Prediction Surface Uncertainty Quantification in Object Detection Models for Autonomous Driving

F. Ozgur Catak and Tao Yue and Shaukat Ali

Simula Research Laboratory,
Department of Engineering Complex Software Systems
Fornebu, Norway
{ozgur,tao,shaukat}@simula.no

AITest 2021

https://github.com/Simula-COMPLEX/pure
https://arxiv.org/abs/2107.04991

simula

Outline

- Monte-Carlo dropout based Uncertainty Quanfitication in DL
- Uncertainty Sources and Quantification Metrics in DL
- The PURE method for quantifying the uncertainty in object detection models
- Experiments
- Future work and Conclusions

simula

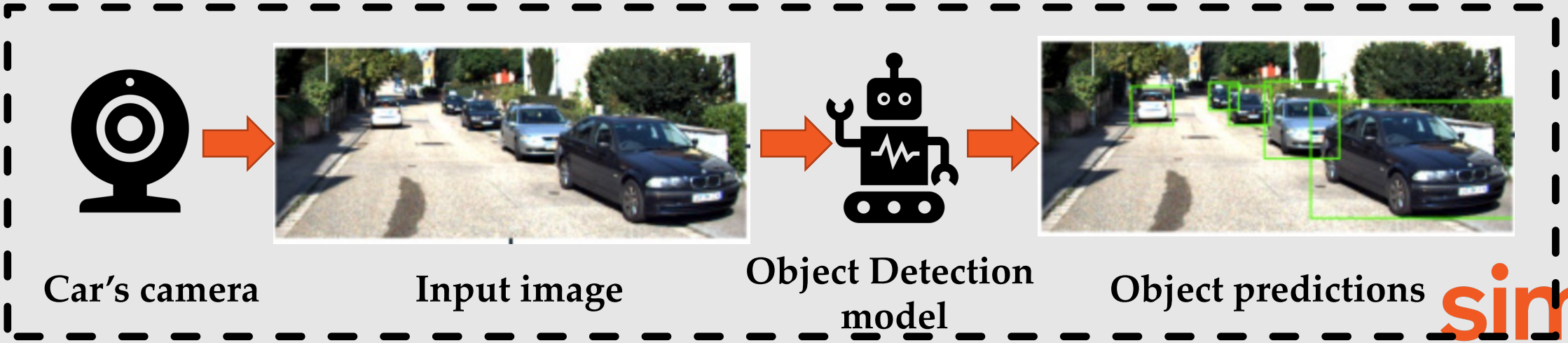
Introduction

Object detection in autonomous cars

- Generally based on Camera and Lidar inputs
- DL models are trained with these inputs
  - object recognition
  - adjusting speed

A mistake in such decision making can be damaging;

- measure the reliability of decisions
- Uncertainty in DL models



simula

Motivation: Can we trust object predictions?

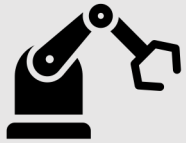
- There is always some uncertainty in DL models' predictions.
- No DL model is perfect
  - Model architecture is not the best one for the problem
  - Not enough training data for all cases in the domain
- With their use in CPSs, the DL models' faults pose a threat to human life.

simula

Uncertainty in DL

- **Uncertainty**: could potentially lead to unreliable (e.g., unsafe) behaviors of CPSs,
  - if such uncertainty is not properly dealt with.

- **DNNs**: black box models (multilayered nonlinear structures)
  - non-transparent
  - Predictions not identifiable by humans

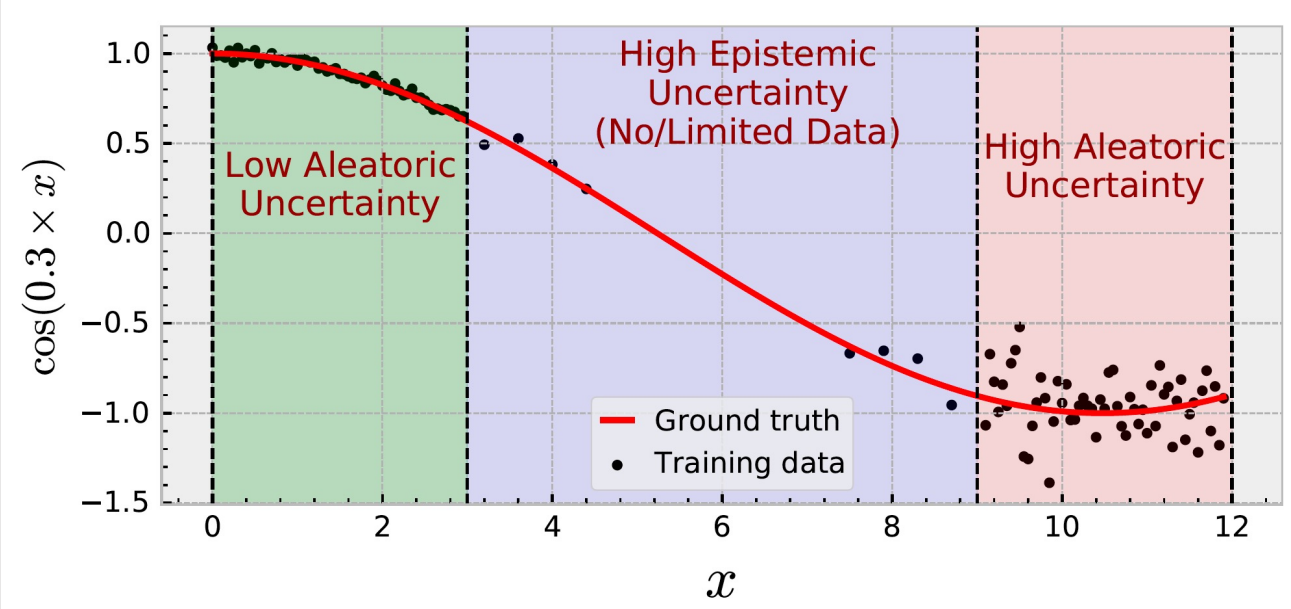


- **CPSs**: black-box DL models have been used to make critical predictions

simula

Preliminaries: Uncertainty Sources

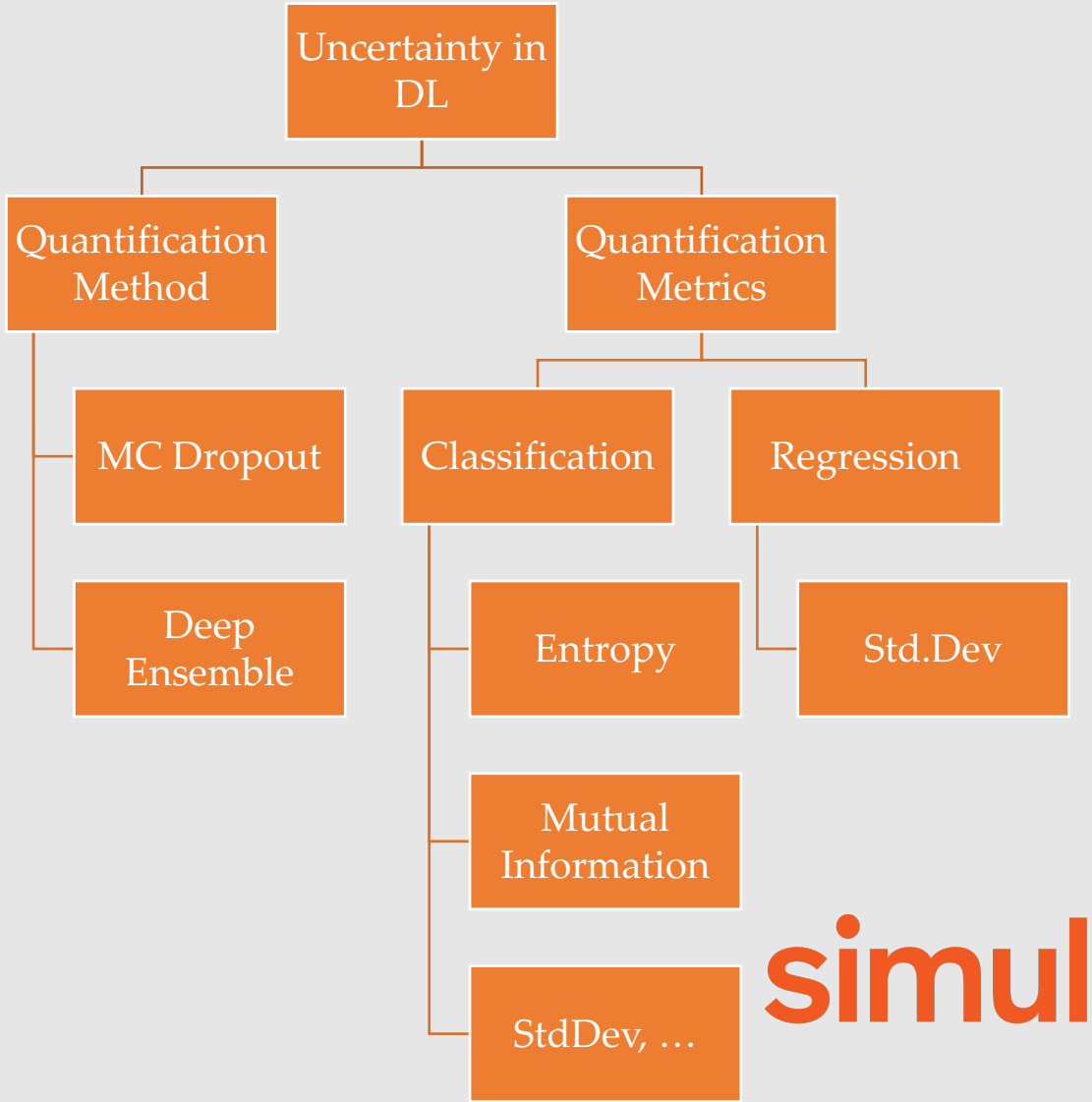
- **Epistemic Uncertainty**
  - Uncertainty in models (not in data)
  - "Epistemic": Greek "episteme" = knowledge
  - **<u>Reducible: more data helps</u>**
  - There are two types Epistemic uncertainty
    - Model Uncertainty
      - Neural network model's neuron weights are not optimized well for the domain
    - Approximation uncertainty
      - Model structure (# of layers, activation functions (ReLU, Tanh, Sigmoid etc), optimizer functions (SGD, RMSProp, Adam, Adamax etc)
- **Aleatoric Uncertainty**
  - because of the noise input dataset.
  - "**Aleatoric**": Latin "aleator" = dice players
  - Noise in the training data
  - **Stochastic**, **irreducible** in data (noise)
    - **More data doesn't help**
  - For instance; noise sensor readings for a CPS application



simula

Uncertainty Quantification in DL

- Deep Learning (DL) is increasingly used in autonomous driving.
- **Current research**: improving accuracy of the DL models
- **Missing part**: Uncertainty quantification for the object detection models



simula

MC dropout Predictions

Important parameters:
- **T**: the number of predictions
- **p**: dropout ratio

$$\begin{bmatrix} y_{11} & y_{21} & y_{31} & y_{41} & \vdots & y_{T1} \\ y_{12} & y_{22} & y_{32} & y_{42} & \vdots & y_{T2} \\ y_{13} & y_{23} & y_{33} & y_{43} & \vdots & y_{T3} \end{bmatrix}$$

Softmax output vectors for each prediction (i.e. 3 classes, T predictions)

simula

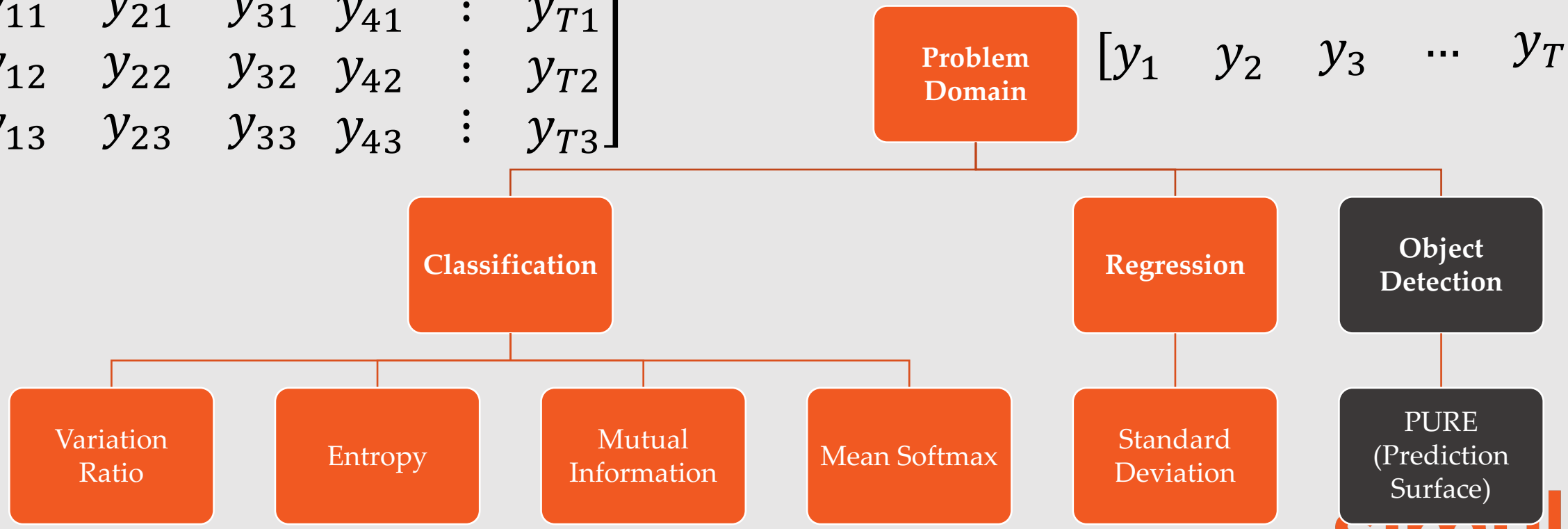
Uncertainty Quantification Metrics

Classification output

$$\begin{bmatrix} y_{11} & y_{21} & y_{31} & y_{41} & \vdots & y_{T1} \\ y_{12} & y_{22} & y_{32} & y_{42} & \vdots & y_{T2} \\ y_{13} & y_{23} & y_{33} & y_{43} & \vdots & y_{T3} \end{bmatrix}$$

Regression output

$$[y_1 \quad y_2 \quad y_3 \quad \cdots \quad y_T]$$



simula

Introduction

Uncertainty quantifiction metrics[1]

- **Classification**: VariationRatio, PredictiveEntropy, MutualInformation, MeanSoftmax
- **Regression**: StandardDeviation
- **Object Detection**: ?

PURE (Prediction sURface uncErtainty)

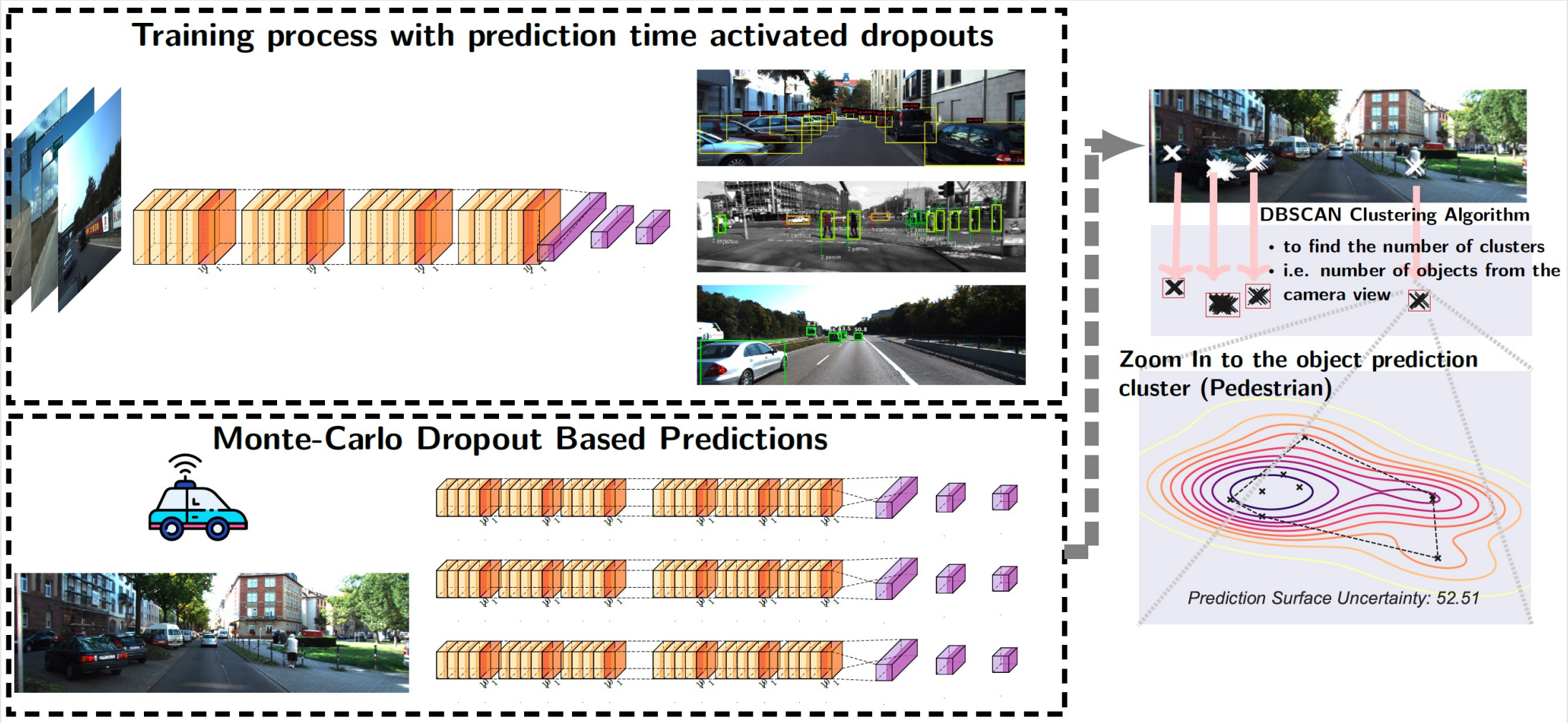
Experiments

- **Models**: YoLo, SSD300, SSD512
- **Datasets**: KITTI, Stanford Cars, Berkeley DeepDrive, NEXET

simula

1. https://uncertainty-wizard.readthedocs.io/en/latest/user_guide_quantifiers.html

The PURE Method



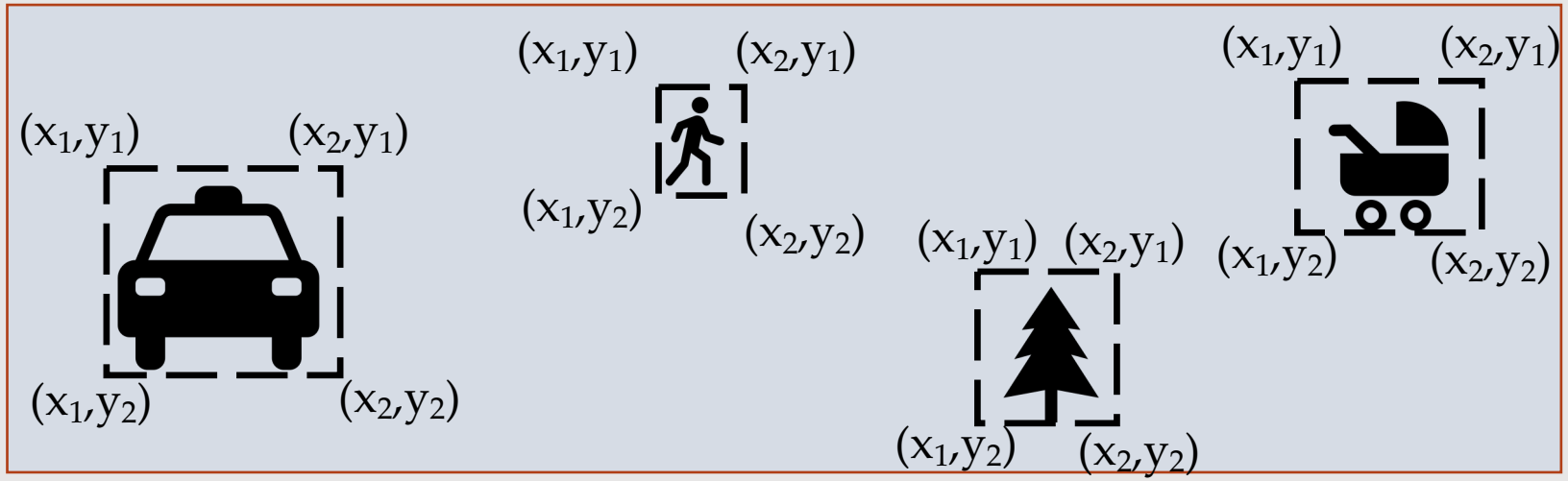
GitHub: https://github.com/Simula-COMPLEX/pure

simula

Problem Formulation

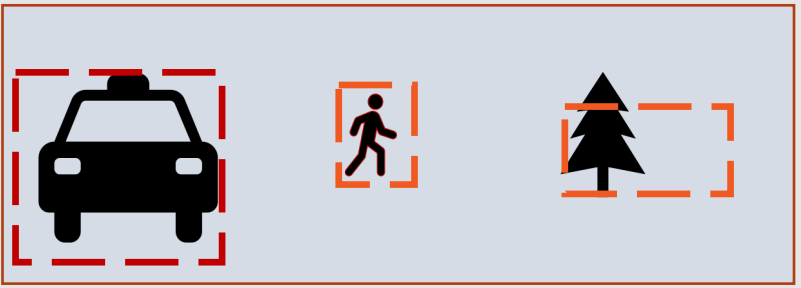
- Object detection task: detect all objects from the camera scene

- Problems:
  - the number of objects
  - Objects sizes

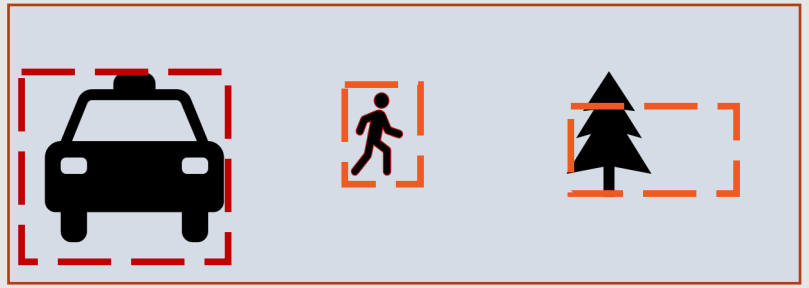


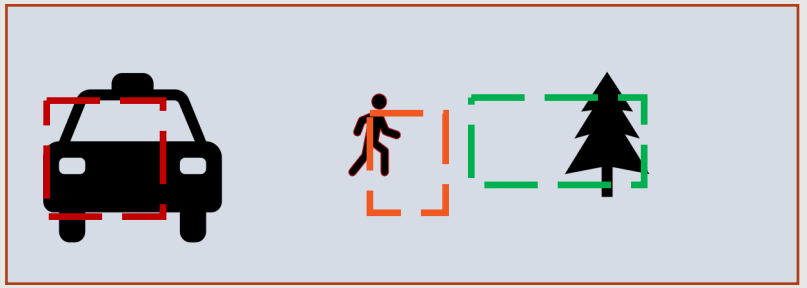
simula

The PURE Method

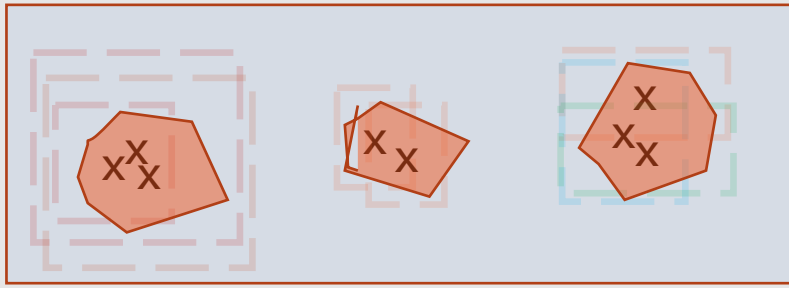


MC dropout predictions with T=3



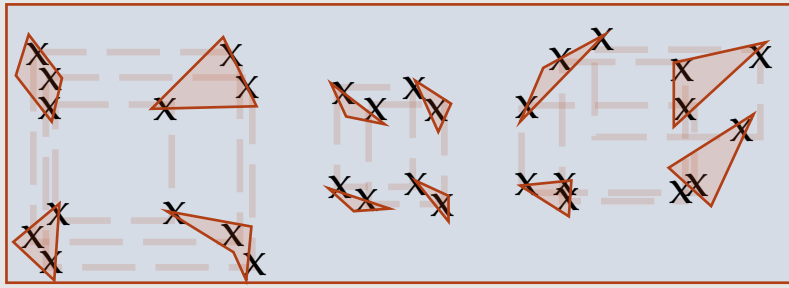


DBScan based clustering algorithm



Prediction centers

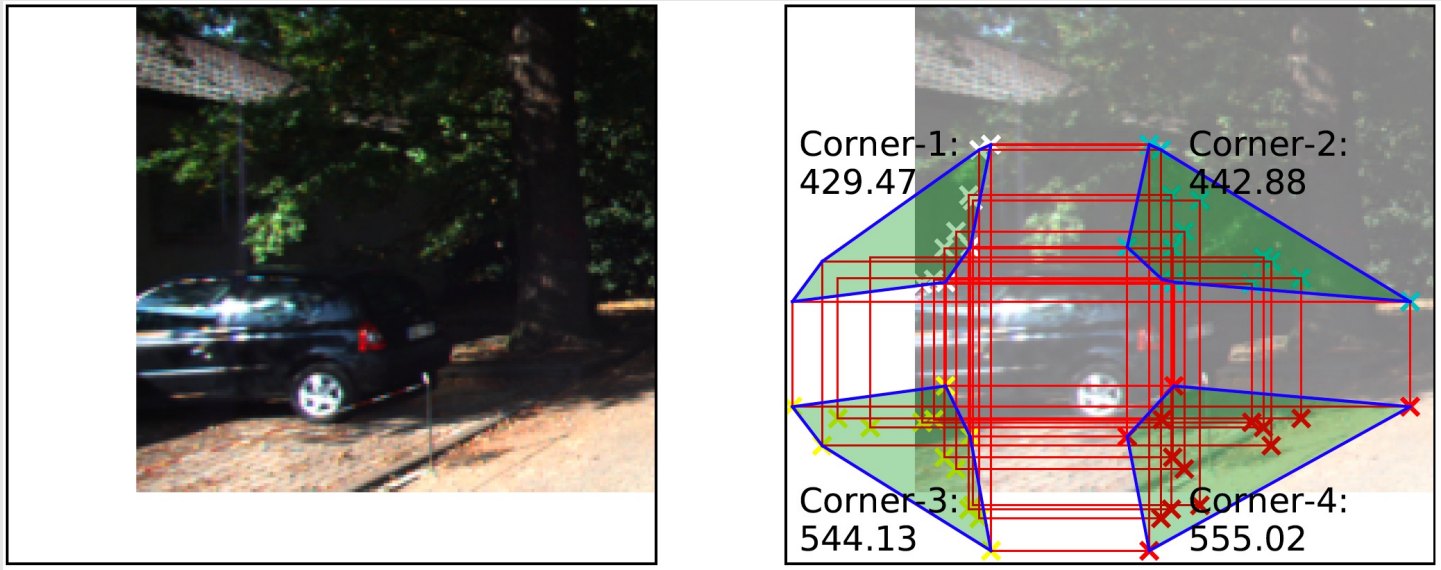
Corners of the each predictions

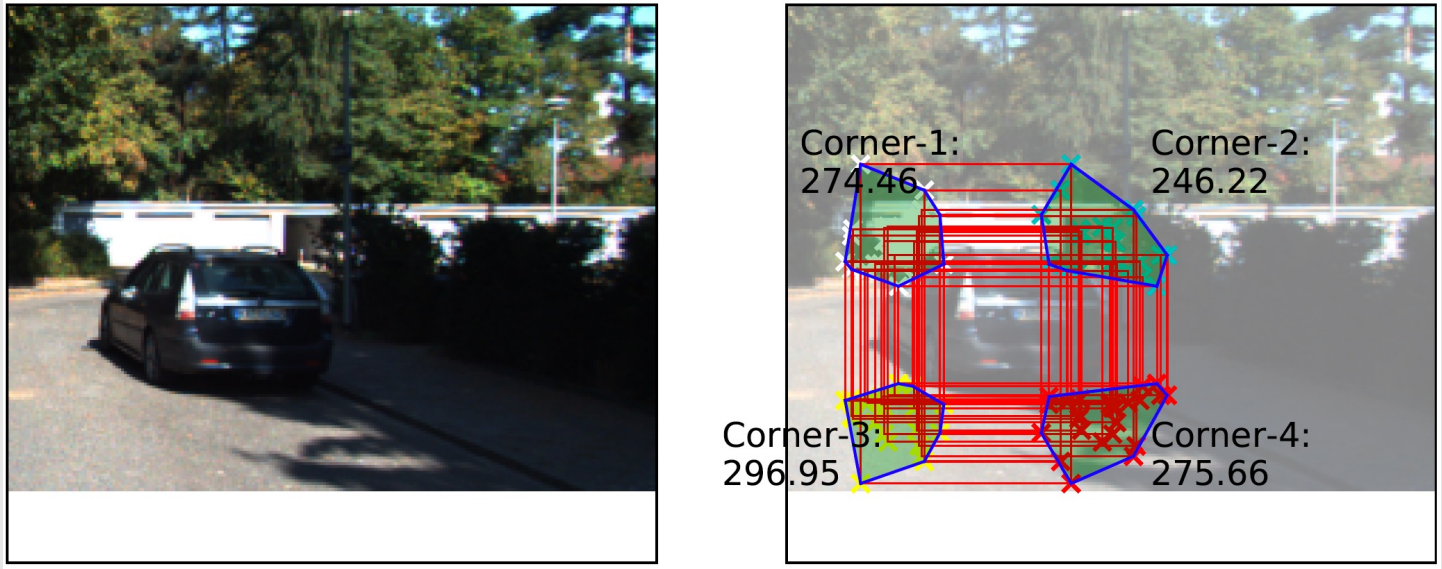


Convex-Hull based are calculation

simula

The PURE Method





simula

The PURE Method

Algorithm 1: MC dropout based object predictions

Input: $X \in \mathbb{R}^{m \times n}$: camera view
h: Object detection model
T: MC dropout size
Output: Predictions *center_lists*

$center\_lists \leftarrow \emptyset$
/* T predictions using the object detection model h */
foreach $t \in T$ **do**
  /* Predict objects with surrounding boxes from image X */
  $VBoxes \leftarrow h(X)$
  foreach $box \in VBoxes$ **do**
    /* Get the corner locations of each surrounding box */
    $x_1, y_1 = box.xmin, box.ymin,$
    $x_2, y_2 = box.xmax, box.ymax$
    /* Find the width and height of the predicted object */
    $width, height = x_2 - x_1, y_2 - y_1$
    $x_{center}, y_{center} \leftarrow x_1 \frac{width}{2}, y_1 + \frac{height}{2}$
    $center\_lists \leftarrow UpdateArchive(x_{center}, y_{center})$
return *center_lists*

Algorithm 2: DBSCAN clustering algorithm based clustering of predictions and uncertainty quantification

Input: Predictions *center_lists*,
ϵ: radius of DBSCAN
Output: Uncertainty of predictions, $\mathcal{U}$

$area \leftarrow \emptyset$
/* Predictions are completed. Find object clusters from the $center_list$ */
$Clusters \leftarrow DBSCAN(center\_list, \epsilon)$
foreach *cluster* $c \in Clusters$ **do**
  /* Calculate the area of each corner of the predictions in cluster c */
  area $x_1 \leftarrow ConvexHull(center\_lists[c].x_1)$
  area $y_1 \leftarrow ConvexHull(center\_lists[c].y_1)$
  area $x_2 \leftarrow ConvexHull(center\_lists[c].x_2)$
  area $y_2 \leftarrow ConvexHull(center\_lists[c].y_2)$
  /* Calculate the cluster's (i.e. the single object's) uncertainty */
  $area \leftarrow$ UpdateArchive(Mean(area x_1, area y_1, area x_2, area y_2))
/* Average uncertainty of all detected objects from all T predictions */
$\mathcal{U} \leftarrow Mean(area)$
/* Return calculated uncertainty $\mathcal{U}$ for the predictions from the image X */
return $\mathcal{U}$

Research Questions

RQ1: Is the prediction surface an effective uncertainty quantification method for object detection predictions? This RQ assesses the effectiveness of PURE.

RQ2: Is there any correlation between prediction surface uncertainty and the object detection performance of DNN models? This RQ tests the hypothesis that the prediction performance of a DNN model decreases with the increase in the uncertainty in images used for object detection.

simula

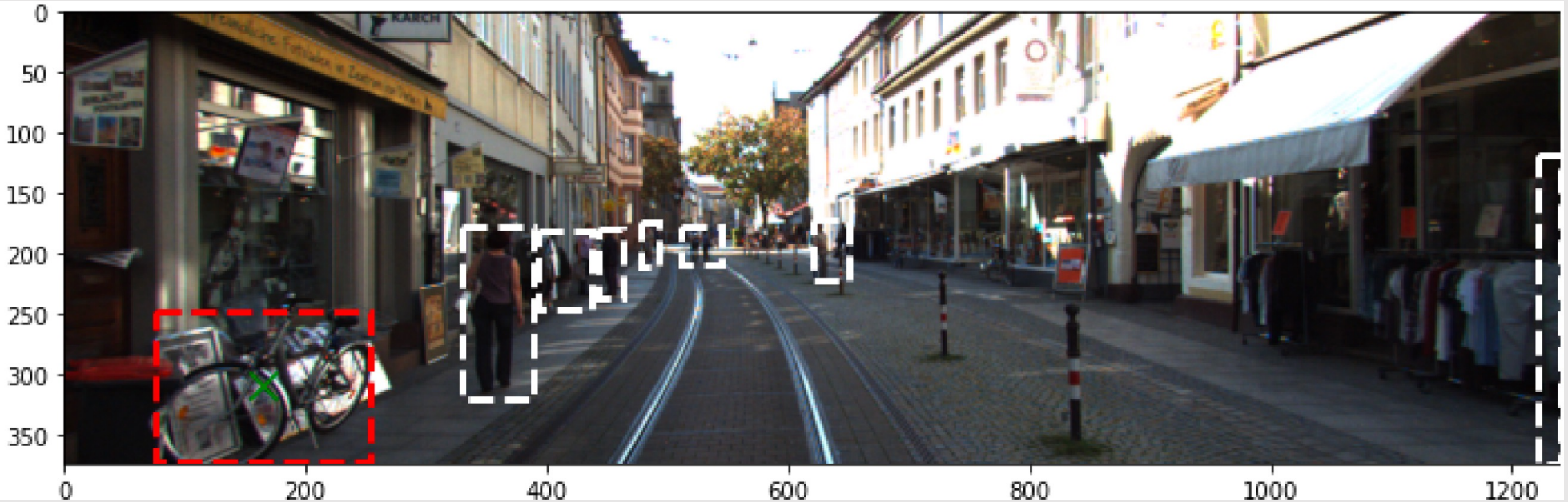
Models & Datasets

MC dropout compatible models:

- SSD300
- SSd512
- YoLo

Tensorflow versions of Python code of all the model and added the dropout layers using Tensorflow.

Dataset	Number of images	Resolution
KITTI	7,481	1224×370
Stanford Cars	8,144	300×200
Berkeley DeepDrive	50,004	1280×720
NEXET	44,527	1280×720



Illustrating YoLo's object detection predictions for KITTI.

simula

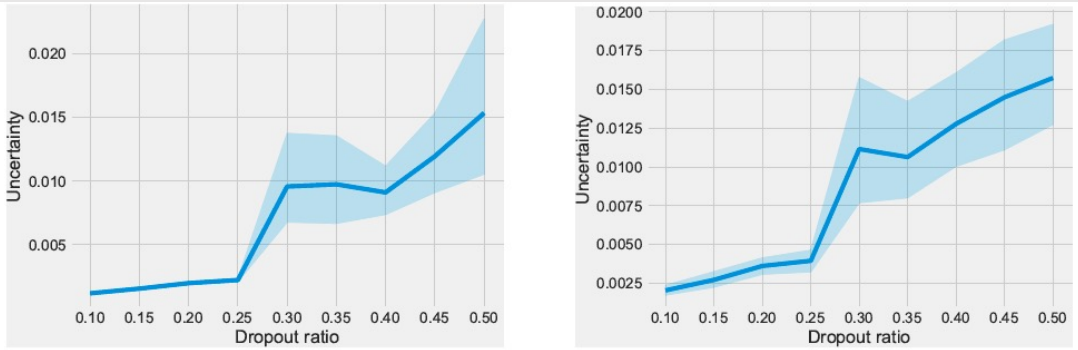
Results for RQ1

TABLE II: Object detection prediction results of all the models for all the datasets

Model	Dataset	Avg. IoU	Precision	Recall	F_1
SSD300	KITTI	0.5506	0.9065	0.2192	0.3558
	Berkeley	0.6600	0.4881	0.5000	0.4940
	NEXET	0.8483	0.6111	0.3548	0.4489
	Stanford	0.8984	0.637	1.0000	0.7783
SSD512	KITTI	0.7313	0.8437	0.3576	0.5023
	Berkeley	0.7053	0.3694	0.5409	0.4390
	NEXET	0.8341	0.6875	0.4583	0.5500
	Stanford	0.9469	0.6608	1.000	0.7957
YoLo	KITTI	0.4453	0.8916	0.1946	0.3195
	Berkeley	0.5968	0.2115	0.1182	0.1517
	NEXET	0.6378	0.7037	0.0902	0.1600
	Stanford	0.8183	0.7931	0.7931	0.7931

simula

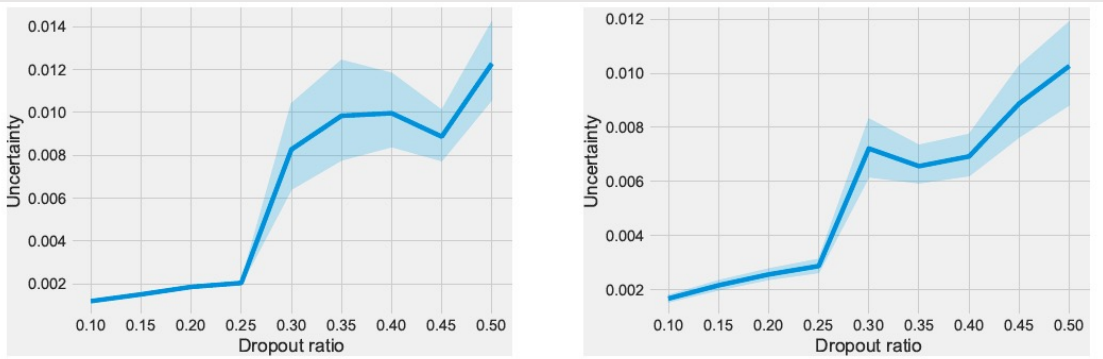
Results for RQ1



(a) Stanford (b) Berkeley

(c) NEXET (d) KITTI

Fig. 6: SSD512 model uncertainty changes with different p.



(a) Stanford cars (b) Berkeley DeepDrive

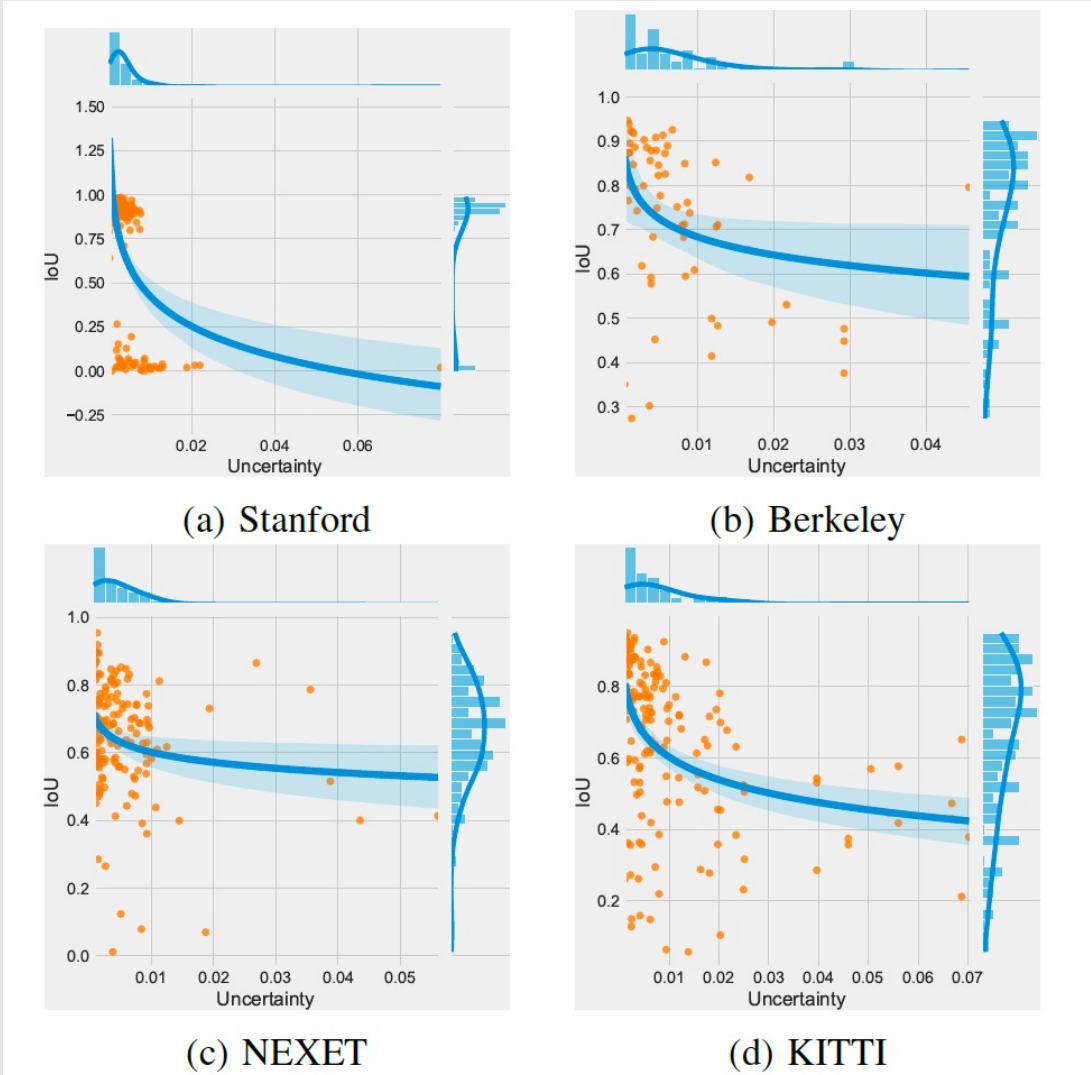
(c) NEXET (d) KITTI

Fig. 7: SSD300 model uncertainty changes with different p.

- if an object detection model has a high dropout ratio (p > 0.3), then the resulting uncertainty value becomes higher.
- a small value p should be chosen.

simula

Results for RQ2



(a) Stanford (b) Berkeley

(c) NEXET (d) KITTI

Fig. 9: SSD512 performance with different uncertainty values.

simula

Results for RQ2

		YoLo		SSD300		SSD512	
	p	r	p-val	r	p-val	r	p-val
KITTI	0.10	-0.2274	4.53e-04	-0.2141	1.86e-04	-0.3116	2.83e-05
	0.15	-0.2279	4.79e-04	-0.3340	3.34e-09	-0.3040	1.63e-05
	0.20	-0.1324	5.36e-02	-0.2710	1.54e-06	-0.2573	2.02e-04
	0.25	-0.3241	5.97e-07	-0.2726	1.21e-07	-0.3848	3.93e-07
	0.30	-0.2483	4.48e-04	-0.2876	2.23e-07	-0.3399	1.09e-06
	0.35	-0.2089	2.01e-03	-0.2579	6.21e-06	-0.3190	5.21e-05
	0.40	-0.2818	4.05e-05	-0.2329	4.24e-05	-0.2310	3.77e-04
	0.45	-0.1650	2.28e-02	-0.4131	1.53e-14	-0.2127	4.82e-03
	0.50	-0.2804	1.93e-04	-0.2256	1.18e-04	-0.4412	2.07e-10
NEXET	0.10	-0.3965	1.92e-06	-0.1970	6.42e-03	-0.1693	4.62e-02
	0.15	-0.2068	3.34e-02	-0.1641	2.18e-02	-0.2957	8.97e-04
	0.20	-0.2943	1.55e-03	-0.3278	1.27e-05	-0.2876	5.43e-04
	0.25	-0.2012	1.92e-02	-0.2302	1.35e-03	-0.2066	1.23e-02
	0.30	-0.1897	3.32e-02	-0.2858	7.00e-05	-0.1667	4.58e-02
	0.35	-0.2571	1.62e-02	-0.1581	2.53e-02	-0.2428	2.48e-03
	0.40	-0.2989	2.93e-03	-0.2438	9.70e-04	-0.3189	1.93e-04
	0.45	-0.3710	4.13e-03	-0.2039	3.76e-03	-0.3289	1.40e-04
	0.50	-0.2788	4.11e-02	-0.2121	8.47e-03	-0.2725	1.03e-03

Berkeley	0.10	-0.3743	2.25e-07	-0.3778	2.72e-08	-0.4940	6.46e-07
	0.15	-0.3265	1.49e-04	-0.4098	6.29e-09	-0.5528	1.20e-05
	0.20	-0.3665	9.57e-05	-0.4394	7.52e-11	-0.6793	1.90e-10
	0.25	-0.4161	1.11e-05	-0.3841	4.49e-07	-0.3670	3.34e-03
	0.30	-0.4247	4.11e-05	-0.4356	1.68e-09	-0.4724	4.75e-05
	0.35	-0.4711	1.98e-05	-0.3156	2.91e-05	-0.3362	9.86e-03
	0.40	-0.1481	5.99e-02	-0.3293	4.70e-06	-0.5052	1.10e-06
	0.45	-0.3007	9.73e-03	-0.4091	7.76e-08	-0.6211	6.82e-06
	0.50	-0.3564	6.42e-05	-0.2122	1.00e-02	-0.2589	1.12e-02
Stanford	0.10	-0.5002	1.160e-07	-0.2807	6.77e-12	-0.4197	1.02e-10
	0.15	-0.6169	2.52e-10	-0.2547	5.52e-10	-0.5766	8.31e-21
	0.20	-0.7152	1.02e-14	-0.3400	4.40e-17	-0.4325	2.11e-11
	0.25	-0.5619	2.19e-08	-0.4810	1.22e-34	-0.4377	1.58e-11
	0.30	-0.7161	5.66e-14	-0.3713	3.83e-20	-0.3506	9.84e-08
	0.35	-0.7816	9.77e-20	-0.3580	9.54e-19	-0.3909	2.25e-09
	0.40	-0.7586	3.04e-18	-0.4667	1.89e-32	-0.5171	2.62e-16
	0.45	-0.5765	4.13e-09	-0.4606	1.73e-31	-0.4123	2.33e-10
	0.50	-0.7769	1.52e-17	-0.4475	1.03e-29	-0.2846	1.97e-05

simula

CONCLUSION AND FUTURE WORKS

PURE calculates the convex hull area of the predicted object localization generated by the MC dropout approach.

Based on our experimental results PURE is an effective to calculate the object detection models' epistemic uncertainty.

Future works:

We will apply the DNN testing metrics, e.g., neuron coverage, and neuron boundary coverage to assess PURE's effectiveness.

simula