Seeing without Looking: Contextual Rescoring of Object Detections for AP Maximization

Lourenço V. Pato[1] lourenco.pato@tecnico.ulisboa.pt
Renato Negrinho[2] negrinho@cs.cmu.edu
Pedro M. Q. Aguiar[1] aguiar@isr.ist.utl.pt

[1]Institute for Systems and Robotics / IST, ULisboa [2]Carnegie Mellon University

Abstract

The majority of current object detectors lack context: class predictions are made independently from other detections. We propose to incorporate context in object detection by post-processing the output of an arbitrary detector to rescore the confidences of its detections. Rescoring is done by conditioning on contextual information from the entire set of detections: their confidences, predicted classes, and positions. We show that AP can be improved by simply reassigning the detection confidence values such that true positives that survive longer (i.e., those with the correct class and large IoU) are scored higher than false positives or detections with small IoU. In this setting, we use a bidirectional RNN with attention for contextual rescoring and introduce a training target that uses the IoU with ground truth to maximize AP for the given set of detections. The fact that our approach does not require access to visual features makes it computationally inexpensive and agnostic to the detection architecture. In spite of this simplicity, our model consistently improves AP over strong pre-trained baselines (Cascade R-CNN and Faster R-CNN with several backbones), particularly by reducing the confidence of duplicate detections (a learned form of non-maximum suppression) and removing out-of-context objects by conditioning on the confidences, classes, positions, and sizes of the co-occurrent detections. Code is available at `https://github.com/LourencoVazPato/seeing-without-looking/`

1. Introduction

The convolutional backbone of current object detectors processes the whole image to generate object proposals. However, these proposals are then classified independently, ignoring strong co-occurrence relationships between object classes. By contrast, humans use a broad range of contextual cues to recognize objects [12], such as class co-occurrence statistics and relative object locations and sizes.

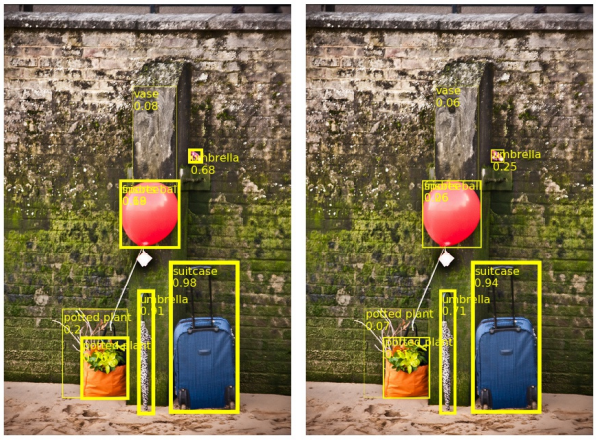


Figure 1: Detection confidences before (*left*) and after (*right*) contextual rescoring. High-confidence detections inform the topic of the image. False positives have their confidences reduced (only suitcase and the umbrella are in the ground truth). The line thickness of a bounding box is proportional to its confidence.

This observation motivates our work, where we exploit contextual information from the whole set of detections to inform which detections to keep.

Through an error analysis, we observe that current object detectors make errors that can be mitigated by the use of context. Errors can be ascribed to two types of problems: non-maximum suppression failing to remove duplicate detections (Figure 3); and local methods making insufficient use of context, e.g., when the object is visually similar to some class but the its context makes it unlikely (Figure 4).

We first study how to improve AP by rescoring detections while keeping the same location and class (Section 4.1). The insight is that detections with higher IoU count as true positives for more IoU thresholds and therefore should be scored higher. These scores are induced with the knowledge of the ground truth labels and lead to im-

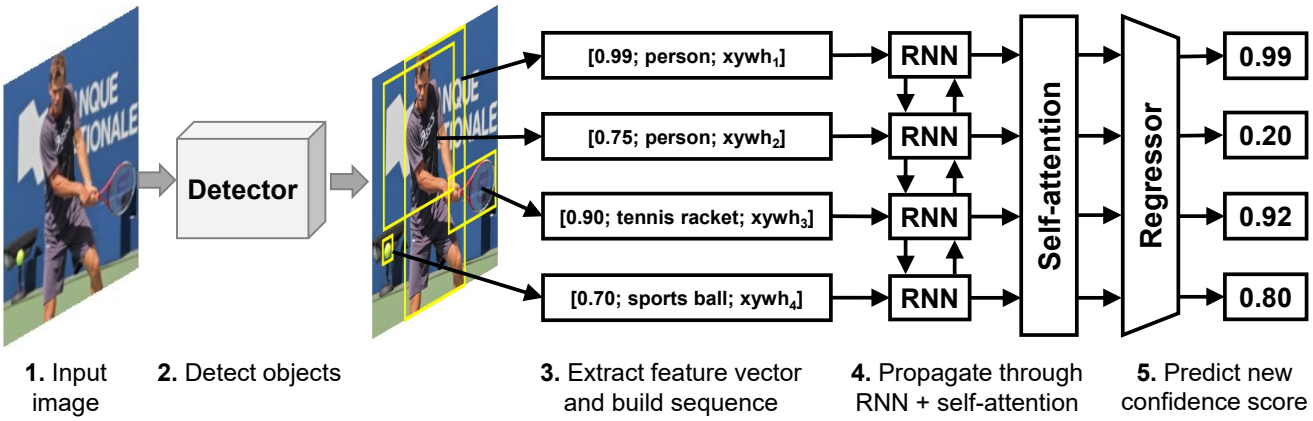


Figure 2: Overview of the contextual rescoring approach. **1-2.** A set of detections is collected by an object detector. **3.** A feature vector is extracted for each detection (by concatenating its confidence, predicted class, and coordinates). **4.** Detections are processed by an RNN with self-attention. **5.** A regressor predicts a new confidence for each detection.

provements of up to 15 AP on MS COCO `val2017` for detections produced by high-performance two-stage detectors (see Table 1). Given a fixed matching between predicted and ground truth detections, to maximize AP, it is optimal to assign score equal to the IoU with the ground truth to each matched predicted detection.

We propose a model to rescore detections of a previous detector using context from all detections in the image (see Figure 2). Each detection is represented by a feature vector with original confidence, predicted class, and bounding box coordinates. While the baseline detectors use only visual information, our model exploits non-visual high-level context, such as class co-occurrences, and object positions and sizes. We use recurrent neural networks (RNNs) with self-attention to induce the contextual representation. We train with a loss that pushes the model towards producing scores that maximize AP for the set of detections being rescored. Our approach is widely applicable as it does not use visual or other detector-specific features.

Results on MS COCO 2017 [22] (see Table 2) show that the proposed model improves AP by 0.5 to 1 across strong region-based baseline detectors (Faster R-CNN [27] and Cascade R-CNN [5]) and different backbone networks (ResNet-101 and ResNet-50 [16]). Although the improvements may seem modest, we consider very strong baselines and obtain consistent improvements across them. An analysis of the rescored detections (Section 5) shows that the model decreases the confidence for out-of-context and duplicate detections, while maintaining it for correct detections. Figure 1 illustrates this: false positives (sports ball, potted plant and umbrella) have their confidences reduced, while keeping high confidences for true positives (suitcase and umbrella). We present additional examples picked systematically, i.e., those with the largest overall confidence changes according to the cosine distance (see Appendix C).

We identify the following contributions of this work:

- A rescoring algorithm to maximize AP given fixed sets of predicted and ground truth bounding boxes. We show that for detections produced by current two-stage object detectors, there is an improvement of approximately 15 AP.
- A contextual rescoring approach that generates a new confidence for each detection by conditioning on the confidences, classes, and positions of all detections. Our model uses RNNs with self-attention to generate a contextual representation for each detection and it is trained to regress the values for AP maximization (i.e., IoU of the bounding box with the ground truth).

2. Related work

Two-stage detectors State-of-the-art object detectors [15, 14, 27, 5] rely on a two-stage approach: select image regions likely to contain objects (e.g., using fixed region proposal algorithms [15, 14] or a region proposal network [27]) and then classify each region independently. These approaches do not use non-visual contextual information.

Object detection with context Existing methods include context either in post-processing (as a rescoring or refinement step) [13, 8, 10, 11, 30, 12, 1] or in the detection pipeline [25, 3, 23, 21, 7, 26]. Existing work has incorporated context through multiple approaches such as logistic regression [12], deformable parts-based models [13, 25], latent SVMs [30], binary trees [10], graphical models [23], spatial recurrent neural networks [7, 26, 3] and skip-layer

connections [3]. Relation Networks [20] introduces a "Object Relation Module" that is incorporated into Faster R-CNN to capture inter-object relations and suppress duplicates. Other work captures context by using RNNs to process visual feature maps [21, 7, 26, 3]. Recently, [2] explored the utility of context by rescoring detections using non-visual context inferred from ground truths. They consider how to improve AP by rescoring and propose an heuristic rule based on the ratio of true and false positives. Their approach does not provide a rescoring model as they condition on ground truth information. To the best of our knowledge, we are the first to use a deep learning model that conditions on non-visual features (confidence, predicted class, and bounding box location) to rescore predictions generated by an arbitrary detector. Furthermore, our model is trained with a loss for AP maximization (see Section 4.1), which is developed based on the insight that better localized detections should be scored higher.

Non-maximum suppression NMS is a crucial component for removing duplicate detections. In addition to traditional NMS, Soft-NMS [4] reduces confidence proportionally to the IoU overlap, while learned NMS [18, 19] learns the NMS rule from data. Both learned NMS approaches use the same matching strategy used in evaluation and use a weighted logistic loss for rescoring (i.e., keep or remove a detection). This loss does not encode preference for detections with better localization. NMS approaches do not remove duplicate detections with different classes (Figure 3 *right*). By contrast, our approach conditions on all the predicted classes, confidences, and positions and therefore, our model can learn class, confidence and position-dependent suppression rules. Furthermore, we formulate a regression problem where the target is the IoU with ground truth such that better localized detections should be given a higher score. In Section 4.1, we compare our rescoring approach (matching and targets) with learned NMS approaches and show that there is large margin for improvement (Table 1).

3. Error analysis

We analyze the errors made by two strong detectors. For this analysis, we use the detections generated by MMDetection's [6] implementation of Faster R-CNN [27] and Cascade R-CNN [5] with a ResNet-101 [16] backbone. The backbone is pre-trained for ImageNet [28] classification and fine-tuned for object detection on COCO `train2017`[1]. Unless mentioned otherwise, all future analyses and examples will use results and examples from COCO `val2017` with Cascade R-CNN and a ResNet-101 backbone.

[1]For more information, please refer to the project's GitHub page `https://github.com/open-mmlab/mmdetection/`

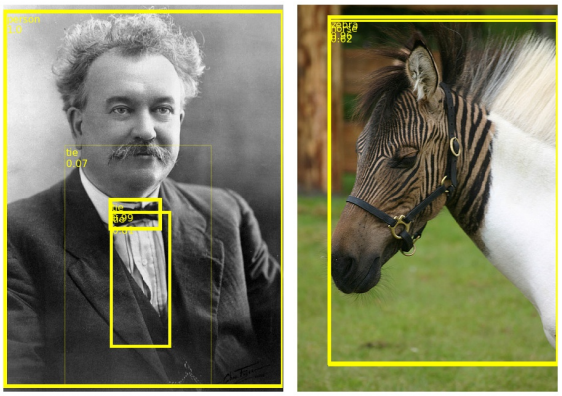


Figure 3: Duplicate detections illustrating failure cases of NMS. *Left:* Two high confidence detections of tie with low IoU. *Right:* Overlapping detections of horse and zebra.

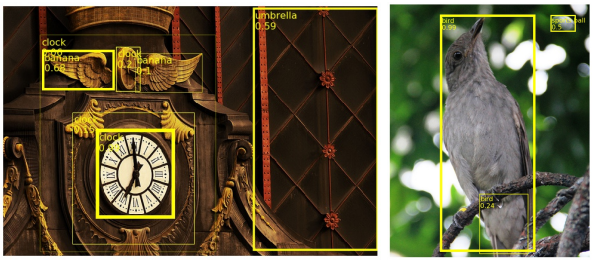


Figure 4: Failure cases of local non-contextual detection. *Left:* Banana and umbrella detected in a clock. *Right:* Sports ball detected in the tree background.

3.1. Detection errors

Localization errors and duplicate detections Localization errors occur when the predicted box has the correct class but low IoU with its ground truth, or when multiple boxes are predicted for the same object (duplicate detections). NMS removes detections whose confidence is lower than any other detection with the same object class and IoU above a threshold (typically 0.7 [27]). Unfortunately, NMS fails to remove duplicate detections with low IoU or with different classes, e.g., in Figure 3, a man with two ties (*left*) and overlapping detections of zebra and horse (*right*). A learned contextual NMS procedure should suppress these false positives as it is unlikely for a person to have two ties and for a horse and a zebra to overlap completely.

Confusions with background and dissimilar class In Figure 4, the detector finds unexpected objects such as an umbrella and a banana in a clock (*left*), and a sports ball in a tree (*right*). A learned rescoring model should be able suppress these false positives due their low probability in their context, e.g., by capturing class co-occurrences. Figure 5 illustrates class co-occurrences for the ground truth objects in

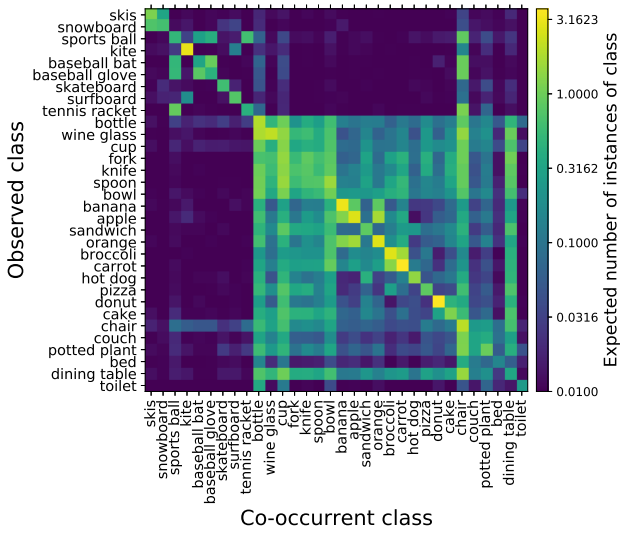


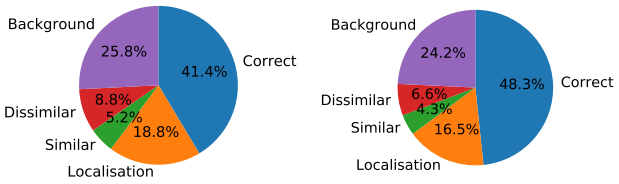
Figure 5: Co-occurrences for a subset of classes in COCO `train2017`. Each cell represents the expected number of instances of the co-occurrent class in an image that has at least one instance from the observed class. Related objects frequently co-occur: skis and snowboard; baseball bat, baseball glove and sports ball; cutlery. Rare co-occurrences are clear: sports objects and food rarely co-occur, bed and toilet appear with few other objects. There are strong diagonal co-occurrences: multiple classes frequently co-occur with themselves. Among these diagonal co-occurrences, toilet, bed and dining table are relatively weak.

`val2017`. Each cell represents the expected number of instances of the co-occurrent class to be encountered in an image given that an instance from the observed class is present. Using context, we can leverage these co-occurrences and decrease confidence for unexpected objects and increase it for detections that are likely correct. The figure with all class co-occurrences can be found in Appendix A.

3.2. Statistical error analysis

Current object detectors place a significant amount of confidence on false positives (Figure 6). We perform an analysis similar to [17], but because our rescoring approach does not change detections, only their scores, we change the metric to reflect the relative amount of confidence on each type of error. Detections are split into five types:

- **Correct:** correct class and location (IoU ≥ 0.5).
- **Localization error:** correct class but wrong location ($0.1 \leq$ IoU < 0.5); or correct location (IoU ≥ 0.5), but ground truth already matched (duplicate detection).
- **Confusion with similar class:** similar class (same COCO supercategory) and IoU ≥ 0.1.



(a) Faster R-CNN. (b) Cascade R-CNN.

Figure 6: Confidence distribution of Faster R-CNN and Cascade R-CNN (ResNet-101 backbone) on `val2017`.

- **Confusion with dissimilar class:** dissimilar class (different COCO supercategory) and IoU ≥ 0.1.
- **Confusion with background:** the remaining false positives (IoU < 0.1).

We iterated over detections by decreasing confidence and matched them with the ground truth with highest overlap, regardless of their class (by contrast, AP matches each class separately). In Figure 6, we accumulate the total confidence placed on each type of detection (i.e., higher confidence detections have higher weight). Both Faster and Cascade R-CNN detectors place the majority of confidence on false positives. In Section 5.2 we compare the same distributions after rescoring and show that our rescoring model reduces the fraction of confidence placed on false positives (Figure 7) and increases AP (Table 2).

4. Proposed approach: Contextual Rescoring

We consider a simple post-processing strategy: keep the class and location of the predicted bounding boxes and change only their confidence. Detections can be removed by driving their confidence to zero. We show that given a set of detections and ground truth annotations, we can rescore detections such that AP is greatly improved (Table 1).

4.1. Rescoring target

AP computation AP is computed for each class separately at various IoU thresholds $(0.5, 0.55, \ldots, 0.95)$. Increasing IoU thresholds reward better localization by requiring a detection to be closer to a ground truth to be considered true positive. For computing AP, we first determine true and false positives by matching each detection with a ground truth. COCO's matching strategy sorts detections by descending confidence order. Following this order, each detection is matched with the ground truth with the highest IoU if the following conditions are met: they have the same class, their IoU is greater or equal than the IoU threshold, and the ground truth was not yet matched. If no match is found, the detection is a false positive.

Then, the interpolated precision-recall curve is computed. Starting from the highest confidence detections, the curve $p(r)$ is traced by filling in the point that corresponds to the precision p at the current recall r for the running set of detections. This curve is then made monotonically decreasing by re-assigning the precision at each recall level as the maximum precision at higher recalls:

$$p_{\text{interp}}(r) = \max_{\tilde{r} \geq r} p(\tilde{r}). \tag{1}$$

AP approximates the area under the interpolated precision-recall curve by averaging the interpolated precision at 101 equally spaced recall levels. For a given class c and IoU threshold t, AP is given by

$$\text{AP}_t^c = \frac{1}{101} \sum_{r \in \{0, 0.01, \ldots, 1\}} p_{\text{interp}}(r, c, t). \tag{2}$$

The final metric for Average Precision is the average AP across the 80 object classes and at 10 different IoU levels,

$$\text{AP} = \frac{1}{10} \sum_{t \in \{0.5, 0.55, \ldots, 0.95\}} \frac{1}{80} \sum_{c \in \text{classes}} \text{AP}_t^c. \tag{3}$$

Greedy maximization of AP Given a set of detections and ground truths, we aim to find the confidences that yield the maximum achievable AP. To achieve this, we divide the maximization into *two steps*: *matching detections with ground truths* and *selecting the optimal score for each detection*. AP is a function of the ordering induced by the confidences but not their absolute values. Rescoring improves performance by reordering detections, assigning higher confidences to true positives than to false positives.

Matching detections with ground truths Matching a detection with a ground truth is non-trivial because several detections can refer to the same ground truth. COCO's AP evaluation computes a different matching for each IoU threshold $(0.5, 0.55, \ldots, 0.95)$. For our rescoring approach, a single matching must be found. A matching strategy that prioritizes detections by their confidence is penalized by AP when the highest confidence detection is not the best localized one. A high-confidence detection may be a true positive for lower IoU thresholds but become a false positive for higher thresholds. We propose an heuristic algorithm, Algorithm 1, that prioritizes IoU with ground truth (i.e., better localization) over confidence. Starting from the highest IoU threshold and gradually reducing it (Line 4), the algorithm iterates over all ground truths (Line 5) and matches each ground truth with the detection with the highest overlap (Line 9) from the set of unmatched detections from the same class and with IoU above the threshold (Line 7). We denote the sets of already-matched predicted detections and ground truth detections as $\hat{B}(M) = \{\hat{b} \mid (\hat{b}, b^*) \in M\}$ and $B^*(M) = \{b^* \mid (\hat{b}, b^*) \in M\}$, respectively.

Algorithm 1 Greedy matching by ground truth overlap

1: **Input:** Predicted detections $\hat{B}$, Ground truth B^*
2: **Output:** Matching $M \subseteq \hat{B} \times B^*$
3: $M \leftarrow \emptyset$
4: **for** $t \in \{0.95, 0.9, \ldots, 0.5\}$ **do**
5: **for** $b^* \in B^*$ **do**
6: **if** $b^* \notin B^*(M)$ **then**
7: $\hat{B}_{t,b^*} \leftarrow \{\hat{b} \in \hat{B} \mid \text{class}(\hat{b}) = \text{class}(b^*), \hat{b} \notin \hat{B}(M), \text{IoU}(\hat{b}, b^*) \geq t\}$
8: **if** $\hat{B}_{t,b^*} \neq \emptyset$ **then**
9: $b \leftarrow \arg\max_{\hat{b} \in \hat{B}_{t,b^*}} \text{IoU}(\hat{b}, b^*)$
10: $M \leftarrow M \cup \{(b, b^*)\}$

Matching	Target	C-101	C-50	F-101	F-50
	baseline	42.1	41.1	39.4	36.4
confidence	binary	47.8	46.9	44.8	42.9
	IoU	55.4	54.5	52.8	51.0
localization	binary	48.6	47.6	45.8	44.1
	IoU	**55.8**	**54.9**	**53.4**	**51.7**

Table 1: Average Precision for the target rescored values on `val2017`. **C:** Cascade R-CNN, **F:** Faster R-CNN, **101:** ResNet-101, **50:** ResNet-50. These rescoring results are computed from ground truths and predictions so they represent improvements achievable by an oracle.

Optimal confidence values For a fixed matching, optimal rescoring orders detections such that those with higher IoUs have higher confidences. This ordering ensures that better localized detections have higher priority in AP's matching algorithm. Our proposed target confidence y^* is the IoU with the matched ground truth for true positives and zero for false positives:

$$y_{\hat{b}}^* = \begin{cases} \text{IoU}(\hat{b}, b^*) & \text{if } \hat{b} \in \hat{B}(M), \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

for $\hat{b} \in \hat{G}$ and b^* is such that $(\hat{b}, b^*) \in M$.

Target AP Table 1 compares the baseline AP obtained by Faster and Cascade R-CNN architectures (using ResNet-101 and ResNet-50 backbones) with the AP obtained if the detections are rescored using the proposed matching algorithms and target confidences. Results are computed from the predictions and ground truths so they are only used to compute improved targets for training models. Combinations in Table 1 correspond to whether bounding boxes are greedily matched by the original confidence or IoU and

whether the target confidence is binary (one if matched and zero otherwise) or its IoU with the ground truth.

Our matching strategy (Algorithm 1) shows an improvement (ranging from 0.5 to 1.5) over a matching strategy that prioritizes confidence. Our target rescoring is around 8 AP better than the training target used by learned NMS approaches [18, 19] (which use binary targets and confidence matching) and shows that large improvements (up to 15 AP) are possible just by rescoring detections. In the following section, we train a rescoring model that uses contextual information to predict these target confidences.

4.2. Model architecture

We incorporate context to rescore detections produced by an earlier object detector (see Figure 2). The set of detections is mapped to a *sequence* of features $\mathbf{x} \in \mathbb{R}^{L \times N}$ that is fed to our model that computes the rescored confidences $\hat{\mathbf{y}} \in \mathbb{R}^{L}$. Each rescored confidence in $\hat{\mathbf{y}}_i$ is generated by conditioning on $\mathbf{x}$ (i.e., the whole set of detections).

Feature extraction A feature vector containing the original predicted confidence, class and location, is extracted for each detection in the image (see Equation 5). Together, they form a contextual representation for the set of detections. For MS COCO, the extracted feature vector is a 85-dimensional ($N = 85$) for detection i is given by

$$\mathbf{x}_i = [score_i] \oplus [\text{one\_hot}\,(class_i)] \oplus \left[\frac{x_i}{W}, \frac{y_i}{H}, \frac{w_i}{W}, \frac{h_i}{H}\right], \tag{5}$$

where $\oplus$ denotes vector concatenation, x_i, y_i are the coordinates of the top left corner of the detection bounding box, w_i, h_i are its width and height, and W, H are the width and height of the image. Features $score_i$ and $class_i$ are the detection confidence score and object class. Function one_hot creates a one-hot vector encoding for the object class. Detections are grouped by image and mapped to a sequence by sorting them by decreasing confidence. Sequences are padded to length 100 (the maximum number of detections often outputted by a detector).

Recurrent neural networks The proposed model uses a bidirectional stacked GRU [9] to compute two hidden states $\overrightarrow{\mathbf{h}_t}$ and $\overleftarrow{\mathbf{h}_t}$ of size n_h, corresponding to the forward and backward sequences, that are concatenated to produce the state vector $\mathbf{h}_t$ of size $2n_h$. We stack n_r GRU layers. The bidirectional model encodes each detection as a function of past and future detections in the sequence.

Self-attention We use self-attention [29] to handle long range dependencies between detections which are difficult to capture solely with RNNs. For each element i, self-attention summarizes the whole sequence into a context vector $\mathbf{c}_i$, given by the average of all the hidden vectors in the sequence, weighted by an alignment score:

$$\mathbf{c}_i = \sum_{j=1}^{L} \alpha_{ij} \mathbf{h}_j, \tag{6}$$

where L is the length of the sequence length before padding, $\mathbf{h}_j$ is the hidden vector of element j, and α_{ij} measures the alignment between i and j. The weights α_{ij} are computed by a softmax over the alignment scores:

$$\alpha_{ij} = \frac{\exp(\text{score}(\mathbf{h}_i, \mathbf{h}_j))}{\sum_{k=1}^{L} \exp(\text{score}(\mathbf{h}_i, \mathbf{h}_k))}, \tag{7}$$

where $\text{score}(\mathbf{h}_i, \mathbf{h}_j)$ is a scoring function that measures the alignment between $\mathbf{h}_i$ and $\mathbf{h}_j$. We use the scaled dot-product [29] function as a measure of alignment:

$$\text{score}(\mathbf{h}_i, \mathbf{h}_j) = \frac{\mathbf{h}_i^\top \mathbf{h}_j}{\sqrt{L}}. \tag{8}$$

Regressor Our model uses a multi-layer perceptron (MLP) to predict a value for the rescored confidence for each detection. The regressor input is the concatenation of the GRU's hidden vector $\mathbf{h}$ and the self-attention's context vector $\mathbf{c}$. Our proposed architecture consists of a linear layer of size $4n_h \times 80$ with ReLU activation, followed by a linear layer of size 80×1 with a sigmoid activation layer to produce an score between 0 and 1.

Loss function We formulate rescoring as regression for the target motivated by AP maximization (Section 4.1). We use squared error:

$$\mathcal{L}(\mathbf{y}, \mathbf{y}^*) = \sum_{i=1}^{L} (\mathbf{y}_i - \mathbf{y}_i^*)^2, \tag{9}$$

where $\mathbf{y}$ are the rescored confidences, $\mathbf{y}^*$ is the target sequence computed by Algorithm 1 and Equation 4.

5. Experimental results

5.1. Implementation details

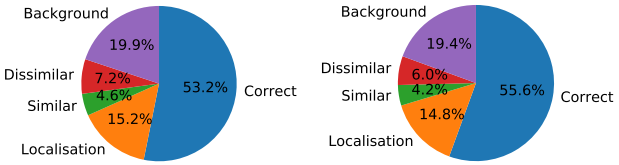
We ran existing detectors on MS COCO [22] to generate detections for `train2017` (118k images) for training, `val2017` (5k images) for model selection, and `test-dev2017` (20k images) for evaluation. As baseline detectors, we used MMDetection's [6] implementations of Cascade R-CNN [5] and Faster R-CNN [27] with ResNet-101 and ResNet-50 [16] backbones. We made our code available at `https://github.com/LourencoVazPato/seeing-without-looking` to easily train models on detections from arbitrary detectors.

Base model (backbone)	rescored	val2017 (5k) AP	AP^{50}	AP^{75}	AP^S	AP^M	AP^L	test-dev2017 (20k) AP	AP^{50}	AP^{75}	AP^S	AP^M	AP^L
Faster R-CNN (ResNet-50)		36.4	58.4	39.1	21.6	40.1	46.6	36.7	58.8	39.6	21.6	39.8	44.9
	✓	**37.4**	**60.0**	**40.1**	**21.8**	**40.7**	**48.7**	**37.4**	**60.2**	**40.3**	**21.8**	**40.4**	**46.1**
Faster R-CNN (ResNet-101)		39.4	60.7	43.0	22.1	43.6	52.0	39.7	61.4	43.2	22.1	43.1	50.2
	✓	**39.9**	**61.6**	**43.5**	**22.4**	**43.8**	**53.0**	**40.1**	**62.2**	**43.5**	22.1	**43.4**	**50.8**
Cascade R-CNN (ResNet-50)		41.1	59.3	44.8	22.6	44.5	54.8	41.5	60.0	45.2	23.3	44.0	53.1
	✓	**41.8**	**60.2**	**45.3**	**23.1**	**45.1**	**56.0**	**42.0**	**60.7**	**45.5**	**23.5**	**44.7**	**54.2**
Cascade R-CNN (ResNet-101)		42.1	60.3	45.9	23.2	46.0	56.3	42.4	61.2	46.2	23.7	45.5	54.1
	✓	**42.8**	**61.5**	**46.5**	**23.9**	**46.7**	**57.5**	**42.9**	**62.1**	**46.6**	**23.9**	**46.1**	**55.3**

Table 2: Performance results before and after rescoring. AP^S, AP^M and AP^L refer to small, medium and large objects.

top positives class	ΔAP	top negatives class	ΔAP
toaster	+ 3.2	wine glass	- 0.4
couch	+ 1.7	person	- 0.3
hot dog	+ 1.6	banana	- 0.3
frisbee	+ 1.4	elephant	- 0.3
microwave	+ 1.4	clock	- 0.3
baseball bat	+ 1.4	zebra	- 0.2
apple	+ 1.3	tennis racket	- 0.2
sandwich	+ 1.2	bicycle	- 0.1
pizza	+ 1.1	bus	- 0.1
cake	+ 1.1	giraffe	- 0.1

Table 3: Classes with highest changes in AP after rescoring.



(a) Faster R-CNN. (b) Cascade R-CNN.

Figure 7: Accumulated confidence distribution on val2017 after rescoring (compare to Figure 6).

Model hyperparameters The best hyperparameters found have hidden size $n_h = 256$ and number of stacked GRUs $n_r = 3$. We present model ablations in Appendix B.

Shuffling detections When a model is trained with input sequences ordered by descending confidence, it is biased into predicting the rescored confidences in the same decreasing order, yielding no changes to AP. We shuffle the input sequences during training with probability 0.75. As future work, it would be interesting to consider models that are invariant to the ordering of the bounding boxes.

Training We use Adam with batch size 256 and initial learning rate 0.003. When AP on the plateaus for more than 4 epochs on val2017 (i.e., the patience hyperparameter), the learning rate is multiplied by 0.2 and the parameters are reverted to those of the best epoch. Training is stopped if validation AP does not improve for 20 consecutive epochs.

5.2. Comparison with baselines

Table 2 compares performance before and after rescoring across different detectors. Rescored detections perform better, with consistent improvements ranging from 0.4 to 1 AP. Bigger objects achieve larger improvements ($\Delta AP^L > \Delta AP^M > \Delta AP^S$). Poorly localized detections have larger AP improvements ($\Delta AP^{50} > \Delta AP^{75}$).

In Figure 7, we compare the total accumulated confidence for each error type, obtained by adding the confidence for all detections in val2017 before and after rescoring (see Section 3.2). Correct detections have an increased share of the total confidence. Background and localization errors have a substantial reduction.

Class AP Table 3 shows the classes with the largest changes in AP for Cascade R-CNN with ResNet-101 backbone. Other detectors can be found in Appendix B. Most classes show a significant and consistent AP increase.

Generalization across architectures and backbones Different architectures have different error profiles. A rescoring model trained for one detector should hopefully generalize for other detectors. Table 4 compares the AP increase obtained by using a model trained on one detector and evaluated on a different one. Although improvements are not as large when tested with different baselines, all models show consistent improvements.

trained on `train2017`	evaluated on (`val2017`) F-50	F-101	C-50	C-101
F-50	**+ 1.0**	**+ 0.6**	+ 0.6	+ 0.5
F-101	+ 0.8	+ 0.5	+ 0.5	+ 0.5
C-50	+ 0.5	+ 0.1	**+ 0.6**	+ 0.6
C-101	+ 0.5	+ 0.3	+ 0.5	**+ 0.7**

Table 4: AP increase for models trained with different detectors (Faster R-CNN and Cascade R-CNN) and different backbones (ResNet-101 and ResNet-50).

5.3. Ablations

Training target Table 5 compares the AP achieved by our model when trained with a binary target and our proposed IoU target. The difference in AP confirms that using the IoU with the ground truth better aligns with AP and produces higher improvements, as expected from Table 1.

target	C-101	C-50	F-101	F-50
baseline	42.1	41.1	39.4	36.4
binary	42.5	41.6	39.6	37.3
IoU	**42.8**	**41.8**	**39.8**	**37.4**

Table 5: Average Precision on COCO `val2017` for binary and IoU training targets.

Feature importance Table 6 explores feature importance by training the models with subsets of all the features. The most important feature is the original confidence, while the least important ones are the bounding box coordinates. Not using the original confidence degrades AP by 2.2.

	conf.	class	coord.	`val2017` AP
baseline				42.1
all features	✓	✓	✓	42.8
no coordinates	✓	✓		42.4
no class	✓		✓	42.3
no confidence		✓	✓	39.9
just confidence	✓			42.2

Table 6: Feature importance. The original confidence contributes the most to performance.

Figure 8: Detections after rescoring. Duplicate detections are suppressed (compare to Figure 3).

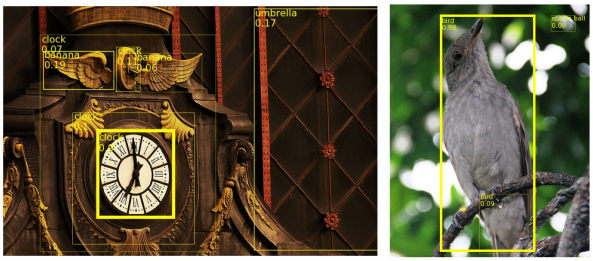


Figure 9: Detections after rescoring. False positives have been substantially suppressed (compare to Figure 4).

6. Conclusions

Current detectors make sub-optimal use of context, e.g., in a two-stage detector, each region is classified independently. Furthermore, NMS is an heuristic algorithm that fails to remove duplicates with low IoU or different classes. We observe that, to optimize AP, detections with better localization must be scored higher than poorly localized detections or false positives. Large increases in AP can be obtained solely by rescoring detections. We train a contextual rescoring model, consisting of a bidirectional GRU with self-attention followed by a regressor, with this AP maximization target on MS COCO. The experiments show that the model improves AP and reduces the total confidence placed on false positives across different baseline detectors. This model improves performance by 0.5 to 1 AP by exploiting solely non-visual context such as the confidences, classes, positions, and sizes of all detections in an image.

Acknowledgments This work was partially supported by LARSyS - FCT Plurianual funding 2020-2023. We thank the anonymous reviewers for helpful comments.

References

[1] Noa Arbel, Tamar Avraham, and Michael Lindenbaum. Inner-scene similarities as a contextual cue for object detection. *arXiv:1707.04406*, 2017.

[2] Ehud Barnea and Ohad Ben-Shahar. Exploring the bounds of the utility of context for object detection. In *CVPR*, 2019.

[3] Sean Bell, C. Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *CVPR*, 2016.

[4] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry Davis. Soft-nms–improving object detection with one line of code. In *ICCV*, 2017.

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: delving into high quality object detection. In *CVPR*, 2018.

[6] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv:1906.07155*, 2019.

[7] Xinlei Chen and Abhinav Gupta. Spatial memory for context reasoning in object detection. In *ICCV*, 2017.

[8] Zhe Chen, Shaoli Huang, and Dacheng Tao. Context refinement for object detection. In *ECCV*, 2018.

[9] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv:1406.1078*, 2014.

[10] Myung Choi, Joseph Lim, Antonio Torralba, and Alan Willsky. Exploiting hierarchical context on a large database of object categories. In *CVPR*, 2010.

[11] Ramazan Cinbis and Stan Sclaroff. Contextual object detection using set-based classification. In *ECCV*, 2012.

[12] Santosh Divvala, Derek Hoiem, James Hays, Alexei Efros, and Martial Hebert. An empirical study of context in object detection. In *CVPR*, 2009.

[13] Pedro Felzenszwalb, Ross Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 2009.

[14] Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[17] Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In *ECCV*, 2012.

[18] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. A convnet for non-maximum suppression. In *German Conference on Pattern Recognition*, 2016.

[19] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *CVPR*, 2017.

[20] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *CVPR*, 2018.

[21] Jianan Li, Yunchao Wei, Xiaodan Liang, Jian Dong, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Attentive contexts for object detection. In *IEEE Transactions on Multimedia*, 2016.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[23] Yong Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Structure inference net: Object detection using scene-level context and instance-level relationships. In *CVPR*, 2018.

[24] Minh-Thang Luong, Hieu Pham, and Christopher Manning. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

[25] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014.

[26] Jimmy Ren, Xiaohao Chen, Jian-Bo Liu, Wenxiu Sun, Jiahao Pang, Qiong Yan, Yu-Wing Tai, and Li Xu. Accurate single stage detector using recurrent rolling convolution. In *CVPR*, 2017.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *IJCV*, 2015.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[30] Ruichi Yu, Xi Chen, Vlad Morariu, and Larry Davis. The role of context selection in object detection. *arXiv:1609.02948*, 2016.