iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** I **Month of publication:** January 2025

DOI: https://doi.org/10.22214/ijraset.2025.66215

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue I Jan 2025- Available at www.ijraset.com

Car Parking System

J Vidya Mounika[1], L Harshitha[2], K Shreeja[3], D Dharani Shreya[4], Mrs. Rachel Evangeline[5]
BITM

Abstract: Car Parking Monitoring System Using VGG16 project is a Car Parking Monitoring System that uses computer vision and deep learning to detect whether parking spaces are occupied or free. The system works by analyzing video footage of a parking lot, using a pre-trained VGG16 model to identify if there are cars in specific parking spots. This system is designed to help parking lot owners or businesses keep track of parking space usage from video recordings. It offers an easy way to monitor parking lot occupancy automatically.

I. INTRODUCTION

Efficient parking space management is a critical aspect of urban infrastructure, especially with the exponential growth of vehicles in cities worldwide. The increasing demand for parking, coupled with the challenges of manual monitoring, has created the need for automated solutions that can provide accurate and real-time updates on parking space availability. Traditional systems often rely on human intervention, which is prone to errors, inefficiency, and higher operational costs.

This project addresses these challenges by developing an Automated Car Parking Monitoring System that utilizes the power of deep learning and computer vision. At the core of this system is a VGG-based convolutional neural network (CNN), a proven architecture for image classification tasks, adapted here to classify parking spaces as either “Free” or “Occupied.” The system leverages pre-annotated Regions of Interest (ROIs) within parking lot video footage to identify and analyze individual parking spaces with high precision.

By combining robust machine learning techniques, efficient data preprocessing, and a practical deployment framework, this system offers a scalable and accurate solution for modern parking management in public and private settings.

II. LITERATURE REVIEW

A. Real-time image-based Parking Occupancy Detection using Deep Learning

The paper "Real-time Image-based Parking Occupancy Detection Using Deep Learning" introduces an innovative approach for automated parking management by utilizing deep learning and image processing techniques to detect parking space occupancy in real-time. The authors leverage Convolutional Neural Networks (CNNs) to analyze images captured from parking lot cameras, automatically classifying parking spaces as either free or occupied.

This approach significantly improves parking space monitoring efficiency, reducing human intervention and minimizing errors in detecting space availability. The system’s real-time capability makes it highly applicable to urban settings where timely updates on parking availability are crucial.

The use of deep learning ensures high accuracy and scalability, making the solution effective for both public and private parking management. Overall, this research provides a practical and modern solution to address the growing challenges of parking space management in densely populated areas.

B. Tiny Vehicle Detection Based on YOLO and Multi-Scale Convolutional Neural Networks

The paper titled "Smart Parking System with IoT and Deep Learning" explores an innovative smart parking system that integrates Internet of Things (IoT) technology with deep learning to efficiently manage parking spaces. The system uses real-time sensor data from IoT devices combined with deep learning algorithms to detect parking space occupancy and guide drivers to available spots. By utilizing deep learning for image and video analysis, the system can accurately detect whether a parking space is occupied or vacant, offering a more efficient and automated approach than traditional parking management methods. The integration of IoT enhances the system's scalability and enables remote monitoring, making it suitable for large urban parking areas. This project offers a promising solution to the growing demand for parking in crowded cities, improving both parking efficiency and the user experience.

©IJRASET: All Rights are Reserved

C. Method For Judging Parking Status Based On Yolov2 Target Detection Algorithm

The paper titled "Smart Parking System with IoT and Deep Learning" explores an innovative smart parking system that integrates Internet of Things (IoT) technology with deep learning to efficiently manage parking spaces. The system uses real-time sensor data from IoT devices combined with deep learning algorithms to detect parking space occupancy and guide drivers to available spots. By utilizing deep learning for image and video analysis, the system can accurately detect whether a parking space is occupied or vacant, offering a more efficient and automated approach than traditional parking management methods. The integration of IoT enhances the system's scalability and enables remote monitoring, making it suitable for large urban parking areas. This project offers a promising solution to the growing demand for parking in crowded cities, improving both parking efficiency and the user experience.

III. PROBLEM DEFINITION

Design a Car Parking Monitoring System that processes a prerecorded video file of a parking lot to determine the status of each parking space. The system should use a pre-trained deep learning model (VGG16) to identify whether a parking space is occupied or vacant based on vehicle detection. The processed results (annotated video frames and a parking status count) will be displayed on a web interface for easy monitoring of parking availability.

IV. METHODOLOGY

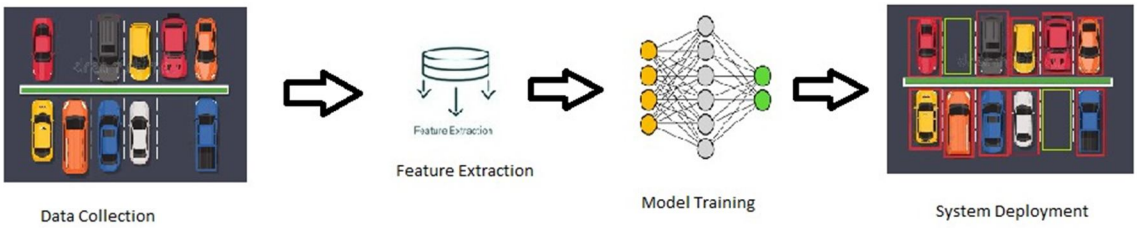


Fig1: Dataflow diagram

A. Data Collection and Preprocessing

Video footage of a parking area was used as the input dataset. The preprocessing involved resizing video frames while maintaining aspect ratios to ensure compatibility with the model. Noise in the images was minimized using padding techniques, ensuring accurate detection of parked and non-parked spaces.

B. Feature Extraction

Regions of Interest (ROIs) were defined for each parking space using bounding boxes stored in a pre-annotated file (carposition.pkl). These regions were cropped from the frames and processed to extract features such as:

1) Resizing the cropped images to a standard size (48x48 pixels).
2) Normalizing pixel values between 0 and 1 to enhance model compatibility.

C. Model Training

The VGG-based convolutional neural network (CNN) was trained on the processed dataset for binary classification:

1) Classes:
2) Parked: The parking space is empty
3) Not Parked: The parking space is occupied.
4) Training Process: The model utilized supervised learning with a binary cross-entropy loss function and an Adam optimizer. The dataset was split into training and testing subsets for performance evaluation.

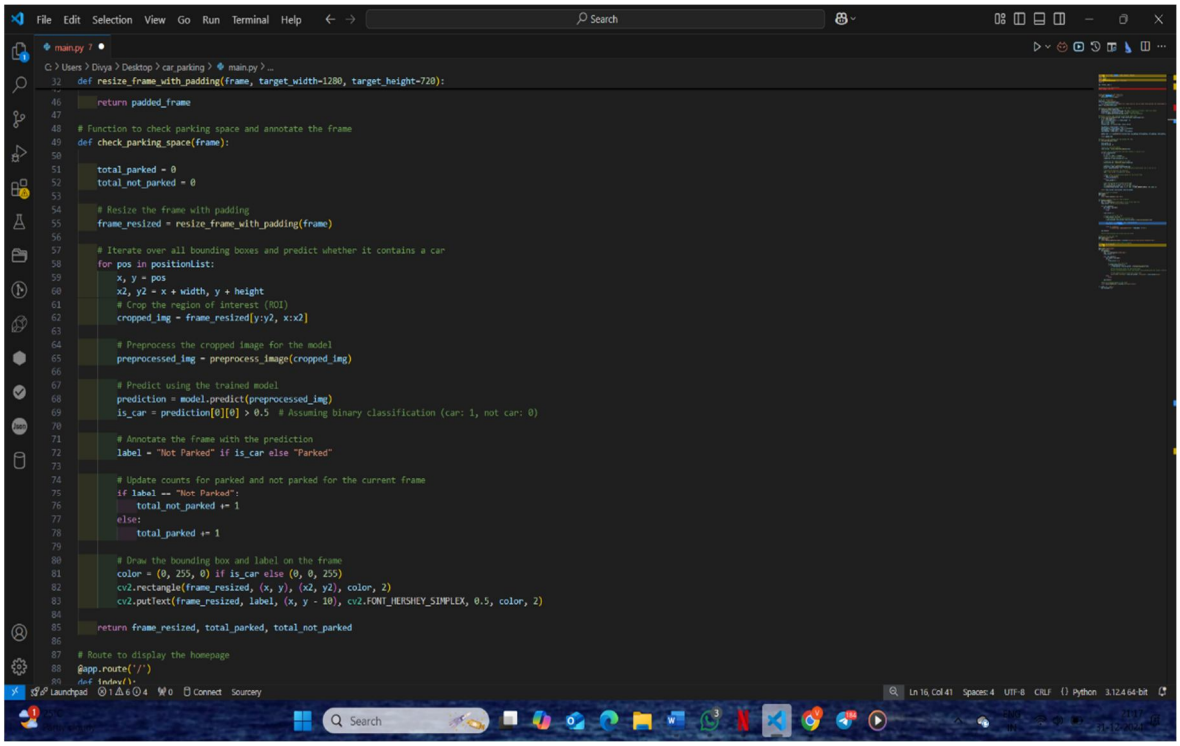
D. System Deployment

The system was implemented as a Flask web application with the following features:

1) Parking Status Detection: Each parking space is classified as “Free” or “Occupied,” and labels are shown directly on the video.
2) Space Count Updates: The system provides live updates about the number of free and occupied parking spaces.

V. RESULTS AND EVALUATION

The Car Parking System using the VGG16 model was implemented to classify parking spaces as either occupied or vacant based on the presence of vehicles in a given parking lot. The system processes frames from a video and annotates them to identify parking space status. Although the system does not provide real-time detection, it demonstrates effective detection from pre-recorded video.



```python
def resize_frame_with_padding(frame, target_width=1280, target_height=720):
    return padded_frame

# Function to check parking space and annotate the frame
def check_parking_space(frame):

    total_parked = 0
    total_not_parked = 0

    # Resize the frame with padding
    frame_resized = resize_frame_with_padding(frame)

    # Iterate over all bounding boxes and predict whether it contains a car
    for pos in positionList:
        x, y = pos
        x2, y2 = x + width, y + height
        # Crop the region of interest (ROI)
        cropped_img = frame_resized[y:y2, x:x2]

        # Preprocess the cropped image for the model
        preprocessed_img = preprocess_image(cropped_img)

        # Predict using the trained model
        prediction = model.predict(preprocessed_img)
        is_car = prediction[0][0] > 0.5  # Assuming binary classification (car: 1, not car: 0)

        # Annotate the frame with the prediction
        label = "Not Parked" if is_car else "Parked"

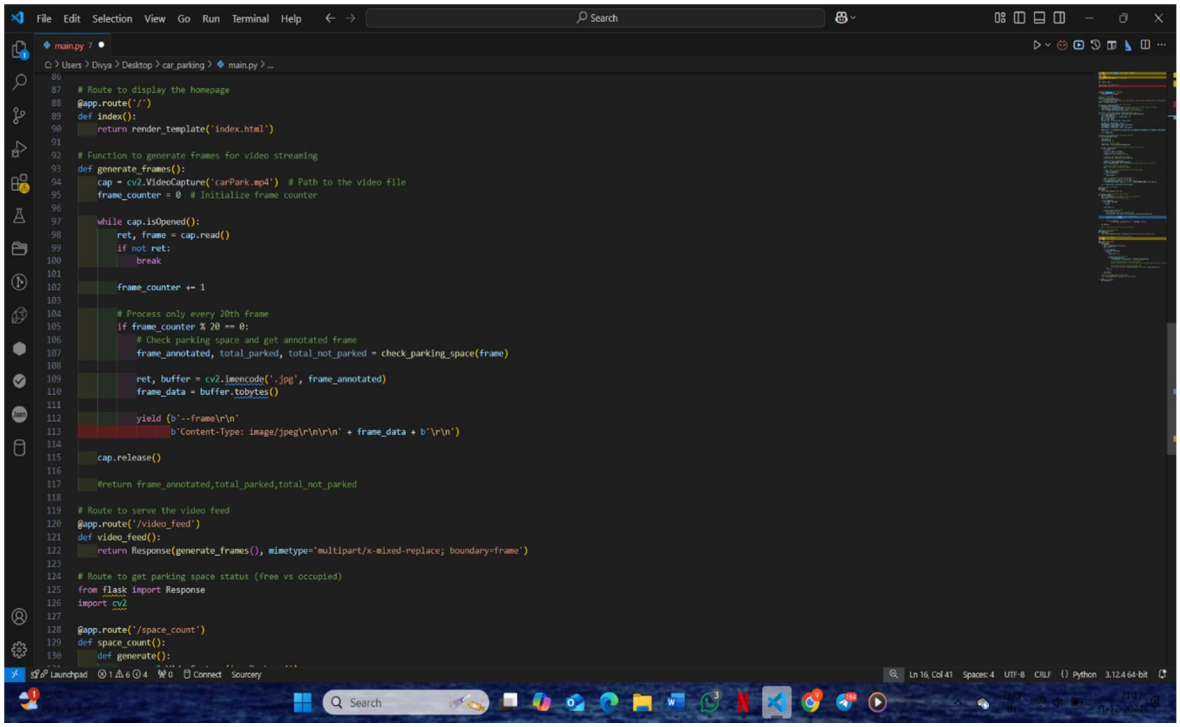
        # Update counts for parked and not parked for the current frame
        if label == "Not Parked":
            total_not_parked += 1
        else:
            total_parked += 1

        # Draw the bounding box and label on the frame
        color = (0, 255, 0) if is_car else (0, 0, 255)
        cv2.rectangle(frame_resized, (x, y), (x2, y2), color, 2)
        cv2.putText(frame_resized, label, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

    return frame_resized, total_parked, total_not_parked

# Route to display the homepage
@app.route('/')
```

Fig2: Function to check parking space



```python
# Route to display the homepage
@app.route('/')
def index():
    return render_template('index.html')

# Function to generate frames for video streaming
def generate_frames():
    cap = cv2.VideoCapture('carPark.mp4')  # Path to the video file
    frame_counter = 0  # Initialize frame counter

    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break

        frame_counter += 1

        # Process only every 20th frame
        if frame_counter % 20 == 0:
            # Check parking space and get annotated frame
            frame_annotated, total_parked, total_not_parked = check_parking_space(frame)

            ret, buffer = cv2.imencode('.jpg', frame_annotated)
            frame_data = buffer.tobytes()

            yield (b'--frame\r\n'
                   b'Content-Type: image/jpeg\r\n\r\n' + frame_data + b'\r\n')

    cap.release()

    #return frame_annotated,total_parked,total_not_parked

# Route to serve the video feed
@app.route('/video_feed')
def video_feed():
    return Response(generate_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')

# Route to get parking space status (free vs occupied)
from flask import Response
import cv2

@app.route('/space_count')
def space_count():
    def generate():
```

Fig 3: Generation of frames for video

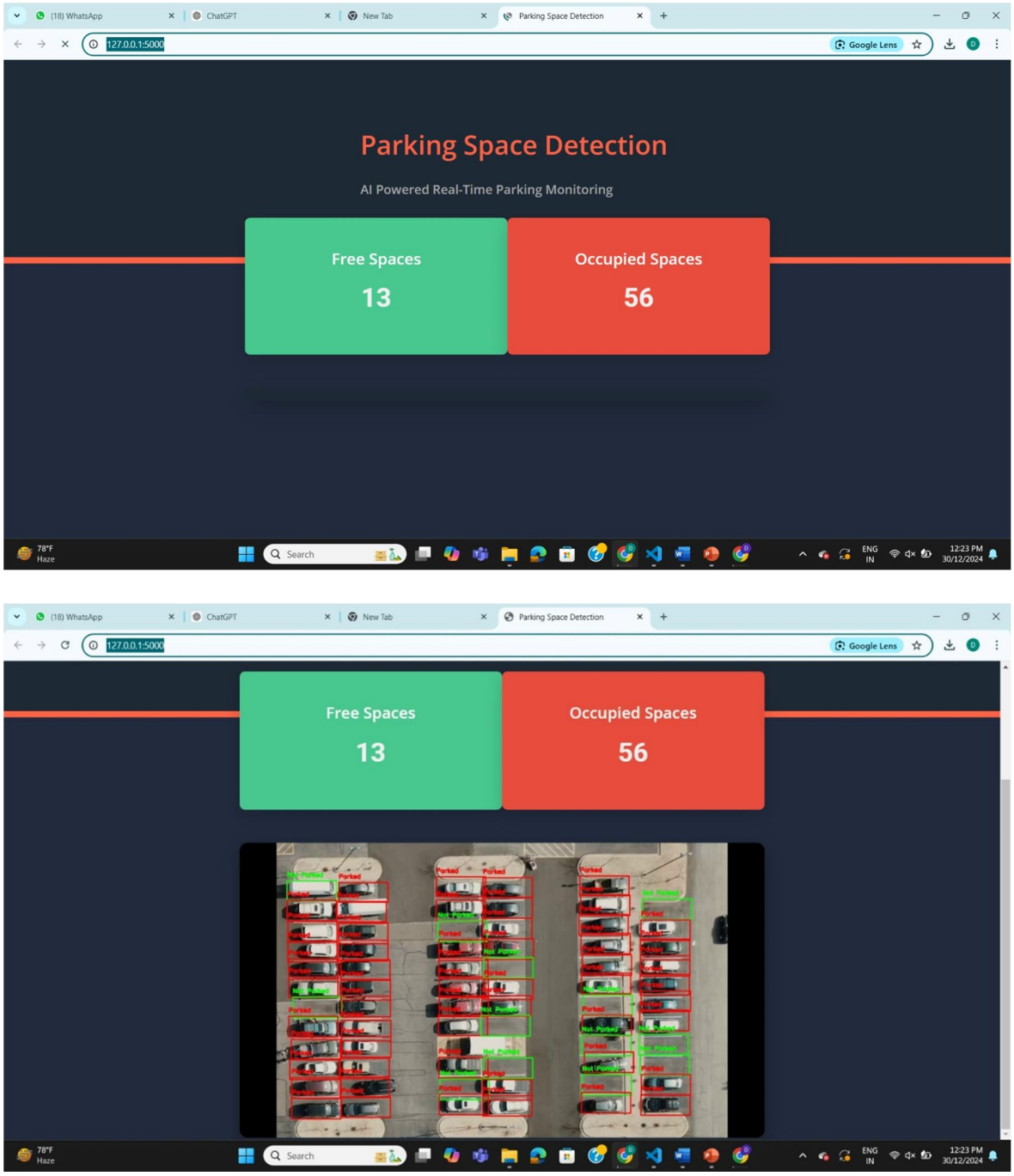


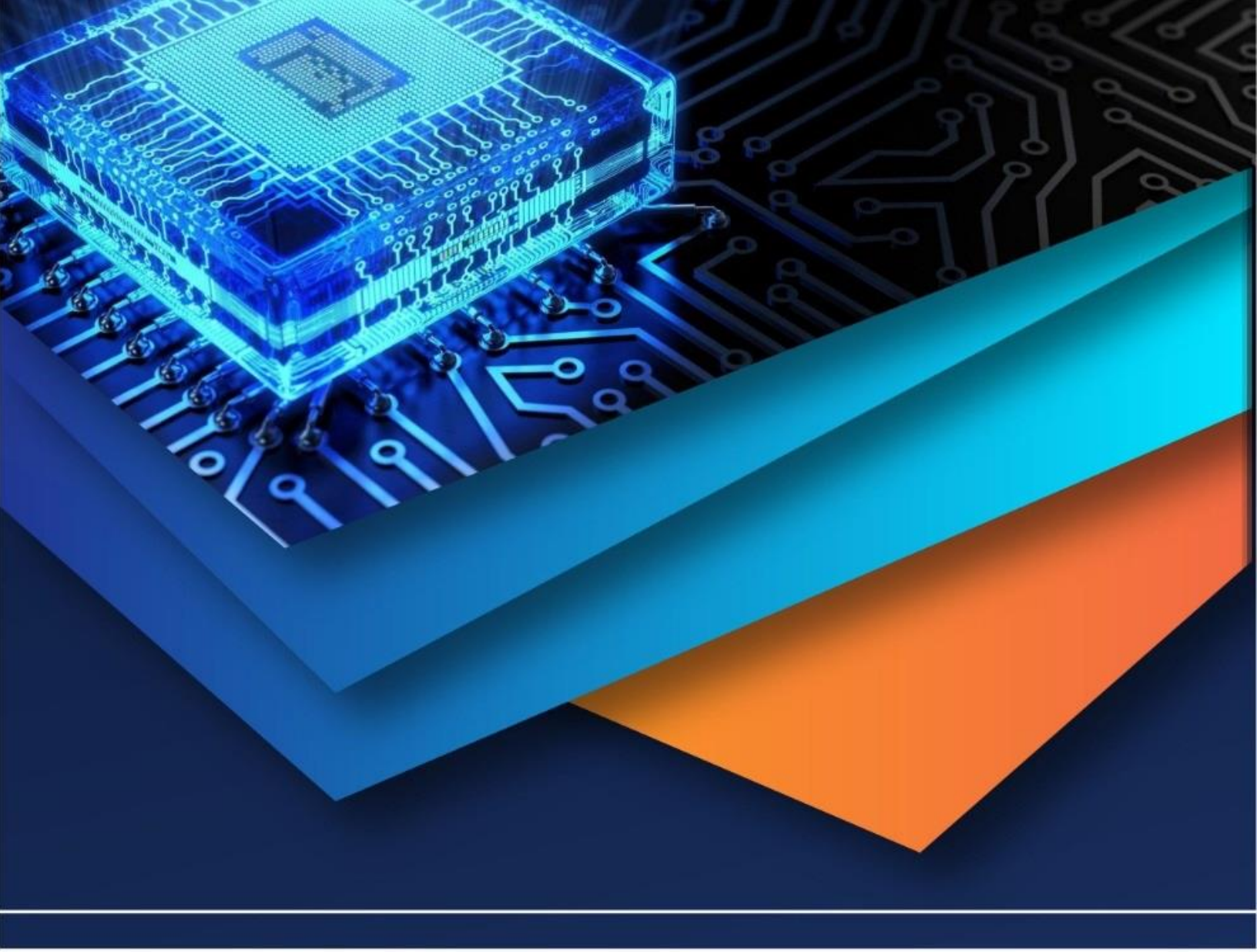
Fig 4: Results

VI. CONCLUSION

In this project, a Car Parking System was developed using VGG16, a deep learning model based on the Convolutional Neural Network (CNN) architecture, for detecting parking space occupancy. The goal was to automate the process of determining whether parking spots are occupied or free using video footage, by classifying each parking space based on the presence of a car.

REFERENCES

[1] Author(s): Mohamed A. A. Abdessalem, Wahiba Benameur, and Sofiene Affes. Year of Publication: 2018Title of the Paper: Real-time image-based parking occupancy detection using deep learning
[2] Method for judging parking status based on yolov2 target detection algorithm Hanbo Zhou, Yiming Zhaob, Wei Xiang,* 2021
[3] A Review: Object Detection Models Author - Aakash K. Shetty ; Ishani Saha; Rutvik M. Sanghvi; Siddhesh A. Save ; Yashkumar J. Patel



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)