

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint

# --- Main Program to Plot the Bee's Path ---

def main():
    """
    Main function to solve the differential equations and plot the 3D path.
    """
    print("Generating the 3D plot of the 'bee' path...")

    # Define the differential equations for the system (the 'bee's' movement)
    # The function takes the current state (x, y, z), time t, and parameters.
    # It returns the derivatives (x_dot, y_dot, z_dot).
    def bee_equations(state, t, a, b, c):
        """
        Defines the system of differential equations.

        Args:
            state (list): The current position [x, y, z].
            t (float): The current time.
            a (float): The first parameter.
            b (float): The second parameter.
            c (float): The third parameter.

        Returns:
            list: The derivatives of the position [x_dot, y_dot, z_dot].
        """
        x, y, z = state

        # The user-provided equations
        x_dot = a * (y - b)
        y_dot = b * x - y - x * z
        z_dot = x * y - c * z

        return [x_dot, y_dot, z_dot]

    # --- Parameters and Initial Conditions ---
    # The parameters 'a', 'b', and 'c' for the equations
    a = 20.0
    b = 58.0
    c = 4.667

    # The initial position of the 'bee' at time t=0
    x0 = 0.0
    y0 = 1.0
    z0 = 1.05
    initial_state = [x0, y0, z0]

    # --- Time Integration ---
    # Create a time array from 0 to 50 seconds with 5000 points
    # This determines the detail and length of the plotted path.
    t = np.linspace(0, 50, 5000)

    # Use the `odeint` function to solve the system of equations.
    # It takes the equation function, initial conditions, time array, and parameters.
    solution = odeint(bee_equations, initial_state, t, args=(a, b, c))

    # Extract the x, y, and z coordinates from the solution
    x = solution[:, 0]
    y = solution[:, 1]
    z = solution[:, 2]

    # --- 3D Plotting ---
    # Create a new figure for the 3D plot
    fig = plt.figure(figsize=(10, 8))
    ax = fig.add_subplot(111, projection='3d')

    # Plot the path using the x, y, and z coordinates
    ax.plot(x, y, z, lw=0.5)

    # Set the labels and title for the plot
    ax.set_title("3D Path of the 'Bee' Over Time")
    ax.set_xlabel("X Position")
    ax.set_ylabel("Y Position")

```

```

ax.set_zlabel("Z Position")

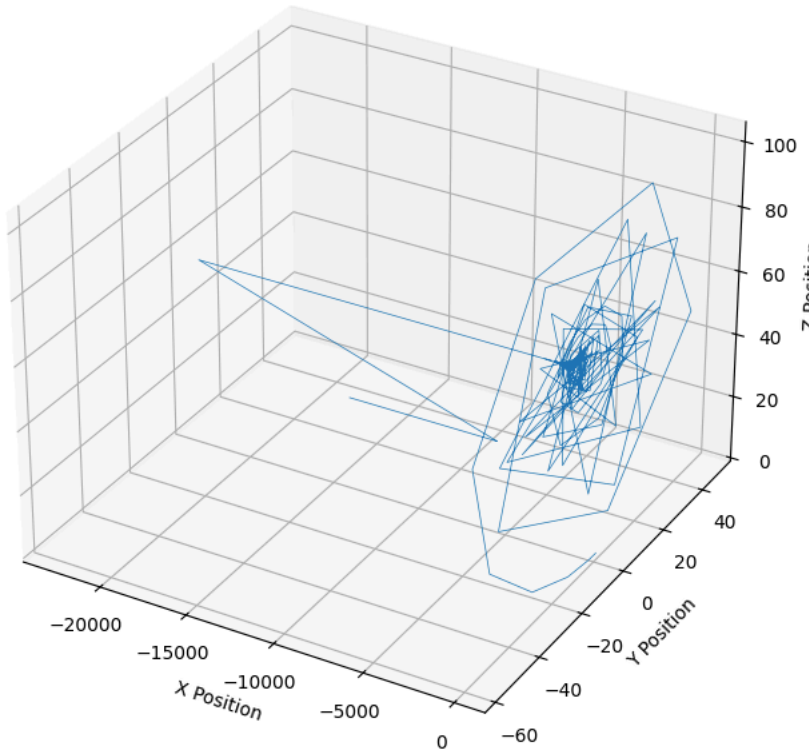
# Show the final plot
plt.show()
print("Plot generation complete.")

if __name__ == "__main__":
    main()

```

Generating the 3D plot of the 'bee' path...
 /tmp/ipython-input-2151827100.py:58: ODEintWarning: Excess work done on this call (perhaps wrong Dfun type). Run with full_output = 1 to
 solution = odeint(bee_equations, initial_state, t, args=(a, b, c))

3D Path of the 'Bee' Over Time



Plot generation complete.

```

# Create a .kaggle directory
!mkdir -p ~/.kaggle

# Move the uploaded kaggle.json to the directory
# This step assumes you have manually uploaded kaggle.json to your Colab session's files.
!mv /content/kaggle.json ~/.kaggle/

# Set file permissions for the API key
!chmod 600 ~/.kaggle/kaggle.json

# Download and unzip the dataset using the Kaggle API
print("Downloading dataset...")
!kaggle datasets download -d pratikbarua/vehicle-detection-dataset

print("Unzipping dataset...")
!unzip -q /content/vehicle-detection-dataset.zip

print("\nDataset is ready. You can now run the rest of the code.")

```

Downloading dataset...
 Dataset URL: <https://www.kaggle.com/datasets/pratikbarua/vehicle-detection-dataset>
 License(s): unknown
 Downloading vehicle-detection-dataset.zip to /content
 98% 1.37G/1.39G [00:11<00:00, 255MB/s]
 100% 1.39G/1.39G [00:11<00:00, 130MB/s]
 Unzipping dataset...

Dataset is ready. You can now run the rest of the code.

```
!pip install ultralytics matplotlib seaborn pandas xmltodict
```

```

ent already satisfied: numpy>=1.23.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (2.0.2)
ent already satisfied: opencv-python>=4.6.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (4.12.0.88)
ent already satisfied: pillow>=7.1.2 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (11.3.0)
ent already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (6.0.2)
ent already satisfied: requests>=2.23.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (2.32.4)
ent already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (1.16.1)
ent already satisfied: torch>=1.8.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (2.8.0+cu126)
ent already satisfied: torchvision>=0.9.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (0.23.0+cu126)
ent already satisfied: tqdm>=4.64.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics) (4.67.1)
ent already satisfied: psutil in /usr/local/lib/python3.12/dist-packages (from ultralytics) (5.9.5)
ent already satisfied: py-cpuinfo in /usr/local/lib/python3.12/dist-packages (from ultralytics) (9.0.0)
ng ultralytics-thop>=2.0.0 (from ultralytics)
adding ultralytics_thop-2.0.16-py3-none-any.whl.metadata (14 kB)
ent already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
ent already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
ent already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.59.1)
ent already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
ent already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (25.0)
ent already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.2.3)
ent already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.9.0.post0)
ent already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
ent already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
ent already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
ent already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests>=2.23.0->ultralytics) (3.4.3)
ent already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests>=2.23.0->ultralytics) (3.10)
ent already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests>=2.23.0->ultralytics) (2.5.0)
ent already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests>=2.23.0->ultralytics) (2025.8.3)
ent already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (3.19.1)
ent already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (4.14.1)
ent already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (75.2.0)
ent already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (1.13.3)
ent already satisfied: networkx in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (3.5)
ent already satisfied: Jinja2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (3.1.6)
ent already satisfied: fsspec in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (2025.3.0)
ent already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (12.6.77)
ent already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (12.6.77)
ent already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (12.6.80)
ent already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (9.10.2.21)
ent already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (12.6.4.1)
ent already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (11.3.0.4)
ent already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (10.3.7.77)
ent already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (11.7.1.2)
ent already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (0.7.1)
ent already satisfied: nvidia-nccl-cu12==2.27.3 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (2.27.3)
ent already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (12.6.77)
ent already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (12.6.85)
ent already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (1.11.1.6)
ent already satisfied: triton==3.4.0 in /usr/local/lib/python3.12/dist-packages (from torch>=1.8.0->ultralytics) (3.4.0)
ent already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch>=1.8.0->ultralytics) (1.37.0)
ent already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from Jinja2->torch>=1.8.0->ultralytics) (3.0.2)
ing ultralytics-8.3.182-py3-none-any.whl (1.0 MB)
1.0/1.0 MB 43.4 MB/s eta 0:00:00
ing xmltodict-0.14.2-py2.py3-none-any.whl (10.0 kB)
ing ultralytics_thop-2.0.16-py3-none-any.whl (28 kB)
ng collected packages: xmltodict, ultralytics-thop, ultralytics
ully installed ultralytics-8.3.182 ultralytics-thop-2.0.16 xmltodict-0.14.2

```

```

import os
import shutil
import random
import xml.etree.ElementTree as ET
from collections import defaultdict
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from ultralytics import YOLO

```

```

# Define the paths to your dataset directories.
# These paths are now correct for the Colab environment after unzipping.
TRAIN_DATA_PATH = '/content/train/Final Train Dataset'
TEST_DATA_PATH = '/content/test1/test'
YOLO_DATA_PATH = '/content/yolo_dataset'

```

```
os.makedirs(YOLO_DATA_PATH, exist_ok=True)
```

```

# =====
# STEP 3: DATA PREPARATION & VISUALIZATION

```

```

# STEP 2: DATA PREPARATION & VISUALIZATION
# =====

# Step 2a: Define the class names and a mapping function for XML to YOLO
CLASS_NAMES = ['car', 'bus', 'motorcycle', 'truck']
CLASS_MAP = {name: i for i, name in enumerate(CLASS_NAMES)}

def convert_xml_to_yolo(xml_path, image_size):
    """
    Parses an XML file and converts its annotations to the YOLO format.
    """
    try:
        tree = ET.parse(xml_path)
        root = tree.getroot()

        yolo_annotations = []

        img_width = float(image_size[0])
        img_height = float(image_size[1])

        for obj in root.findall('object'):
            class_name = obj.find('name').text
            if class_name not in CLASS_MAP:
                continue

            class_id = CLASS_MAP[class_name]

            bndbox = obj.find('bndbox')
            xmin = float(bndbox.find('xmin').text)
            ymin = float(bndbox.find('ymin').text)
            xmax = float(bndbox.find('xmax').text)
            ymax = float(bndbox.find('ymax').text)

            x_center = ((xmin + xmax) / 2) / img_width
            y_center = ((ymin + ymax) / 2) / img_height
            width = (xmax - xmin) / img_width
            height = (ymax - ymin) / img_height

            yolo_annotations.append(f"{class_id} {x_center} {y_center} {width} {height}")

        return yolo_annotations
    except Exception as e:
        print(f"Error processing XML file {xml_path}: {e}")
        return []

print("--- Data Processing ---")

image_files = [f for f in os.listdir(TRAIN_DATA_PATH) if f.endswith('.jpg')]
random.seed(42)
random.shuffle(image_files)

class_counts = defaultdict(int)

for split_type in ['train', 'val']:
    os.makedirs(os.path.join(YOLO_DATA_PATH, 'images', split_type), exist_ok=True)
    os.makedirs(os.path.join(YOLO_DATA_PATH, 'labels', split_type), exist_ok=True)

split_index = int(len(image_files) * 0.8)
train_files = image_files[:split_index]
val_files = image_files[split_index:]

for file_list, split_dir in zip([train_files, val_files], ['train', 'val']):
    for img_filename in file_list:
        base_filename = os.path.splitext(img_filename)[0]
        xml_filename = base_filename + '.xml'
        xml_path = os.path.join(TRAIN_DATA_PATH, xml_filename)

        if os.path.exists(xml_path):
            try:
                tree = ET.parse(xml_path)
                size = tree.getroot().find('size')
                img_width = int(size.find('width').text)
                img_height = int(size.find('height').text)

                yolo_annotations = convert_xml_to_yolo(xml_path, (img_width, img_height))

                if yolo_annotations:
                    shutil.copy(

```

```

        os.path.join(TRAIN_DATA_PATH, img_filename),
        os.path.join(YOLO_DATA_PATH, 'images', split_dir, img_filename)
    )
    with open(os.path.join(YOLO_DATA_PATH, 'labels', split_dir, f"{base_filename}.txt"), 'w') as f:
        f.write('\n'.join(yolo_annotations))
    for annotation in yolo_annotations:
        class_id = int(annotation.split(' ')[0])
        class_counts[CLASS_NAMES[class_id]] += 1
    except Exception as e:
        print(f"Could not process {xml_filename}: {e}")
    else:
        print(f"Skipping {img_filename} as no corresponding XML file was found.")

print(f"\nProcessed {len(train_files)} training images and {len(val_files)} validation images.")
print("Class Counts:", dict(class_counts))

# Prepare the test set
os.makedirs(os.path.join(YOLO_DATA_PATH, 'images', 'test'), exist_ok=True)
test_image_files = [f for f in os.listdir(TEST_DATA_PATH) if f.endswith('.jpg')]

for filename in test_image_files:
    shutil.copy(
        os.path.join(TEST_DATA_PATH, filename),
        os.path.join(YOLO_DATA_PATH, 'images', 'test', filename)
    )

print(f"Copied {len(test_image_files)} test images to {os.path.join(YOLO_DATA_PATH, 'images', 'test')}")

# Create the data.yaml configuration file
yaml_content = f"""
train: {os.path.join(YOLO_DATA_PATH, 'images', 'train')}
val: {os.path.join(YOLO_DATA_PATH, 'images', 'val')}
test: {os.path.join(YOLO_DATA_PATH, 'images', 'test')}

nc: {len(CLASS_NAMES)}
names: {CLASS_NAMES}
"""
with open('data.yaml', 'w') as f:
    f.write(yaml_content)

print("\ndata.yaml created successfully.")

# =====
#                               STEP 3: MODEL TRAINING AND PREDICTION
# =====

print("\n--- Model Training ---")

model = YOLO('yolov8n.pt')

model.train(
    data='data.yaml',
    epochs=50, # Set to a lower number for faster testing
    imgsz=640,
    project='vehicle_detection',
    name='yolov8_custom'
)

print("\n--- Prediction on Test Set ---")
best_model_path = os.path.join('/content/vehicle_detection/yolov8_custom/weights/best.pt')

if os.path.exists(best_model_path):
    model = YOLO(best_model_path)
    model.predict(
        source=os.path.join(YOLO_DATA_PATH, 'images', 'test'),
        save=True,
        save_txt=True,
        project='vehicle_detection',
        name='predictions'
    )
    print("\nPredictions complete. The results are saved in 'vehicle_detection/predictions' directory.")
else:
    print(f"Error: The best model file was not found at {best_model_path}. Training might have failed.")

```



```

image 452/500 /content/yolo_dataset/images/test/sabiha(260).jpg.rf.88a6ff8f5fcd603679dd4e27e78f9e5.jpg: 640x640 2 buss, 7.1ms
image 453/500 /content/yolo_dataset/images/test/sabiha(261).jpg.rf.743c1c6a6aa0640ab841c6a29d10b613.jpg: 640x640 (no detections), 7.1m
image 454/500 /content/yolo_dataset/images/test/sabiha(262).jpg.rf.5d3ccab3f3a933c9e7e3e57299234bc6.jpg: 640x640 (no detections), 7.1m
image 455/500 /content/yolo_dataset/images/test/sabiha(263).jpg.rf.03d0db5aa784c807bf9533db22290b6d.jpg: 640x640 (no detections), 7.1m
image 456/500 /content/yolo_dataset/images/test/sabiha(264).jpg.rf.10b6523bab965856aed3e6f92dbe52da.jpg: 640x640 (no detections), 7.1m
image 457/500 /content/yolo_dataset/images/test/sabiha(265).jpg.rf.779c5064fd24b4016dc45075040e93ee8.jpg: 640x640 (no detections), 7.1m
image 458/500 /content/yolo_dataset/images/test/sabiha(266).jpg.rf.fe35645467b87d79e05b7ac67743a5e8.jpg: 640x640 (no detections), 7.1m
image 459/500 /content/yolo_dataset/images/test/sabiha(267).jpg.rf.8fab973fa509682786b1da707718c910.jpg: 640x640 1 car, 1 bus, 7.1ms
image 460/500 /content/yolo_dataset/images/test/sabiha(268).jpg.rf.c384821ae960f93302304821d3fd9fa5.jpg: 640x640 1 car, 1 bus, 7.1ms
image 461/500 /content/yolo_dataset/images/test/sabiha(269).jpg.rf.d6731f50ee76e29dd737953ee7fc8cc.jpg: 640x640 1 car, 1 bus, 7.1ms
image 462/500 /content/yolo_dataset/images/test/sabiha(270).jpg.rf.afbe1ee8f34834d46cbca0711b33c2f5.jpg: 640x640 1 car, 1 bus, 7.1ms
image 463/500 /content/yolo_dataset/images/test/sabiha(271).jpg.rf.7e229e02ca290a09082e1f3104815064.jpg: 640x640 1 car, 1 bus, 1 truck
image 464/500 /content/yolo_dataset/images/test/sabiha(272).jpg.rf.84ee4cf90b760d131ac7edbf82c76c8b.jpg: 640x640 3 cars, 1 bus, 7.1ms
image 465/500 /content/yolo_dataset/images/test/sabiha(273).jpg.rf.917d0093d1684b05ca7a2db3d47fed3b.jpg: 640x640 1 car, 7.1ms
image 466/500 /content/yolo_dataset/images/test/sabiha(274).jpg.rf.8026836d96ed746ba17cb4b4cbd450c2.jpg: 640x640 (no detections), 7.1m
image 467/500 /content/yolo_dataset/images/test/sabiha(275).jpg.rf.ca2094c002d9b6e4bb35213403d0b540.jpg: 640x640 1 car, 1 bus, 7.1ms
image 468/500 /content/yolo_dataset/images/test/sabiha(276).jpg.rf.1d899834608964a21eca056262a8495.jpg: 640x640 1 bus, 7.8ms
image 469/500 /content/yolo_dataset/images/test/sabiha(277).jpg.rf.cc58f2ca32198d901da4962a9b8e388d.jpg: 640x640 1 car, 1 bus, 9.4ms
image 470/500 /content/yolo_dataset/images/test/sabiha(278).jpg.rf.0307226bdf5adcae4c23f3e35ef09430.jpg: 640x640 2 buss, 7.1ms
image 471/500 /content/yolo_dataset/images/test/sabiha(279).jpg.rf.1b73844e91805f851f80e116a2cdd436.jpg: 640x640 1 bus, 7.1ms
image 472/500 /content/yolo_dataset/images/test/sabiha(280).jpg.rf.a9cc77f7ae5411843f074197e5c8900d.jpg: 640x640 (no detections), 7.1m
image 473/500 /content/yolo_dataset/images/test/sabiha(281).jpg.rf.7c528a33de06a0ee9c86d6b09f5f556e.jpg: 640x640 (no detections), 7.2m
image 474/500 /content/yolo_dataset/images/test/sabiha(282).jpg.rf.841abeed169d40b6ad99fbaf322cd67.jpg: 640x640 1 bus, 7.1ms
image 475/500 /content/yolo_dataset/images/test/sabiha(283).jpg.rf.665d72b975d92aed831a2b3d3ba2bd1b.jpg: 640x640 (no detections), 7.1m
image 476/500 /content/yolo_dataset/images/test/sabiha(284).jpg.rf.f0d8a3083e94eb27c678d1ab590db8b4.jpg: 640x640 (no detections), 7.1m
image 477/500 /content/yolo_dataset/images/test/sabiha(285).jpg.rf.f4407d5e4601fe675015f3010c5d3bb.jpg: 640x640 (no detections), 7.1m
image 478/500 /content/yolo_dataset/images/test/sabiha(286).jpg.rf.457cd435b19e687b14f9e6d08f2100bd.jpg: 640x640 (no detections), 7.1m
image 479/500 /content/yolo_dataset/images/test/sabiha(287).jpg.rf.654a76b46af4f8e693747ff4d1329ccf.jpg: 640x640 (no detections), 7.1m
image 480/500 /content/yolo_dataset/images/test/sabiha(288).jpg.rf.4ff913df539f8238963ae6647085a479.jpg: 640x640 1 bus, 7.1ms
image 481/500 /content/yolo_dataset/images/test/sabiha(289).jpg.rf.ffa59002e097172ae82c9acd9808f942.jpg: 640x640 1 car, 1 bus, 7.1ms
image 482/500 /content/yolo_dataset/images/test/sabiha(290).jpg.rf.610ba4043e25a173509f60d3f89f9a08.jpg: 640x640 1 bus, 1 truck, 7.1ms
image 483/500 /content/yolo_dataset/images/test/sabiha(291).jpg.rf.e53eb2b14b4829010586b7a1ac0de107.jpg: 640x640 2 buss, 7.1ms
image 484/500 /content/yolo_dataset/images/test/sabiha(292).jpg.rf.1918842e4b4e3f3fa3094e3abf5c42a9.jpg: 640x640 1 car, 1 bus, 7.1ms
image 485/500 /content/yolo_dataset/images/test/sabiha(293).jpg.rf.c77bb8cf2a094a91b11f90265cd50ae8.jpg: 640x640 1 bus, 7.1ms
image 486/500 /content/yolo_dataset/images/test/sabiha(294).jpg.rf.a893d0d0095536e232867a893a7a7bba.jpg: 640x640 2 buss, 7.1ms
image 487/500 /content/yolo_dataset/images/test/sabiha(295).jpg.rf.11ce3b1228e313784ff802dce9fbedc0.jpg: 640x640 1 bus, 7.1ms
image 488/500 /content/yolo_dataset/images/test/sabiha(297).jpg.rf.3cd624e290aefcc04e9f4e4f7e2cf869.jpg: 640x640 (no detections), 7.1m
image 489/500 /content/yolo_dataset/images/test/sabiha(298).jpg.rf.facade9e2a68b70d04ee4c8278efaa83.jpg: 640x640 1 bus, 7.1ms
image 490/500 /content/yolo_dataset/images/test/sabiha(299).jpg.rf.374710a574064d7a97efe09acacd2275.jpg: 640x640 1 bus, 7.1ms
image 491/500 /content/yolo_dataset/images/test/sabiha(300).jpg.rf.5c95cec0762afb2c66e8e438911788b4.jpg: 640x640 1 car, 1 bus, 7.2ms
image 492/500 /content/yolo_dataset/images/test/sabiha(301).jpg.rf.1170ed9bae1473adcc41da0a8d92463c.jpg: 640x640 1 bus, 7.1ms
image 493/500 /content/yolo_dataset/images/test/sabiha(302).jpg.rf.f3d2d1ed17291539560eede57601ba64.jpg: 640x640 1 car, 2 buss, 7.1ms
image 494/500 /content/yolo_dataset/images/test/sabiha(303).jpg.rf.66506cde12b3e190df17bc28c0fcd878.jpg: 640x640 2 buss, 7.1ms
image 495/500 /content/yolo_dataset/images/test/sabiha(304).jpg.rf.43ab27f27507ac52a04623ba76ea08a2.jpg: 640x640 1 bus, 7.1ms
image 496/500 /content/yolo_dataset/images/test/sabiha(305).jpg.rf.074c9622acab9462d3684ac016a418cc.jpg: 640x640 2 buss, 7.1ms
image 497/500 /content/yolo_dataset/images/test/sabiha(306).jpg.rf.805d44e365ff6c92ea18f247dabf3406.jpg: 640x640 1 bus, 7.1ms
image 498/500 /content/yolo_dataset/images/test/sabiha(307).jpg.rf.9254b96641d2c1493221251d6f8dfc17.jpg: 640x640 1 car, 1 bus, 7.1ms
image 499/500 /content/yolo_dataset/images/test/sabiha(308).jpg.rf.e32e5aec95792c69753f42b72724057c.jpg: 640x640 4 buss, 7.1ms
image 500/500 /content/yolo_dataset/images/test/sabiha(309).jpg.rf.93e77c106e2f415529533bacade2485e.jpg: 640x640 1 bus, 7.1ms

```

Speed: 3.4ms preprocess, 7.5ms inference, 1.3ms postprocess per image at shape (1, 3, 640, 640)

Results saved to **vehicle_detection/predictions**

328 labels saved to vehicle_detection/predictions/labels

Predictions complete. The results are saved in 'vehicle_detection/predictions' directory.

◆ Gemini

```

# Zip the output directory
!zip -r /content/vehicle_detection.zip /content/vehicle_detection

print("\nDirectory zipped. You can now download vehicle_detection.zip from the files panel.")

```



```
adding: content/vehicle_detection/predictions/Sabiha(194)_jpg.rf.c122373/0008/e808/e4e2ac/e97a431.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/Sabiha(3)_jpg.rf.3e18a5ba51674a4be06858ab6252d43d.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/Shykat_5_(54)_jpg.rf.0d5886ce1d1e71c3522957da635b1062.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/Shykat_5_(68)_jpg.rf.5718dea3dcb99362d60f5e4a360fb54c.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/Sabiha(287)_jpg.rf.654a76b46af4f8e693747ff4d1329ccf.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/Shykat_5_(28)_jpg.rf.54734e64d93f5311efa478458d981015.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/Shykat_5_(14)_jpg.rf.4b2383070e0446ce1a10a3526ebe7105.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/sabiha(241)_jpg.rf.fd2c8189c9e0d74ffc8b34c85aa3ddca.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/Shykat_4_(43)_jpg.rf.457ec12ec835929047e4c011e9361e9b.jpg (deflated 6%)
adding: content/vehicle_detection/predictions/Shykat_4_(39)_jpg.rf.c57524355aeca23e0c2ab2cda25aaa5c.jpg (deflated 3%)
adding: content/vehicle_detection/predictions/Shykat_5_(94)_jpg.rf.eb35130704f5929f0ba32a26ce0315ff.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/Shykat_5_(69)_jpg.rf.b54c84a84a6c91927b5afba60d0dd0e3.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/Shykat_02_015.jpg.rf.90e353daaaad993d1c60ee502e48793f.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/Sabiha(8)_jpg.rf.24900113770665f45ff44a6bc6c36911.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/sabiha(133)_jpg.rf.86a2b77a32589d81bca3820c5b1912d4.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/sabiha(299)_jpg.rf.374710a574064d7a97efe09acac2275.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/Shykat_4_(13)_jpg.rf.987c7fb1045e8a757ca5740dea508c7b.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/Sabiha(14)_jpg.rf.801d6d37254ef2d3d43876bdcf5d9d54.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/Sabiha(24)_jpg.rf.66060ac19e1ebaf7aaebf7b638a17178.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/Shykat_5_(3)_jpg.rf.0448dd6ebc43fd8fc4d2d2f8d0a9dc9b.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/Shykat_5_(64)_jpg.rf.6e16a7cab17e310481abe9635f5afa47.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/Sabiha(01)_jpg.rf.f935c7fc51a14c64e34c17a17c41cb1f.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/Shykat_02_010.jpg.rf.9bf70d019ae3d0e74921341c4f6eb39c.jpg (deflated 7%)
adding: content/vehicle_detection/predictions/Shykat_5_(97)_jpg.rf.64d22026d38be7b14963a464abec6304.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/sabiha(284)_jpg.rf.f0d8a3083e94eb27c678d1ab590db8b4.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/Sabiha(10)_jpg.rf.d163e2a53d5ed22112e9ead62807acf4.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/sabiha(306)_jpg.rf.805d44e365ff6c92ea18f247dabf3406.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/sabiha(167)_jpg.rf.64c558bc072f848330231c8c89fd0bf6.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/Shykat_02_020.jpg.rf.deddb99792c580f276d35d5aa50e0247.jpg (deflated 7%)
adding: content/vehicle_detection/predictions/Shykat_5_(42)_jpg.rf.b9c0eaf47758f42ac76cfb5fc0520c29.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/Shykat_02_030.jpg.rf.4b789303eddebb18dbf3d4a90c45cc43.jpg (deflated 7%)
adding: content/vehicle_detection/predictions/sabiha(222)_jpg.rf.4eac8e3b730dafc1b2acf62d6e7e32b3.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/Shykat_03_014.jpg.rf.29404b71284ea2d6d2b6a2e0ff9832d7.jpg (deflated 5%)
adding: content/vehicle_detection/predictions/Sabiha(38)_jpg.rf.0938b234884f5978919bd69e2aa4cb10.jpg (deflated 4%)
adding: content/vehicle_detection/predictions/Shykat_5_(110)_jpg.rf.71606778c62285beb0c0db0ab3480680c.jpg (deflated 6%)
```

Directory zipped. You can now download vehicle_detection.zip from the files panel.