A Systematic Analysis of a Context Aware Deep Learning Architecture for Object Detection

Kevin Bardool[1], Tinne Tuytelaars[1], and José Oramas[1,2]
[1]KU Leuven,ESAT-PSI [2]UAntwerpen, IDLab

Abstract. The utility of exploiting contextual information present in scenes to improve the overall performance of deep learning based object detectors is a well accepted fact in the computer vision community. In this work we propose an architecture aimed at learning contextual relationships and improving the precision of existing CNN-based object detectors. An off-the-shelf detector is modified to extract contextual cues present in scenes. We implement a fully convolutional architecture aimed at learning this information. A synthetic image generator is implemented that generates random images while implementing a series of predefined contextual rules, allowing the systematic training of such relationships. Finally, a series of experiments are carried out to evaluate the effectiveness of our design in recognizing such associations by measuring the improvement in average precision.

1 Introduction

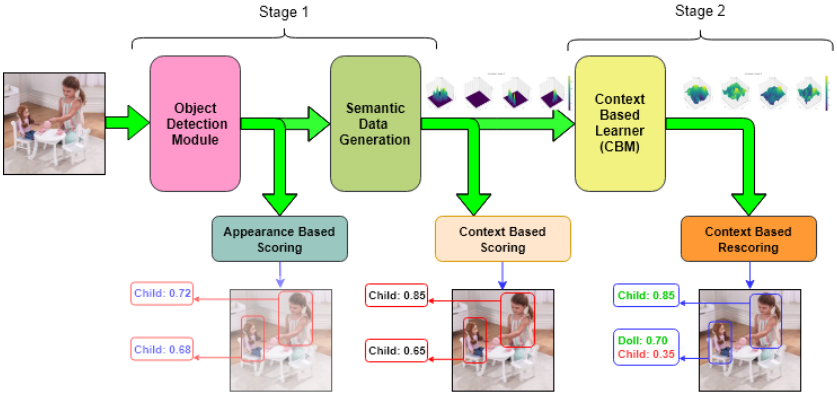


Fig. 1: High level overview of our proposed design

A notable feature of our visual sensory system is its ability to exploit contextual cues present in a scene. Various works on visual cognition have shown that humans naturally exploit such information to enhance their perception and understanding of the image [1–5].

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)

CEUR-WS.org/Vol-2491/paper90.pdf

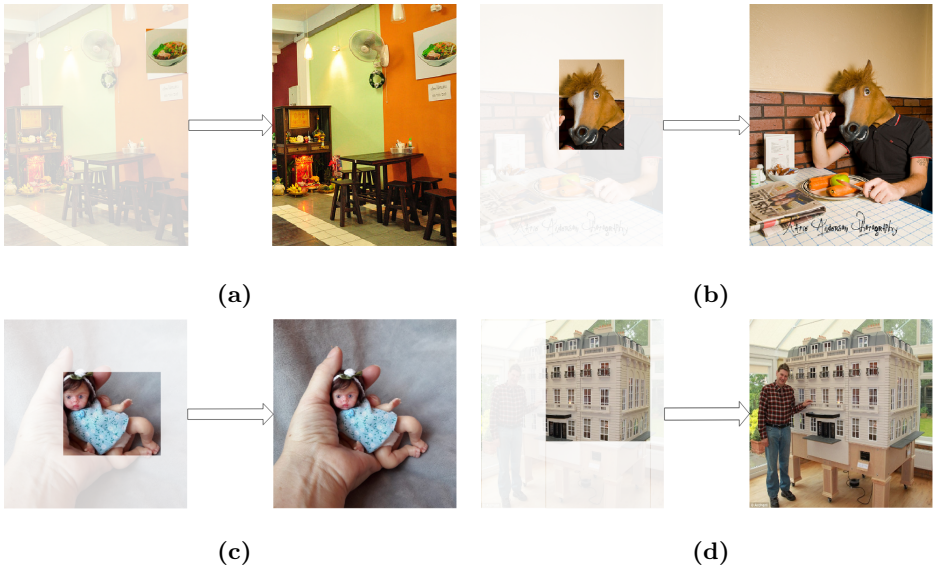


Fig. 2: Examples illustrating the effect of contextual cues in object detection (a): Expected position. **(b):** Occurrence probability. **(c, d):** Size

While most object detection models have focused on intrinsic feature descriptors for classification and localization tasks [6–8], it is generally accepted that context can aid in improving the performance of such models [9–12]. Exploring methods of incorporating and learning contextual information in these tasks has been a continuous focus of research, both in conventional object detectors [13–15] as well as more recent CNN-based architectures [16–18]. It is in this context that we propose a deep CNN architecture focused on learning such contextual relationships.

The main contributions of this paper are: a) proposing a methodology to extract contextual cues present in a scene, as well as a fully convolutional architecture aimed at learning such relationships, and b) a synthetic image generator used to generate random images while enforcing a series of predefined contextual relationships. We also evaluate the proposed architecture to determine its robustness in learning contextual relationships.

2 Related Work

2.1 Contextual Cues in Vision

There have been various attempts to categorize sources of contextual information [19, 10, 9]. Biederman groups relationships between an object and its surroundings into five classes: *interposition, support, occurrence probability, position, and size* [19]. It is the three latter relationships which are of our interest: *occurrence probability*, the likelihood of an object appearing in a particular scene; *position*,

the expectation that when certain objects are present, they occupy predictable positions; and *size*, the expectation that an object's size follows a propotional relationship with other objects and the general scene [9]. Such relationships are called *contextual features*, as they require access to the referential meaning of an object and its context.

Figure 2 illustrates how various contextual cues can influence object detection. For example, in Figure 2(b), the small image patch on the left can be perceived as a horse, however when the full image is revealed, we see the object is actually a mask. Occurrence context could assist in such a scenario. In Figure 2(c), the image patch on the left can be identified as an infant, while the complete image reveals the object in question is actually a doll. Here, relative size between the object and the hand can assist in correcting the detection.

2.2 CNN-based Object Detectors

Since the resounding success of the AlexNet [6] architecture, the performance of CNN-based object detectors has been improving at an astounding rate. This success can be attributed to incremental advancements such as deeper network architectures [20–22], the shift from sliding window [23] to region proposal methods [24, 25, 8], and feature sharing for multi-scale object detection [26].

Faster R-CNN. Faster R-CNN [27] is a two-stage object detector sharing a common backbone. The first stage is the Region Proposal Network (RPN), a small fully convolutional network [28] that uses feature maps generated in the shared backbone to identify a series of rectangular object proposals *(region of interest (RoI))*, along with an objectness score. The second stage is a Fast R-CNN [29] detector. Receiving the highest scoring proposals from RPN, the *RoI Pooling* component uses the common backbone to extract fixed sized feature maps for the selected proposals, passing them to two fully connected layers responsible for object classification and bounding box localization.

Mask R-CNN. Mask R-CNN [30] is one of the most recent iterations of the R-CNN family of object detectors [8, 29, 31]. It introduces instance segmentation to the Faster R-CNN architecture by adding a branch to predict segmentation masks over each proposed RoI. To accommodate this, the RoI Pooling layer is replaced with RoI Align layer, which provides a more accurate alignment between feature maps and their corresponding RoIs.

2.3 Exploiting Context in Object Detection Tasks

In works aimed at exploiting contextual information in DNN-based object detectors, two main approaches stand out.

The first approach uses contextual information as a feedback method that guides the generation of initial object proposals. Chen et al. [17] propose a *memory network* [32]: a Faster R-CNN detector coupled with external memory that

can be read and written to, augmenting the detector with a long-term memory component. This *spatial memory* is iteratively updated with feature maps of detected objects. The memory content is then used as input to a CNN-based contextual reasoning component, iteratively producing scores to assist the Faster R-CNNs region proposal process.

A second approach involves the extraction and use of contextual information after proposal selection, and during the classification stage [33, 34, 18]. Zeng et al. [33] exploit contextual features by establishing a message sharing mechanism among proposal feature maps of multiple regions and resolutions. RoI feature maps generated in backbone are passed through a network of Gated Bi-Directional (GBD) units which form the message sharing system. The gated feature of GBD units is trained to only pass along information useful to other units. This architecture allows feature maps of different scales and resolutions, which correspond to local and contextual features, cooperatively share supporting information in a way that improves the confidence of proposed hypotheses and detection accuracy.

Some works integrate both methods in their design. Shrivastava et al. [16] employ semantic segmentation as a source of contextual information between objects and global structures in the image and incorporate it into a Faster R-CNN architecture, using it as a top-down feedback signal to guide *both* the region proposal and object classification modules.

3 Proposed Methodology

In the aboved mentioned approaches, use of contextual information is intertwined with the object detection architecture. We take a different approach: the separation of appearance detection and contextual reasoning. While in some works a secondary model was used as a source of contextual information flowing *into* the object detector (e.g. [35]), our design reverses this: we adopt a two stage pipeline, where contextual information from the primary stage flows to a secondary model, trained to learn such relationships. At inference time, the secondary model is used to re-evaluate object detector proposals based on the contextual relationships it has learned. Figure 3 illustrates our proposed method.

3.1 Stage One: Object Detection and Feature Map Generation

The first stage of our architecture is an off-the-shelf object detector, responsible for generating appearance-based feature maps, as well as classification and localization. Additionally, it will be used to construct *contextual feature maps* that are passed on to the next stage. We select the Mask R-CNN model for this stage.

3.2 Contextual Feature Maps

A new network layer is responsible for generating per-class contextual feature maps. These are constructed using class confidence scores and bounding boxes

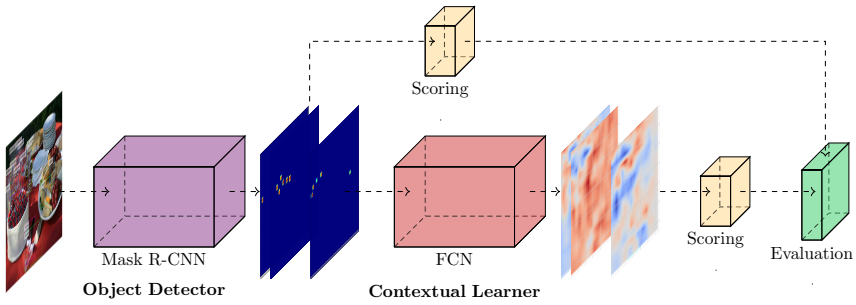


Fig. 3: Proposed two stage pipeline architecture.

produced by the Mask R-CNN object detection and localization heads. Each proposal is represented as a Gaussian kernel with a covariance matrix proportional to the bounding box size:

Proposal Feature Maps. For each RoI proposal b, a multivariate Gaussian distribution is generated:

$$f_{\mathcal{N}}(x, y) \sim \mathcal{N}((x, y); \mu_b, \Sigma_b) \tag{1}$$

where:

$$\mu_b = [x_{cb}, y_{cb}], \quad \sigma_{bx}^2 = \sqrt{\frac{w_b}{2}}, \quad \sigma_{by}^2 = \sqrt{\frac{h_b}{2}}, \quad \Sigma_b = \begin{bmatrix} \sigma_{bx}^2 & 0 \\ 0 & \sigma_{by}^2 \end{bmatrix}, \quad b = [0, 1, .., B]$$

Here, B is the total number of proposed bounding boxes and the tuple $(x_{cb}, y_{cb}, w_b, h_b)$ designates the x, y coordinates of the center, width, and height of bounding box b. Since detected bounding boxes are axis aligned, the covariance matrix is considered diagonal.

To generate the feature map for an individual bounding box, FM_b, each resulting distribution is passed through a masking stage which suppresses the probability values for areas outside of a tight region surrounding the bounding box centroid to zero (Eq. 2).

$$fm_b(x, y) = \begin{cases} f_{\mathcal{N}}(x, y) & where \quad \begin{cases} x_{cb} - \sigma_{bx}^2 \leq x \leq x_{cb} + \sigma_{bx}^2 \\ y_{cb} - \sigma_{by}^2 \leq y \leq y_{cb} + \sigma_{by}^2 \end{cases} \\ 0 & otherwise \end{cases} \tag{2}$$

$b=[0,1,..,B]$

$f_{\mathcal{N}}$ is the normal distribution defined for the bounding boxes as defined in (Eq. 1). Each individual heatmap fm_b is normalized and multipled by ncs_b, the class-normalized score of the bounding box generated by Mask R-CNN classifier (Eq. 3).

$$FM_b = \frac{fm_b}{max(fm_b)} * ncs_b \tag{3}$$

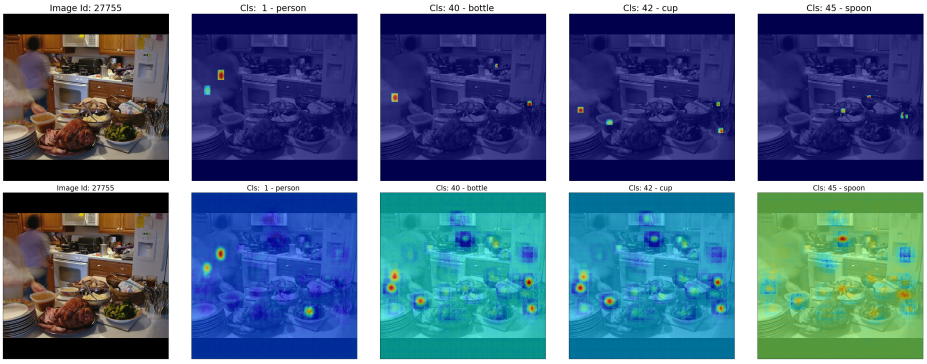


Fig. 4: Contextual heatmaps generated for a sample image. **Top:** Feature maps generated by stage one object detector **Bottom:** Predictions generated by the contextual learner.

The motivation behind this is to give higher weights to bounding boxes the detector is more confident about. The final result, FM_b, is the contextual feature map for proposal b.

Class Contextual Feature Maps. Heatmaps for all bounding boxes predicted for each class are summed and normalized to $[0, 1]$ (Eqs. 4,5). The result, CFM_c, is the contextual feature map for class c across the full spatial extent of the image.

$$cfm_c(x, y) = \sum_{cls(b)=c}^{\substack{b=[1,\ldots,B] \\ c=[1,\ldots,C]}} hm_b(x, y) \tag{4}$$

$$CFM_c = \frac{cfm_c}{max(cfm_c)}^{c=[1,\ldots,C]} \tag{5}$$

The output of the contextual feature map layer is a $(H_c \times W_c \times C)$ tensor, where H_c and W_c are the heatmap dimensions, and C is the number of classes.

When training on datasets that involve a high number of classes and/or large image sizes, the memory requirements for our architecture substantially increase and can become computationally prohibitive. For example, when processing COCO dataset images, the dimensions of the generated contextual feature map tensor are $(batchsize \times 1024 \times 1024 \times 81)$. For a batch size of 2, this tensor alone requires around 640 MB of storage.

To address this, we allow a configurable downscaling of the generated feature maps to reduce memory requirements when necessary. For example, feature maps generated for COCO images are scaled down by a factor of 4, which results in a 16 fold reduction in memory requirements. In the scenario described above, the space required needed for the feature maps is reduced to 40 MBs.

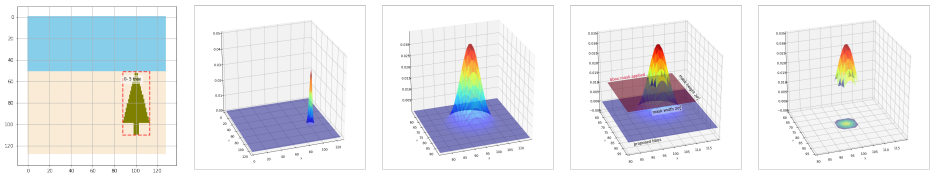


Fig. 5: Object proposals are represented using Gaussian kernels. Scoring is computed as an integration within a tight bandwidth surrounding the bounding box center.

3.3 Contextual Scoring

Another contribution is the design of a scoring function to measure the contextual relevance of detections in relation to other objects present in the image and the general scene. Scores are computed using the contextual feature maps generated in each stage of our pipeline, and are used as the ranking score in AP calculations to measure whether contextual learner confirms or refutes detections passed to it, based on learned semantic relationships. The scoring process is designed as a new network layer, and appended to the end of each stage in our pipeline. We define two alternative scoring methods, as described below.

Scoring Method 1. The first score is calculated using bounding box individual feature maps, FM_b (Eq. 3). The score is calculated as the summation of the FM_b values within the mask region, divided by the area of the bounding box's mask (Eq. 6).

$$Score_1(b) = \frac{\sum FM_b(x,y)}{2\sigma_{bx}^2 \times 2\sigma_{by}^2} \qquad where \quad \begin{cases} x_{cb} - \sigma_{bx}^2 \leq x \leq x_{cb} + \sigma_{bx}^2 \\ y_{cb} - \sigma_{by}^2 \leq y \leq y_{cb} + \sigma_{by}^2 \\ b = [1, .., B] \end{cases} \tag{6}$$

Scoring Method 2. This score is calculated using the heatmap generated for each class, CFM_c (Eq. 5). For each bounding box, we use tight mask around the center coordinates as before, apply the summation on the corresponding class heatmap CFM_c within the mask, and divide by the mask area (Eq. 4).

$$Score_2(b) = \frac{\sum CFM_c(x,y)}{2\sigma_{bx}^2 \times 2\sigma_{by}^2} \qquad where \quad \begin{cases} x_{cb} - \sigma_{bx}^2 \leq x \leq x_{cb} + \sigma_{bx}^2 \\ y_{cb} - \sigma_{by}^2 \leq y \leq y_{cb} + \sigma_{by}^2 \\ b = [1, .., B] \\ class(b) = c \end{cases} \tag{7}$$

3.4 Stage Two: Contextual Learner

The second stage, context-based model is trained to learn semantic relationships using the contextual feature maps generated by the primary object detector. For this stage, a CNN model based on the Fully Convolutional Network (FCN) architecture [28] was implemented.

Fully Convolutional Networks Fully convolutional networks are architectures that only consist of convolutional layers (i.e, no dense layers are used). We adopt the architecture proposed by Shelhamer et al. [28], which itself is based on a modified VGG-16 architecture [20]. The final softmax layer is removed, and the two fully connected layers are converted to convolutional layers. Additionally, a 1×1 convolutional layer consisting of C filters is added, where C is the number of classes and depends on the training data. The result is a per-class prediction map built from the VGG-16 `Pool5` feature map, downscaled by a factor of 32. To further improve accuracy, feature maps produced from intermediate pooling layers of the VGG16 backbone are utilized as skip layers and fused with the $\times 32$ downscaled output to improve performance. Since these feature maps have smaller receptive fields, they also provide finer details.

The output of this model is also series of contextual belief maps, representing the its confidence on the original detections based on contextual relationships it has learned (Figure 4).

4 Evaluation

4.1 Implementation Details

Our design uses an implementation [36] of the Mask R-CNN model built on the Keras [37] and TensorFlow [38] frameworks. The segmentation head of this model was removed, and two new layers were added to generate contextual feature maps and corresponding scores. For the second stage contextual learner, we implement an FCN8 architecture [28] with trainable deconvolution layers. A scoring layer is also added to the bottom of this model to produce the second stage contextual scores.

Training the object detector is performed using the same loss function as specified in [27]. For the contextual learner, training is exercised using a Binary Cross Entropy loss.

4.2 Dataset

For training and evaluation, a synthetic image generator is implemented that generates random images consisting of objects from eight different object classes. Image characteristics such as size, minimum and maximum number of objects in an image, and the maximum number of instances of each class appearing in an image are parameterized and adjustable based on our needs.

A variety of contextual relationships are predefined and enforced during image generation. All objects appear within a spatial range determined by their corresponding class, adjusted relative to a randomly selected horizon line in each image, as well as other objects present in the scene. This also includes certain inter-class spatial relationships; for example, instances of two classes Person and Car maintain a fixed spatial distance with each other. Size based context is enforced by classes maintaining proportionate sizes relative to each other. Additionally, objects are scaled based on their vertical position in the image, through

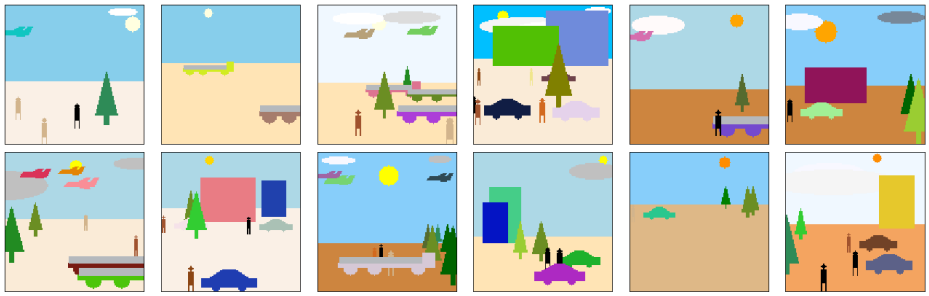


Fig. 6: Sample images from the synthetic toy dataset.

which a notion of depth is simulated. Co-occurrence relationships are established between groups of object classes such that the presence of instances from one group in an image preempts the presence of instances from the other group.

Such a dataset allows us to train the pipeline with relatively simple content, introducing contextual cues in a controlled manner. We use collections of 15,000, 2,500, and 500 images as our training, validation, and test datasets, respectively. Figure 6 shows samples of such randomly generated images. The software is also publicly available[1].

4.3 Experiments and Results

In addition to evaluation on test datasets, we implement a number of experiments to measure the capacity of our design in learning various contextual relationships enforced by the synthetic image generator. The contextual scores are used to compute the average precision (AP) and mean average precision (mAP) as defined in the evaluation protocols for Pascal VOC challenge [39].

Exp 1. Evaluation on Toy Dataset. We evaluate our model on on a set of 500 test images, and observe a contextual-score based mAP improvement of approximately 1.3 points using scoring method 1 (Table 1).

Performance of the baseline Mask R-CNN detector is significantly higher than all contextual-based scoring methods. This is not surprising: the Mask R-CNN detector has access to multiple-scale feature maps based on intrinsic characteristics of object proposals, covering multiple effective receptive fields. On the other hand, our contextual model receives heatmaps that only cover class, location, and size characteristics, in addition to a probabilistic component based on the detector's confidence score. It is up to the model to extract contextual relationships purely based on this information.

Exp 2. Detecting Spatially Out-of-Context Objects. We measure the model's capacity in learning the expected spatial context of objects. For each image, a controlled set of hypotheses that include object proposals positioned out

[1] https://github.com/kbardool/Contextual-Inference-V2

	SM	**mAP**	person	car	sun	building	tree	cloud	airplane	truck
Mask R-CNN[30]	–	83.24	79.49	86.16	90.66	78.07	80.83	79.60	88.26	79.82
Detector	1	77.95	75.84	82.91	88.83	**73.72**	**79.55**	73.67	81.89	67.21
Contextual Learner	1	**79.27**	**76.44**	**83.93**	**89.11**	71.21	79.16	**74.70**	**86.30**	**73.30**
Detector	2	**77.89**	**75.83**	**82.91**	**89.11**	**73.53**	**79.38**	**73.23**	82.00	67.25
Contextual Learner	2	77.57	75.77	81.85	88.23	70.32	78.33	71.73	**83.98**	**70.37**

Table 1: Exp 1 - Detection results on toy dataset. Legend: **SM 1:** Scoring Method 1 **SM 2:** Scoring method 2

	SM	**mAP**	person	car	sun	building	tree	cloud	airplane	truck
MR-CNN Baseline	–	46.77	45.28	46.51	45.62	45.51	45.84	47.39	50.11	47.88
Detector	1	49.24	50.56	46.56	47.46	51.11	51.83	46.30	49.55	50.52
Contextual Learner	1	**79.16**	**76.53**	**88.06**	**95.67**	**65.64**	**52.65**	**94.54**	**83.70**	**76.52**
Detector	2	49.22	45.96	46.56	47.46	55.34	47.49	50.87	49.55	50.53
Contextual Learner	2	**80.53**	**79.27**	**87.09**	**95.61**	**66.27**	**56.60**	**94.80**	**85.87**	**78.76**

Table 2: Exp 2 - In-class spatial context detection. Legend: **SM 1:** Scoring Method 1 **SM 2:** Scoring method 2

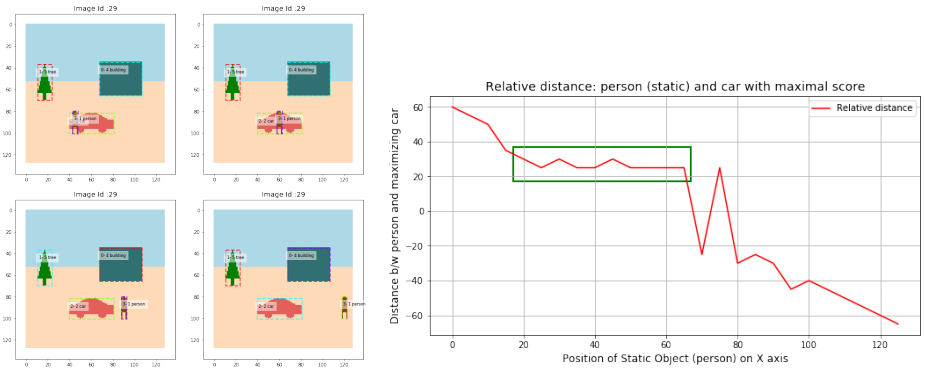
of their natural spatial range is generated and passed to the contextual reasoning model. An average class score is assigned to both true and false positives.

As results in Table 2 show, the contextual model significantly improves the mAP by down-scoring the false positive proposals. The success of the contextual model in rejecting false positives indicates that it is able to learn class-based spatial constraints.

Exp 3. Inter-class Spatial Context. During the toy dataset image generation, a series of spatial relationships were enforced between certain classes. As mentioned previously, instances of the Person class always occur at a fixed location to the left of Car instances (30 pixels to the left of the car's bounding box center point). Here, we will focus on this spatial relationship.

We fix one of the two objects, and move the second object across the horizontal axis. We conduct this experiment in two settings. In each setting the instance of one class is stationary while the other object is moved across the horizontal axis. The second stage contextual scores are plotted to observe score changes as the non-stationary object is moved across the horizontal axis. We run these experiments repeatedly, gradually moving the stationary object across the screen, to evaluate the detection of this spatial relationship across different spatial configurations.

Figure 7 shows the result of this experiment. Figure 7(b) plots the results in the form of relative distance between the two objects which results in a maximal contextual score. We observe that within the area enclosed by the green box, the maximal score for a Car occurs when the Person instance is 30 pixels to its left, which is indeed the spatial distance enforced between these two classes



(a) **(b)**

Fig. 7: Exp. 3 - Inter-class spatial relationships. (a): Sample images of various spatial positioning between car and person objects.**(b)**: Relative distance between static and moving object with maximal score.

by the image generator. We note, however, that this correspondence does not span the complete horizontal extent of the image; this can be attributed to fewer training examples reflecting this relationship around the horizontal boundaries of training images.

This experiment confirms that the contextual learner is able to recognize such spatial relationships, although in a limited area of the image.

Exp 4. Inter-Class Co-Occurrence. Here we focus on semantic rules enforced between different classes during the toy dataset generation process. Co-occurrence relationships between class groups ***A:***{Car, Building} and ***B:***{Airplane, Truck} have been enforced such that the presence of objects from different groups in an image is mutually exclusive (i.e., objects from opposite groups do not co-appear in any image).

If the stage one detector is uncertain about an object proposal p, we expect the addition of an object from the same group would result in the contextual model confirming the presence of p, and consequently an increase its stage two contextual score. Conversely, if an object from an opposing group is added to the image, we expect the rescoring module to down-score one of the conflicting detections.

Results in Figure 8(c) shows that a Car's contextual score is minimally affected by the presence or absence of the non-semantically related Airplane. Similarly, Figure 8(f) shows the presence or absence of the semantically related Airplane class has little to no impact on a Truck's contextual score. We conclude the model is unable to learn such relationships.

Exp 5. Size vs. Object Depth in Image In the toy dataset images, a sense of depth is created by scaling object sizes relative to a horizon line present in the image. When positioned further up in the image, objects appears smaller,

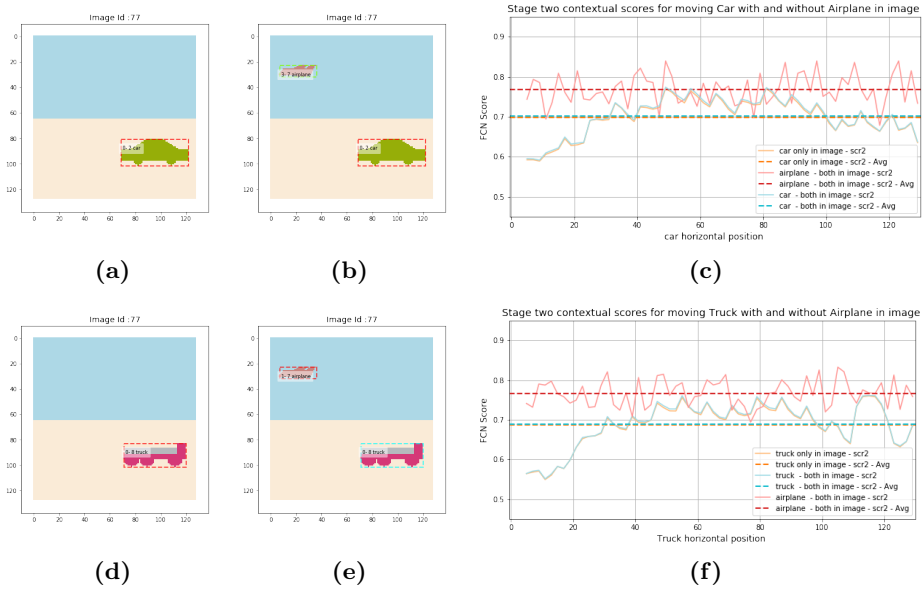


Fig. 8: Exp. 4 - Inter-class semantic co-occurrence. (a,b): Sample images for Car, with and without semantically non-correlated object presence. **(c):** Contextual scores for **(a,b)**. Orange plot: Car scores, without Airplane in image. Blue: Car scores, both objects present. Red: Airplane scores, both objects present in image. **(d,e):** Sample images for Truck, with and without semantically correlated object presence. **(f):** Contextual scores for **(d,e)**. Orange plot: Truck scores, without Airplane in image. Blue: Truck scores, both objects present. Red: Airplane scores, both objects present in image

simulating a further distance from the observer. We test the capacity of the model in learning this relationship by scoring objects at various scales over their minimum and maximum vertical extent.

Figure 9 shows the result of our experiments for the Car class. In Figure 9(a,b), the Car instance is positioned at its minimum vertical position (furthest away from the observer) with different scales. We observe the model prefers smaller sized objects (blue plot in Figure 9(c)). Conversely, in Figure 9(d,e), where the object appears closer to the observer, larger sized Cars receive a higher average score (red plot in Figure 9(f)). Our contextual model is able to learn the relationship between relative size and vertical location for different classes, favoring larger size objects when positioned lower in the image (i.e., closer to the observer).

5 Conclusion

A two stage architecture is proposed that extracts underlying contextual information in images, and represents such relationships using a series of context-

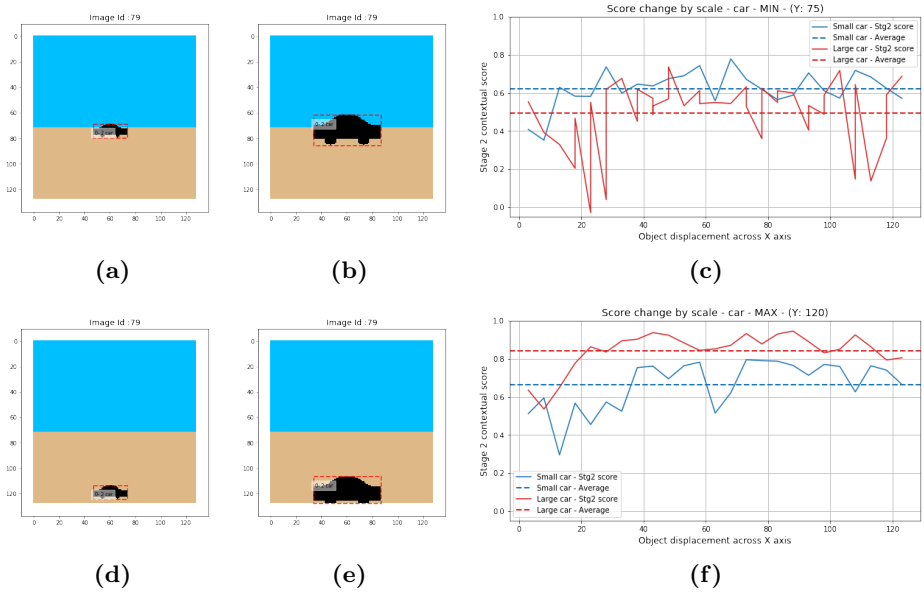


Fig. 9: Exp. 5 - Size vs. depth in image. (a,b): Small and large size car at furthest depth. **(c)**: Contextual scoring results. Blue plot: small sized car. Red plot: large sized car. **(d,e):** Small sized car and large size car at closest depth. **(f)**: Contextual scoring results. Blue plot: small sized car. Red plot: large sized car.

based feature maps. By passing this through the second stage of our architecture, we attempt to train a contextual model to learn representations of such relationships.

Results of our experiments show that the contextual model is successful in learning intra-class spatial relationships, e.g., it is able to detect and reduce the contextual score of objects positioned out of their natural spatial context. We also found that inter-class spatial relationships are recognized, although in a limited capacity, which could be due to the limited spatial range of provided training examples. Additionally, it is able to learn the relation between the size of an image and its depth in the scene: smaller sized object average higher scores when appearing further away from the observer. However, we have not seen robustness towards learning co-occurrence of semantically related objects, which we consider one of the more critical forms of context in assisting object detection tasks.

Continuing experiments on the more challenging COCO dataset, and investigating methods to induce learning of semantic co-occurrence relationships are open avenues for future work.

References

1. I. Biederman, "Perceiving Real-World Scenes," *Science (80-.).*, vol. 177, pp. 77–80, jul 1972.
2. I. Biederman, "On the Semantics of a Glance at a Scene," *Percept. Organ.*, pp. 213–233, 1981.
3. M. Bar, "Visual objects in context," *Nat. Rev. Neurosci.*, vol. 5, pp. 617–629, aug 2004.
4. S. E. Palmer, "The effects of contextual scenes on the identification of objects," *Mem. Cognit.*, vol. 3, no. 5, pp. 519–526, 1975.
5. A. Oliva and A. Torralba, "The role of context in object recognition," *Trends Cogn. Sci.*, vol. 11, no. 12, pp. 520–527, 2007.
6. A. Krizhevsky, I. Sutskever, and G. E. G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *NIPS*, pp. 1097–1105, 2012.
7. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks," *arXiv Prepr. arXiv*, p. 1312.6229, 2013.
8. R. Girshick, J. Donahue, T. Darrell, J. Malik, U. C. Berkeley, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2–9, 2014.
9. C. Galleguillos and S. Belongie, "Context based object categorization: A critical survey," *Comput. Vis. Image Underst.*, vol. 114, no. 6, pp. 712–722, 2010.
10. S. Divvala, D. Hoiem, J. Hays, A. Efros, and M. Hebert, "An empirical study of context in object detection," in *CVPR 2009*, pp. 1271–1278, IEEE, jun 2009.
11. A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in Context," *Proc. IEEE Int. Conf. Comput. Vis.*, 2007.
12. K. Murphy, A. Torralba, and W. Freeman, "Using the forest to see the trees: a graphical model relating features, objects and scenes," *Adv. Neural Inf. . . .*, vol. 53, no. 3, pp. 107—-114, 2003.
13. R. Perko and A. Leonardis, "A framework for visual-context-aware object detection in still images," *Comput. Vis. Image Underst.*, vol. 114, no. 6, pp. 700–711, 2010.
14. R. Mottaghi, X. Chen, X. Liu, N.-g. Cho, S.-w. Lee, R. Urtasun, and A. Yuille, "The Role of Context for Object Detection and Semantic Segmentation in the Wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 891–898, 2010.
15. B. Alexe, N. Heess, Y. Teh, and V. Ferrari, "Searching for objects driven by context," *NIPS*, pp. 1–9, 2012.
16. A. Shrivastava and A. Gupta, "Contextual priming and feedback for faster R-CNN," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9905 LNCS, pp. 330–348, 2016.
17. X. Chen and A. Gupta, "Spatial Memory for Context Reasoning in Object Detection," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, pp. 4106–4116, 2017.
18. S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks," 2015.
19. I. Biederman, R. J. Mezzanotte, and J. C. Rabinowitz, "Scene Perception : Detection and Judging Objects undergiong relational violations," *Cogn. Psychol.*, vol. 177, no. 2, pp. 143–177, 1982.
20. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv Prepr. arXiv1409.1556*, pp. 1–14, 2014.

21. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," *arXiv Prepr. arXiv1409.4842*, 2014.
22. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, pp. 770–778, 2016.
23. P. Viola and M. Jones, "Robust Real-time Object Detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 1–25, 2001.
24. C. Gu, J. J. Lim, P. Arbeláez, and J. Malik, "Recognition using regions," *2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work. CVPR Work. 2009*, vol. 2009 IEEE, pp. 1030–1037, 2009.
25. J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, pp. 154–171, sep 2013.
26. T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 936–944, 2017.
27. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, pp. 1137–1149, jun 2017.
28. E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2016.
29. R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1440–1448, 2015.
30. K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, pp. 2980–2988, 2017.
31. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.
32. J. Weston, S. Chopra, and A. Bordes, "Memory Networks," oct 2014.
33. X. Zeng, W. Ouyang, J. Yan, H. Li, T. Xiao, K. Wang, Y. Liu, Y. Zhou, B. Yang, Z. Wang, H. Zhou, and X. Wang, "Crafting GBD-Net for Object Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8828, no. c, pp. 1–16, 2017.
34. J. Li, Y. Wei, X. Liang, J. Dong, T. Xu, J. Feng, and S. Yan, "Attentive contexts for object detection," *IEEE Trans. Multimed.*, vol. 19, no. 5, pp. 944–954, 2017.
35. A. Shrivastava and A. Gupta, "Contextual priming and feedback for faster R-CNN," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9905 LNCS, pp. 330–348, 2016.
36. Walled Abdulla, "Mask R-CNN for Object Detection and Instance Segmentation on Keras and TensorFlow." Accessed on: 2017-12-01 [Online] Available: https://github.com/matterport/Mask_RCNN.
37. F. Chollet, "Keras: The Python Deep Learning Library." Accessed on: 2018-06-21 [Online] Available: https://keras.io, 2015.
38. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," *12th {USENIX} Symp. Oper. Syst. Des. Implement. ({OSDI} 16)*, pp. 265–283, 2016.
39. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, jun 2010.