Improving Object Detection with Reinforcement Learning

Siddharth Nayak Balaraman Ravindran

Indian Institute of Technology Madras

The Problem Statement

- Given an image try to find the optimal set of digital transformations to be applied on the image such that the object detection performance of a pre-trained detector improves.

Pipeline

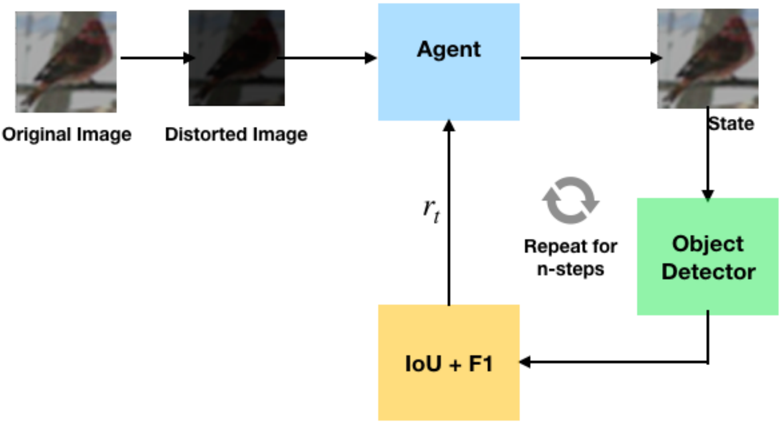


Figure: Pipeline

Digital Distortions

Although we work with digital distortions, ObjectRL can be extended to choose the camera parameters to capture the images by using the image formation model proposed by Hassinoff et al.[1]

- Brightness: $I(x, y) \leftarrow \min(\alpha I(x, y), 255))$
- Color: $gray = (I(r) + I(g) + I(b))/3$, where I(r), I(g) and I(b) are the R, G & B pixel values respectively.
  $I(x, y) \leftarrow \min(\alpha I(x, y) + (1 - \alpha)gray(x, y), 255)$
- Contrast: $\mu_{gray} = mean(gray)$
  $I(x, y) \leftarrow \min(\alpha I(x, y) + (1 - \alpha)\mu_{gray}, 255)$

[1] Hasinoff, Durand, and Freeman, "Noise-optimal capture for high dynamic range photography".

Scales of Distortion

Scales of Distortion: We perform experiments with the following two degrees of distortion in the image:

- Full-scale distortion: The random distortion in the images $\alpha \in [0, 2]$.
- Minor-scale distortion: The random distortion in the images $\alpha \in [0.5, 1.8]$. This constraint limits the images to not have distortions which cannot be reverted back with the action space, the agent has access to.

Distortion Scales

Reward

$$d_t(x) = \gamma(IoU(x)) + (1-\gamma)(F1(x)) \tag{1}$$

Evaluate:

- $d_{o,t} = d_t(\text{original image})$
- $d_{d,t} = d_t(\text{distort image})$
- $d_{s,t} = d_t(\text{state})$

$$\beta_t = 2d_{s,t} - d_{o,t} - d_{d,t} \tag{2}$$

$$r_t = \begin{cases} +1, & \text{if } \beta_t \geq -\epsilon \\ \text{-1}, & \text{otherwise} \end{cases}$$

Motivation for ObjectRL

- In real-time detection applications, lighting conditions and subject speeds can change quickly.
- Single operation mode on cameras will not work well.
- In these cases it would not be possible to create new datasets with images obtained from all the possible combinations of camera parameters along with manually annotating them with bounding-boxes.
- Also, due to the lack of these annotated images we cannot fine-tune the existing object-detection networks on the distorted images.

Motivation

- We propose an extension to *ObjectRL* (for future work) where we have an RL agent which initially captures images by choosing random combinations of camera parameters (exploration phase).
- A human would then give rewards according to the objects detected in the images in the current buffer and these rewards would then be used to update the policy to improve the choice of camera parameters.
- This method of assigning a $\{\pm 1\}$ reward is comparatively much faster than annotating the objects. This methodology is quite similar to the DAgger method (Dataset Aggregation) by Ross et al.[2] where a human labels the actions in the newly acquired data before adding it into the experience for imitation learning.

[2] Ross, Gordon, and Bagnell, "No-Regret Reductions for Imitation Learning and Structured Prediction".

Metric: TP-Score

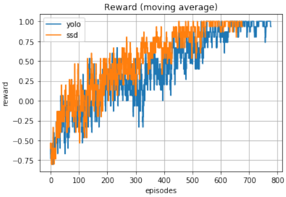
To the best of our knowledge, we believe no suitable measure is defined for this problem and hence we define a measure called *TP-Score(k)* (True Positive Score).

- TP-Score is the number of images in an image set $\mathcal{I}$ in which $k-$or more true positives were detected which were not detected in the image before transformation.
- The *TP-Score(k)* is initialised to zero for a set of images $\mathcal{I}$.
- For example: Let the number of true-positives detected before the transformation be 3 and let the number of true-positives detected after the transformation be 5. Then we have one image where 2 extra true-positives were detected which were not detected in the input image. Thus, we increase *TP-Score(1)* and *TP-Score(2)* by one.

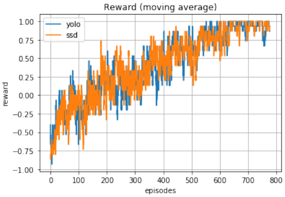
Baselines

- To obtain the baselines, we first distort the images in the original dataset with α being randomly chosen from the set $\mathcal{S} = \{0.1, \ldots, 1.9, 2.0\}$ or $\mathcal{S} = \{0.5, \ldots, 1.7, 1.8\}$ depending on the scale. The set of available actions to be applied on on these images are: $\hat{\mathcal{S}} = \{\frac{1}{s} \forall s \in \mathcal{S}\}$.
- We evaluate the *TP-Score(k)* on the distorted images by applying the transformations by performing a grid-search over all $\alpha \in \hat{\mathcal{S}}$ and report the scores obtained with the best-performing actions for different types and scales of distortions.

Learning Curves

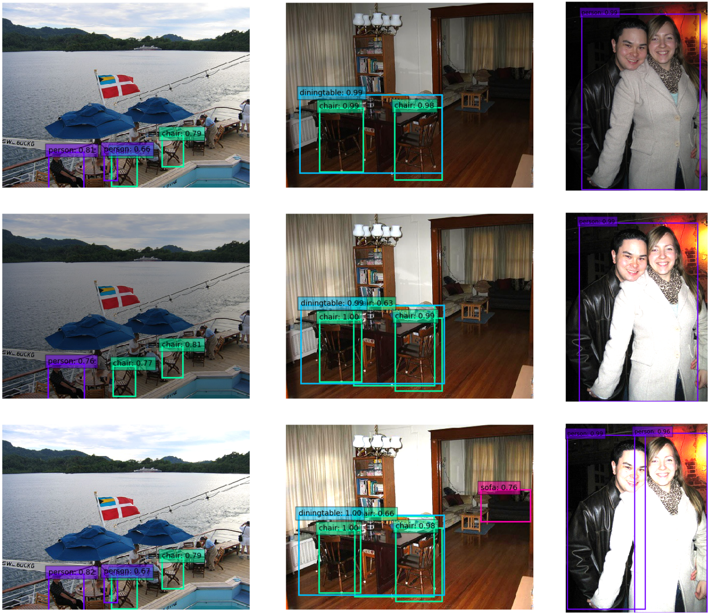


Brightness



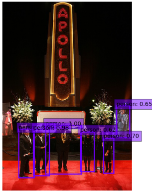
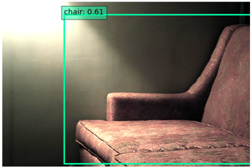
Color

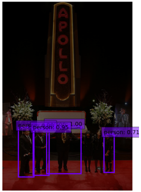
Contrast


chair: 0.79
person: 0.81
person: 0.66
diningtable: 0.99
chair: 0.99
chair: 0.98
chair: 0.81
person: 0.76
chair: 0.77
diningtable: 0.99
chair: 1.00
chair: 0.99
chair: 0.79
person: 0.82
person: 0.67
sofa: 0.76
diningtable: 1.00
chair: 1.00
chair: 0.98

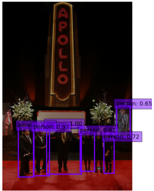
Results


sofa: 0.65

bus: 0.96
car: 0.99

person: 0.65
person: 0.98
person: 0.70

chair: 0.61

bus: 0.96
car: 1.00
person: 0.60

person: 0.95
person: 0.71

sofa: 0.75

bus: 0.95
car: 1.00
person: 0.62

person: 0.65
person: 0.97
person: 0.72

Results

k	Brightness							
	Full-scale				Minor-scale			
	SSD		YOLO		SSD		YOLO	
	GS	*ObjectRL*	*GS*	*ObjectRL*	*GS*	*ObjectRL*	*GS*	*ObjectRL*
1	955 ± 14	532 ± 20	1360 ± 22	976 ± 18	435 ± 25	428 ± 23	1025 ± 23	883 ± 24
2	154 ± 6	87 ± 3	202 ± 15	118 ± 15	87 ± 12	80 ± 9	85 ± 15	63 ± 15
3	49 ± 3	32 ± 4	52 ± 8	18 ± 6	14 ± 5	12 ± 3	8 ± 2	5 ± 1
4	18 ± 3	7 ± 1	17 ± 2	4 ± 1	5 ± 1	3 ± 0	2 ± 0	0
5	7 ± 2	2 ± 0	4 ± 1	2 ± 0	0	0	0	0

Table: *TP-Score(k)* with brightness distortion. GS stands for Grid-Search.[3]

[3] The scores reported are averaged over 10 image sets $\mathcal{I}$, each containing 10,000 images. The means and standard deviations are rounded to the nearest integers.

Results

k	Color							
	Full-scale				Minor-scale			
	SSD		YOLO		SSD		YOLO	
	GS	*ObjectRL*	*GS*	*ObjectRL*	*GS*	*ObjectRL*	*GS*	*ObjectRL*
1	973 ± 17	672 ± 19	1250 ± 23	1103 ± 21	561 ± 18	532 ± 22	974 ± 21	930 ± 22
2	123 ± 7	84 ± 4	210 ± 16	135 ± 13	43 ± 9	37 ± 9	83 ± 12	82 ± 12
3	53 ± 4	31 ± 3	63 ± 7	23 ± 6	1 ± 0	0	15 ± 2	10 ± 1
4	11 ± 2	3 ± 1	19 ± 2	5 ± 1	0	0	6 ± 1	3 ± 0
5	5 ± 1	1 ± 0	6 ± 1	2 ± 0	0	0	0	0

Table: *TP-Score(k)* with color distortion. GS stands for Grid-Search.

Results

k	Contrast							
	Full-scale				Minor-scale			
	SSD		YOLO		SSD		YOLO	
	GS	*ObjectRL*	*GS*	*ObjectRL*	*GS*	*ObjectRL*	*GS*	*ObjectRL*
1	955 ± 15	532 ± 20	1360 ± 21	976 ± 19	680 ± 22	663 ± 24	1038 ± 23	975 ± 24
2	163 ± 8	101 ± 4	213 ± 16	134 ± 15	62 ± 10	49 ± 9	104 ± 13	85 ± 15
3	55 ± 4	36 ± 4	67 ± 7	39 ± 6	14 ± 3	6 ± 2	19 ± 3	16 ± 2
4	21 ± 2	11 ± 1	28 ± 2	13 ± 1	1 ± 0	1 ± 0	5 ± 0	3 ± 0
5	4 ± 1	2 ± 0	5 ± 1	2 ± 0	0	0	0	0

Table: *TP-Score(k)* with contrast distortion. GS stands for Grid-Search.

Cross Policies

To check for the dependence of the policy learned by the agents on the detector it was trained on, we test π_{yolo} with SSD, (denoted as π_{yolo}^{ssd}) and π_{ssd} with YOLO, (denoted as π_{ssd}^{yolo}).

We report the number of images where $k-$or lesser true positives were detected with the swapped policy than what were detected using the original policy on their corresponding detectors.

Cross-Policies

k	Brightness		Color		Contrast	
	π_{yolo}^{ssd}	π_{ssd}^{yolo}	π_{yolo}^{ssd}	π_{ssd}^{yolo}	π_{yolo}^{ssd}	π_{ssd}^{yolo}
1	582 ± 13	1045 ± 24	800 ± 15	1249 ± 26	813 ± 15	1243 ± 26
2	36 ± 6	73 ± 11	72 ± 8	138 ± 11	65 ± 8	145 ± 12
3	2 ± 0	9 ± 4	10 ± 1	13 ± 3	2 ± 0	19 ± 4

Table: *TP-Score(k)* by crossing the policies. π_{SSD} on YOLO is worse than π_{YOLO} on SSD. This is because the range of values for which SSD gives optimal performance is bigger than the range of values for which YOLO gives optimal performance. In essence, YOLO is more sensitive to the image parameters than SSD.

Future work

- Combining all the distortions together instead of one at a time.
- Making local manipulations in the images.
- Extending ObjectRL for choosing camera parameters.

Acknowledgements

I would like to thank Hannes Gorniaczyk, Rahul Ramesh and Manan Tomar for their insights.

The End