Smart Bus Tracker: Enhancing Public Transit with IoT-Integrated Application

Shivam Tiwari†, Shashank Patel†, Parita Jain†

[1] CSE, Delhi-NCR, Ghaziabad, UP, India.
[2] CSE, Delhi-NCR, Ghaziabad, UP, India.
[3] CSE, Delhi-NCR, Ghaziabad, UP, India.

Contributing authors: shivamtiwaritiwari0704@gmail.com; shashank.patel0908@gmail.com; parita.jain@kiet.edu;
†These authors contributed equally to this work.

Abstract

This paper presents the design and implementation of a mobile application integrating bus tracking, ticket booking, and seat availability checking. Leveraging Internet of Things (IoT) technologies such as GPS modules and sensors, the system dynamically tracks bus locations in real-time. Users access the application to view buses on a map, book tickets, and check seat availability. Additionally, the application includes a secure payment gateway and an SOS feature for emergencies. By combining IoT infrastructure with mobile technology, this system enhances public transportation services efficiently. Data is transmitted to a centralized server and made accessible to users through the mobile application.
Index Terms—Internet of Things, bus tracking, ticket booking, seat availability, mobile application, GPS, sensors, payment gateway, SOS feature.

1 Introduction

Public transportation is crucial for urban mobility, yet issues like congestion and inefficiency often plague bus services, leading to passenger dissatisfaction and long wait times. Timely information regarding seat occupancy and bus locations can significantly enhance the passenger experience and enable service providers to optimize operations. However, traditional methods of collecting passenger data, such as manual counting or expensive automated systems, fall short in terms of efficiency and cost-effectiveness.

In response to these challenges, this research proposes a novel approach to bus monitoring and passenger flow prediction, leveraging modern technologies. By utilizing APIs for tracking, a user-friendly interface developed with React Native, and machine learning algorithms for seat occupancy checks, the system aims to offer real-time information on bus location and seat availability. Additionally, the integration of an SOS feature activated by shaking the phone ensures passenger safety during emergencies. Inspired by existing automated fare collection and GPS tracking systems, the proposed solution seeks to overcome limitations such as high costs and the impracticality of manual counting.

By harnessing data-driven decision-making and real-time monitoring, the system aims to enhance the efficiency and reliability of bus services, ultimately improving the overall passenger experience.

2 Literature Review

In their study, Murdan et al. [1] designed and tested an IoT-based Bus Seating Information Technology system to address challenges in public transport reliability. Using sensors like infrared and force-sensitive resistors, along with GPS tracking, the system dynamically monitored vehicle occupancy in real-time. Results showed promising capabilities in tracking both bus location and occupancy status, offering potential solutions to the lack of comprehensive real-time data in public transportation.

In their endeavor, Patel et al. [2] address the challenges stemming from road conditions and operational uncertainties in bus systems. They propose a Real-Time Bus Tracking System (RTBTS) to alleviate passenger inconvenience by furnishing real-time updates on bus locations and arrival estimates. Employing GPS-module devices installed on college buses, the RTBTS transmits location data to a computer, which is then stored in a text file. This data is subsequently retrieved and stored on a web server for user access, offering real-time insights into bus locations. The RTBTS stands as an independent solution tailored to enhance the efficiency of bus services provided by the college.

In their study, Sharma and Sawant [3] delve into the realm of Internet of Things (IoT) applications, focusing on bus seat management. They present a solution combining hardware and web applications to address the challenge of verifying bus seat occupancy. The proposed system enables passengers to check seat availability either through their connected devices or via display systems at bus entrances. Utilizing pressure sensors and Arduino Uno modules for seat monitoring, the prototype system transmits real-time seat utilization data to a web application interface. Through student evaluations, the research demonstrates the feasibility and significance of the developed system, achieving a remarkable 90% accuracy rate for the physical prototype and a 90% real-time reporting accuracy for seat utilization via the web application.

3 Algorithms, Program codes and Listings

3.1 MOBILENET-SSD ARCHITECTURE

MobileNet represents a convolutional neural network (CNN) architecture tailored specifically for image classification tasks in mobile applications. Unlike traditional CNN architectures, MobileNet utilizes Depthwise Separable Convolution layers, which significantly reduce computational complexity. This distinctive architecture distinguishes itself by requiring minimal computational power, making it highly suitable for deployment in resource-constrained environments. Below, we provide code snippets demonstrating the implementation of Depthwise Separable Convolution and the loading of MobileNet in TensorFlow for image classification tasks. These examples serve to illustrate the practical application of the MobileNet architecture in real-world scenarios.

Single Shot Multibox Detector (SSD) is a groundbreaking algorithm capable of

```python
import tensorflow as tf

# Define Depthwise Separable Convolution layer
def depthwise_separable_conv(inputs, filters, kernel_size, strides=1):
    # Depthwise Convolution
    x = tf.keras.layers.DepthwiseConv2D(kernel_size, strides=strides, padding='same', use_bias=False)(inputs)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.ReLU()(x)

    # Pointwise Convolution
    x = tf.keras.layers.Conv2D(filters, kernel_size=1, strides=1, padding='same', use_bias=False)(x)
    x = tf.keras.layers.BatchNormalization()(x)
    x = tf.keras.layers.ReLU()(x)

    return x

# Example of loading MobileNet model for image classification
def load_mobilenet_model(input_shape, num_classes):
    base_model = tf.keras.applications.MobileNet(input_shape=input_shape, include_top=False, weights='imagenet')
    for layer in base_model.layers:
        layer.trainable = False

    x = tf.keras.layers.GlobalAveragePooling2D()(base_model.output)
    x = tf.keras.layers.Dense(num_classes, activation='softmax')(x)

    model = tf.keras.models.Model(inputs=base_model.input, outputs=x)
    return model

# Example usage
input_shape = (224, 224, 3)
num_classes = 1000
model = load_mobilenet_model(input_shape, num_classes)
model.summary()
```

Fig. 1: Depthwise Separable Convolution and MobileNet model loading

detecting multiple objects in an image with just one shot, using a multibox approach. Utilizing a single deep neural network, SSD operates across various scales to identify objects of different sizes within the image.

The SSD algorithm, described earlier, provides efficient and accurate object detection capabilities, making it widely used in

computer vision applications. These code examples illustrate how to construct an SSD model and perform object detection on images using TensorFlow.

In real-time applications on low-cost devices, deploying and running complex and power-intensive neural networks poses practical limitations. To address this challenge, the Single Shot Multibox Detector (SSD) leverages MobileNet as its base network, resulting in what is known as MobileNet SSD.

When integrating MobileNet V1 with SSD, the final layers, including FC, Maxpool, and Softmax, are excluded. Instead, the outputs from the last convolutional layer in MobileNet are utilized, further undergoing convolution operations to generate a stack of feature maps. These feature maps serve as inputs for the detection heads of the SSD architecture. The modular nature of MobileNet SSD allows for flexible

```python
import tensorflow as tf
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Conv2D

def ssd_model(input_shape, num_classes):
    base_model = VGG16(weights='imagenet', include_top=False, input_shape=input_shape)

    # Omitting last few layers of VGG
    base_output = base_model.layers[-5].output

    # Additional convolution layers for feature maps of varying sizes
    conv6 = Conv2D(1024, kernel_size=(3, 3), padding='same', activation='relu')(base_output)
    conv7 = Conv2D(1024, kernel_size=(1, 1), padding='same', activation='relu')(conv6)

    # Construct SSD model
    predictions = Conv2D(num_classes * 4, kernel_size=(3, 3), padding='same', activation='sigmoid')(conv7)
    predictions = tf.keras.layers.Reshape((-1, num_classes, 4))(predictions)

    model = tf.keras.models.Model(inputs=base_model.input, outputs=predictions)
    return model

# Example usage
input_shape = (300, 300, 3)
num_classes = 80
model = ssd_model(input_shape, num_classes)
model.summary()
```

Fig. 2: Construction of a Single Shot Multibox Detector (SSD) model for object detection tasks

modifications to adapt its architecture to specific requirements and constraints.

The architectural diagram illustrates the integration of MobileNet with the Single Shot Multibox Detector (SSD) framework, forming MobileNet SSD. This diagram visually depicts the workflow and components of the MobileNet SSD

architecture, showcasing the utilization of MobileNet as the base network for feature extraction and its seamless integration with the SSD framework for efficient object detection.

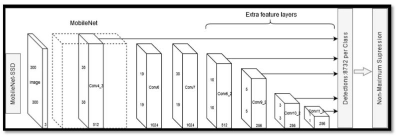

Fig. 3: Architecture of SSD MobileNet model.

4 Methodology

Our proposed system represents a significant advancement in the realm of local bus booking applications, aimed at revolutionizing the passenger experience through the integration of cutting-edge technologies and innovative features. By leveraging state-of-the-art solutions such as MobileNet SSD architecture, GPS tracking, and real-time security measures, our system strives to address key challenges faced by passengers while enhancing convenience, safety, and efficiency.

In the following subsections, we delve into the core components and features of our proposed system, outlining how each element contributes to an enhanced local bus booking experience.

4.1 Real-Time Seat Vacancy Details

Leveraging the MobileNet SSD architecture, our system provides passengers with real-time seat vacancy details. By analyzing bus interior images using machine learning algorithms, users can instantly view available seats, enhancing convenience and informed decision-making.

4.2 Cross-Platform Compatibility and User Interface

Built using React Native for cross-platform compatibility, our app offers a user-friendly interface and seamless performance across various devices. Integration with Google Maps further enhances the user experience by providing accurate location data and navigation assistance.

The system architecture presents an overview of our local bus booking application. It integrates modern technologies to offer passengers an efficient and user-friendly experience. This diagram illustrates the main components and interactions within the system, highlighting its key functionalities.

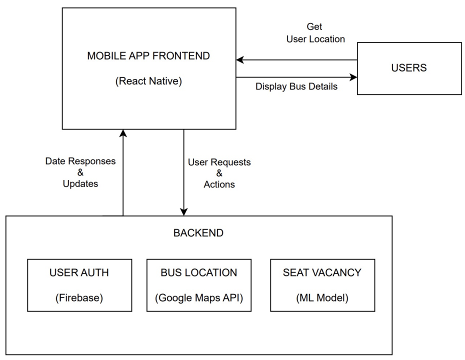

Fig. 4: Proposed System Architecture Diagram

5 Conclusion

In conclusion, the development and implementation of our local bus booking application represent a significant advancement in public transportation technology. By leveraging modern technologies such as React Native for frontend development, machine learning for real-time seat vacancy detection, and GPS integration for bus tracking, we have created a robust and user-friendly platform for passengers. The integration of features such as emergency response mechanisms further enhances the safety and convenience of our application. Through this research, we have demonstrated the effectiveness of utilizing MobileNet SSD architecture for real-time seat vacancy detection, providing valuable insights into improving public transportation services. Moving forward, we envision continued enhancements and optimizations to our application, driven by feedback from users and advancements in technology.

References

[1] A. P. Murdan, V. Bucktowar, V. Oree, and M. P. Enoch, ”Low-cost bus seating information technology system,” IET Intelligent Transport Systems, doi: 10.1049/iet-its.2019.0529.

[2] D. Patel, R. Seth, V. Mishra, and R. Pathari, ”Real-Time Bus Tracking System,” International Research Journal of Engineering and Technology (IRJET), vol. 04, no. 03, Mar. 2017.

[3] R. Sharma and Y. Sawant, ”SMART BUS: Seat Occupancy Detection System,” International Research Journal of Modernization in Engineering Technology and Science, vol. 03, no. 05, May 2021.

[4] J. Zhang, D. Shen, L. Tu, F. Zhang, C. Xu, Y. Wang, C. Tian, X. Li, B. Huang, and Z. Li, ”A Real-Time Passenger Flow Estimation and Prediction Method for Urban Bus Transit Systems,” IEEE Transactions on Intelligent Transportation Systems.

[5] K. Sujatha, P. V. Nageswara Rao, K. J. Sruthi, and A. Arjuna Rao, ”Design and Development of Android Mobile-Based Bus Tracking System,” in Proceedings of the 2014 First International Conference on Networks & Soft Computing (ICNSC2014), IEEE.

[6] O. Obulesu, A. Hanshika, A. Lahari, P. Arunodaya, and R. Ashwi, ”People Counting and Tracking System in Real-Time Using Deep Learning Techniques,” International Journal for Research in Applied Science and Engineering Technology, vol. 02, no. 04, Jun. 30, 2023.

[7] A. Bakliwal, A. Puranik, A. Modi, A. Jain, A. Jaiswal, D. Godani, P. Bhanodia, and P. Jangde, ”Crowd Counter: An Application of Centroid Tracking Algorithm,” International Research Journal of Modernization in Engineering Technology and Science, vol. 02, no. 04, Apr. 2020.

[8] Book- Puneet Kumar Aggarwal, Parita Jain, Jaya Mehta, Riya Garg, Kshirja Makar, and Poorvi Chaudhary, “Machine Learning, Data Mining and Big Data Analytics for 5G-Enabled IoT”, Blockchain for 5G enabled IoT: the new wave for Industrial Automation, pp. 351-375, Springer, 2021.

[9] N. Gautam, A. Singh, K. Kumar, P K Aggarwal, Anupam, “Investigation on performance analysis of support vector machine for classification of abnormal regions in medical image”, Journal of Ambient Intelligence and Humanized Computing, 2021.