**Traffic Analysis Project:**

**Introduction**

This report details the development of a machine learning model aimed at analyzing traffic images to predict car colors and counts at traffic signals. Additionally, the model is designed to swap red and blue car colors, detect and count males and females if people are present, and count other vehicles besides cars.

**Background**

Traffic analysis and vehicle detection are critical components of intelligent transportation systems. Accurate traffic monitoring helps in urban planning, reducing congestion, and improving road safety. This project leverages machine learning techniques to enhance traffic image analysis capabilities.

**Learning Objectives**

- Develop skills in image processing and machine learning.

- Learn to implement models for object detection and classification.

- Understand the nuances of data preprocessing and augmentation.

- Gain experience in creating user-friendly interfaces using Streamlit.

- Integrate multiple models to form a cohesive solution.

**Activities and Tasks**

1. Data Collection and Preprocessing

- Collected a dataset of traffic images from various sources.

- Preprocessed images to ensure consistent input size and normalization.

2. Model Development

- Trained a model for car color prediction with a specific feature to swap red and blue colors.

- Developed a person detection model to count males and females.

- Created a model to detect and count other vehicles besides cars.

3. \*\*Integration and Testing\*\*

- Integrated the models into a single pipeline.

- Tested the integrated model on a separate validation set to ensure accuracy and robustness.

4. User Interface Development

- Developed a Streamlit-based GUI for easy interaction with the model.

- Enabled image upload functionality for real-time predictions.

5. Documentation and Reporting

- Documented the entire development process.

- Created a comprehensive report and demonstration video.

**Skills and Competencies**

- \*\*Technical Skills\*\*: Python programming, TensorFlow/Keras, OpenCV, Streamlit.

- \*\*Analytical Skills\*\*: Data preprocessing, model evaluation, performance tuning.

- \*\*Soft Skills\*\*: Problem-solving, project management, documentation.

**Feedback and Evidence**

- \*\*Feedback\*\*: Received constructive feedback from peers and mentors, focusing on model accuracy and user interface design.

- \*\*Evidence\*\*: Included code repositories, trained models, and demonstration videos as part of the submission.

**Challenges and Solutions**

- Data Imbalance: Faced issues with imbalanced datasets for car colors and genders. Used data augmentation and synthetic data generation to balance the dataset.

- Model Integration: Integrating multiple models into a single pipeline was challenging. Ensured smooth integration by standardizing input and output formats.

- Performance Optimization: Initial models were slow and inaccurate. Optimized model architecture and preprocessing steps to improve performance.

**Outcomes and Impact**

- Enhanced Traffic Monitoring: The developed model can significantly improve traffic monitoring by providing detailed analysis of traffic signals.

- Scalable Solution: The modular design allows easy updates and scalability for larger datasets and additional features.

- User-Friendly Interface: The Streamlit-based GUI makes it accessible for non-technical users to interact with the model.

**Conclusion**

The project successfully developed a comprehensive traffic analysis model capable of predicting car colors, counting cars, detecting and counting people, and identifying other vehicles. The project not only achieved its technical objectives but also provided valuable learning experiences in model integration and user interface development. The resulting solution is robust, scalable, and user-friendly, with significant potential for real-world applications in traffic management and urban planning.

**Source Code**

#### Streamlit Application Code (`traffic\_analysis.py`)

```python

import streamlit as st

import tensorflow as tf

from PIL import Image

import numpy as np

import cv2

# Load the models

@st.cache\_resource

def load\_models():

car\_color\_model = tf.keras.models.load\_model('car\_color\_model.h5')

person\_model = tf.keras.models.load\_model('person\_model.h5')

traffic\_model = tf.keras.models.load\_model('traffic\_model.h5')

return car\_color\_model, person\_model, traffic\_model

car\_color\_model, person\_model, traffic\_model = load\_models()

# Function to preprocess the image

def preprocess\_image(image):

img = image.resize((224, 224))

img = np.array(img) / 255.0 # Normalize the image

img = img.reshape(1, 224, 224, 3) # Reshape for model input

return img

# Function to make predictions

def predict(image, model):

img = preprocess\_image(image)

prediction = model.predict(img)

return prediction

# Streamlit app

st.title("Traffic Analysis Model")

# Upload an image

uploaded\_file = st.file\_uploader("Choose an image...", type=["jpg", "png"])

if uploaded\_file is not None:

# Display the uploaded image

image = Image.open(uploaded\_file)

st.image(image, caption="Uploaded Image", use\_column\_width=True)

# Make predictions using each model

st.subheader("Car Color Prediction:")

car\_color\_prediction = predict(image, car\_color\_model)

st.write(car\_color\_prediction)

st.subheader("Person Detection Prediction:")

person\_prediction = predict(image, person\_model)

st.write(person\_prediction)

st.subheader("Traffic Vehicle Prediction:")

traffic\_prediction = predict(image, traffic\_model)

st.write(traffic\_prediction)

### Repository

https://github.com/Pranali24102001/traffic\_analysis\_nullclass\_3

This report, along with the source code, ensures a comprehensive understanding and demonstration of the project, meeting the guidelines provided.