# MATLAB evaluations

## Map Validation

Validation of driven route’s corridor info.

Segmentation profile: Map\_Validation

Evaluation profile: Map\_Validation

Output:

* correlation plots of node point data with validator and the test file (LDM correlation plots).
* Comparison plots on the corridor coefficients (c0,c1 and c2).
* Numerical statistical comparison on node point data and coefficients

Parameters:

* Localization: no filter / rural road / highway
* Objective function:
  + retrigger cycle
* Modell fitting:
  + curve detection thresholds

Two types of measurements are expected:

1. Validator: containing map data from a Bosch vehicle measurement, and all other necessary data defined in the configpool:
   1. GPS data (Longitudinal and lateral positions, GPS time) maximum 50ms sample time
   2. Vehicle kinematics (Longitudinal speed, acceleration, yawrate, steering angle)
   3. Lane information (at least 2nd order polynomial representation of the ego lane edges: c\_0, c\_1 and c\_2, standing for distance, orientation and curvature) for both lane edges.

This measurement file shall contain “validator” in its name.

1. Test file: measurement file containing the same data as the validator. Its naming must not contain “validator” in its name. No other rule shall be followed.

If more than two files are added, the evaluation will not run, and a warning message will be thrown.

The two measurements must have enough overlap on localization to provide good results on validation.

## Node point definition

Node points are identified on different drivers and road sections.

Segmentation profile: Driver\_Model

Evaluation profile: nodePointDefinition

Output:

* Optimal node point distances for each measurements.
* Comparison of fitted trajectories to driver trajectories in terms of plots and numerical statistical indicators using global parameters.
* Global parameters sets.
* Optimization progress plots.

Parameters:

* Localization:
  + Road type (no filter / rural road / highway)
* Objective function:
  + retrigger cycle
* Optimization parameters:

Various measurements containing the following data shall be added:

1. GPS data (Longitudinal and lateral positions, GPS time) maximum 50ms sample time
2. Vehicle kinematics (Longitudinal speed, acceleration, yawrate, steering angle, driver torque)
3. Lane information (at least 2nd order polynomial representation of the ego lane edges: c\_0, c\_1 and c\_2, standing for distance, orientation and curvature) for both lane edges.

Multiple measurement files can be added.

## Driver model analysis

Driving data is analysed for Linear Driver Model (LDM) and correlations are revealed.

Segmentation profile: Driver\_Model

Evaluation profile: driverModelAnalysis

Output:

* Correlation plots for LDM
* Regression confidence factor matrix

Parameters:

* Localization:
  + Road type (no filter / rural road / highway)
* Objective function:
  + retrigger cycle

Various measurements containing the following data shall be added:

1. GPS data (Longitudinal and lateral positions, GPS time) maximum 50ms sample time
2. Vehicle kinematics (Longitudinal speed, acceleration, yawrate, steering angle, driver torque)
3. Lane information (at least 2nd order polynomial representation of the ego lane edges: c\_0, c\_1 and c\_2, standing for distance, orientation and curvature) for both lane edges.

Multiple measurement files can be added.

## Driver model learning

Driving data is analysed for Linear Driver Model (LDM) and correlations are revealed.

Segmentation profile: Driver\_Model

Evaluation profile: driverModelAnalysis

Output:

Parameters:

* Localization:
  + Road type (no filter / rural road / highway)
* Objective function:
  + retrigger cycle
  + node point parameters

Various measurements containing the following data shall be added:

1. GPS data (Longitudinal and lateral positions, GPS time) maximum 50ms sample time
2. Vehicle kinematics (Longitudinal speed, acceleration, yawrate, steering angle, driver torque)
3. Lane information (at least 2nd order polynomial representation of the ego lane edges: c\_0, c\_1 and c\_2, standing for distance, orientation and curvature) for both lane edges.

Multiple measurement files can be added.

# Python evaluations

## Basic evaluation

Evaluation of Lane Change data with a lane change approve button.

**Main Options**:

Evaluation profile: Python\_LaneChange\_basic

Vehicle configurations: VW\_Golf\_Bp\_2201\_for\_mapping, VW\_Golf\_Bp\_2211\_ES910\_CAN\_only, VW\_Golf\_Bp\_CAN\_only

**Python Evaluation Options:**

Training: False Training Step: -

Gentle Threshold/Strict Threshold: float number in s

If the prediction deviates less from the ground truth than the threshold.

The start prediction has to be in range:

[ground truth start - gentle threshold, ground truth start + strict threshold]

The end prediction has to be in range:

[ground truth end – strict threshold, ground truth end + gentle threshold]

Classification Model: Classification\_ResNet-20\_100

Regression Model: Simple\_Regression\_model

Output:

* Results csv including the start, end and direction of the Lane Change given by the kinetic labelling and the regression prediction. One csv for every measurement file.
* Evaluation matrix displaying the number of True Positives, False Positives, False Negatives and True Negatives

Expected types of measurements:

* mf4 or mat file with the following signals:

"GPS\_time", "GPS\_status", "LongPos\_abs", "LatPos\_abs" ,"c01\_left", "c01\_right", "c1", "c2", "q\_T0", "VelocityX\_ESP", "AccelerationX\_ESP", "AccelerationY\_ESP", "yawRateESP", "SteeringAngle", "objectAccelerationFront", "objectVelocityFront", "objectDistanceFront", "Left\_Index", "Right\_Index", "LaneChange\_Approved"

The measurement file is converted into a csv file and cut into short 9s sequences containing individual lane changes.

The classifier will then infer on the small sequence files.

After that the regression model will predict the start and end time of the lane change. If the classifier finds no lane changes in a measurement file, no regression prediction is performed.

## Light evaluation

Evaluation of Lane Change data without a lane change approve button

**Main Options**:

Evaluation profile: Python\_LaneChange\_light

Vehicle configurations: VW\_Golf\_Bp\_2201\_for\_mapping, VW\_Golf\_Bp\_2211\_ES910\_CAN\_only, VW\_Golf\_Bp\_CAN\_only, VW\_Golf\_DASy\_JP

**Python Evaluation Options:**

Training: False Training Step: -

Gentle Threshold/Strict Threshold: float number in s

If the prediction deviates less from the ground truth than the threshold.

The start prediction has to be in range:

[ground truth start - gentle threshold, ground truth start + strict threshold]

The end prediction has to be in range:

[ground truth end – strict threshold, ground truth end + gentle threshold]

Classification Model: Classification\_ResNet-20\_100

Regression Model: Simple\_Regression\_model

Output:

* Results csv including the start, end and direction of the Lane Change given by the kinetic labelling and the regression prediction. One csv for every measurement file.
* Evaluation matrix displaying the number of True Positives, False Positives, False Negatives and True Negatives of comparing the regression prediction to the kinetic labels

Expected measurement file:

mf4 or mat file with these measured signals:

"c01\_left", "c01\_right", "c1", "c2", "VelocityX\_ESP", "AccelerationX\_ESP", "AccelerationY\_ESP", "yawRateESP", "SteeringAngle", "objectAccelerationFront", "objectVelocityFront", "objectDistanceFront", "Left\_Index", "Right\_Index"

The measurement file is converted into a csv file and cut into short 9s sequences containing individual lane changes.

The classifier will then infer on the small sequence files.

After that the regression model will predict the start and end time of the lane change. If the classifier finds no lane changes in a measurement file, no regression prediction is performed.

## Online Evaluation

Evaluation of Lane Change data with an approve button and with the predictions of the online lane change detection.

**Main Options**:

Evaluation profile: Python\_LaneChange\_online

Vehicle Configuration: VW\_Golf\_Bp\_2211\_ES910\_CAN\_only

**Python Evaluation Options:**

Training: False Training Step: -

Gentle Threshold/Strict Threshold: float number in s

If the prediction deviates less from the ground truth than the threshold.

The start prediction has to be in range:

[ground truth start - gentle threshold, ground truth start + strict threshold]

The end prediction must be in range:

[ground truth end – strict threshold, ground truth end + gentle threshold]

Classification Model: Classification\_ResNet-20\_100

Regression Model: Simple\_Regression\_model

Output:

* Results csv including the start, end and direction of the Lane Change given by the kinetic labelling and the regression prediction. One csv for every measurement file.
* Evaluation matrix csv comparing the online prediction to the kinetic labels
* Evaluation matrix csv comparing the online prediction to the regression predictions
* ‘True Positives’ csv displaying all the online predictions that were True Positives when comparing to the kinetic label displaying the start, end and direction of the Ground truth and the prediction as well as whether the direction of the prediction is correct
* ‘True Positives’ csv displaying all the online predictions that were True Positives when comparing to the regression prediction displaying the start, end and direction of the Ground truth and the prediction as well as whether the direction of the prediction is correct

Expected measurement file:

mat file containing the following measured signals:

"c01\_left", "c01\_right", "c1", "c2", "q\_T0", "VelocityX\_ESP", "AccelerationX\_ESP", "AccelerationY\_ESP", "yawRateESP", "SteeringAngle", "objectAccelerationFront", "objectVelocityFront", "objectDistanceFront", "Left\_Index", "Right\_Index", "LaneChange\_Approved", "**laneChangeDirectionOnline**", "**laneChangeStateOnline**"

The measurement file is converted into a csv file and cut into short 9s sequences containing individual lane changes.

The classifier will then infer on the small sequence files.

After that the regression model will predict the start and end time of the lane change. If the classifier finds no lane changes in a measurement file, no regression prediction is performed.

## Python Training

### Preprocessing

Evaluation Profile: Python\_train

Vehicle Configurations: VW\_Golf\_Bp\_2211\_ES910\_CAN\_only, VW\_Golf\_Bp\_CAN\_only, VW\_Golf\_Bp\_2201\_for\_mapping,

Training: True

Training Step: preprocessing

(Other Python Evaluation Options are irrelevant here)

The datasets needed to train the classification and regression models are generated in this step. This step has to be executed before training any model.

### Classification Training

Evaluation Profile: Python\_train

Training: True

Training Step: train\_classification

Classification model: Classification\_ResNet-20\_100, class\_model1, class\_model2, class\_model3

Output: Trained classification model chosen above as an .hdf5 file in ‘models’ folder

The classification model is trained in this step. To execute this step, the datasets from the preprocessing step has to have been generated.

### Regression Training

Evaluation Profile: Python\_train

Training: True

Training Step: train\_regression

Regression model: Simple\_Regression\_model, reg\_model1, reg\_model2, reg\_model3

Output: Trained classification model chosen above as a .hdf5 file in ‘models’ folder

The regression model is trained in this step. To execute this step, the datasets from the preprocessing step has to have been generated.

### Cleanup

Evaluation Profile: Python\_train

Training: True

Training Step: cleanup

The datasets generated in the preprocessing step can be deleted here.

**Document History**

* Note: The section document history is marked as “HIDDEN text”; hence it will not be printed: For this, you have to click the check box “Hidden Text” under “Tools -> Options”, in the tab “Print”.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Document Version | Compo-nent Version | Date | Author | Description | Status |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
| Keywords |  |
| Roles |  |
| Screens |  |
| Process steps |  |
| Document type |  |