```python
# ------------------------------
# Project: AI-powered Eco-friendly Traffic Management
# Author: [Your Name]
# Description: Detects traffic density from uploaded images or videos
#              and suggests optimal traffic light duration.
# Note: This project demonstrates a solution-driven AI application
#       for reducing traffic congestion and carbon emissions.
#       Deployable using Streamlit for live demonstration.
# Date: [Today's Date]
# ------------------------------

# ------------------------------
# Step 0: Install required packages
# ------------------------------
# pip install streamlit opencv-python numpy pillow

# ------------------------------
# Step 1: Import libraries
# ------------------------------
import streamlit as st
import cv2
import numpy as np
from PIL import Image

# ------------------------------
# Step 2: Streamlit Page Configuration
# -------------------------------
st.set_page_config(page_title="Eco-friendly Traffic Management", layout="wide")
st.title("🚦 AI-powered Eco-friendly Traffic Management")
st.write("Upload a traffic video or image to detect vehicles and suggest optimal traffic light duration.")


# -------------------------------
# Step 3: Upload Video or Image
# -------------------------------
uploaded_file = st.file_uploader("Upload Traffic Video (MP4) or Image (JPG/PNG)", type=["mp4", "jpg", "png"])


if uploaded_file:
  # Check if file is image or video
  if uploaded_file.type in ["image/jpeg", "image/png"]:
    image = Image.open(uploaded_file)
    st.image(image, caption="Uploaded Traffic Image", use_column_width=True)
    frame = np.array(image)


    # Simulated vehicle detection using contours
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    _, thresh = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)
    contours, _ = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    vehicle_count = len(contours)
    st.success(f"Detected vehicles: {vehicle_count}")
    st.info(f"Suggested Green Light Duration: {max(10, vehicle_count * 2)} seconds")

else:
    # Process video
    tfile = uploaded_file.name
    with open(tfile, "wb") as f:
        f.write(uploaded_file.read())

    st.video(tfile)
    st.info("Processing video for vehicle detection (simulated)...")

    # Video capture
    cap = cv2.VideoCapture(tfile)
    total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
    st.write(f"Total frames: {total_frames}")
    vehicle_counts = []
    frame_number = 0

    while cap.isOpened() and frame_number < total_frames:
        ret, frame = cap.read()
        if not ret:
            break
        # Simulated vehicle detection
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        _, thresh = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)
        contours, _ = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
```

```
            vehicle_counts.append(len(contours))
            frame_number += 30  # Skip frames for faster simulation

        cap.release()
        avg_vehicles = int(np.mean(vehicle_counts))
        st.success(f"Average detected vehicles: {avg_vehicles}")
        st.info(f"Suggested Green Light Duration: {max(10, avg_vehicles * 2)} seconds")

# -------------------------------
# End of AI-powered Eco-friendly Traffic Management
# -------------------------------
```