Recurrent Attention for Deep Neural Object Detection

Georgios Symeonidis
Aristotle University of Thessaloniki
Department of Informatics
gasymeon@csd.auth.gr

Anastasios Tefas
Aristotle University of Thessaloniki
Department of Informatics
tefas@aiia.csd.auth.gr

ABSTRACT

Recent advances in deep learning have achieved state-of-the-art results for object detection by replacing the traditional detection methodologies with deep convolutional neural network architectures. A contemporary technique that is shown to further improve the performance of these models on tasks ranging from optical character recognition and neural machine translation to object detection is based on incorporating an *attention mechanism* within the models. The idea behind the attention mechanism and its variations was to improve the information quality extracted for any confronted task by focusing on the most relevant parts of the input. In this paper we propose two novel deep neural architectures for object recognition that incorporate the idea of the attention mechanism in the well-known faster-RCNN object detector. The objective is to develop attention mechanisms that can be used for small objects detection as they appear when using Drones for covering sport events like bicycle races, football matches and rowing races. The proposed approaches include a class agnostic method that applies the same predetermined context for every class, and a class specific method which learns to include context that maximizes the class's precision individually for each class. The proposed methods are evaluated in the VOC2007 dataset, improving the performance of the baseline faster-RCNN architecture.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**;

ACM Reference Format:
Georgios Symeonidis and Anastasios Tefas. 2018. Recurrent Attention for Deep Neural Object Detection . In *SETN '18: 10th Hellenic Conference on Artificial Intelligence, July 9–15, 2018, Rio Patras, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3200947.3201024

1 INTRODUCTION

The current rapid advancements on the field of artificial intelligence have a massive impact on many other traditional fields of computer science, providing them with better results while radically changing the underlying problem structures and the deployed algorithms. Arguably the most favored field from this trend is computer vision and its branches, such as facial recognition, image segmentation, and the main focus of this paper, object detection. Especially, the deployment of robust and lightweight deep object detectors in Drones is crucial for their increased autonomy when used for covering specific sport events or in Cinematography in general. The objective on this paper is to investigate ways to introduce an attention mechanism to deep object detectors in order to increase their performance for challenging scenarios like those appearing in Drone Cinematography.

The advancements on object detection are mainly attributed to the application of deep learning architectures [15] incorporating deep convolutional neural networks (CNNs)[10] on the field. CNNs introduce non linearity in the architecture and perform feature extraction on the image, attempting to learn features which are equivariant to scale and translation. Current state-of-the-art object detection systems implement different variations of a standard detection pipeline which is comprised of a region proposal module and a detection module [6].

The region proposal module implements the artificial neural network equivalent of the common attention mechanism existing within the human visual object recognition system. As human object recognition benefits from focusing on particular regions of an object, the detection module's efficacy for an object can benefit from applying its classification and regression operations on features extracted from an area approximating that object. The region proposal module can be a conventional region proposal method, e.g. Selective Search [16] and Edgeboxes [21], or an independent neural network as was firstly proposed in the faster R-CNN model [12].

The faster R-CNN model closely represents the typical detection pipeline described above, incorporates a novel attention mechanism, and is one of the best performing object detection systems available today. It utilizes a deep convolutional neural network, which in the context of computer vision is referred as the *base network* ([14], [20]), to perform feature extraction on the image and shares these learned features between the region proposal network (RPN) and its detection module, eliminating the speed bottleneck emerging on most other object detection systems from applying a conventional region proposal method on the pixels of the image. This speedup is attributed mainly to the RPN module which is comprised of three parts, a set of convolutional filters that fit a predefined set of anchor boxes on the center of every location on the image where their sliding windows are applied, and two sibling fully connected layers which predict the probability that an object exist inside the anchor box and the regression to the nearest ground truth bounding box for that particular anchor box respectively. The *anchor boxes* are a set of predefined bounding boxes in various scales and aspect ratios that are used to detect objects within the RPN module. The regions of interest (ROIs) produced by the RPN module correspond to specific areas on the shared feature maps of the base network, thus a ROI

SETN '18, July 9–15, 2018, Rio Patras, Greece

ACM ISBN 978-1-4503-6433-1/18/07…$15.00
https://doi.org/10.1145/3200947.3201024

pooling layer [6] is necessary to facilitate the mandatory transition from areas on the image to features appropriate for detection in the detection module. Unfortunately, despite the speedup allowed by the RPN module, the delay introduced at this feature resampling stage renders the faster R-CNN model inappropriate for many real time applications.

The best performing model among models that do not suffer this delay is the single shot detector (SSD) model. SSD was proposed in [11] and is the first deep neural network based object detector that does not resample pixels or features for bounding box hypotheses and is as accurate as approaches that do so. The core idea behind SSD is predicting category scores and bounding box offsets for a fixed set of default bounding boxes using small convolutional filters applied to feature maps, similarly to the anchor boxes in the faster-RCNN model, but instead applied to multiple feature maps allowing for detection at multiple scales. SSD is deployed into two different variants, depending on the input image's dimensions. These are SSD300 and SSD512, which are trained and tested on input images of dimensions 300x300 and 512x512 respectively. SSD300 is the first real-time method to achieve above 70% mAP in the VOC2007 dataset [4]. SSD does not incorporate any attention mechanism, sacrificing accuracy for the gain of real time performance when deployed on high-end Graphics Processing Units (GPU).

Unlike SSD, the models we propose incorporate an attention mechanism and focus on the effects of context inclusion into the regions of interest proposed by an attention mechanism, in order to evaluate how the average precision per object class is affected from context inclusion. We designed models that expand and modify the faster R-CNN architecture aiming to increase the efficacy on the classification task of object detection. We propose two different methods for expanding an attention mechanism, a class agnostic and a class specific method.

In the class agnostic method, we extend the faster R-CNN's detection module by adding multiple neural streams, structurally identical to the original detection stream. The idea is to simulate the way humans pay attention to specific objects by observing the object parts in different scales and also the context of the object. This can be achieved if the neural network is able to analyze the object at different scales and cropped regions. The outcome of the object analysis at different scales should then be combined in order to produce a final prediction about the object class and regress the object size. To do so, a Long-Short Term Memory (LSTM) layer is added to the architecture and is trained to output the object label combining the outputs of three different neural streams. These neural streams have their input feature areas sampled from regions of interest where context inclusion has been applied. The size of the context inclusion is predetermined for each context inclusion branch, and is presented here in the form of a percentage along the newly included or excluded region's dimensions with respect to the initial ROI's corresponding dimensions. The highest level features extracted within each branch are then combined to surpass the accuracy of the faster R-CNN model on the object classification task.

The class specific method is developed based on the hypothesis that different classes need different attention areas for optimal object classification. For example, the class *train* might benefit from a larger context where the train lines will also be visible whereas a class that can be found in many different contexts (e.g., *person*) might require precise attention without any additional context. Thus, we modify the faster R-CNN model to allow it to learn the best extension factors for each object class by incorporating a context inclusion layer that calculates gradients with respect to the ROI's dimensions and uses them to approximate the optimal attention areas. As *extension factors* we refer to a pair of values for each class that extend or shrink the produced ROI's dimensions according to the object class of the ROI's included object. The two values of each pair apply context inclusion along the height and the width of the ROI respectively.

We test our models on the VOC2007 dataset and compare them with the faster R-CNN model which was used to derive our baseline results.

2 THE PROPOSED ATTENTION APPROACHES

This paper focuses on the effect of context inclusion in the provided regions of interest, either they are produced from a conventional algorithmic method as in [16] and [21], or they are the output of an independent neural network as proposed in the faster R-CNN model.

Generally, those methods aim to produce regions of interest that match the ground truth bounding boxes of the image's objects as closely as possible. Our hypothesis states that there can be certain classes of objects for which the optimal detection will be performed when the proposed regions of interest are extended or shrinked. The improvement can emerge either on the classification task, which intuitively seems more likely, or the bounding box regression or both.

We propose two different approaches regarding the context inclusion per class, a class agnostic where the class subtleties are not taken into account and our model applies a combination of values, on its context inclusion branches, hardwired from a predetermined set of indicative percentage values, and a class specific where for each class our model tries to learn the appropriate extension factors that maximize the average precision for the class. Our class agnostic model is presented in Figure 1 and our two proposed models for the class specific approach deployed on the training and testing phase are illustrated in Figures 4 and 3 respectively.

2.1 Class agnostic approach

Our proposed model extends the faster R-CNN architecture by inserting two parallel context inclusion neural streams in its detection module. Each stream applies a predetermined attention (i.e., context inclusion percentage) on every ROI proposed by the RPN module and contains a ROI pooling layer and two fully connected layers, identically structured to the layers in the faster R-CNN model's detection module. A LSTM layer [9] follows the last fully connected layers of each branch. The LSTM layer was invented as an attempt to tackle the *vanishing gradient* problem [2] by implementing a memory emulating mechanism which allows it to retain information for longer periods of time than a typical recurrent neural network [8]. In our model, it is responsible for combining the highest level features learned from the context inclusion branches and the original detection branch to improve the efficacy on the object classification

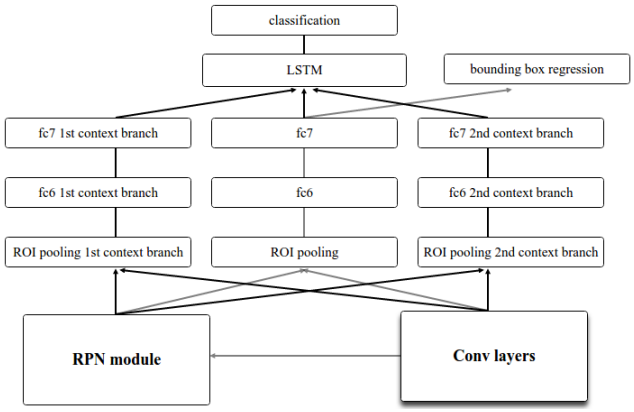


Figure 1: Our proposed class agnostic model deployed on both the training and the testing phase. The grey arrows point out connections already present in the original faster R-CNN model. For the sake of simplicity the two context inclusion layers preceding each context branch are ignored.

task, thus its output is fed to a typical fully connected output layer. The bounding box regression layer is trained on features from the last fully connected layer of the original detection branch, leaving the architecture of the faster R-CNN model intact within our model. A typical ROI proposal's progression within our highest performing class agnostic model until the classification phase is illustrated in Figure 2.

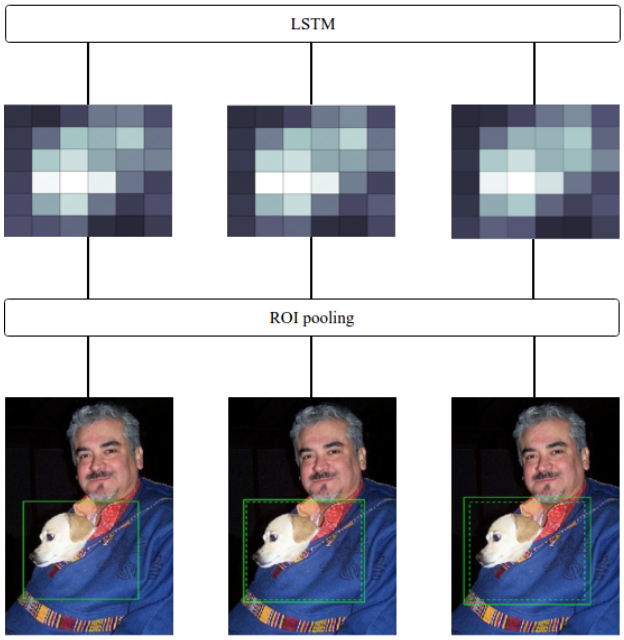


Figure 2: From left to right: Original ROI, 5% extended and 10% extended. Dashed outlines inside context included ROIs indicate the original ROI. For simplicity, the context inclusion layers on the two rightmost branches are ignored.

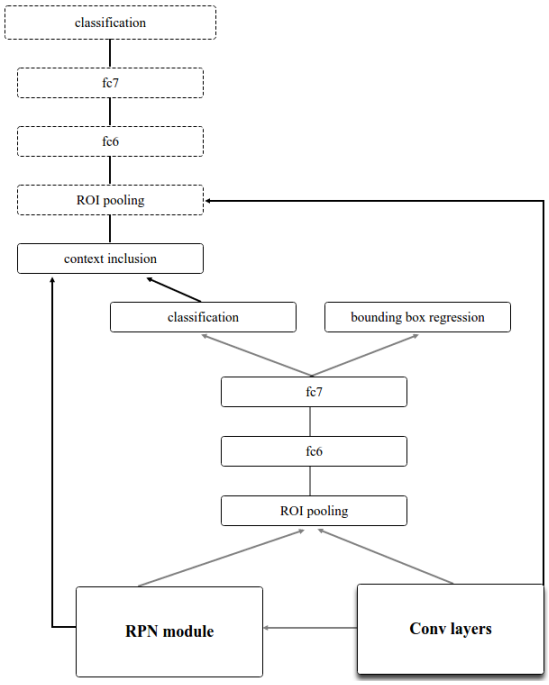


Figure 3: Our proposed class specific model used during the testing phase. The grey arrows point out connections already present in the original faster R-CNN model and dashed outlines indicate weight sharing between the layer and its equally named layer in the original faster R-CNN detection module.

2.2 Class specific approach

Our proposed model modifies the faster R-CNN architecture by replacing the ROI pooling layer in the detection module with a differentiable ROI warping layer introduced in [3]. The motivation behind the ROI warping layer was to refine the regions of interest proposed by the RPN module. Instead, we embed it in our model to make use of the gradients it returns with respect to the proposed ROIs' dimensions in our context inclusion layer to learn the optimal attention parameters (i.e., extension factors) (a_{i_h}, a_{i_w}) for each class i. As extension factors we define a set of pairs of values $\{(a_{i_h}, a_{i_w})\}$, each of whom corresponds to a class formally defined in the input image dataset. The pairs are used to calculate the optimal extension or shrinkage for the dimensions of the ROIs containing objects of any class, e.g., a pair of values $(a_{i_h} = 1.1, a_{i_w} = 1.2)$ for a class i would indicate that the ROIs containing objects of class i should have their height extended by 10% relative to the ROI's total height and their width extended by 20% relative to the ROI's total width. The extension of a dimension is equivalently interpreted as half of that extension on both the dimension's sides.

The updates of each class's extension factors are carried out **asynchronously**, that is, each update of an extension factor of a specific class takes place independently from other updates of the same extension factor for other classes. The attention parameters' update formulas that are used to learn the optimal extension factors

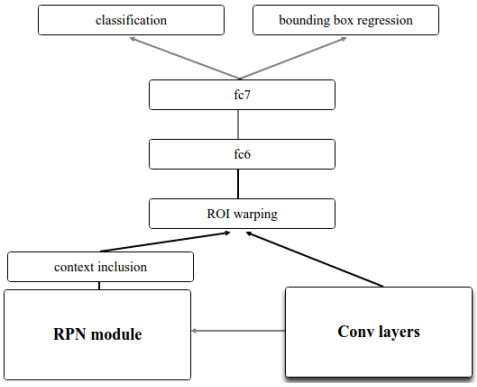


Figure 4: Our proposed class specific model used during the training phase in order to learn the extension factors for each class. The grey arrows point out connections already present in the original faster R-CNN model.

during training are the following:

$$a_{i_h}^{t+1} = a_{i_h}^{t} - n * \frac{dh}{height}$$

$$a_{i_w}^{t+1} = a_{i_w}^{t} - n * \frac{dw}{width}$$

where

- dh, the gradient of the total error with respect to the ROI's height.
- dw, the gradient of the total error with respect to the ROI's width.
- height, the height of the ROI.
- width, the width of the ROI.
- n, the learning rate used for the extension factors during training.

Our proposed class specific models deployed on the training phase and on the testing phase are presented in Figure 4 and Figure 3 respectively.

3 EXPERIMENTAL RESULTS

To examine how the context inclusion mechanism should be proposed we compared an ideal case consisting of images containing non overlapping bounding boxes of a single class, with a realistic scenario of overlapping bounding boxes containing every class existing in the original image. The remaining area of each image for both image groups was filled black. The dataset used for all the experiments of our project is the VOC2007 dataset which contains 20 different object classes, with the training phase being performed on the 5011 images of 12608 annotated objects and the testing phase performed on the 4952 images of 12032 annotated objects, comprising the official "trainval" and "test" sets of VOC2007 images respectively. A detailed description of the object classes distribution among the images is provided in [4].

The model used to perform the following preliminary set of experiments is the faster R-CNN deployed with the default parameter configuration when utilizing ZF [20] as its base network and trained using the *approximate joint training* algorithm, as described in [12].

We use as baseline for our comparisons presented in Table 1 and Table 2, the results provided by the faster R-CNN model realized with the aforementioned parameters and training algorithm.

Table 1: Models trained and tested on images containing non overlapping bounding boxes of a single class.

VOC classes	Baseline	NE	ATE	TTE
Aeroplane	57.9	73.4	64.7	58.2
Bicycle	67.2	84.3	57.9	51
Bird	52.3	71.1	49.9	45.2
Boat	37.1	70.9	41.8	37.9
Bottle	32.1	66	21.9	22
Bus	61.9	79.8	61	51.5
Car	71.7	85.7	68.8	59.7
Cat	63.2	79	68	62.1
Chair	34.1	70.8	29.1	23.6
Cow	54.8	62.6	58.5	50
Dining table	60.1	75.7	53.9	39.8
Dog	56.2	72.6	50.2	50.3
Horse	74.1	77.4	68.8	50.8
Motorbike	67.7	77.3	56.3	44.8
Person	61.6	78.1	58.3	47.6
Potted plant	29.1	73.9	34.7	24.6
Sheep	52.5	56.7	50.3	47.4
Sofa	48.9	72.3	50.3	44.5
Train	67.2	85.3	73.4	62.3
TV monitor	58.6	86	62.5	48.2
mAP%	55.4	74.9	54	46.1

The column acronyms, which describe each model's input images, stand for:

- **NE**: Not Extended. Each object on the image is cropped exactly on its ground truth bounding box, providing perfectly attended regions of interest to the detection module.
- **ATE**: A Third Extended. Each object on the image is cropped on an extended by a third version on each dimension of its original ground truth bounding box. The extension can be thought as a third on each dimension or a sixth on each dimension's side.
- **TTE**: Two Thirds Extended. Each object on the image is cropped on an extended by two thirds version on each dimension from its original ground truth bounding box. The extension can be thought as two thirds on each dimension or a third on each dimension's side.

We note that, when having a perfect attention model ([19], [18]), meaning to attend each object exactly on its ground truth bounding box, there is a significant increase on mAP for images which contain only one class with clearly separated bounding boxes for the class's objects, versus the realistic scenario where each image contains an arbitrary number of different classes' objects with overlapping bounding boxes (74.9% vs 70.2%). As the attention model becomes more erroneous though, both ATE and TTE models trained with all the original images' classes' objects outperform the ATE and TTE models trained on the heavily preprocessed images with 57.3% vs

Table 2: Models trained and tested on images containing multiple classes with overlapping bounding boxes.

VOC classes	Baseline	NE	ATE	TTE
Aeroplane	57.9	73.1	64.5	58.5
Bicycle	67.2	72.9	69.6	71
Bird	52.3	71.6	52.2	47.6
Boat	37.1	70.1	42.4	41.1
Bottle	32.1	45.4	26.9	28.9
Bus	61.9	77.4	61.3	59.3
Car	71.7	81.8	71.4	67.3
Cat	63.2	77.9	70.6	66.8
Chair	34.1	57.9	34.2	33
Cow	54.8	59	59.5	55.4
Dining table	60.1	70.9	63.3	59.5
Dog	56.2	67.5	55.3	58.6
Horse	74.1	76	72.2	70.6
Motorbike	67.7	73.8	64.1	62.2
Person	61.6	68	63.4	61.5
Potted plant	29.1	69.1	37.8	30.1
Sheep	52.5	57.2	49.8	50.4
Sofa	48.9	66.9	49.6	50.6
Train	67.2	84.1	72	63.6
TV monitor	58.6	84.1	66.3	57.4
mAP%	55.4	70.2	57.3	54.7

54% and 54.7% vs 46.1% respectively. This suggests that as long as the attention model is not perfect, context inclusion can be applied on the regions of interest as soon as those regions are produced by the region proposal method utilized within the architecture, without the need for a sophisticated preprocessing on the original images. Based on this result, the rest of the experiments conducted on our project applied context inclusion without preprocessing the dataset's images.

Using the same parameter configuration for the faster R-CNN model as used for the previous set of experiments, we trained a number of indicative models to test the effect of context inclusion in the object detection task for the 20 object classes contained in the VOC2007 dataset. Each model differs by the amount of extension or shrinkage of context space applied around the ground truth bounding boxes of the images' objects used for its training. Five indicative percentages were used to train each model, namely 5% (shrinkage and extension) and (10,15,20)% extension. The percentages refer to both sides of every dimension, e.g 10% means a 5% extension for the lower and upper sides calculated with respect to the ROI's height, and 5% extension for the left and right sides calculated with respect to the ROI's width.

The results for the aforementioned models on the VOC2007 "test" set are presented in Table 3, where it can be observed that specific classes benefit from context inclusion whereas other classes are negatively affected by it. For example, the class *sofa* goes from 48.9% to 56.2% average precision when the ground truth bounding boxes are extended by 10%, when in comparison the class *cat* is always negatively affected by context inclusion. Thus, our proposed deep architectures are trained in order to learn how the various attention areas should be combined using a LSTM layer in the class agnostic case, or how a class dependent attention can be learned in the class specific case.

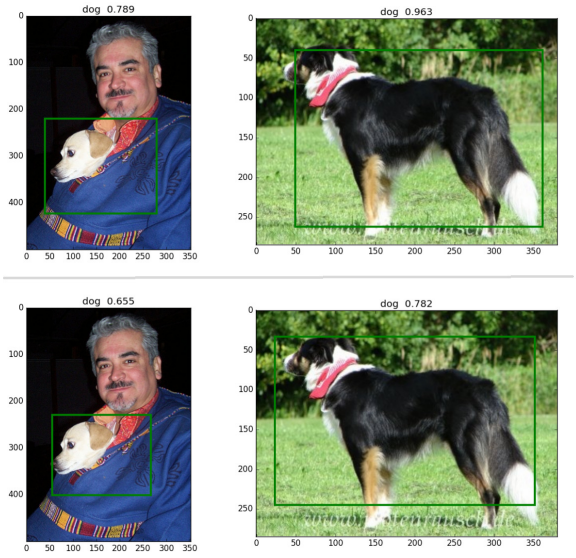


Figure 5: Indicative examples of a class where our model's precision exceeds the baseline. Upper row: Our model's detections for the class *Dog*. Lower row: Baseline's detections for the class *Dog*. Our model has improved confidence but the bounding boxes are less accurate.

Figure 6: Indicative examples of a class where our model's precision is worse than the baseline. Upper row: Our model's detections for the class *Bird*. Lower row: Baseline's detections for the class *Bird*. Our model has improved confidence but the bounding boxes are less accurate.

3.1 Class agnostic approach

We trained our proposed model applying in the context inclusion branches the combinations of the percentage values used to train

Table 3: Average precision on VOC2007 test set of models trained with applied ground truth bounding box context inclusion. The percentages of the relative change between the transformed and the original bounding boxes are recorded as the column headers.

VOC classes	Baseline	5% shrinked	5% extended	10% extended	15% extended	20% extended
Aeroplane	57.9	57.45	59.3	56.8	55.1	48.6
Bicycle	67.2	61.36	65.3	67	66.5	63.2
Bird	52.3	50.86	50	49.7	44.7	38.3
Boat	37.1	38.13	39.5	36.6	32.6	29.9
Bottle	32.1	27.43	28.9	22.7	12.8	12.4
Bus	61.9	61.11	61.4	62.7	62.3	62.5
Car	71.7	66.99	68.7	65.6	58.2	55.7
Cat	63.2	61.54	66.3	68.2	66.9	65
Chair	34.1	29.75	33.1	31.4	28.4	26.5
Cow	54.8	51.35	60.1	55.5	47.8	40.8
Dining table	60.1	58.47	56.3	60.5	62.9	61.7
Dog	56.2	61.8	58.6	61.7	61.1	59.2
Horse	74.1	70.96	73.8	74.7	75.2	69.3
Motorbike	67.7	66.43	63.3	66	64.1	60.3
Person	61.6	61.15	61.8	56.1	49.6	47.6
Potted plant	29.1	30.9	30.4	28.3	21.5	21.8
Sheep	52.5	52.97	50.6	47.6	40.1	38.4
Sofa	48.9	47.56	48.7	56.2	54.1	51.5
Train	67.2	67.34	64.2	66.6	67.2	64.1
TV monitor	58.6	56.9	58.5	57.8	47.3	44.1
mAP%	55.4	53.99	54.9	54.6	50.9	48

the models presented in Table 3. The results of the models achieving the highest mean average precision values are shown in Table 4.

The training of these models was conducted into three phases. In the first phase only the LSTM and and its following fully connected output layer incur weight updates. In the second phase both our context inclusion branches are trained in addition to the ones mentioned in the first phase, and in the final phase the whole network is trained end to end. Until the third phase, the faster R-CNN architecture contained within our model is initialized with weights learned from finetuning the alexNet classification network [10] trained on the imagenet dataset [13] on the VOC2007 "trainval" set.

Comparing the Tables 3 and 4 an important remark can be made, that the two single branch models achieving the best results among our modified models, which have their RPN's trained with 5% and 10% ground truth bounding box extensions respectively, when combined together into one model with three branches provide the highest mean average precision among our other best performing models. On a class level interpretation, these two models have the highest precision per class for the majority of the classes, suggesting a positive correlation between the per class performance of a single branch model and the precision improvement for the three branch model including it.

In general our model fulfills its purpose, that is, it increases the class confidence of correctly detected objects. The reduction of its average precision for some classes is mainly attributed to worsened performance on the bounding box regression task for these classes. Examples for the aforementioned statements are presented in Figures 5 and 6. An illustration of our model's overall improvement during the training phase compared to the faster R-CNN model is shown in Figure 7.

3.2 Class specific approach

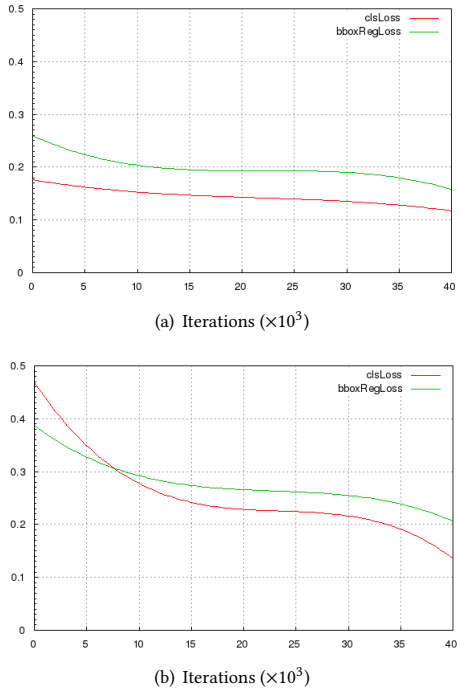
Our proposed model is trained using the default parameter configuration of the faster R-CNN model, when utilizing ZF [20] as its base network and trained using the *approximate joint training* algorithm, as described in [12]. During the training phase, we initialize our model with both extension factors for each class equal to values that correspond to the context inclusion percentage that provided the best precision for the class, shown in Table 3, in order to facilitate the convergence to the values which maximize that precision. This optimization scheme converges to the extension factors illustrated in Table 5.

The baseline results are derived by replacing the ROI pooling layer in the faster R-CNN architecture with the ROI warping layer and without applying context inclusion to the proposed ROIs. A comparison between the baseline and our proposed class specific model when tested with the optimized extension factors provided in Table 5 is presented in Table 6.

Comparing the Tables 3 and 6 it is derived that when the precision for a class is clearly benefited from context inclusion compared to the baseline, refining its extension factors can significantly improve its performance if these factors have been initialized near their optimal values during training (e.g *cat*, *dog*). In case context inclusion does not achieve notable improvement for the precision of a class, refining the class's extension factors starting from moderately good initial values leads to small insignificant changes of the

Table 4: Average precision on VOC2007 test set of models trained with the context inclusion configurations which provided the highest mAP values. The percentages of context inclusion on each branch are recorded as the column headers.

VOC classes	Baseline	(-5 , 5)%	(5 , 10)%	(10 , 15)%
Aeroplane	57.9	61.6	62	62.4
Bicycle	67.2	68.8	68.5	67.6
Bird	52.3	53.8	50.1	49.7
Boat	37.1	35.3	44.2	41.3
Bottle	32.1	29.8	34.2	32.1
Bus	61.9	62.6	63.7	66.2
Car	71.7	72.1	72.7	73.3
Cat	63.2	69.8	70	61.3
Chair	64.1	33.7	36.6	35.7
Cow	54.8	59.1	61.3	62.4
Dining table	60.1	57.4	57.9	58.5
Dog	56.2	63	64.8	58.5
Horse	74.1	73.6	73.6	74.8
Motorbike	67.7	66.2	66.3	66.2
Person	61.6	65.4	66.1	66
Potted plant	29.1	29.6	33.1	31.7
Sheep	52.5	54.1	51.4	53
Sofa	48.9	45.5	52.4	48.4
Train	67.2	68.1	70	65
TV monitor	58.6	55.6	60.9	59.8
mAP%	55.4	56.3	58	56.7



(a) Iterations ($\times 10^3$)

(b) Iterations ($\times 10^3$)

Figure 7: Classification and bounding box regression loss progression during the training phase. (a): Our model's loss during training. (b): Baseline's loss during training. Our model performs better than the baseline in both tasks.

precision around the baseline result, or even significant decrease of the precision (e.g *bird*, *potted plant*).

4 CONCLUSIONS

This paper proposes two approaches to incorporate context inclusion within the attention mechanism of a CNN based architecture specialized on object detection. The class agnostic approach suggests a model indifferent to individual class subtleties, and manages to outperform the state-of-art faster R-CNN model with respect to the mean average precision metric. The class specific approach utilizes previous knowledge about the effect of context inclusion on each class and learns to approximate the optimal extension factors for the class in order to maximize its performance.

Experiments conducted using the VOC2007 dataset indicate the improved performance achieved by the proposed approaches.

5 ACKNOWLEDGMENT

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 731667 (MULTIDRONE). This publication reflects the authorsâĂŹ views only. The European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] 2016. *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7776647

[2] Y. Bengio, P. Simard, and P. Frasconi. 1994. Learning Long-term Dependencies with Gradient Descent is Difficult. *Trans. Neur. Netw.* 5, 2 (March 1994), 157–166. https://doi.org/10.1109/72.279181

[3] Jifeng Dai, Kaiming He, and Jian Sun. 2016. Instance-Aware Semantic Segmentation via Multi-task Network Cascades, See [1], 3150–3158. https://doi.org/10.1109/CVPR.2016.343

[4] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* 88, 2 (01 Jun 2010), 303–338. https://doi.org/10.1007/s11263-009-0275-4

[5] Felix A. Gers and JÃijrgen Schmidhuber. 2000. Recurrent Nets that Time and Count.. In *IJCNN (3)* (2004-04-19). 189–194. http://dblp.uni-trier.de/db/conf/ijcnn/ijcnn2000-3.html#GersS00

[6] Ross B. Girshick. 2015. Fast R-CNN. *CoRR* abs/1504.08083 (2015). arXiv:1504.08083 http://arxiv.org/abs/1504.08083

[7] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR* abs/1311.2524 (2013). arXiv:1311.2524 http://arxiv.org/abs/1311.2524

[8] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. 2009. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 5 (May 2009), 855–868. https://doi.org/10.1109/TPAMI.2008.137

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9, 8 (Nov. 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. *SSD: Single Shot MultiBox Detector*. Springer International Publishing, Cham, 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

Table 5: Our proposed model's learned extension factors when it is initialized with the highest precision extension factors for each class among the tested models in Table 3.

VOC classes	a_{height}	a_{width}
Aeroplane	1.054	1.054
Bicycle	1.008	1.008
Bird	1.013	1.013
Boat	1.060	1.060
Bottle	0.983	0.983
Bus	1.094	1.094
Car	1.015	1.015
Cat	1.088	1.088
Chair	0.997	0.997
Cow	1.050	1.050
Dining table	1.146	1.146
Dog	0.955	0.955
Horse	1.148	1.148
Motorbike	1.009	1.009
Person	1.080	1.080
Potted plant	0.952	0.952
Sheep	0.947	0.947
Sofa	1.094	1.094
Train	0.952	0.952
TV monitor	1.000	1.000

Table 6: Average precision on VOC2007 test set of our proposed model trained with class specific context inclusion.

VOC classes	Baseline	Optimized extension factors
Aeroplane	51.7	60.4
Bicycle	63.4	63.3
Bird	47.3	47.5
Boat	36.2	34.9
Bottle	30.9	30.5
Bus	57.2	52.3
Car	69.4	67
Cat	47.1	61.2
Chair	30.7	28.9
Cow	53	52.6
Dining table	43.8	48.7
Dog	50.2	59.5
Horse	69.6	69.5
Motorbike	62.3	61.2
Person	61.7	60.5
Potted plant	29.8	27.2
Sheep	45.7	51
Sofa	43.2	41.6
Train	57.6	58
TV monitor	59.3	62.3
mAP%	50.5	51.9

[12] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *CoRR* abs/1506.01497 (2015). arXiv:1506.01497 http://arxiv.org/abs/1506.01497

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. 2014. ImageNet Large Scale Visual Recognition Challenge. *CoRR* abs/1409.0575 (2014). arXiv:1409.0575 http://arxiv.org/abs/1409.0575

[14] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014). http://arxiv.org/abs/1409.1556

[15] Leslie N. Smith and Nicholay Topin. 2016. Deep Convolutional Neural Network Design Patterns. *CoRR* abs/1611.00847 (2016). arXiv:1611.00847 http://arxiv.org/abs/1611.00847

[16] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. 2013. Selective Search for Object Recognition. *International Journal of Computer Vision* 104, 2 (2013), 154–171. https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013

[17] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and Tell: A Neural Image Caption Generator. (2014). http://arxiv.org/abs/1411.4555 cite arxiv:1411.4555.

[18] Wenguan Wang and Jianbing Shen. 2017. Deep Visual Attention Prediction. *CoRR* abs/1705.02544 (2017). arXiv:1705.02544 http://arxiv.org/abs/1705.02544

[19] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Francis Bach and David Blei (Eds.), Vol. 37. PMLR, Lille, France, 2048–2057. http://proceedings.mlr.press/v37/xuc15.html

[20] Matthew D. Zeiler and Rob Fergus. 2014. *Visualizing and Understanding Convolutional Networks*. Springer International Publishing, Cham, 818–833. https://doi.org/10.1007/978-3-319-10590-1_53

[21] C. Lawrence Zitnick and Piotr Dollár. 2014. Edge Boxes: Locating Object Proposals from Edges. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 391–405.