2021 embedded vision summit
VIRTUAL | MAY 25-28

Robust Object Detection Under Dataset Shifts

Dr. Partha Maji
Arm Machine Learning Research Lab

arm

Outline

- What are the challenges with object detection (OD) in real life?
- What's wrong with the current approach?
- How do we improve robustness in OD?
- Introducing Stochastic-YOLO OD model
- How do we simulate dataset shift and validate it?
- Key takeaways and guidelines

Challenges with Object Detection (OD) in Real Life

- In real-life predicting bounding-boxes accurately are difficult due to **dataset shift** – occlusions, lighting condition, camera imperfections etc.
- In OD tasks **"spatial quality"** is very important in addition to **"label quality"**

Figure: 1

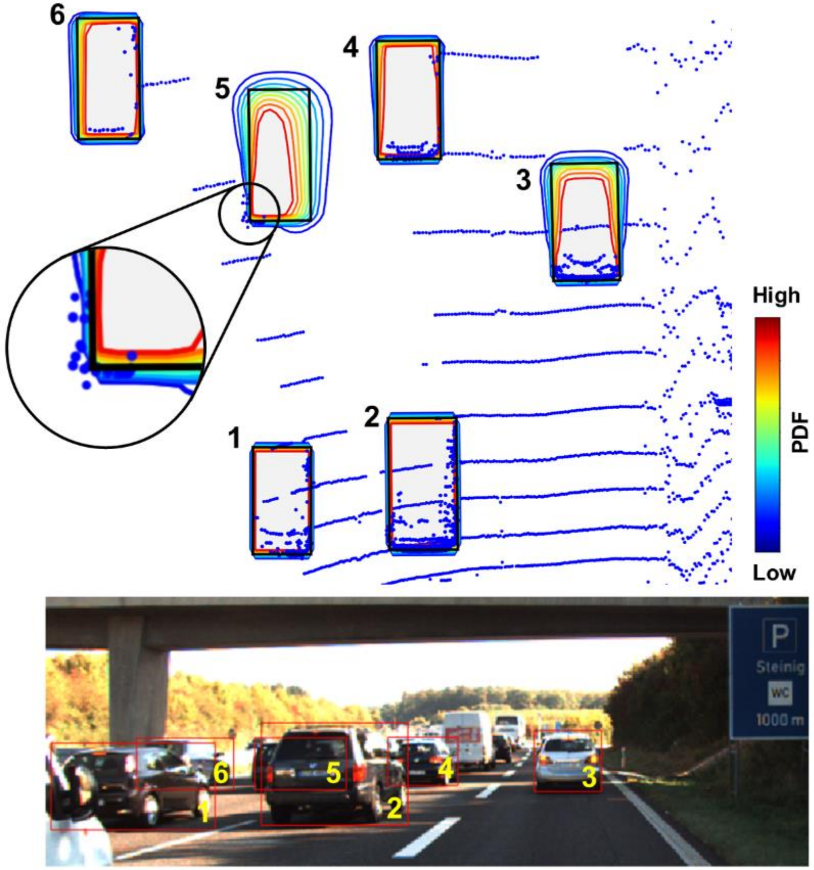


Figure: 2

Challenges with Object Detection (OD) in Real Life

- IoU metric could often be misrepresenting – it does capture **what** very well but not **where**
- In widely used model such as YOLO often score very low in spatial quality
- **pPDQ** – probabilistic detection quality is a new metric that captures both **what(Q_L)** and **where(Q_S)** – proposed by David Hall et al.

$$\mathrm{pPDQ}(\mathcal{G}_i^f, \mathcal{D}_j^f) = \sqrt{Q_S(\mathcal{G}_i^f, \mathcal{D}_j^f) \cdot Q_L(\mathcal{G}_i^f, \mathcal{D}_j^f)}$$

Example of a poor prediction that uses **IoU** score

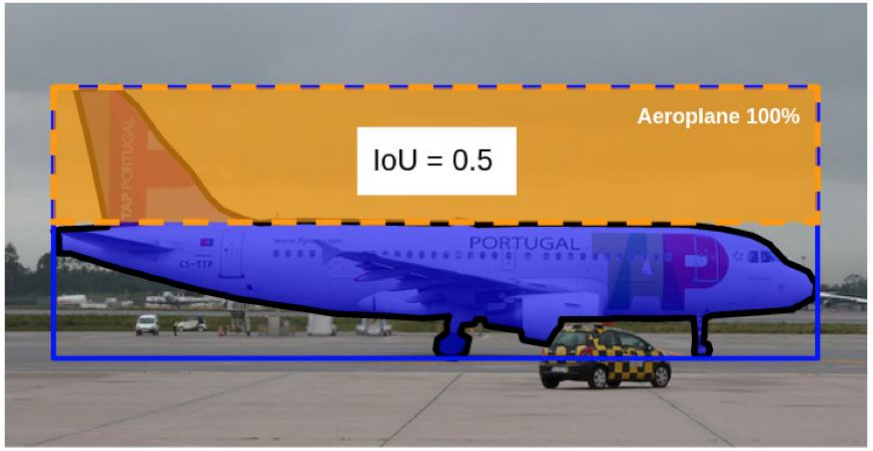


Figure: 3

Source: Hall et al., Probabilistic Object Detection: Definition and Evaluation, WACV 2020

How Do We Improve Robustness in OD Task?

- We need a way to capture ambiguity in bounding box predictions
- One option is to use multiple OD models to draw a number of proposals (e.g. Deep Ensemble)
- Is there a way to do it efficiently on hardware?
- We used Monte Carlo Dropout during inference to draw multiple proposals

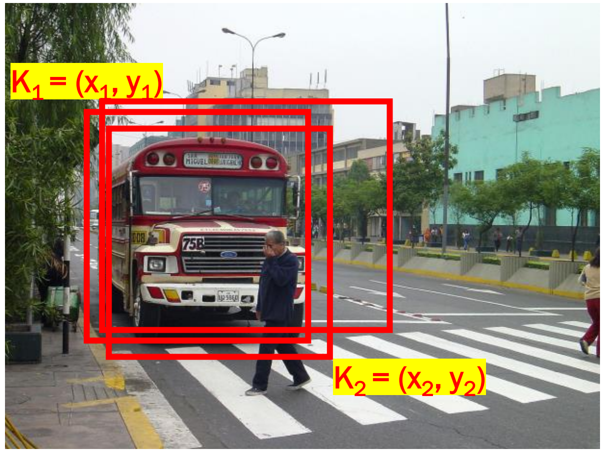
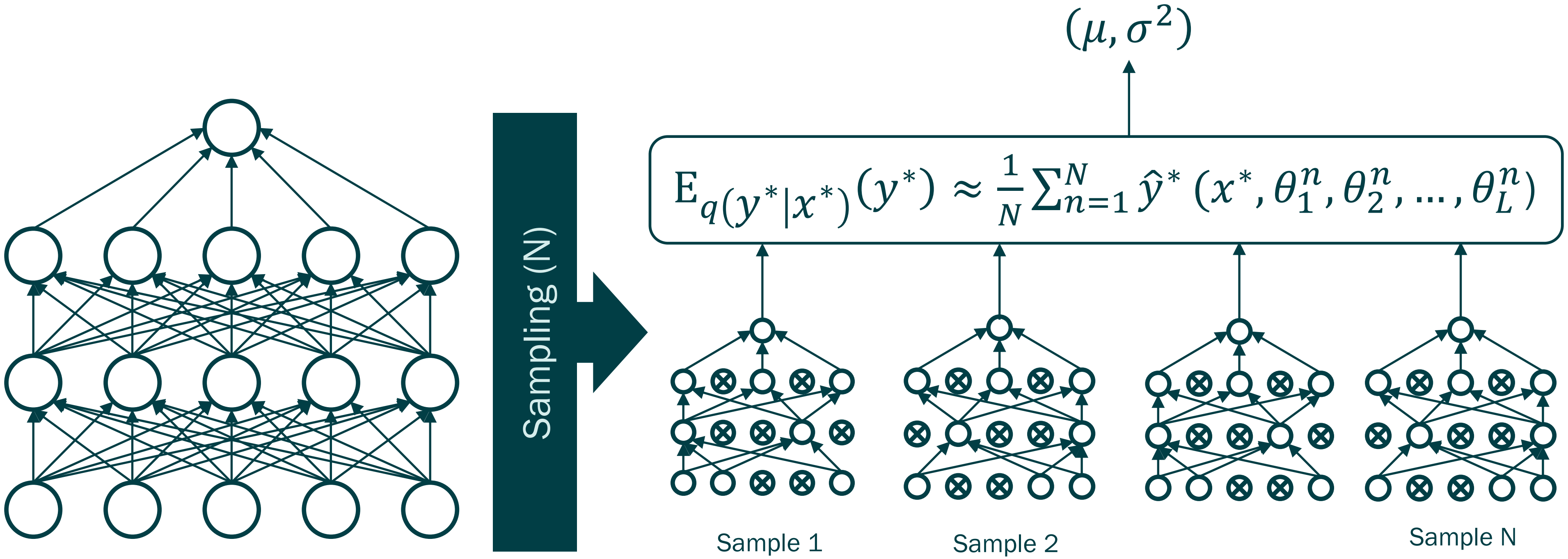


Figure: 4

$$K_{11} = \begin{bmatrix} \sigma_{xx}^1 & \sigma_{xy}^1 \\ \sigma_{yx}^1 & \sigma_{yy}^1 \end{bmatrix} \qquad K_{22} = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 \end{bmatrix}$$

Introduction to Monte Carlo Dropout based Inference

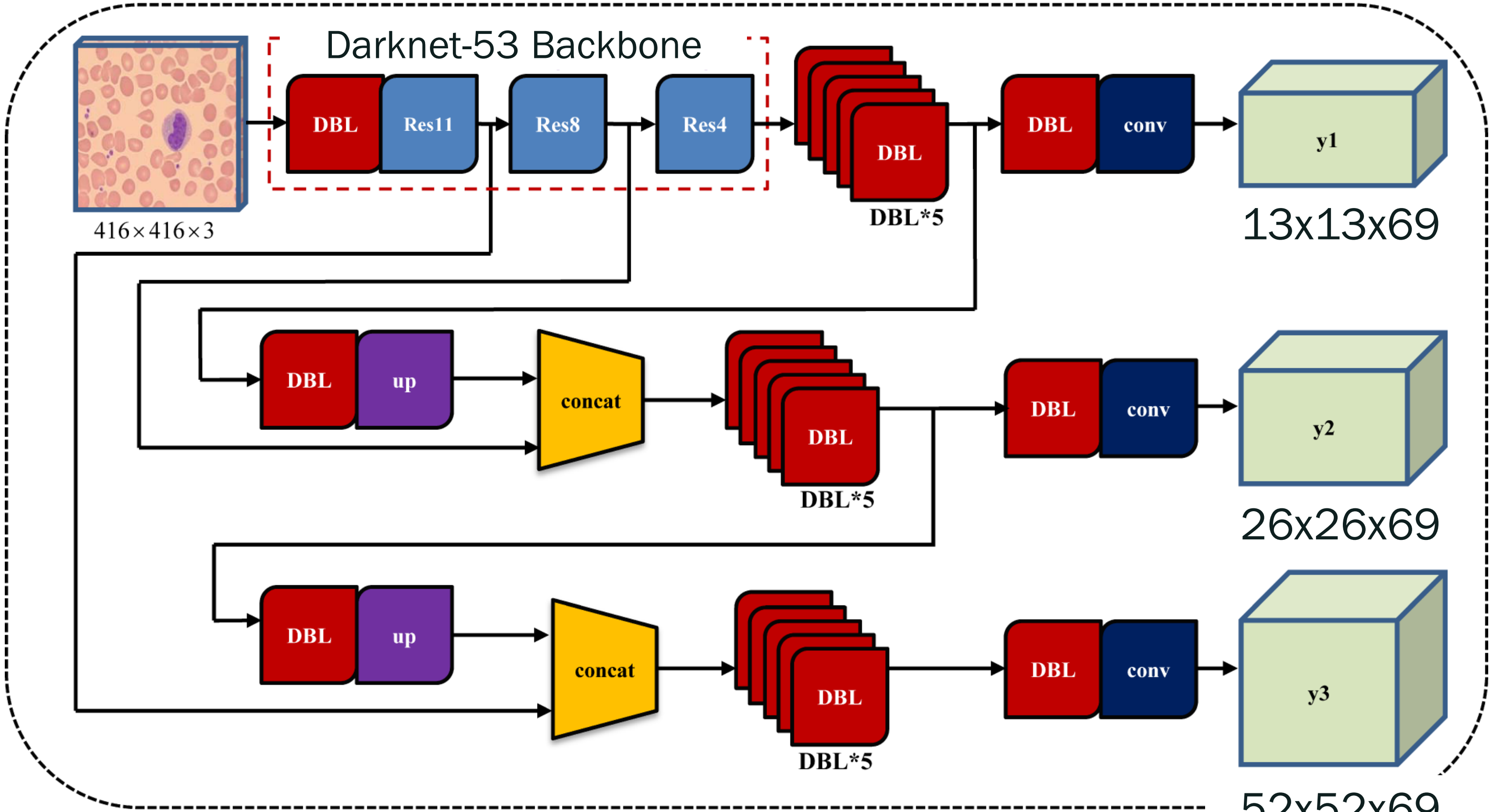


Super Model

Monte-Carlo Averaging

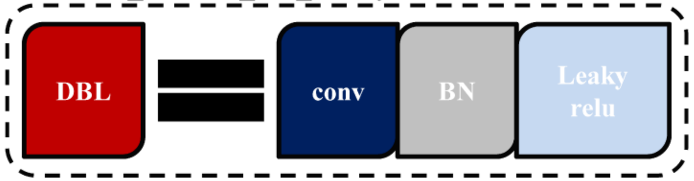
Figure: 5

Incorporation of Monte Carlo Dropout Layer in YOLOv3



YOLO v3

Figure: 6



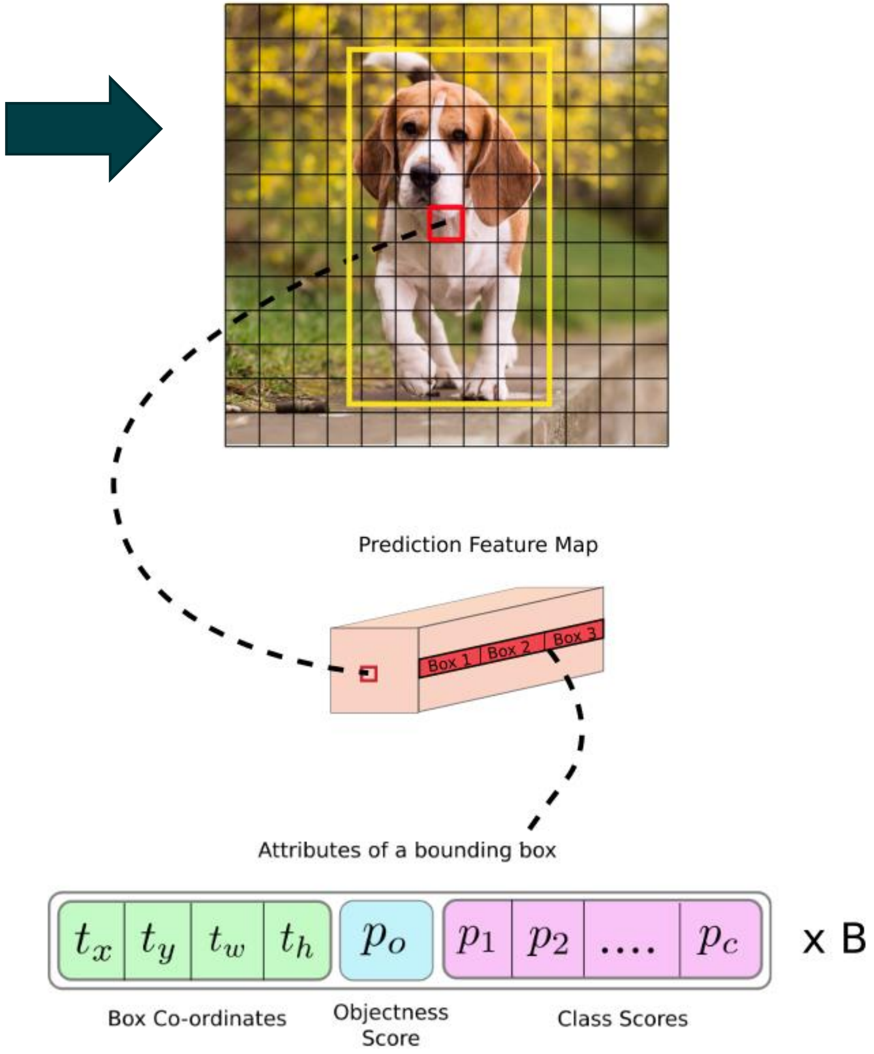
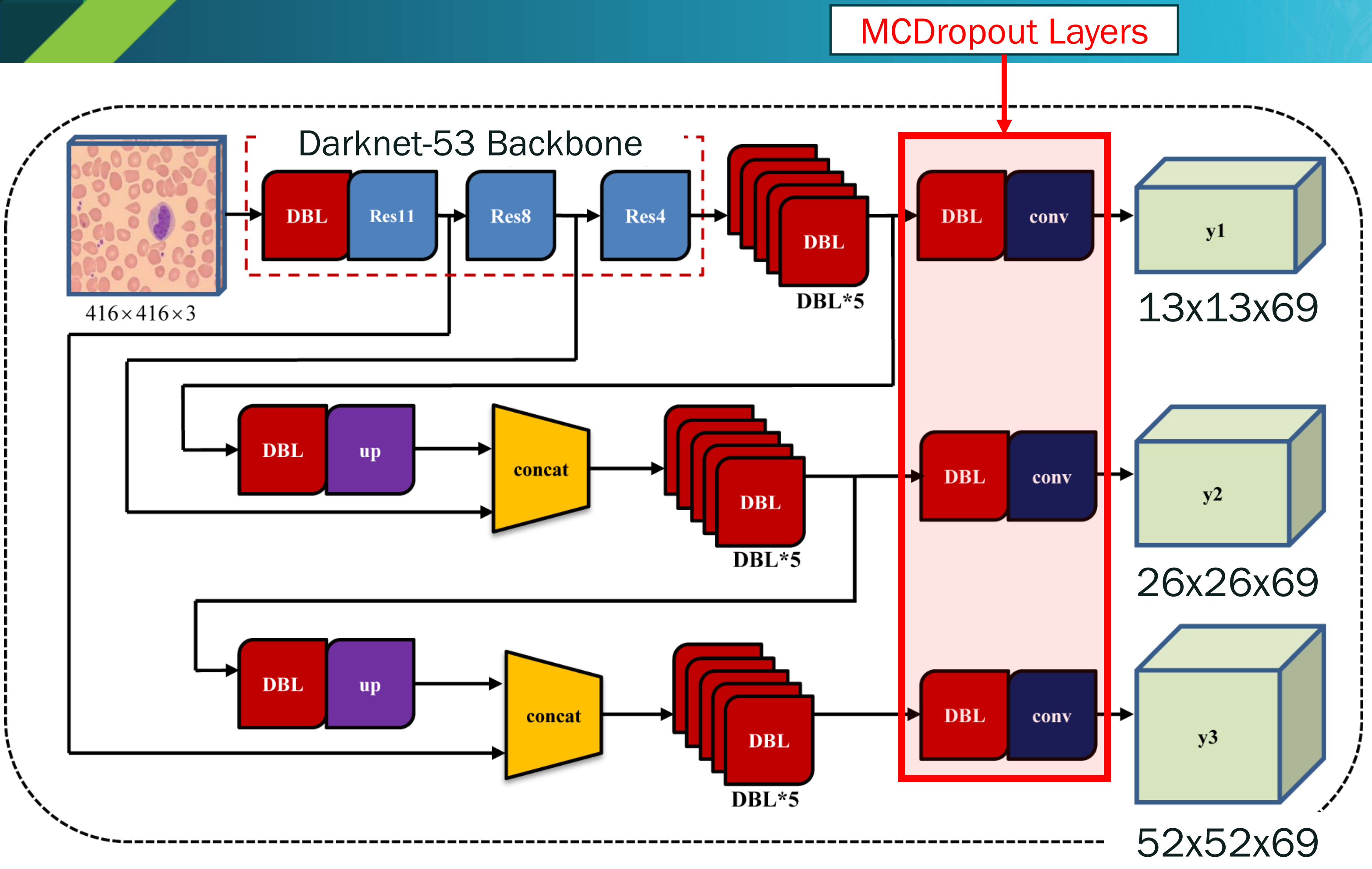
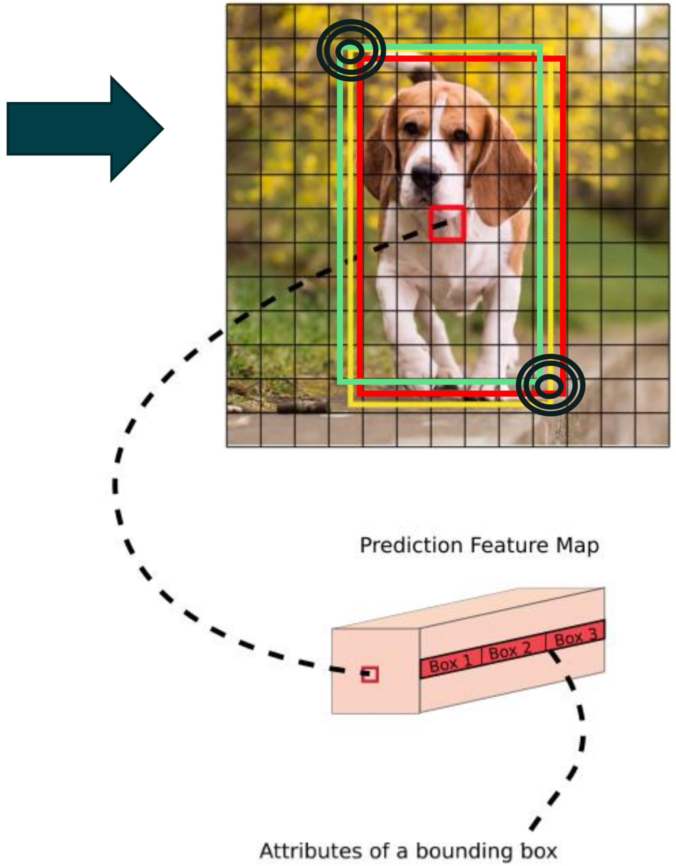


Image Credit: https://doi.org/10.1371/journal.pone.0218808.g005

Incorporation of Monte Carlo Dropout Layer in YOLOv3

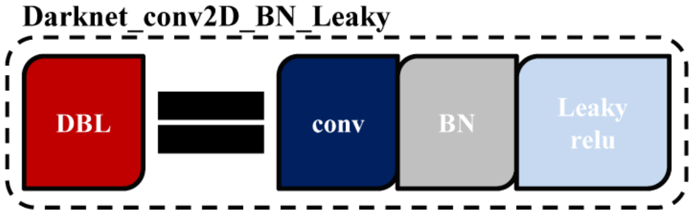


YOLO v3



$\bar{t}_{x1}$	$\bar{t}_{y1}$	$\bar{t}_{x2}$	$\bar{t}_{y2}$	σ_1	σ_2	$\bar{p}_1$	$\bar{p}_2$	...	$\bar{p}_c$

Figure: 7

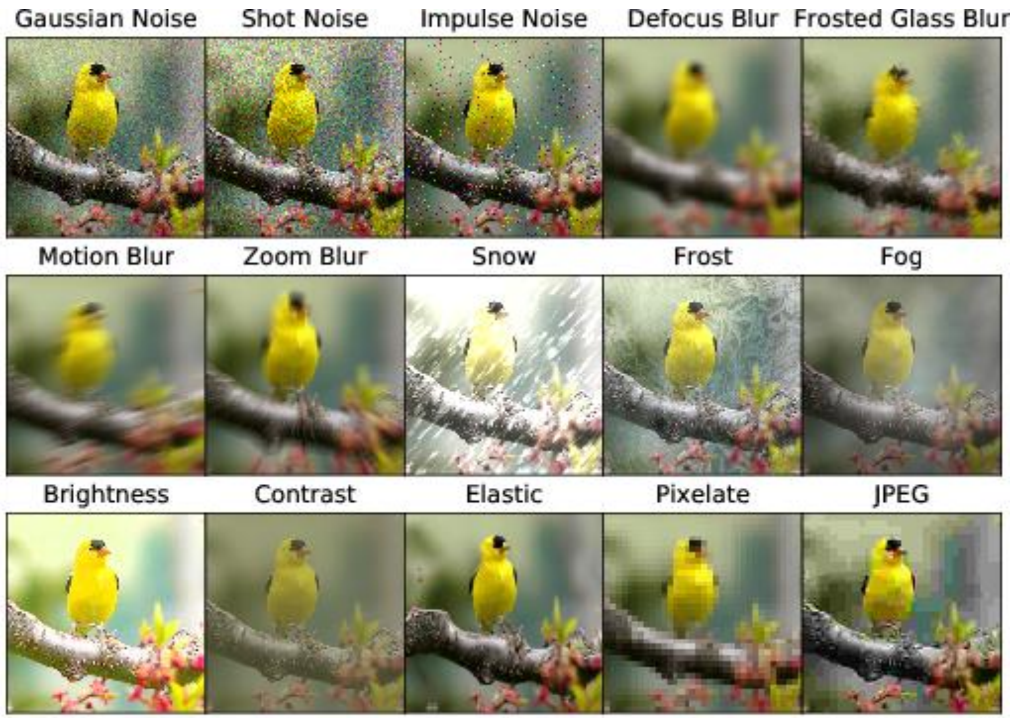


ch Lab

Image Credit: https://doi.org/10.1371/journal.pone.0218808.g005

Validating on Corrupted Input Scene – Dataset Creation

- How do we model dataset shift?
- Training Dataset – Not changed
- Validation Dataset – Corrupted dataset
- 15 corruptions on 5 severity levels
- We validated this on CIFAR-10-C, COCO-C



Download from:
https://github.com/hendrycks/robustness

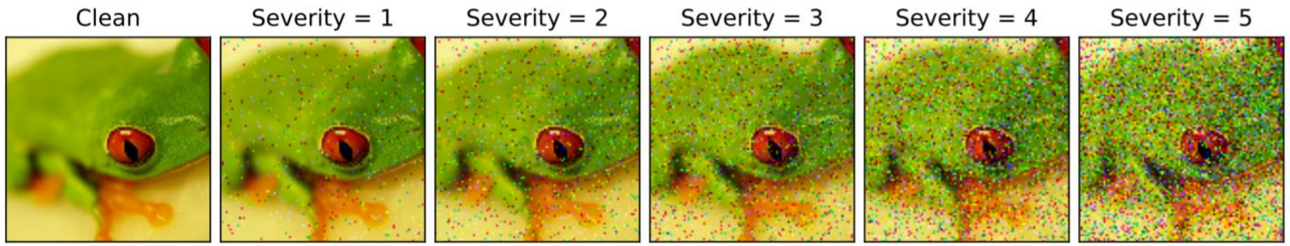


Figure: 8

Improved Prediction Quality of YOLOv3

	PDQ (%)	rPC_{PDQ} (%)	Sp (%)	rPC_{Sp} (%)
Baseline (0.1)	7.19	4.88	14.53	12.18
Baseline (0.5)	9.26	5.4	18.42	16.11
MCD-25 (0.1)	17.73	13.4	37.27	34.84
MCD-25 (0.5)	20.27	12.46	45.78	44.95
Ensemble (0.1)	18.53	13.35	39.9	38.53
Ensemble (0.5)	18.86	11.0	49.97	49.82

PDQ (%): Probabilistic Detection Quality

Sp (%): Spatial Detection Quality

Figure: 9

BBOX Spatial Quality

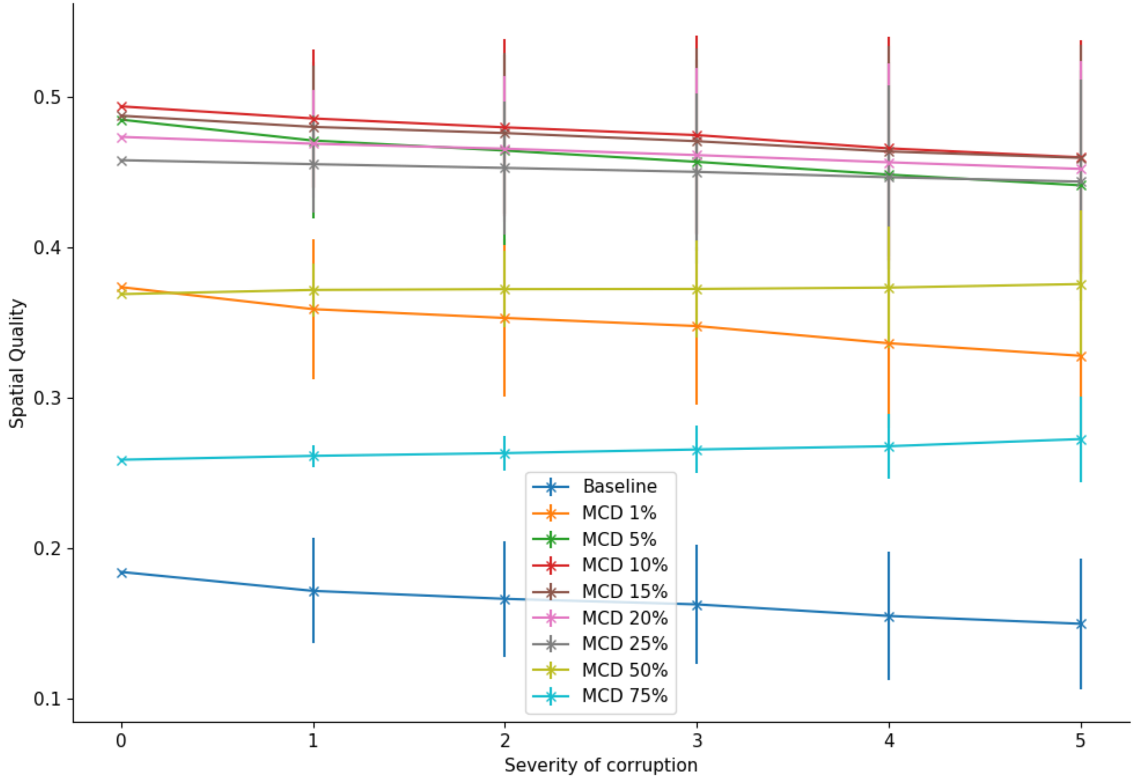


Figure: 10

Performance Impact of Adding MC-Dropout

- Inference time increases by 2x
  - Require intermediate activation caching
  - Better than costly ensemble baseline
  - Model Size does not change
- Trade-off between robustness and performance is possible
  - Apply MCDropout towards the end
  - Choose different dropout rate per layer

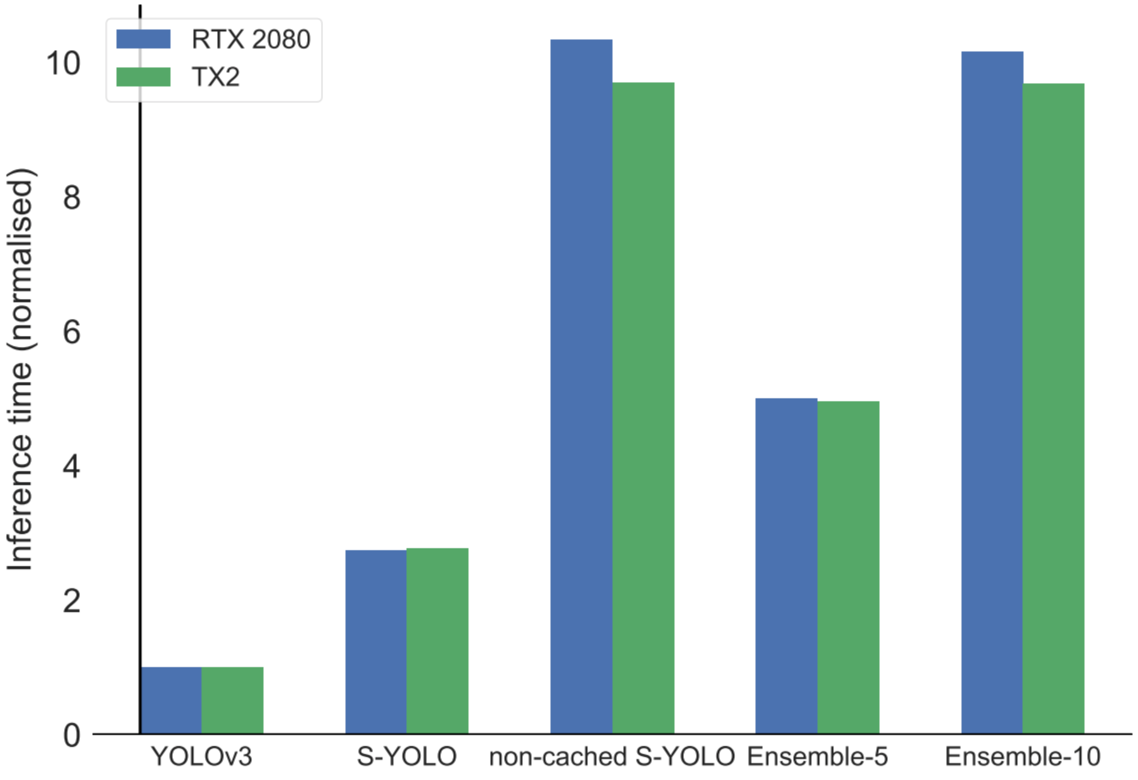


Figure: 11

Key Takeaways

- **Spatial quality** of bounding boxes are very important in Object Detection tasks
- Current quality metric **IoU** does not capture the spatial quality very well
- **PDQ** is a new emerging metric that captures both label and spatial quality
- To capture ambiguity in the bounding box prediction **stochasticity** is required
- **Monte Carlo Dropout** is a simple and elegant way to capture this uncertainty
- **Stochastic-YOLO** is an example model that shows how to use this technique to improve robustness in object detection tasks

More Resources:

Stochastic-YOLO

NeurIPS Paper:

https://ml4ad.github.io/#papers

Code:

https://github.com/tjiagoM/stochastic-YOLO/

Other Probabilistic Models:

https://www.arm.com/resources/research/ml