National University of Computer and Emerging Sciences

Campus Landmarks Detection Model Report

Group Number: 03

Phase 2 Members

Hafsah Zulqarnain	21L-5315
Fatima Pirzada	21L-5487

Group Members

Muhammad Waleed Malik	21L-5248
Tania Waseem	21L-5480
Hafsah Zulqarnain	21L-5315
Fatima Pirzada	21L-5487
Muhammad Sami Khokher	21L-1868
Muhammad Arsalan Kashif	21L-7630
Saad Khan	21L-1867
Muhammad Faizan Majid	21L-5229

FAST School of Computing

National University of Computer and Emerging Sciences

Lahore, Pakistan

1. Introduction

This report documents the design, training, and evaluation of a pre-trained model for recognizing landmarks in images of different buildings in FAST-NU Lahore. The model uses transfer learning with MobileNet to achieve high accuracy.

2. Model Architecture

Base Model

Architecture: MobileNet (pretrained on ImageNet).

Input Shape: (224, 224, 3)

Removed Layers: Our new model is derived from the original MobileNet, truncated at the fifth-to-last layer, thereby omitting the last four layers of our pretrained network. These layers were removed to allow the model to learn on our own FAST University buildings dataset. Through extensive experimentation, we obtained the best results by removing the last four layers.

```
x = mobile.layers[-5].output
```

Reshaping: To adapt the output of the Global Average Pooling layer for classification, we first reshape it into a 1D vector (shape: (1024,)).

```
x = tf.keras.layers.Reshape(target_shape=(1024,))(x)
```

Output Layer: This reshaped output is then passed to a Dense layer with 6 units and a softmax activation function, which generates probability distributions for the 6 target classes corresponding to 6 buildings in our dataset.

```
output = Dense(units=6, activation='softmax')(x)
```

Model Initialization: To assemble the new model, we initialize an instance of the Model class. We designate the input layer of the original MobileNet as the input for this model and assign the output layer (previously defined in the code) as its final output.

```
model = Model(inputs=mobile.input, outputs=output)
```

Layer Freezing and Fine-tuning: To retain the pretrained knowledge MobileNet acquired from ImageNet, we freeze the weights of all layers, excluding last 22 layers. The early layers capture general features like edges and textures. However, in the last 22 layers, the model is trained to adapt to nuances in our dataset. The decision to fine-tune our model using the last 22 layers has been determined after experimentation with many different values.

```
for layer in model.layers[:-22]:
    layer.trainable = False
```

3. Split Ratios:

- Training: 70% of the data.
- Validation: 20% of the data.
- Test: 10% of the data.

4. Training Configuration:

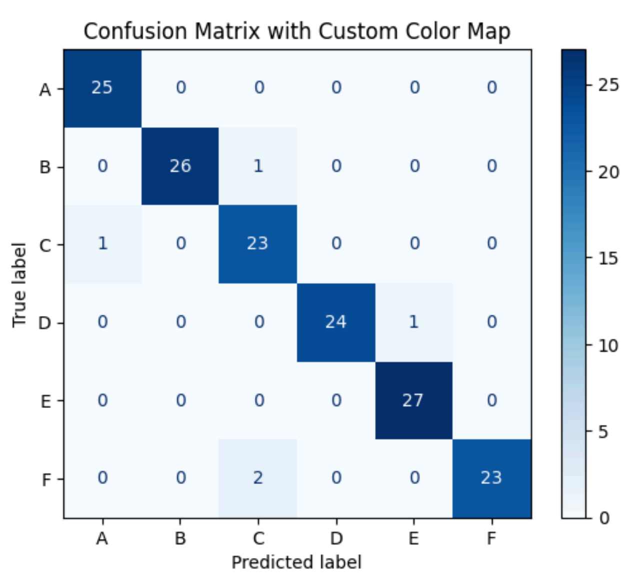
- Optimizer: Adam (learning_rate = 0.0001).
- Loss: Categorical cross-entropy.
- Epochs: 30
- We tried different values of batch_size like 8, 16 & 32. The model produced best results at batch_size=16.

5. Evaluation Results

Performance Metrics

Metric	Value
Test Accuracy	96.73%

Confusion Matrix



Classification Report

	precision	recall	f1-score	support
0	0.96	1.00	0.98	25
1	1.00	0.96	0.98	27
2	0.88	0.96	0.92	24
3	1.00	0.96	0.98	25
4	0.96	1.00	0.98	27
5	1.00	0.92	0.96	25
accuracy			0.97	153
macro avg	0.97	0.97	0.97	153
weighted avg	0.97	0.97	0.97	153

6. Dataset

The dataset used for the training of this model was taken from Phase-1 which consisted of labeled images of six prominent buildings at FAST-NU Lahore: A, B, C, D, E, and F. The number of images were quite small with 305 images so we used data augmentation techniques like random shifts, zooming and rotations using the ImageDataGenerator class from tensorflow library. Using this we made a final dataset which included approximately 250 images for each building.

To ensure high model performance and generalization, several preprocessing steps were applied to the dataset. First, lighting adjustments were performed using CLAHE (Contrast Limited Adaptive Histogram Equalization) from OpenCV, which normalized brightness and contrast to counter varying illumination conditions. Additionally, random blurring was applied to around 30% of the images to simulate real-world effects such as camera shake or motion blur. The model was tested both with and without these filters, and it performed better when they were included.

Finally, all the images were sent to a preprocessing function where they were resized to a uniform size of 224×224 pixels, matching the input requirements of the MobileNet architecture. The image dimensions were also expanded to meet the input format requirements, and pixel values were normalized to the range [-1, 1], as expected by the MobileNet architecture. These preprocessing steps helped ensure that the dataset was clean, balanced, and well-prepared for efficient and effective model training.