ContextShift: A Controlled Benchmark for Context Dependence in Object Detection

Dan Zlotnikov, Alex Lazarovich, Ohad Ben-Shahar

Paper Code Dataset

Ben-Gurion University of the Negev

NEURAL INFORMATION PROCESSING SYSTEMS

Is that an upside-down GIRAFFE?

State-of-the-art object detectors achieve strong performance on standard benchmarks, but their robustness to contextual variation remains insufficiently investigated.

ContextShift is a controlled benchmark for evaluating context dependence by manipulating object–context relationships across 5 SOTA models - **Def. DETR, D-FINE, Faster R-CNN, RF-DETR,** and **YOLO26M**



Dataset & Benchmark

Prior benchmarks leave two major gaps:

1. They rely on aggregate metrics such as AP, obscuring substantial changes in prediction dynamics
2. They confound multiple factors and do not provide a controlled axis of object-background compatibility

We measure the context-dependent degradation, pointing to *where* and *how* it happens, across multiple context manipulation families, at different severity levels.
Our published dataset consists of 131,885 images across 38 manipulation variants.

Manipulation	Folders	Images/folder	Subtotal
background/low_freq_noise	5 (1 original + 4 scales)	4,952	24,760
background/smooth_gradient	5 (1 original + 4 variants)	4,952	24,760
background/solid_color	6 (1 original + 5 colors)	4,952	29,712
geometric/enlarged	6 (1 original + 5 scales)	1,376	8,256
geometric/offset	5 (1 original + 4 variants)	1,355	6,775
geometric/rotated	5 (1 original + 4 angles)	1,582	7,910
geometric/shrinked	6 (1 original + 5 scales)	4,952	29,712
Total	38		131,885

The Failure Mechanism

We find and quantify a **prediction suppression pattern** (FN↑, predictions↓, FP≈), with candidate-level analysis experiments supporting reduced candidate formation, and not confidence degradation.

	Rotation	Shrinkage	Enlargement	Translation
AP rAUC	**0.57 [0.51, 0.62]**	0.81 [0.77, 0.83]	1.05 [1.02, 1.06]	0.94 [0.89, 1.02]
ΔFN/img (%)	+93.5 [61.9, 147.0]	**+106.2 [41.4, 168.1]**	−8.5 [−12.3, −5.4]	−16.4 [−0.2, 40.7]
ΔFP/img (%)	**+6.1 [1.0, 11.7]**	−17.9 [−36.2, 5.4]	+4.7 [−2.9, 8.9]	−19.8 [−28.9, −2.5]
Δpred/img (%)	**−30.2 [−41.3, −21.1]**	−18.9 [−27.1, −12.2]	+4.6 [3.0, 6.5]	−9.3 [−17.7, −5.7]

	Low Freq Noise	Smooth Gradient	Solid Color
AP rAUC	**0.82 [0.68, 0.96]**	0.91 [0.83, 0.95]	0.95 [0.87, 1.05]
ΔFN/img (%)	**+32.4 [10.8, 67.1]**	+19.6 [6.2, 51.3]	+13.3 [−2.7, 42.6]
ΔFP/img (%)	−52.2 [−62.5, −43.9]	**−36.8** [−51.5, −25.6]	−41.4 [−53.1, −35.0]
Δpred/img (%)	**−27.6 [−41.5, −17.0]**	−17.2 [−33.4, −9.7]	−15.8 [−32.4, −9.2]

Natural Background Substitution

To test what happens along an object-context compatibility axis, we compute **Normalized Pointwise Mutual Information (NPMI)** over COCO object classes and Places365 scene categories, scanned by an open-vocabulary detector (YOLO-World) to estimate co-occurrence of object classes in scene contexts.

The correlation analysis between the five benchmarked models and the NPMI compatibility prior shows moderate-strong agreement with the constructed compatibility matrix, which is **available as part of the published dataset.**

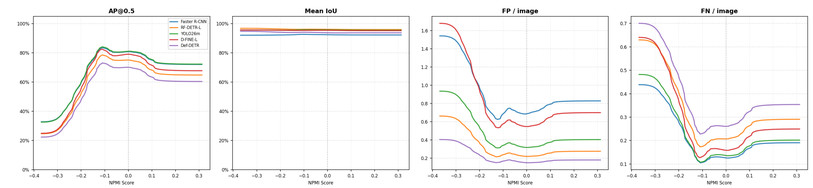
	Def. DETR	D-FINE	Faster R-CNN	RF-DETR	YOLO26M
NPMI Corr. (%)	67.7	68.4	67.7	78.1	77.8

Overall performance on images with natural background substitutions exhibits the same prediction suppression patterns:

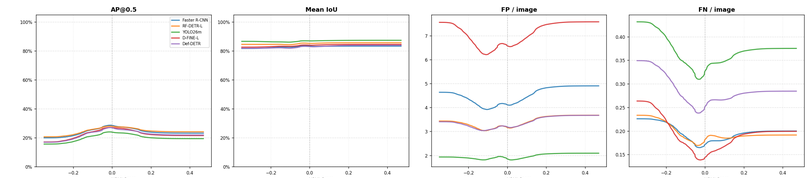
	Def-DETR	D-FINE-L	Faster R-CNN	RF-DETR-L	YOLO26M
AP rAUC	0.81	0.87	0.89	**0.78**	0.93
ΔFN/img	+90.2%	+151.5%	**+227.1%**	+183.9%	+158.6%
ΔFP/img	−60.8%	**−33.2%**	−41.8%	−53.2%	−47.8%
Δpred/img	**−44.2%**	−31.2%	−34.9%	−42.9%	−36.3%

U-Shaped Behavior

Interestingly, behavior along the compatibility axis shows a non-monotonic U-shaped pattern:

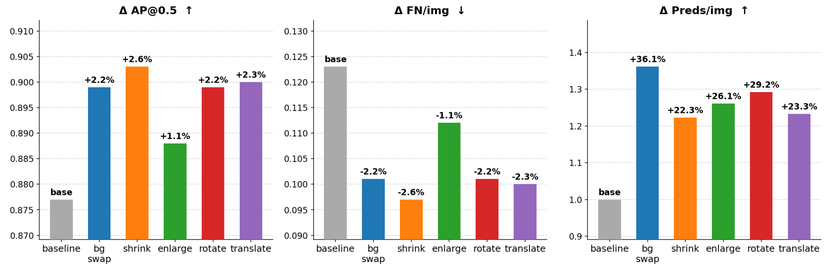


Which repeats (in weaker form) **even on original, unmanipulated images.**



Mitigation

We show that the performance degradation can be partially mitigated by a simple training recipe augmentation, mixing into the training data augmented variants that allow models to decouple objects from their common contextual cues.



Now we can see the difference!

Using **ContextShift,** we can differentiate between failure modes in models that were previously indistinguishable:
~2% overall mAP difference in performance can hide ~50% in FN/image.

