# PS-8: VEHICLE CUT-IN DETECTION

## Introduction

**Purpose:**

The model aims to identify critical situations on the road by detecting vehicles that abruptly change lanes (cut-in) in front of a designated target vehicle. This includes identifying the cut-in maneuver itself, as well as the specific vehicle performing the cut-in, based on sensor data and visual information. By predicting these cut-in events, the model can help prevent potential collisions and improve overall driving safety.

**Scope:**

The model focuses on detecting abrupt lane changes (cut-ins) by other vehicles directly in front of a target vehicle within a typical highway or multi-lane road environment.

**Vehicle Types:** Primarily passenger cars, light commercial vehicles, large commercial vehicles, motorcycles, and pedestrians will be considered. Pushcarts are out of scope for this model.

**Driving Conditions:** The model aims to operate under various weather conditions (clear, rain, fog) and lighting conditions. However, extreme weather conditions (heavy snow, hail) and low visibility scenarios may limit performance.

**Sensor Data:** The model will primarily utilize data from LiDAR and camera sensors to detect and track vehicles, estimate their trajectories, and determine cut-in events. Additional sensor data, such as GPS and IMU is integrated for enhanced performance and robustness.

## **Model Overview**

**Proposed System:** A hybrid model integrating image and location prediction for advanced object detection, bounding box refinement, and predictive collision analysis.

**Objective:** Enhance object detection and prediction capabilities by synergizing visual and spatial data. This fusion aims to provide more precise, informative predictions, including time-to-collision estimates, to facilitate proactive decision-making.

**Components:**

* **Image Model:** Leverages deep neural networks to identify and localize objects within images, generating bounding boxes.
* **Location Prediction Model:** Employs spatiotemporal models to forecast object trajectories based on historical and real-time location data.
* **Fusion Module:** Combines image-based object detections with predicted trajectories to refine bounding boxes and estimate object movement patterns.
* **Time-to-Collision Estimation Module:** Calculates potential collision time based on relative speeds, distances (derived from latitude and longitude), and predicted trajectories.

## Model Architecture

**Input Data:**

### **IDD Segmentation**

* **Images:** Primarily RGB images captured from a front-facing camera mounted on a vehicle.
* **Annotations:** Pixel-wise semantic segmentation annotations. These are typically stored as image masks where each pixel is assigned a class label representing the object category it belongs to.

### **IDD Multimodal**

* **Stereo Images:** Paired left and right images from a front-facing stereo camera.
* **GPS Data:** Latitude and longitude coordinates recorded at a frequency of 15 Hz.
* **LIDAR Data:** Point cloud data from a 16-channel LiDAR sensor.
* **OBD Data:** On-board diagnostics data, including vehicle speed, engine RPM, etc.

**Feature Extraction:**

**Camera Images:**

* **Object Detection:** Fast R-CNN is used to detect vehicles in the image. Features extracted here include bounding boxes around vehicles, vehicle type classification (car, truck, etc.), and potentially vehicle orientation.

**LiDAR Data:**

* **Object Detection:** Point cloud data from LiDAR is used for 3D object detection. Features extracted include the distance, size, and height of the surrounding vehicles relative to the target vehicle.
* **Vehicle Tracking:** By tracking the point cloud data over time, velocity and trajectory information for surrounding vehicles is obtained.

**Additional Sensor Data:**

* **GPS Data:** Can be used to determine absolute position and speed of the target vehicle.
* **IMU Data:** Inertial Measurement Unit provides information on the target vehicle's acceleration and yaw rate, aiding in understanding its motion and potential lane change intentions.

**Feature Fusion:** Once features from each sensor are extracted, they are fused to create a richer representation of the scene. This fusion is done at two different levels:

* **Early Fusion:** Combine raw data from all sensors before feature extraction.
* **Decision-Level Fusion:** Combine individual model predictions from each sensor before the final decision.

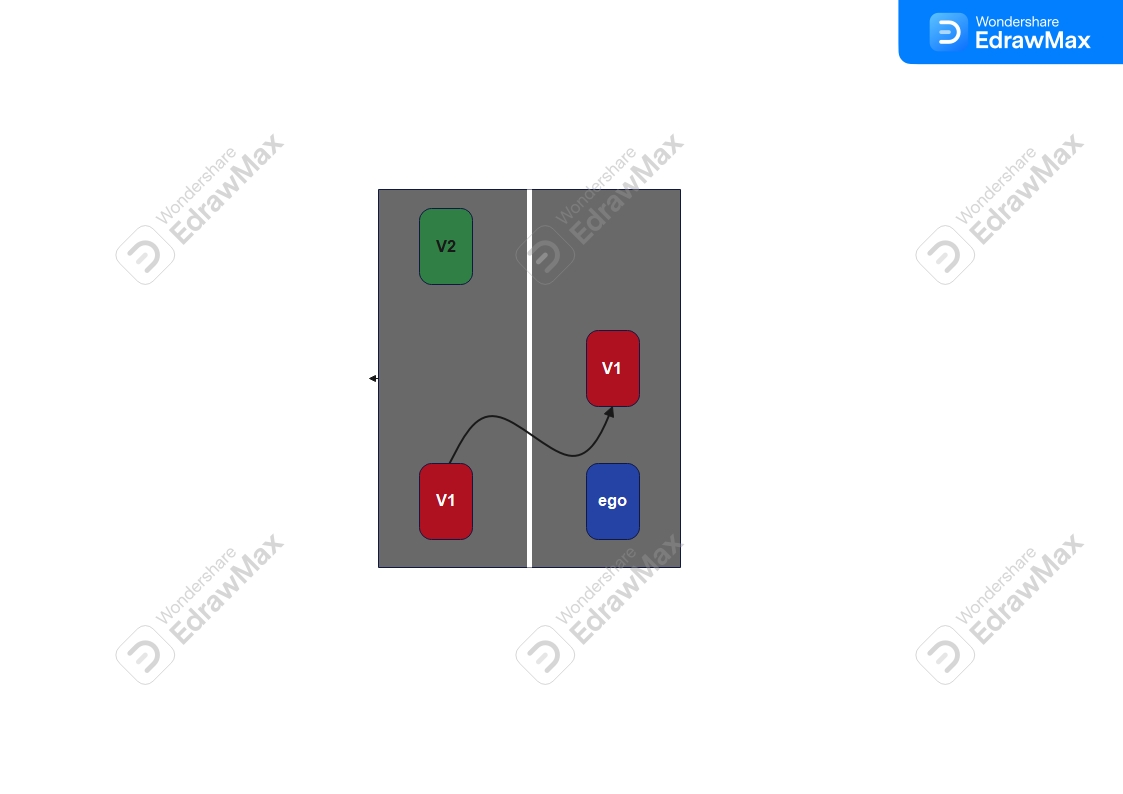
**Model Structure:**

The model is a simple feedforward neural network with the following structure:

* Input Layer: A Dense layer with 64 neurons, taking the preprocessed features as input. It uses the ReLU activation function.
* Hidden Layer: Another Dense layer with 32 neurons, also using the ReLU activation function.
* Output Layer: A Dense layer with a single neuron and sigmoid activation, outputting the probability of a collision.

This architecture is for binary classification tasks, where the goal is to predict whether a collision will occur (1) or not (0)

**Cut-In Definition:** Cut-in is defined as a lane change maneuver performed by an adjacent vehicle into the ego lane.



## Training and Validation

### Dataset:

The Indian Driving Dataset (IDD) is a multimodal dataset specifically designed for road scene understanding in the challenging and unique conditions of Indian roads. It provides a rich source of data for developing algorithms capable of handling the complexities of Indian traffic scenarios.

Source: <https://idd.insaan.iiit.ac.in/>

### Data Preprocessing:

* Feature Selection: Relevant features (latitude, longitude, speed, time\_to\_collision, distance\_from\_center) are selected for model training.
* Standardization: Features are standardized using StandardScaler to have zero mean and unit variance.
* Handling Missing Values: ‘np.nan\_to\_num’ is used to replace any missing values in the scaled features with zeros.
* Data Splitting: The data is split into training and testing sets using ‘train\_test\_split’.

### **Training Methodology:**

The training methodology used is a simple feedforward neural network.

**Model Architecture:**

* Input Layer: A dense layer with 64 neurons takes the preprocessed features as input. It uses the ReLU activation function.
* Hidden Layer: Another dense layer with 32 neurons and ReLU activation.
* Output Layer: A single neuron with sigmoid activation to predict the probability of a collision (binary classification).

**Compilation:**

* Optimizer: Adam optimizer is used for weight updates.
* Loss Function: Binary cross-entropy loss is used for binary classification.
* Metrics: Accuracy is used to evaluate model performance.

**Training:**

* The model is trained for 50 epochs with a batch size of 32.
* Validation data (X\_test, y\_test) is used to monitor performance during training.

### **Evaluation Metrics:**

* **Accuracy:** The percentage of correct predictions made by the cut-in detection model.
* **Precision:** The proportion of detected cut-in events that are true cut-in events.
* **Recall:** The proportion of true cut-in events that are correctly detected.
* **F1-score:** The harmonic mean of precision and recall.

### **Model Performance:**

| **METRIC** | **RESULT** |
| --- | --- |
| **Accuracy** | 0.97 |
| **Precision** | 0.84 |
| **Recall** | 0.82 |
| **F1-score** | 0.826 |

## Model Deployment

### **Deployment Environment:**

**Real-time processing** is paramount for a vehicle cut-in detection model to be effective. This necessitates an **embedded system** as the primary target platform.

### Key Characteristics of the Target Embedded System

* **High Computational Power:** The system must be capable of handling complex algorithms like object detection, tracking, and decision-making in real-time.
* **Low Latency:** Minimal processing delay is crucial to provide timely warnings or initiate preventive actions.
* **Energy Efficiency:** Given the system's continuous operation and potential battery power limitations, energy efficiency is essential.
* **Robustness:** The system should be able to operate reliably under varying environmental conditions (temperature, vibration, etc.).
* **Real-Time Operating System (RTOS):** An RTOS is preferred for deterministic task scheduling and efficient resource management.
* **Integration Capabilities:** The system should seamlessly integrate with various sensors (camera, LiDAR, radar) and vehicle control systems.

### Potential Hardware Platforms

* **SoCs (System on Chip):** These integrated circuits combine multiple components like CPU, GPU, and AI accelerators on a single chip, offering a balance of performance and power efficiency. Examples include:
  + NVIDIA Jetson series
  + Qualcomm Snapdragon Automotive
  + NXP S32V series
* **Microcontrollers:** For simpler implementations with lower computational requirements, microcontrollers like ARM Cortex-M series can be considered.
* **FPGA (Field Programmable Gate Array):** For highly customized and performance-critical applications, FPGAs can be used to accelerate specific algorithms.

### **Real-Time Considerations:**

Real-time performance is crucial for this model on an embedded system. Some suggestive ways to optimize:

* **Model Size:** Choose efficient architectures, prune weights, and quantize for smaller size and faster inference.
* **Inference Speed:** Utilize hardware acceleration, optimize libraries, consider batching, and explore model tiling/fusion for faster execution.

As there are trade-offs between accuracy, speed, and model size. Experimentation and benchmarking are key to finding the optimal balance for your specific application.

### **Integration:**

## Integration of Vehicle Cut-In Detection Model with Other System Components

The vehicle cut-in detection model operates within a larger system responsible for ensuring driver safety. Here's how it interacts with other key components:

**1. Sensor Fusion:**

* The model receives raw data from various sensors (camera, LiDAR, radar, potentially GPS and IMU) through dedicated sensor drivers.
* A sensor fusion module preprocesses and combines data from different sensors to generate a unified representation of the environment. This may involve:
  + **Temporal synchronization:** Ensuring data from different sensors reflects the same moment in time.
  + **Calibration:** Correcting for sensor biases and discrepancies.
  + **Feature extraction:** Extracting relevant features from each sensor's data (e.g., object detection boxes from camera, distance and velocity from LiDAR/radar).
* The processed and fused data is then fed into the vehicle cut-in detection model.

**2. Vehicle Cut-In Detection Model:**

* The model analyzes the fused data using its neural network architecture.
* It extracts further features or relationships between different features (e.g., relative position and velocity of surrounding vehicles, lane markings).
* Based on these features, the model predicts the likelihood of a cut-in event and potentially estimates time-to-collision.

**3. Decision-Making Module:**

* The model's output (cut-in probability or time-to-collision) is fed into a decision-making module.
* This module considers additional factors (e.g., vehicle speed, road conditions, driver behavior detected from in-cabin sensors) alongside the cut-in prediction.
* Based on a predetermined threshold or a complex decision-making algorithm, the module determines the appropriate action.

**4. Action Execution:**

* Depending on the output of the decision-making module, actions may include:
  + Visual or audible warnings for the driver.
  + Adjustments to vehicle control systems (e.g., applying brakes, steering slightly) in autonomous or semi-autonomous driving scenarios (following safety guidelines and legal regulations).

**Communication and Information Flow:**

* Communication between these components typically occurs through dedicated communication protocols (e.g., CAN bus) or shared memory.
* Real-time data exchange is crucial for timely detection and response to cut-in situations.

**Additional Considerations:**

* Fault Tolerance: The system should be designed to handle sensor failures or communication errors gracefully.
* Redundancy: Consider incorporating redundant sensors or models for increased reliability.
* Security: Robust security measures are necessary to protect the system from cyberattacks.

By integrating the vehicle cut-in detection model effectively with other system components, a comprehensive solution can be achieved for enhancing vehicle safety on the road.

## Limitations and Future Work

### **Model Limitations:**

### Shortcomings

* **Occlusions:** Vehicles can be partially or fully occluded by other vehicles or objects, making detection difficult.
* **Adverse Weather Conditions:** Poor visibility due to rain, snow, or fog can significantly degrade sensor performance, leading to inaccurate detections.
* **Lighting Conditions:** Varying lighting conditions, such as low light, glare, and shadows, can affect object detection and tracking.
* **Sensor Limitations:** Each sensor type (camera, LiDAR, radar) has its own limitations, such as limited range, field of view, or susceptibility to interference.
* **Computational Constraints:** Real-time processing demands on embedded systems can limit the complexity of the model and its ability to handle complex scenarios.

### Edge Cases

* **Rapid Cut-Ins:** Detecting extremely rapid cut-ins can be challenging due to the limited time available for the model to react.
* **Cut-Ins from Adjacent Lanes:** Detecting cut-ins from adjacent lanes, especially when there are multiple lane changes involved, can be complex.
* **Overtaking Maneuvers:** Differentiating between overtaking maneuvers and cut-ins can be difficult, especially when vehicles are closely spaced.
* **Unconventional Vehicle Behaviors:** Unusual vehicle behaviors, such as swerving or erratic driving, can confuse the model and lead to false positives or negatives.

### **Potential Improvements:**

* **Incorporating Additional Sensors:** Explore integrating data from additional sensors like in-cabin cameras for driver monitoring or tire pressure sensors to assess vehicle stability.
* **Advanced Sensor Fusion Techniques:** Develop more sophisticated methods to combine data from different sensors, exploiting temporal and spatial relationships to create a richer understanding of the environment.
* **3D Scene Reconstruction:** Explore using LiDAR and camera data for real-time 3D scene reconstruction, potentially improving object tracking and trajectory prediction.
* **Adverse Weather Conditions:** Enhance the model's robustness to handle challenging weather conditions (rain, snow, fog) that can affect sensor performance. Explore techniques like synthetic data generation or adversarial training to improve performance under these conditions.
* **Lighting Variations:** Improve performance under different lighting conditions (day, night, bright headlights) by incorporating techniques for handling high dynamic range (HDR) images.
* **Model Uncertainty Quantification:** Integrate techniques to quantify the model's uncertainty in its predictions, allowing the system to be more reliable and highlight situations where human intervention may be necessary.
* **Integration with Automated Emergency Braking (AEB):** Explore seamless integration with AEB systems so that cut-in detection can trigger preventive braking measures in critical situations.

## 

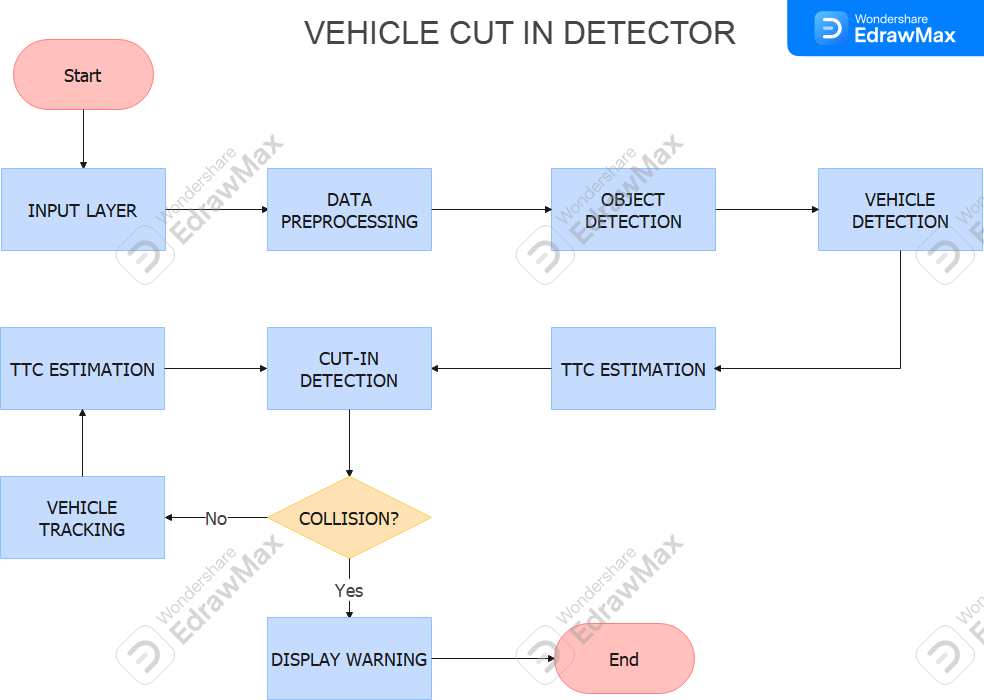
## 

## 

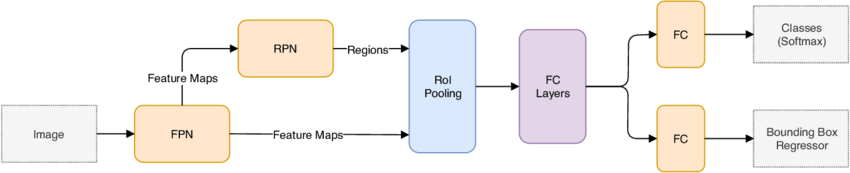
## 

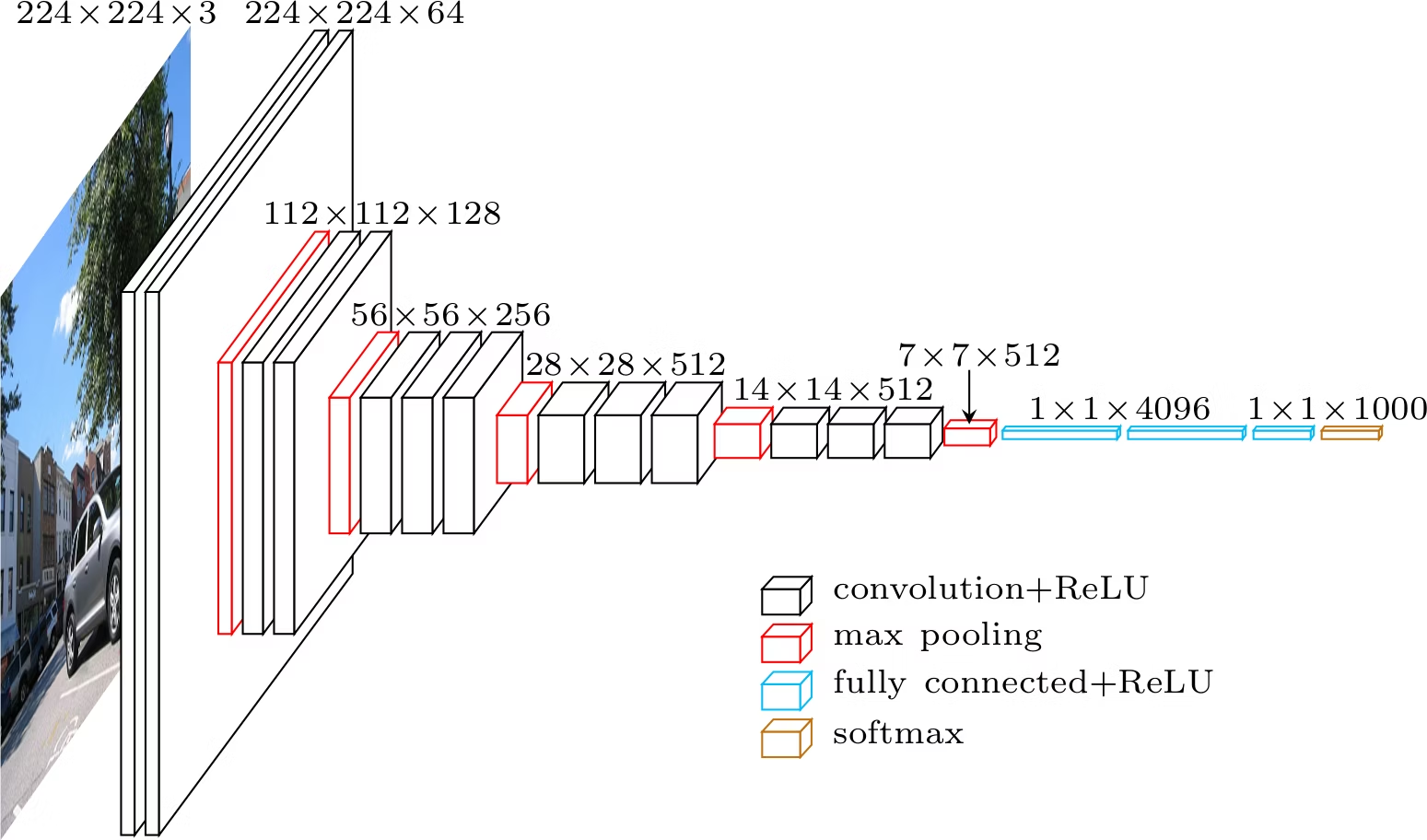
## Appendix

### Detailed Architecture Diagrams:



**Faster R-CNN Inception ResNet v2 model architecture:**





### 

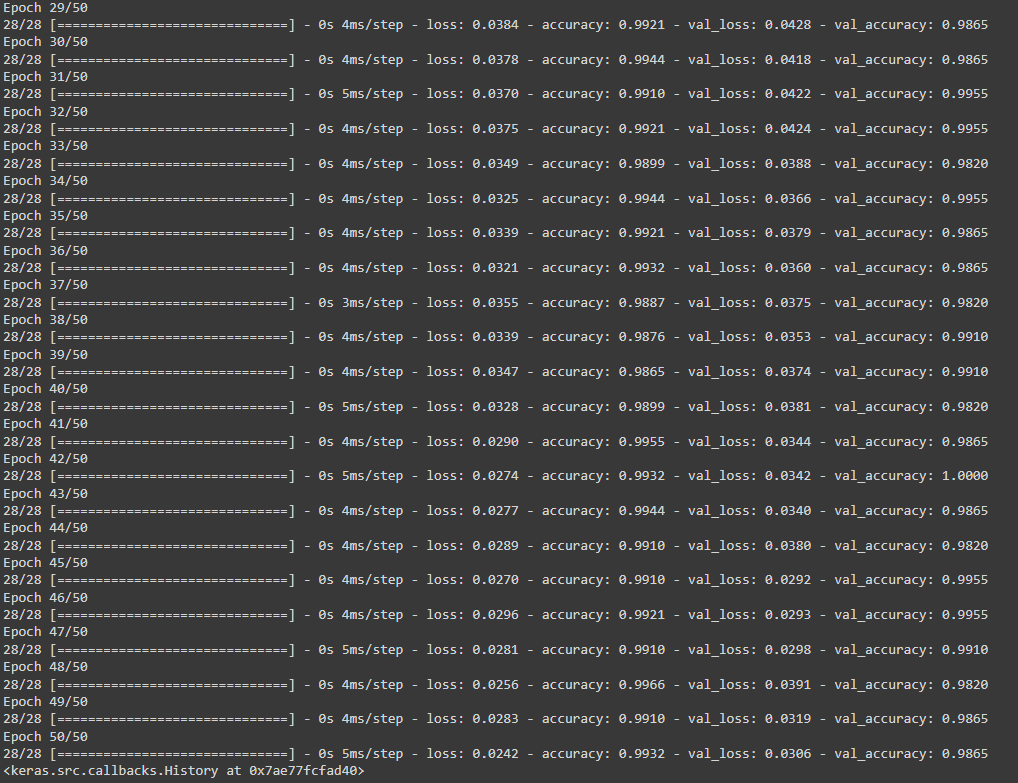
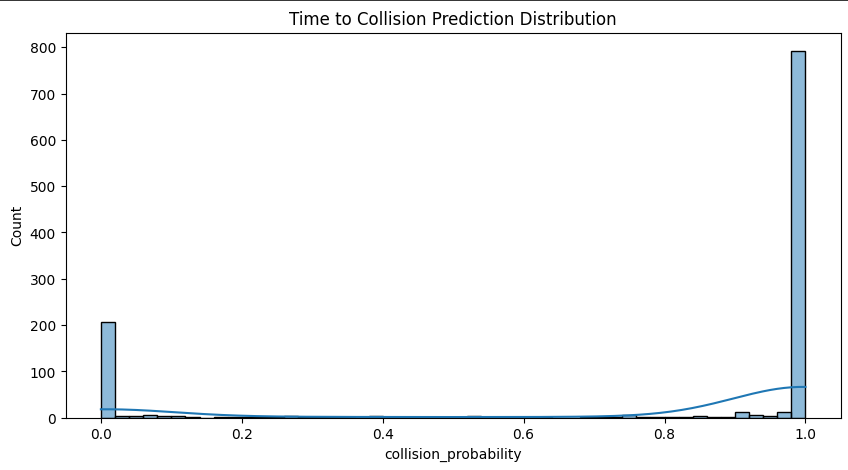
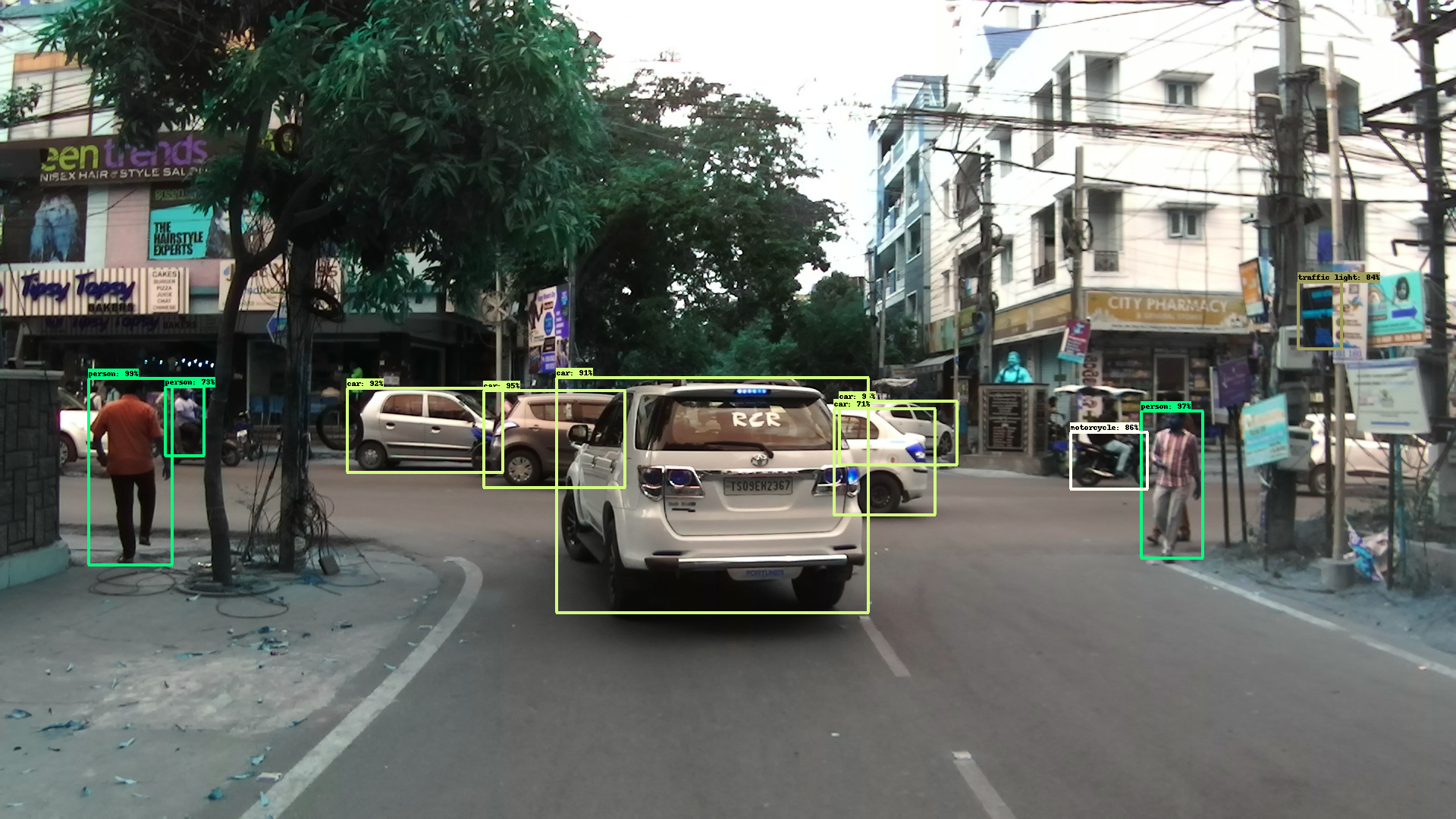
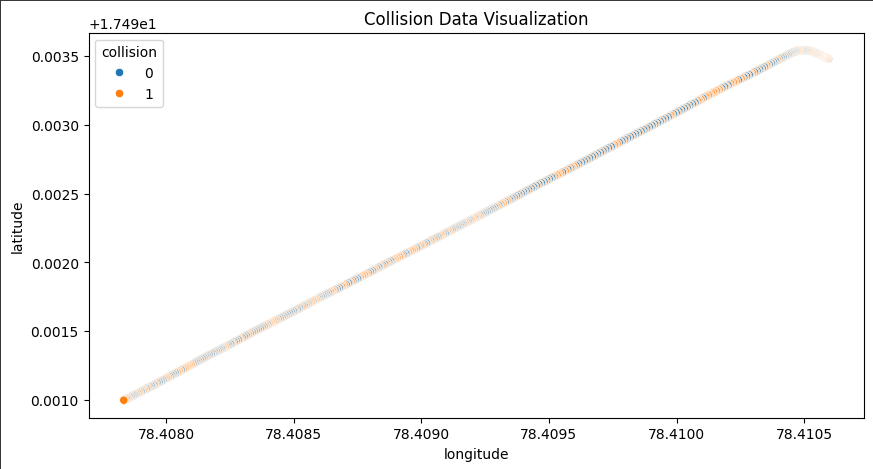
### 

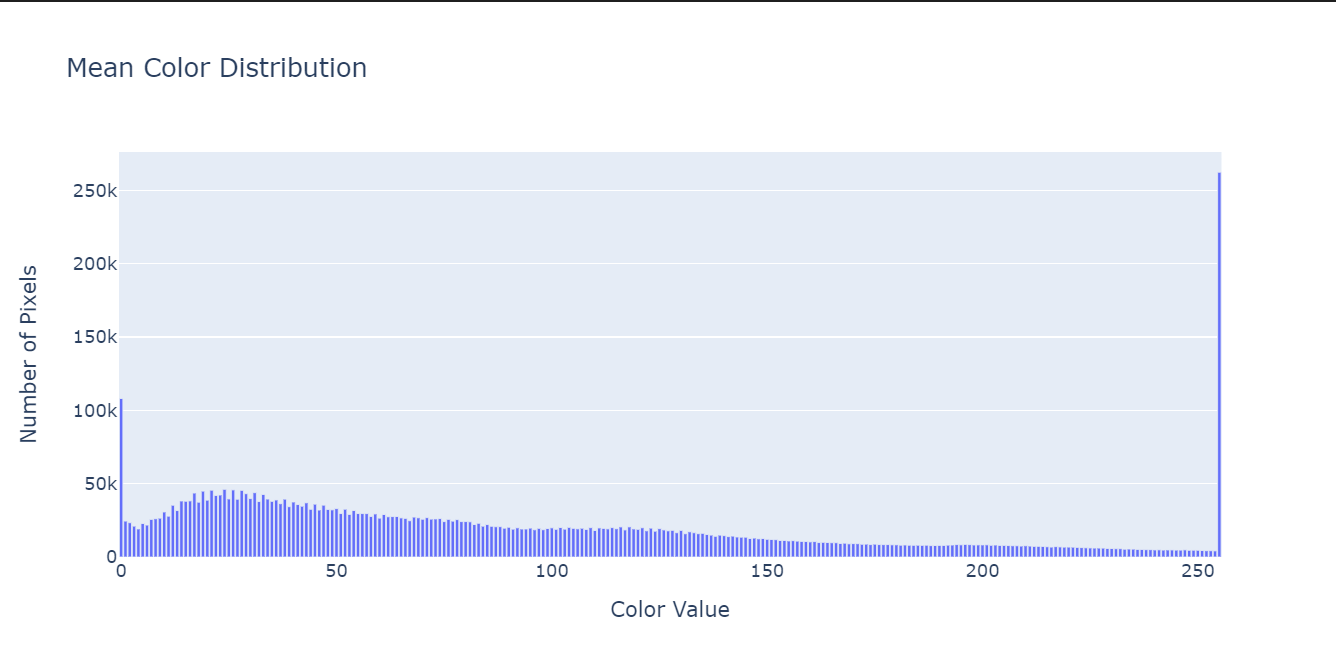
### 

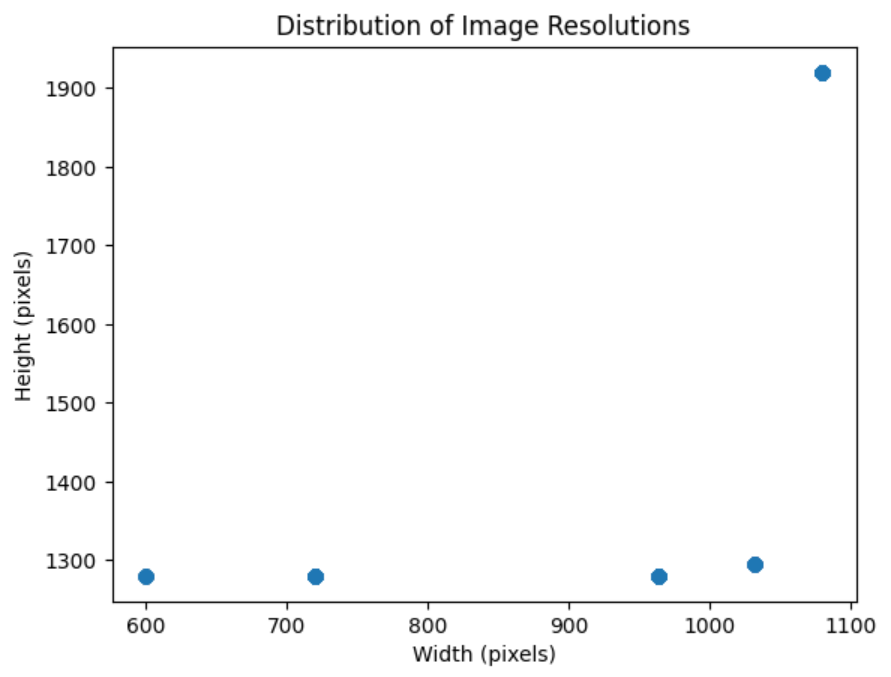
### 

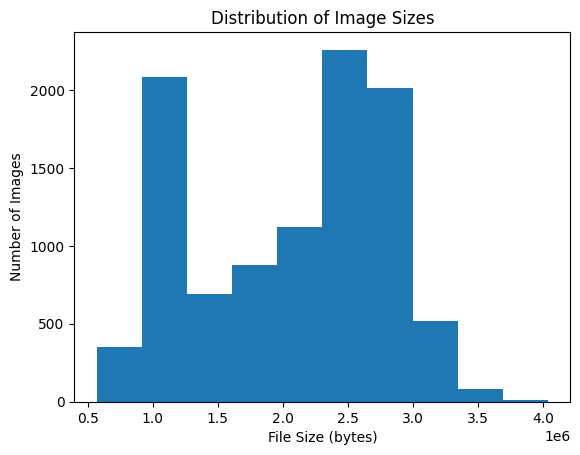
### 

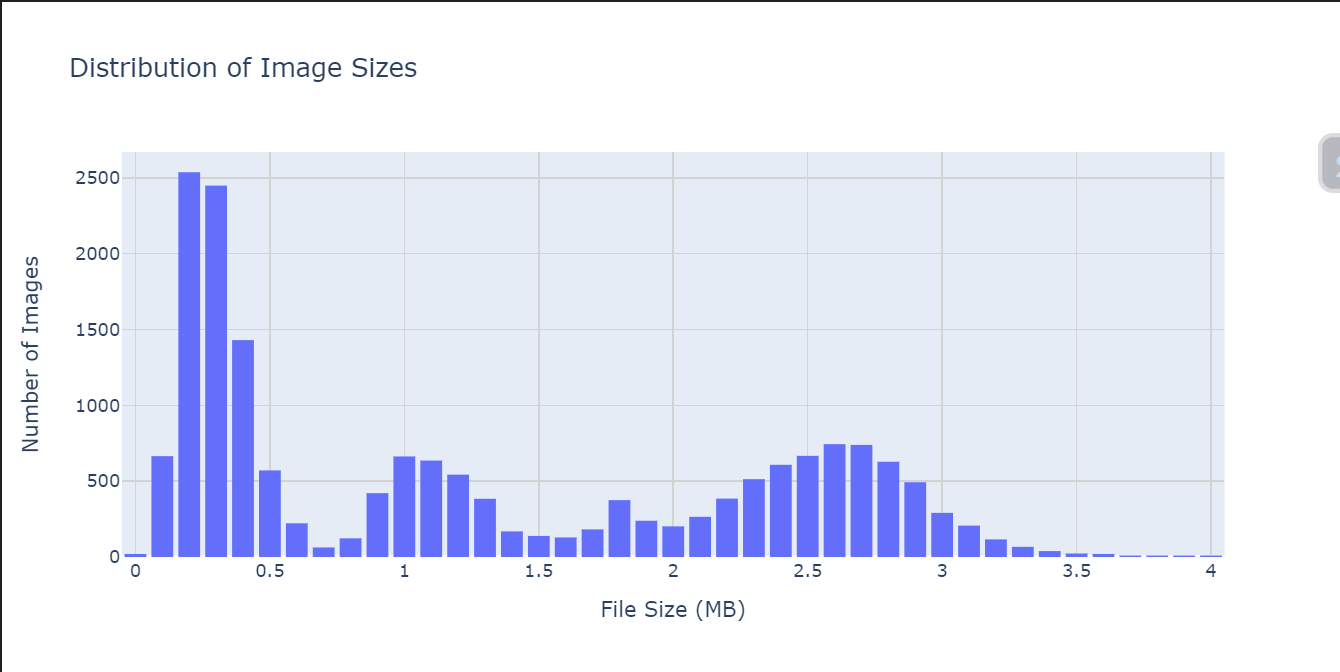
### **Experimental** Visualizations and Outputs**:**







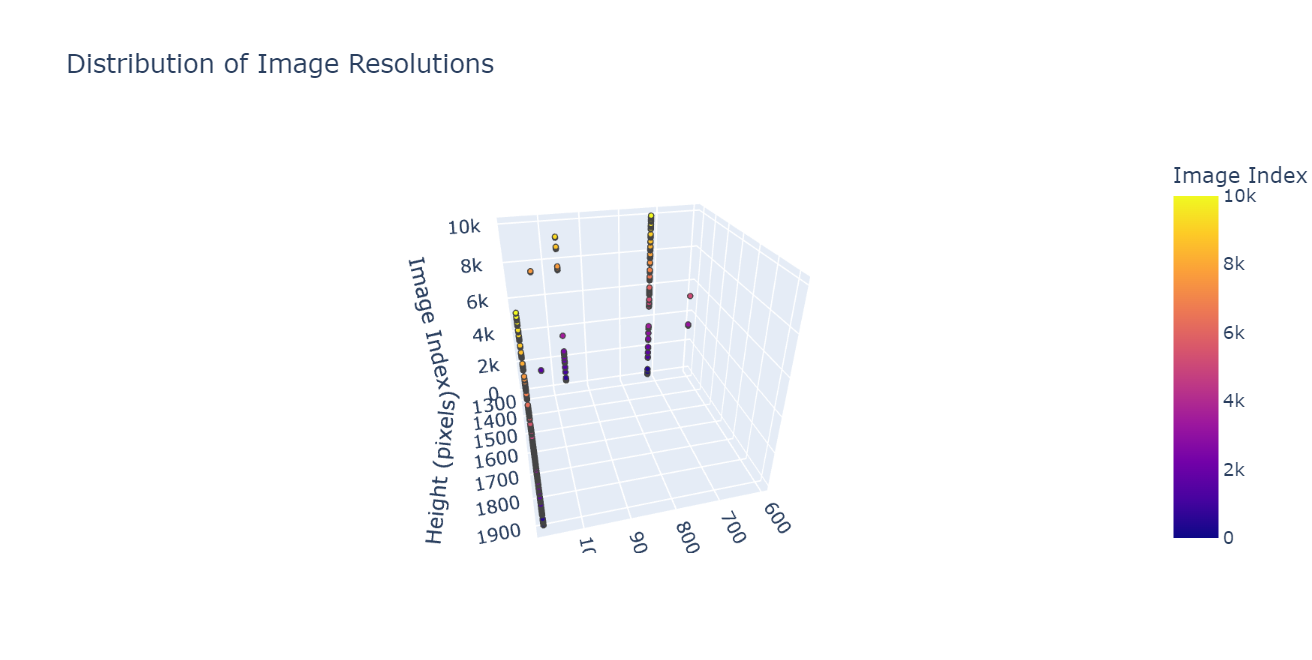
****



## 

## 

## 



## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## Additional Considerations

### **Potential Biases:**

**Data Bias:** The training data is not representative of diverse driving conditions (e.g., weather, various lighting, road types), the model may perform poorly or exhibit biases in certain scenarios.

**Sensor Bias:** Different sensor types (camera, LiDAR, radar) can have varying accuracies and limitations under different conditions. The IDD dataset heavily relies on one sensor type(LiDAR and camera) that can introduce bias.

### **Safety Criticality:**

* **Robustness to Adverse Conditions:** The model should be tested under various adverse conditions such as poor weather, heavy traffic, and different lighting conditions to ensure consistent performance.
* **Sensor Failures:** The model should be refined to gracefully handle sensor failures or degraded sensor data.
* **Redundancy:** Incorporate redundant sensors or models to increase system reliability and reduce the risk of catastrophic failures.
* **Fail-Safe Mechanisms:** Fail-safe mechanisms should be implemented to prevent the model from causing unintended consequences in case of malfunctions or errors and should also adhere to the functional safety standard ISO 26262 for automotive systems to ensure the highest level of safety.

### **Ethical Considerations:**

**Privacy Concerns:** The model may process personal data, such as vehicle location and behavior. Ensuring data privacy and security is crucial.

**Safety Concerns:** While the model aims to improve safety, incorrect or biased predictions could lead to accidents.

**Fairness and Discrimination:** The model should avoid discriminating against certain vehicle types i.e. the model should not be more likely to falsely accuse certain types of vehicles of cutting in.

**Liability:** Determining liability in case of an accident involving a vehicle equipped with such a model is a complex legal and ethical issue.

## References

* YOLO
* SSD
* RCNN
* CNN-LSTM
* Background subtraction
* Trajectory analysis
* Frame difference models
* Histogram based methods
* Optical flow models
* AD
* ADAS
* ACC
* V2V
* V2X
* AV
* LDW
* HA
* UMD-DMED
* Two stream technique
* Late fusion
* Slow-fast networks
* Discriminative based models
* Interaction aware models
* Maneuver based models
* Linear vs Non-linear transitions

Considerations taken into account for cut-in:

* Vehicle distance
* Headways
* Indicators
* Lateral velocity
* Position of surrounding vehicles
* Bounding boxes

Motivations for cut in:

* Urgency (e.g., being late)
* Impatience
* Unawareness
* Aggressive driving
* Emergencies