV2 SSDLite Configuration
  - Epochs: 80 (Early stopped at 69)
  - Batch size: 16
  - Learning rate: 1E-4 up to 60 epochs, then 1E-5
  - Regularization strength (L2): 0.002
  - IoU Threshold: 0.55
  - Confidence Threshold: 0.7
  - Momentum (Adam) beta1: 0.9
  - RMS prop (Adam) beta2: 0.999
- YOLOv5s Configuration
  - Epochs: 70
  - Batch size: 32
  - Initial learning rate: 1E-3
  - Learning rate step size: 1E-3
  - Final learning rate: 0.1
  - Optimizer weight decay: 5E-4
  - IoU Threshold: 0.2
  - Confidence Threshold: 0.45
  - Momentum (Adam) beta1: 0.937
  - RMS prop (Adam) Beta2: 0.999

Results
(Loss Curves)

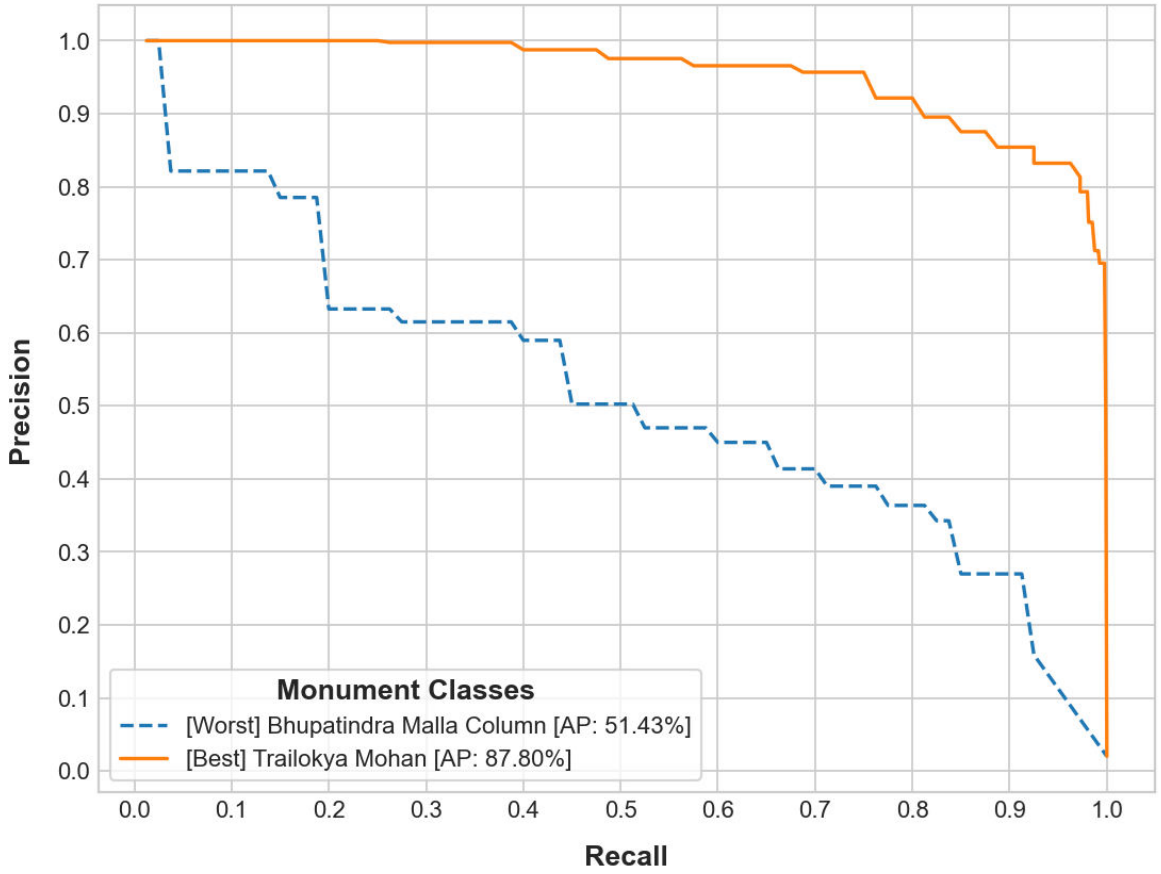


MobileNetV2 SSDLite

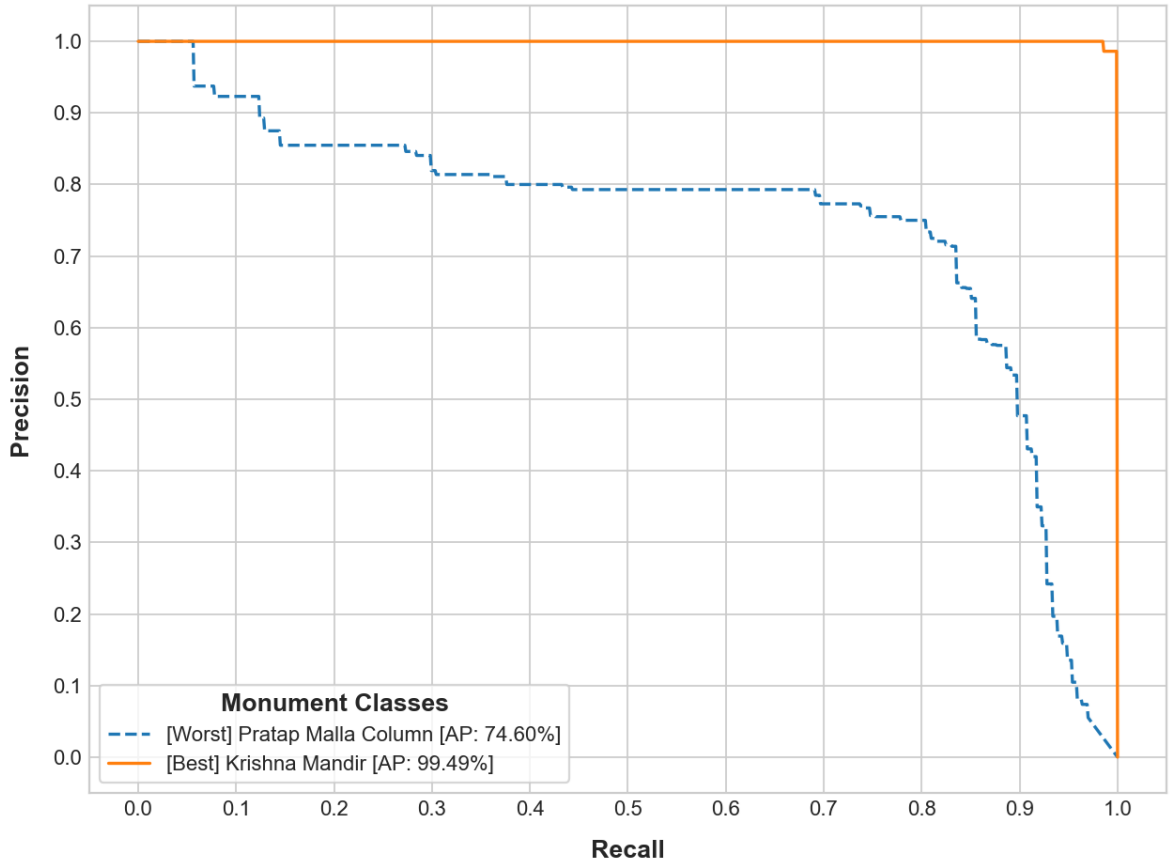


YOLOv5s

Results
(Precision vs. Recall Curves)

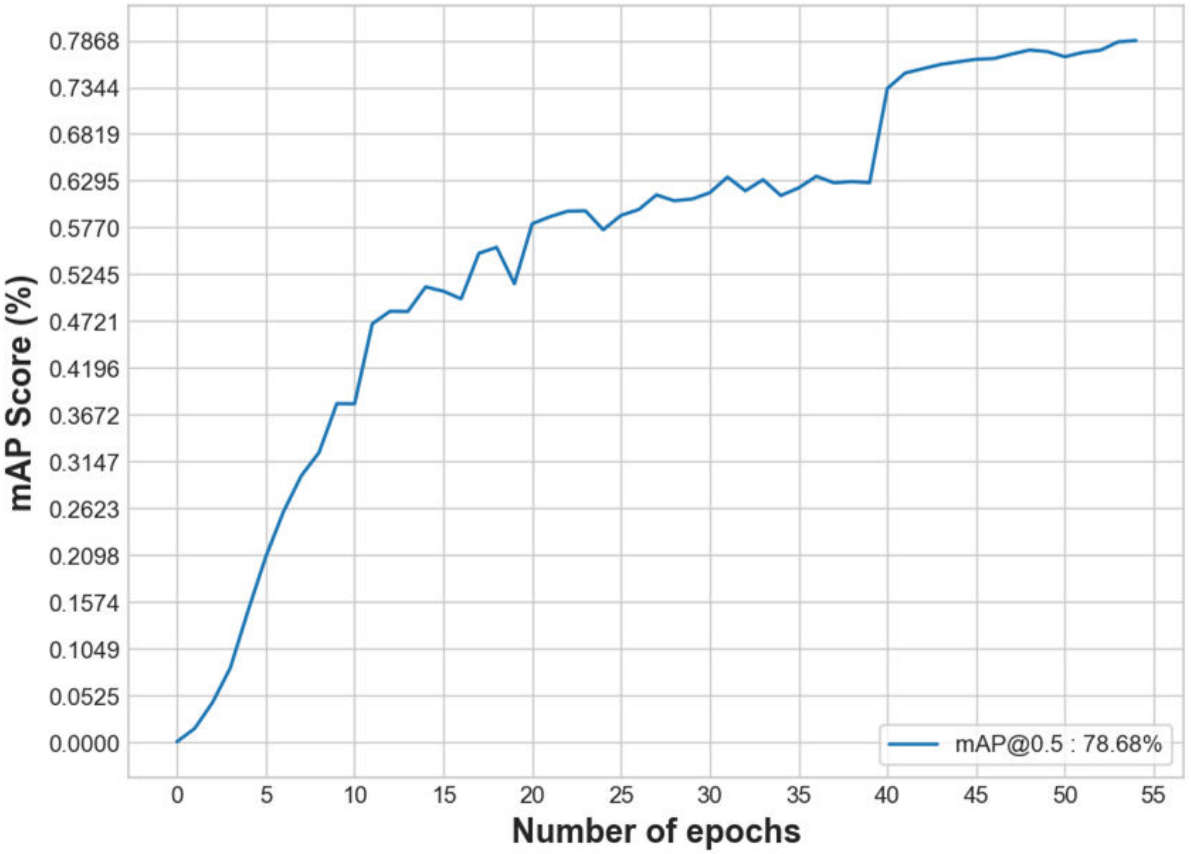


MobileNetV2 SSDLite

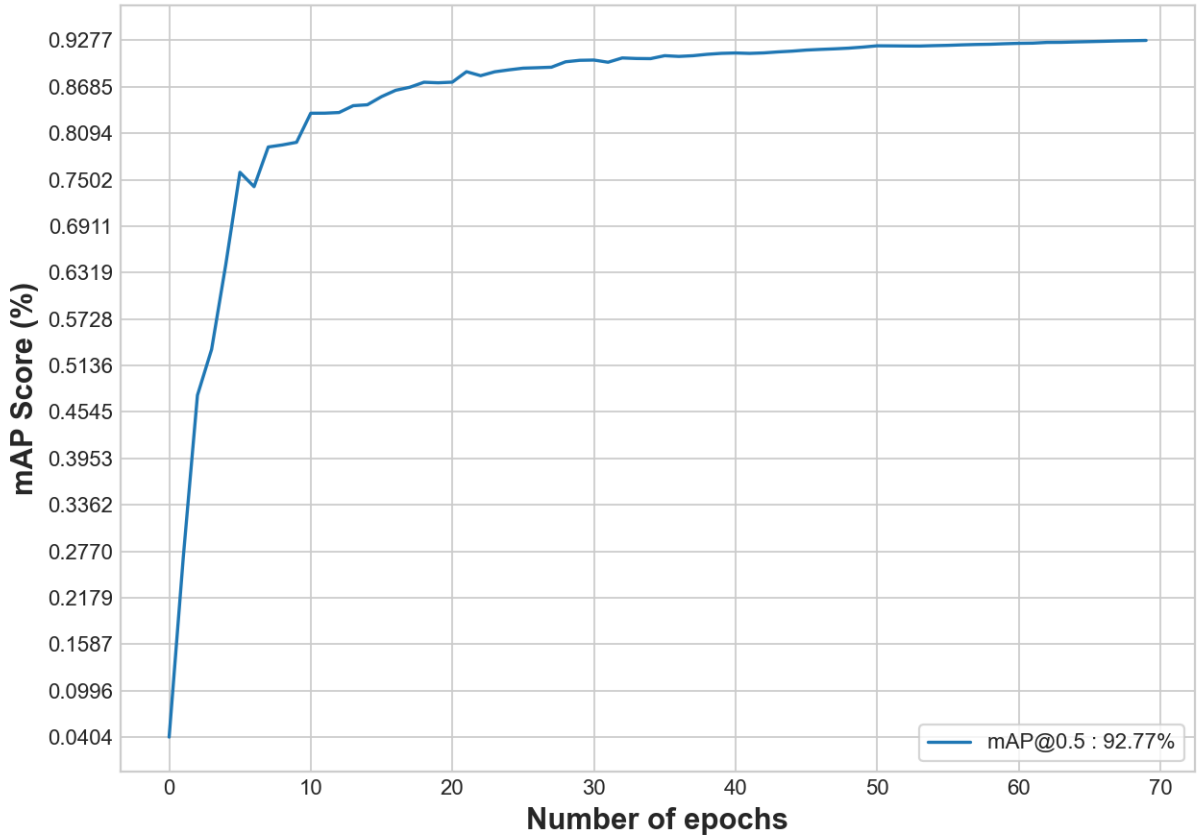


YOLOv5s

Results
(mAP Score Curves on Test Dataset)

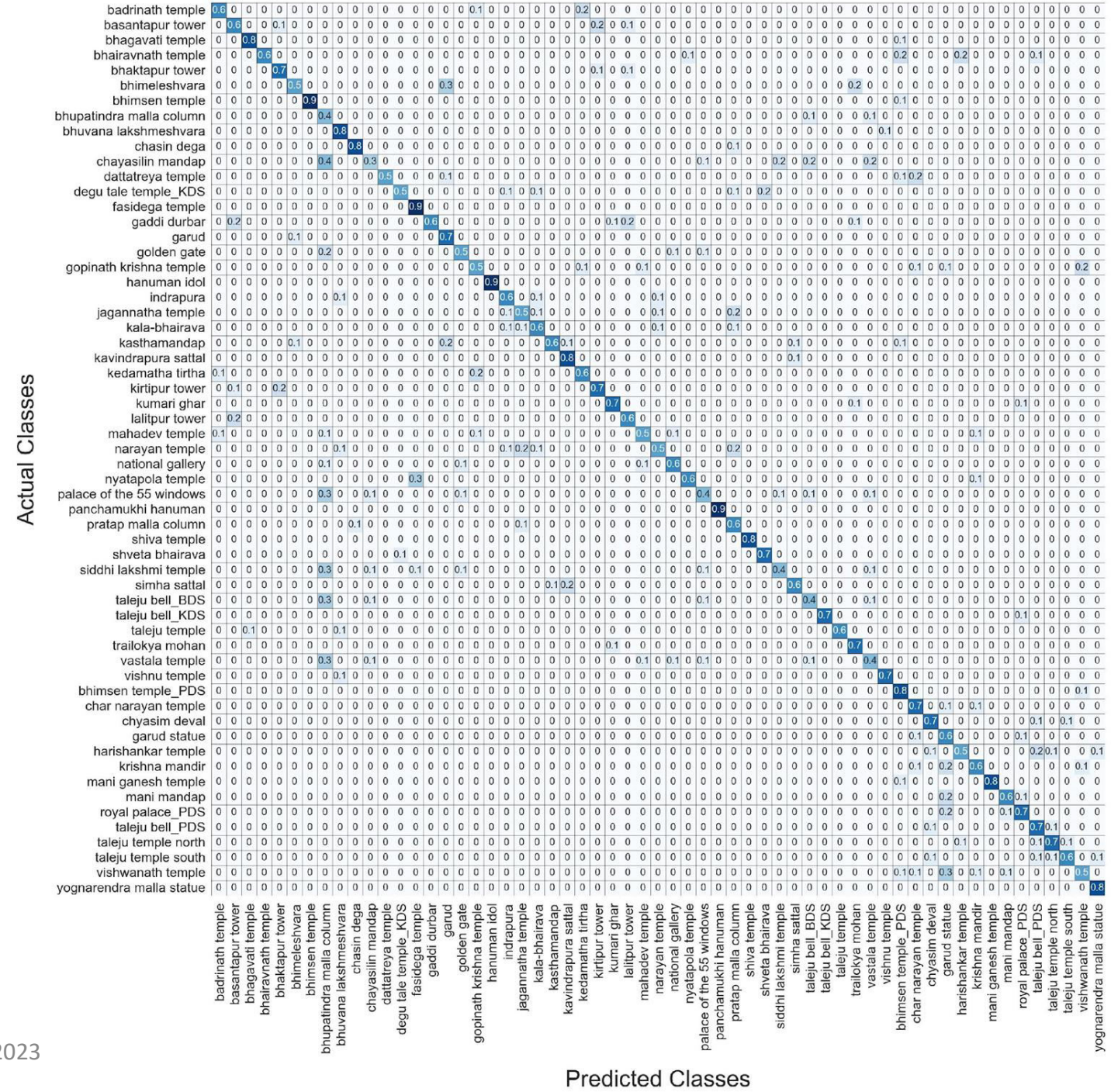


MobileNetV2 SSDLite



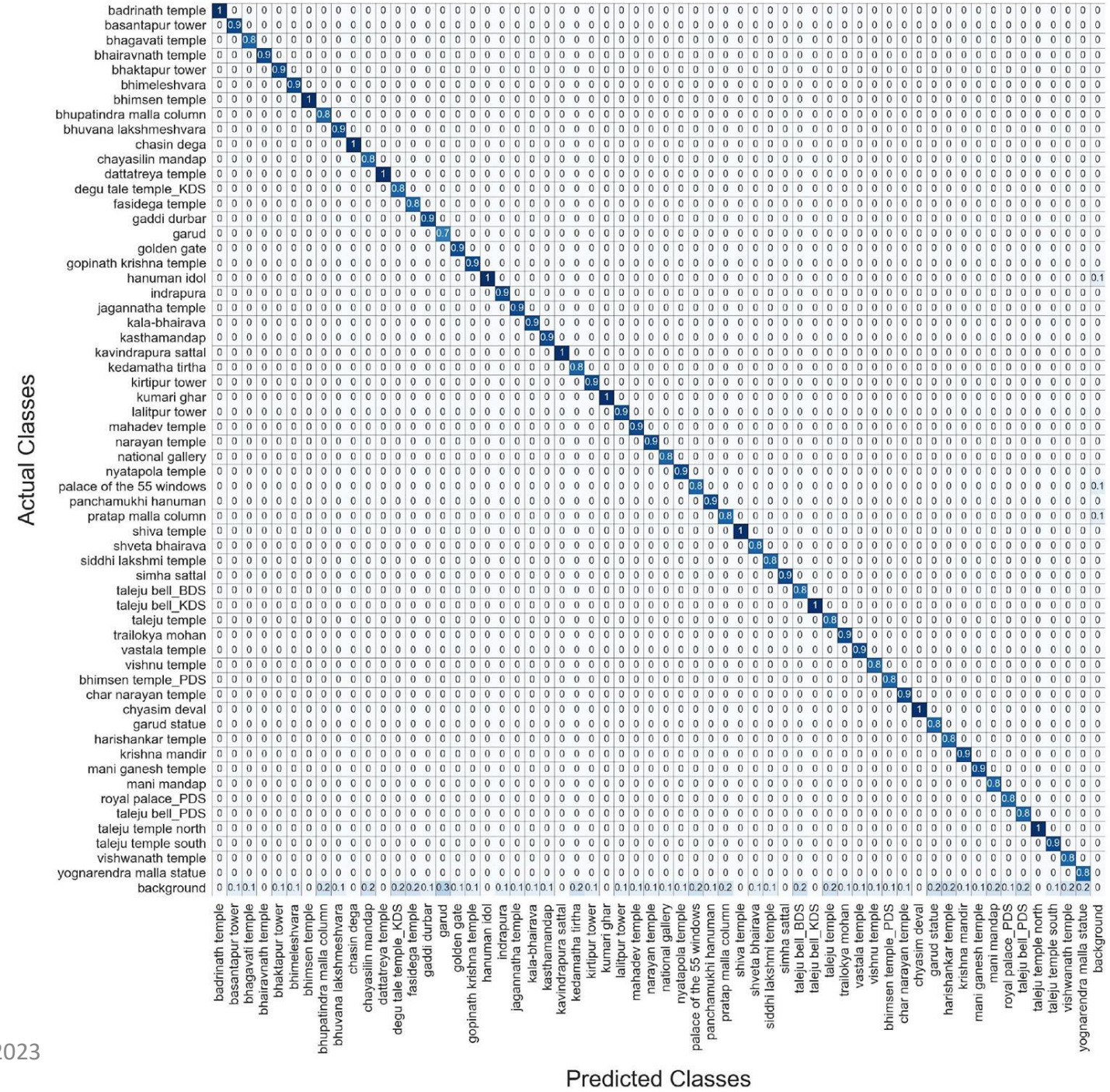
YOLOv5s

Results
(MobileNetV2 SSDLite Confusion Matrix)



- Best Performing Classes
  - Bhimsen Temple (Bhaktapur)
  - Fasidega Temple (Bhaktapur)
  - Hanuman Idol (Kathmandu)
  - Panchamukhi Hanuman (Kathmandu)
  - Possible reason
    - Typically isolated
- Worst Performing Classes
  - Chayasilin Mandap (Bhaktapur)
  - Bhupatindra Malla Column (Bhaktapur)
  - Possible reason
    - Occluded Structure

Results
(YOLOv5s Confusion Matrix)



- Best Performing Classes
  - Badrinath Temple (Bhaktapur)
  - Bhimsen Temple (Bhaktapur)
  - Chasin Dega (Kathmandu)
  - Hanuman Idol (Kathmandu)
  - Chyasim Deval (Patan)
  - Taleju Tower North (Patan)
  - Possible reason
    - Typically isolated
- Worst Performing Class
  - Garud (Kathmandu)
  - Possible reason
    - Dark shade and color of the monument

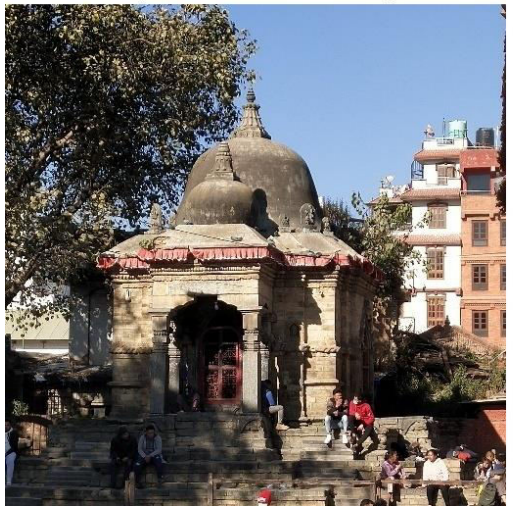
Results
(MobileNetV2 SSDLite Inference Cases)

Original Annotated Image

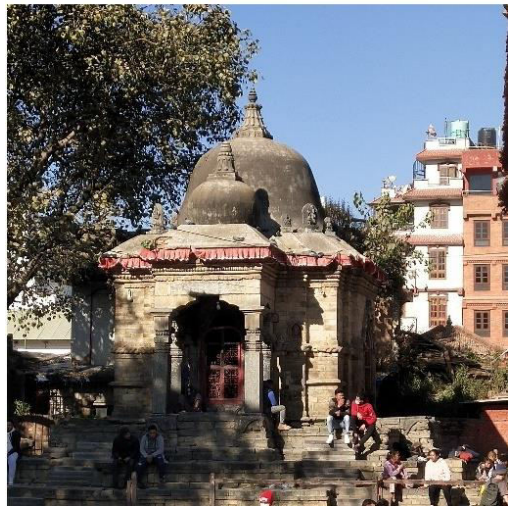
MobileNetV2 SSDLite Model Prediction

True Positive Example

Original Annotated Image



MobileNetV2 SSDLite Model Prediction



True Negative Example

Original Annotated Image

MobileNetV2 SSDLite Model Prediction

False Positive Example

Original Annotated Image

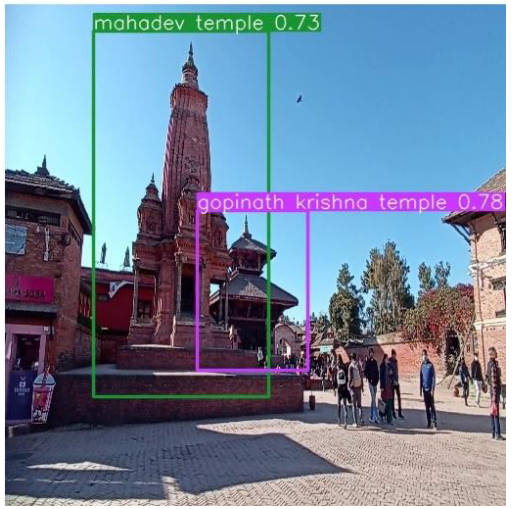
MobileNetV2 SSDLite Model Prediction

False Negative Example

Results
(YOLOv5s Inference Cases)

Original Annotated Image

YOLOv5s Model Prediction



True Positive Example

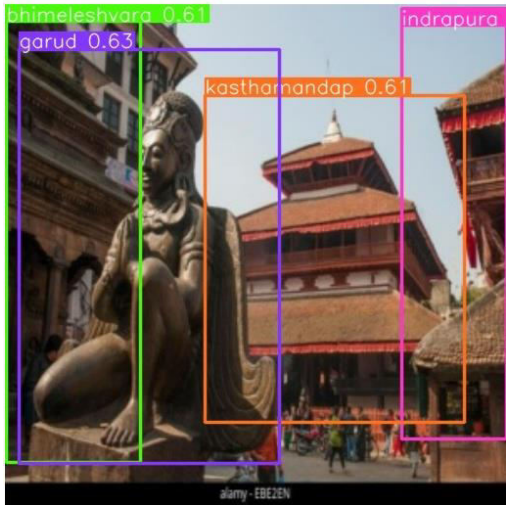
Original Annotated Image

YOLOv5s Model Prediction

True Negative Example

Original Annotated Image

YOLOv5s Model Prediction



False Positive Example

Original Annotated Image

YOLOv5s Model Prediction

False Negative Example

Discussion of Results
(MobileNetV2 SSDLite Model)

- High Confidence Score
  - Slight overfit in detection head for labels scores
  - Synthetically scaled dataset (repetitive angles)

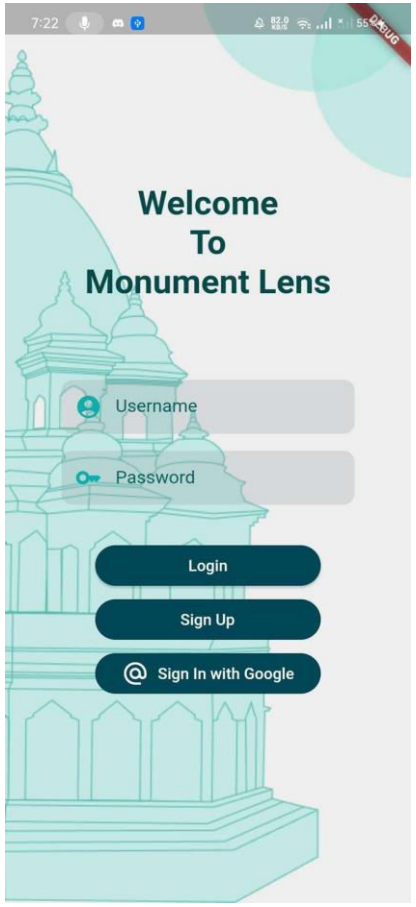
- False Positive Prediction
  - Inefficient handling of background classes by the architecture
  - Fewer instances of background images in the dataset (only 10%)

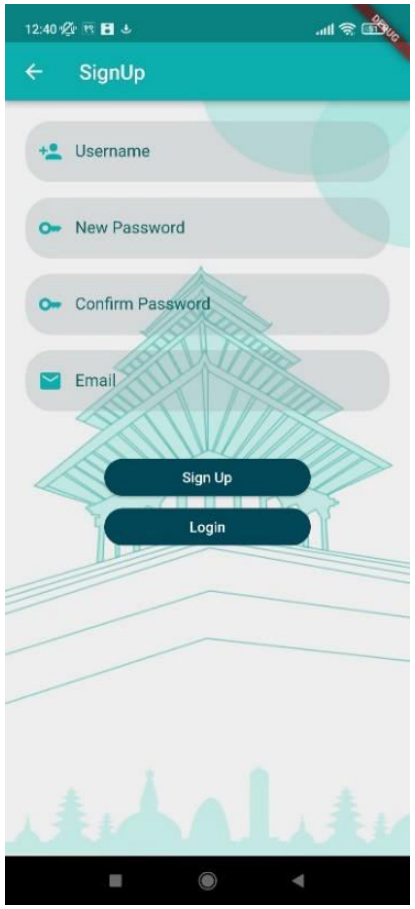
- Lack of Dynamic Anchor Boxes
  - Non-scalable fixed aspect ratio prior boxes

Discussion of Results
(YOLOv5s Model)

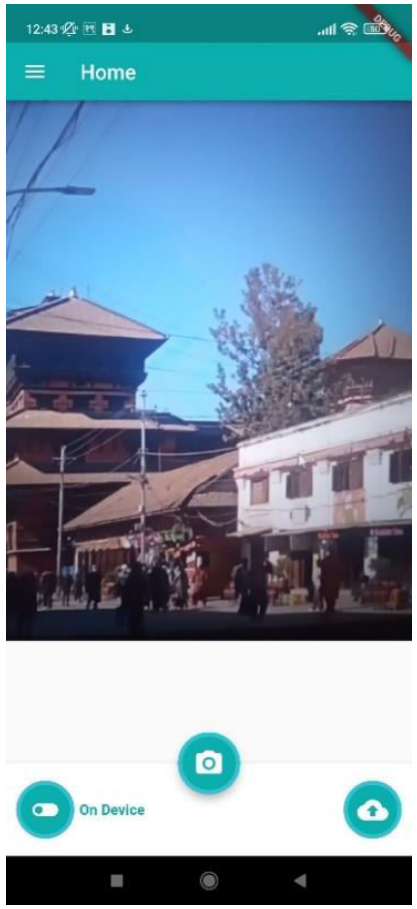
- Dynamic Auto-Anchor Boxes
  - Uses K-Means to analyze provided dataset and dynamically scales the aspect ratio of default prior boxes.
- Handles False Positive Cases Efficiently
  - Maintains objectness score to handle false positive instances
- Mosaic Data Augmentation
  - Combines four images into a single image to improve model performance

Mobile Application
(User Interface)





Login and Sign-Up View



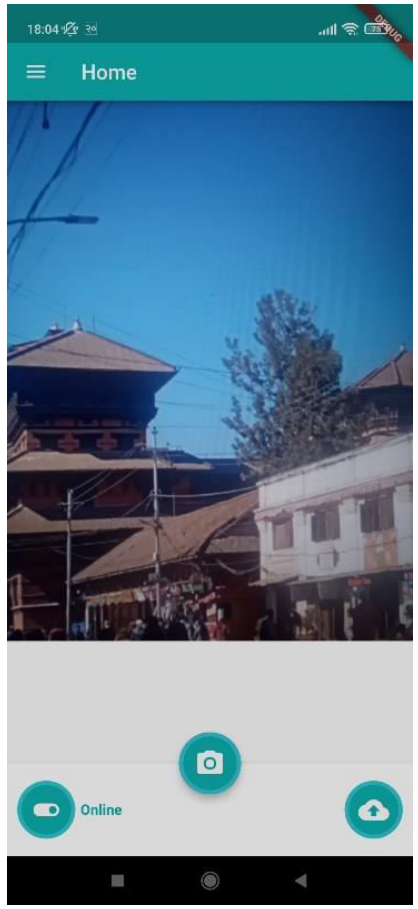
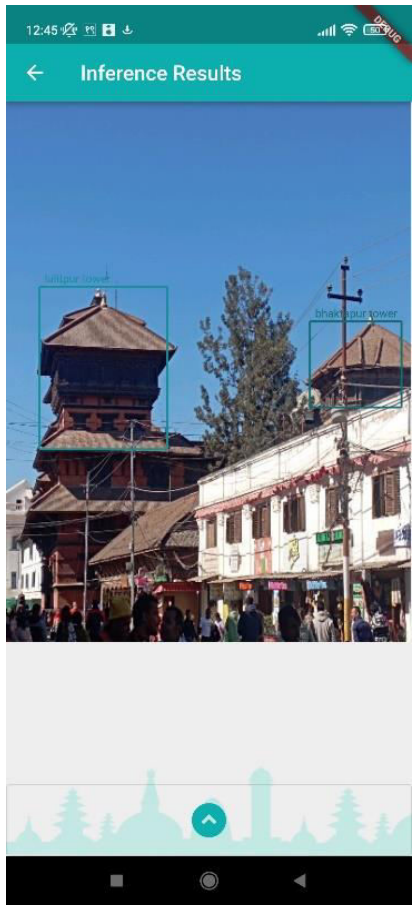
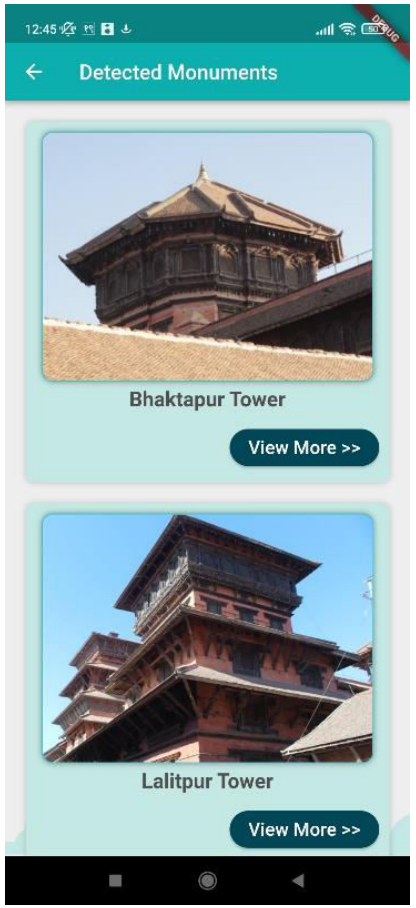


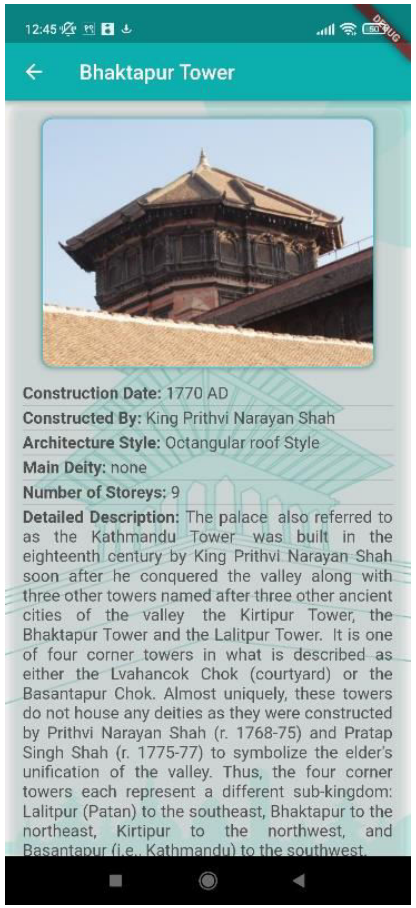
Image Clicking and Loading View

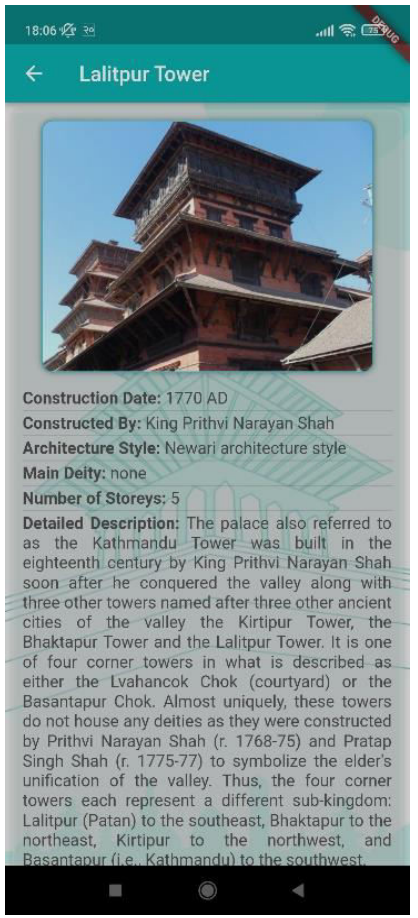
Mobile Application
(User Interface)





Detected Monuments View





Monument Details View

Future Enhancements

- Improving Dataset
  - Diversify dataset with different types of images
  - Explore variety of data augmentation techniques
- Validation and Hyperparameter tuning
  - Using K-folds cross validation
  - Using grid search for hyperparameter tuning
- Upgrading Mobile Application
  - Include interfacing support for both portrait and landscape orientation
  - Display real-time user and monument location in a map
  - Incorporate recommendation system

Conclusion

- Created a dataset of prominent monuments in three Durbar squares of Kathmandu Valley.
- Successfully trained MobileNetV2 SSD Lite and YOLOv5s models for monument detection.
- Achieved mAP@0.5 scores of 78.68% and 92.77% for MobileNetV2 SSD Lite and YOLOv5s, respectively, on the test dataset.
- Successfully developed a mobile application that incorporates both single-shot detectors to detects monuments.
- After detection, the mobile application displays information about the detected monument to the user.

References - [1]

- U. Kulkarni, S. M. Mena, S. V. Gurlahosur and U. Mudengudi, "Classification of Cultural Heritage Sites Using Transfer Learning," in 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM), Singapore, 2019.
- M. Sandhler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Slat Late City, UT, USA, 2018.
- V. Palma, "TOWARDS DEEP LEARNING FOR ARCHITECTURE: A MONUMENT RECOGNITION MOBILE APP," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* Vols. XLII-2/W9, pp. 551-556, 2019.
- W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *European Conference on Computer Vision*, Amsterdam, 2016.
- Y.-C. Chiu, C.-Y. Tsai, M.-D. Ruan, G.-Y. Shen and T.-T. Lee, "Mobilenet-SSDv2: An Improved Object Detection Model for Embedded Systems," in *2020 International Conference on System Science and Engineering (ICSSE)*, Kagawa, Japan, 2020.

References - [2]

- S. N, A. M. P and H. P. V, "Object Detection using YOLO And Mobilenet SSD: A Comparative Study," *International Journal of Engineering Research & Technology (IJERT),* vol. 11, no. 06, pp. 134-138, 2022.

- J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," CoRR, vol. abs/1804.02767, 2018.

- A. Bochkovskiy, C.-Y. Wang and M. H.-Y. Liao, "YOLOv4: Optimal Speed and Acuracy of Object Detection," CoRR, vol. abs/2004.10934, 2020.

- K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," CoRR, vol. abs/1406.4729, 2018.

Thank You