# RESCORING DETECTIONS BASED ON CONTEXTUAL SCORES IN OBJECT DETECTION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ERSAN VURAL ZORLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JULY 2019

Approval of the thesis:

**RESCORING DETECTIONS BASED ON CONTEXTUAL SCORES IN OBJECT DETECTION**

submitted by **ERSAN VURAL ZORLU** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _______________

Assist. Prof. Dr. Emre Akbaş
Supervisor, **Computer Engineering, METU** _______________

**Examining Committee Members:**

Assoc. Prof. Dr. Sinan Kalkan
Computer Engineering, METU _______________

Assist. Prof. Dr. Emre Akbaş
Computer Engineering, METU _______________

Assist. Prof. Dr. Hacer Yalım Keleş
Computer Engineering, Ankara University _______________

Date:

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Surname: Ersan Vural Zorlu

Signature :

# ABSTRACT

## RESCORING DETECTIONS BASED ON CONTEXTUAL SCORES IN OBJECT DETECTION

Zorlu, Ersan Vural
M.S., Department of Computer Engineering
Supervisor: Assist. Prof. Dr. Emre Akbaş

July 2019, 60 pages

To detect objects in an image, current state-of-the-art object detectors firstly define candidate object locations, and then classify each of them into one of the predefined categories or as background. They do so by using the visual features extracted locally from the candidate locations; omitting the rich contextual information embedded in the whole image. Contextual information can be utilized to complement the information extracted locally and thereby to improve object detection accuracy. Researchers have proposed many models that exploit scene-level and/or instance-level context by using non-local features from the same image. In this work, we propose models to improve object detection by utilizing contextual information embedded in the confidence scores of detections in the whole image without using any visual features. Our models use object-to-object spatial and scale-related relationships and work as a post-processing step that can be plugged into any object detector. Specifically, for a reference detection output by the base object detector, our model first defines a variety of spatial and scale-based regions relative to the location of the reference detection. Then, each of these regions is summarized by the confidence scores of detections inside it. Next, the confidence scores of the reference detection and the contextual confidence scores are processed by our models. We propose three variants based on multilayer perceptrons. We evaluate our models in conjunction with the state-of-the-art RetinaNet object detector on the widely used MSCOCO benchmark dataset, where we show that our models improve average precision by up to %1.8 points.

Keywords: Object Detection, Context, Object Recognition, Deep Learning, Artificial Neural Networks

# ÖZ

## NESNE ALGILAMA YÖNTEMLERİNDEN ELDE EDİLEN SEZİMLERİN SKORLARININ BAĞLAM BİLGİSİ KULLANILARAK YENİDEN HESAPLANMASI

Zorlu, Ersan Vural
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Dr. Öğr. Üyesi. Emre Akbaş

Temmuz 2019 , 60 sayfa

Modern nesne algılama yöntemleri, problemi çözmek için öncelikle verilen bir görüntüde aday nesne bölgeleri belirler, daha sonra bu aday bölgelerin görsel özniteliklerini kullanarak sınıflarını kestirmeye çalışır. Bu yöntemler, sadece aday nesne bölgelerinden elde edilen görsel öznitelikler üzerinde çalıştıklarından, resimdeki bağlam bilgisini gözardı etmektedirler. Nesne tanıma başarımını artırmak amacıyla bölgesel özniteliklere ek olarak resimlerdeki bağlam bilgisinden faydalanılabilir. Bu amaçla geliştirilen yöntemler, görüntü seviyesinde ve/veya nesne seviyesinde bağlam bilgisini aynı görüntüdeki alakalı başka bölgelerin görsel özniteliklerini de hesaplamaya katarak değerlendirmiş olurlar. Bu çalışmada, nesne tanıma başarımını artırmak amacıyla, aynı görüntüde tespit edilen diğer nesnelerin güven puanlarını bağlam bilgisi olarak kullanan yöntemler önerilmektedir. Bu çalışmada önerilen yöntemler, herhangi bir nesne algılama yönteminin sonuçları üzerinde uygulanabilir ve nesnelerin arasındaki konumsal ve ölçeğe dayalı ilişkileri kullanır. Açıklamak gerekirse, nesne algılama yöntemi tarafından tespit edilen her nesne için, o nesneye göreceli olacak şekilde çeşitli konumsal ve ölçeğe dayalı bölgeler belirlenir. Bu bölgelerin özet bilgisi, bölgenin içerisine düşen nesnelerin güven puanları kullanılarak çıkartılır. Referans nesnenin güven puanları ve tanımlanan bölgelerin özetleri işlenerek referans nesne için yeni güven puanları hesaplanır. Performans artırımını sağlayabilmek amacıyla bu çalışmada çok katmanlı algılayıcı tabanlı üç model önerilmektedir. Bu modeller RetinaNet modelinin sonuçları kullanılarak MSCOCO veri kümesi üzerinde değerlendirilmiş ve ortalama hassasiyet değerinin temel alınan RetinaNet modeline göre %1.8'e kadar artırıldığı gözlenmiştir.

Anahtar Kelimeler: Nesne Algılama, Bağlam, Nesne Tanıma, Derin Öğrenme, Yapay Sinir Ağları

To my family

# ACKNOWLEDGMENTS

I would like to thank my supervisor Dr. Emre Akbaş. He did not only guide me to accomplish this work, he also encouraged me when I was struggled. This work could not be completed without his constant support. It is an honor for me to be his student.

My dearest thanks to Duygu, who made this work possible with her support and love. She is the one who made me keep going when I was struggled and stressed.

Finally, I would like to express my gratitude to each member of my family for supporting all the way through my education and academic life.

# TABLE OF CONTENTS

CHAPTERS

# LIST OF TABLES

TABLES

# LIST OF FIGURES

FIGURES

# LIST OF ABBREVIATIONS

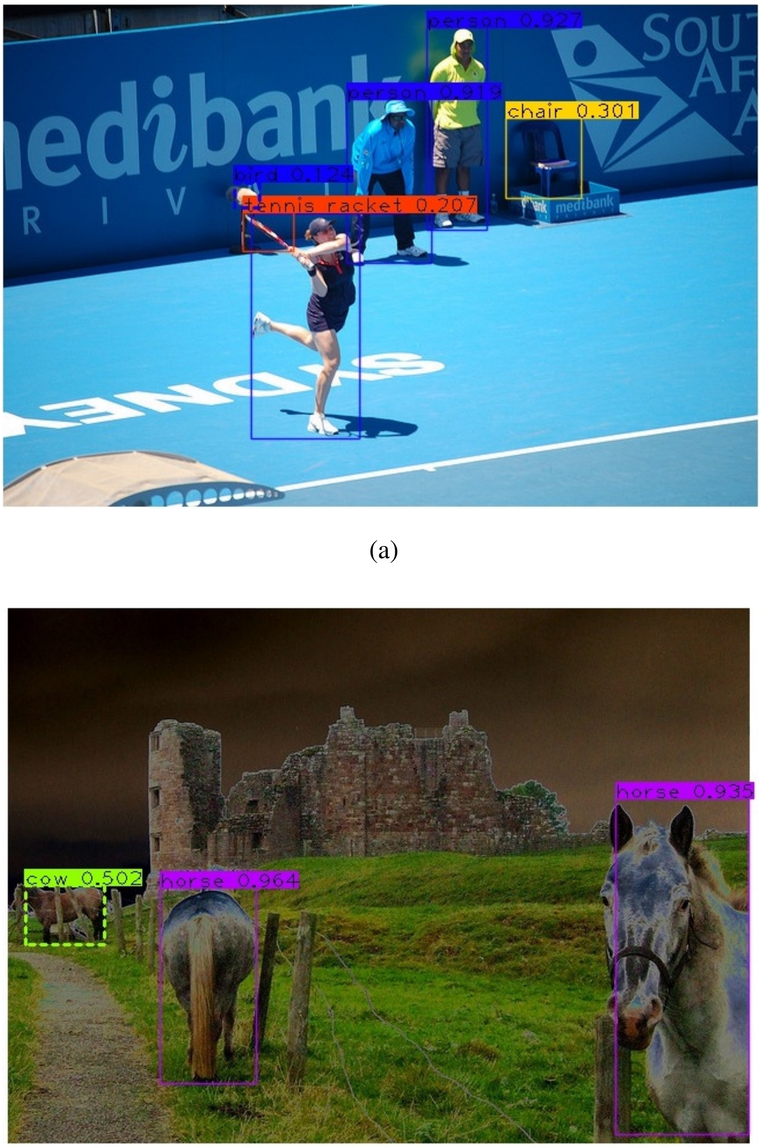
| | |
|---|---|
| AP | Average Precision |
| BAN | Boundary Aware Network |
| CNN | Convolutional Neural Network |
| COCO | Common Objects in Context |
| FC | Fully Connected |
| FPN | Feature Pyramide Network |
| GRU | Gated Recurrent Units |
| IoU | Intersection over Unioin |
| LRP | Localization Recall Precision |
| mAP | Mean Average Precision |
| MLP | Multilayer Perceptron |
| MMSE | Minimum Mean Square Error |
| NMS | Non Maximum Suppression |
| RoI | Region of Interest |
| RPN | Region Proposal Network |
| R-CNN | Region based Convolutional Neural Network |
| SIN | Structure Inference Network |
| SS | Selective Search |
| SSD | Single Shot Multibox Detector |
| VGG | Visual Geometry Group |
| YOLO | You Look Only Once |

# CHAPTER 1

# INTRODUCTION

Object detection is the problem of detecting object instances from a predefined set of classes and their locations in an image. This problem is challenging because unlike image classification, it requires to process localization of an excessive number of candidate object locations and then refine those candidate locations to match locations of ground truth objects precisely. Current state-of-the-art object detection methods mostly follow one of the two paradigms; one-stage object detection or two-stage object detection. Both of these methods make use of convolutional neural networks (CNNs) to extract visual features from images and show great success in performance and accuracy. However, they still make errors due to a various number of reasons including intra-class variation, not enough training data, low-resolution (small) objects, occlusion and changes in viewpoints. To exemplify such cases, some detection errors of state-of-the-art RetinaNet [2] object detector are demonstrated in Figure 1.1. In the first scene, a sports ball object is mislabeled as a bird. In the second scene, RetinaNet mislabels a horse object as a cow although it successfully labels other horse objects in the same scene. These objects are hard to detect by visual appearance only since they are very small or low-resolution objects, so it is not a surprise that RetinaNet mislabels them. If context information embedded in images such as spatial and scale-related relationships between objects are used, false labels in such cases could be eliminated.

When a human looks at a scene, s/he first scans the general structure to interpret it and then focuses on the objects in the scene. Humans classify occluded or low-resolution objects by making inferences based on the rest of the scene. For example, when we see tiny objects appearing in a scene with the sky in the background, we expect them



(a)

(b)

Figure 1.1: **Example erroneous detections of RetinaNet**. Mislabeled detections are drawn with dashed-line boundaries. Images are from the MSCOCO dataset [1]. (a) Sports ball is mislabeled as bird (b) Horse is mislabeled as cow with high confidence.

to be birds or aeroplane; or boats generally appear with the sea in the background. Similarly, when we see a relatively small object next to a keyboard, we expect that object to be a mouse. With the help of context information embedded in images, this type of inferences based on the surroundings of objects help us to interpret images and discover objects appear in it even if they are blurry or occluded. However, most of the currently popular object detectors try to classify bounding box regions based on only the visual features extracted from the region inside of the reference bounding box ignoring the rich contextual information embedded in images. Therefore, integrating context information to object detectors is currently a popular topic. Two types of context have been used in literature. The first type makes use of scene level context or context around the object while the second utilizes instance level object to object relationships. For example, the first type is used to infer that tiny objects in the sky are birds and the second type is used to infer that relatively small object next to a keyboard is quite likely a mouse. Most of the current context models are integrated to a common object detector training pipeline after region proposal step. For each region of interest (RoI), local features are combined with the features of context regions by applying various methods. Then, RoI classifications and bounding box regressions are evaluated using combined features instead of the local features that are the visual features extracted from RoI.

## 1.1 Proposed Method

In this work, we try to exploit context information embedded in images to improve the detection performance of popular object detectors. To this end, we utilize instance level context and use the detections predicted by RetinaNet. Our aim is to improve the detection performance of a base object detector by applying a post-processing step on its predictions so unlike other context models, our model is not trained end to end with the base detector, but it is lightweight and has a simple structure. The post-processing step re-scores each detection given scores, relative locations and relative scales of other detections in the same image. Specifically, for each detection of RetinaNet, we define multiple spatial and scale-based regions such as up, down, bigger, smaller, etc. with respect to that detection. Each region is summarized by first finding a subset of detections which are belong to that region and then taking maximum scores of those detections for each object category. Finally, summary scores for regions are concatenated with the scores of query detection and fed to a neural network to re-score the query detection. Thus, objects contradicting with the spatial or scale-related context are refined while scores of objects approved by context are increased. Figure 1.2 presents the processing pipeline of our proposed method.

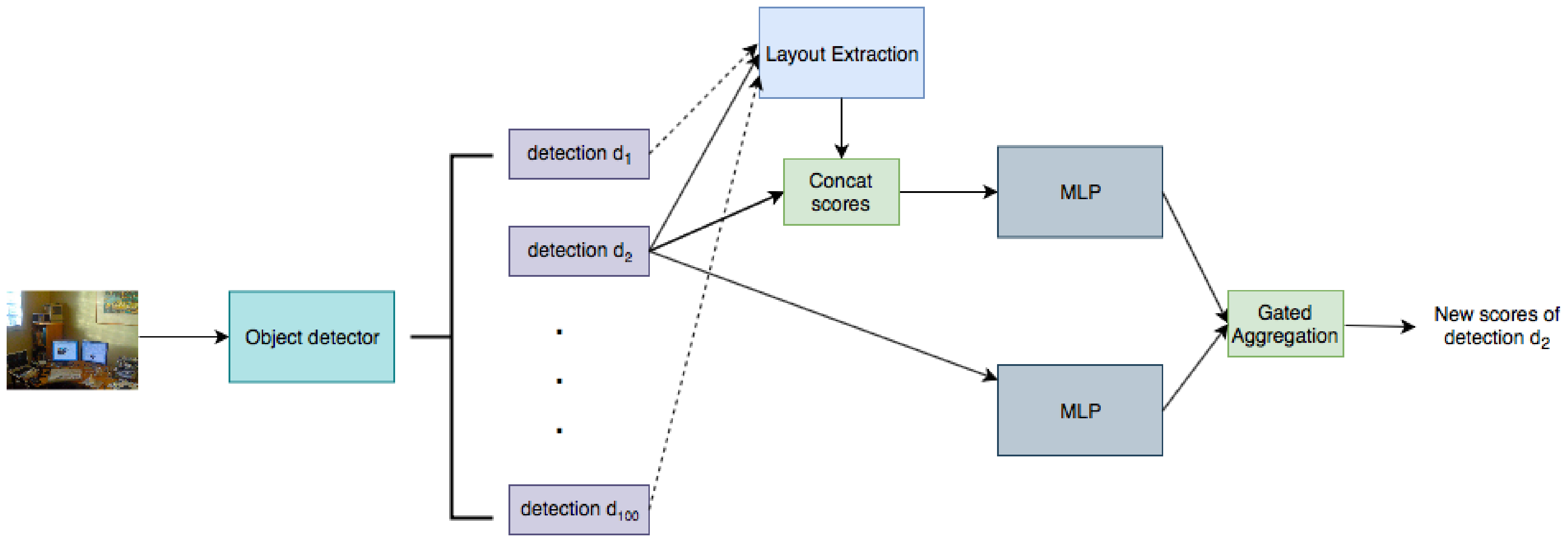


Figure 1.2: **Processing pipeline of our proposed method.** First, we obtain top 100 scored detections from the base object detector for each image. Score summaries of regions based on relative locations and relative scales of detections with respect to the query detection, which is d2 in this example, are extracted. Score summaries are concatenated with the scores of query detection. The concatenated vector is processed by MLP to re-score the query detection. Also, the scores of the query detection is fed to another MLP. The results of 2 MLPs are aggregated using a sigmoid gate. In this figure, we demonstrate re-scoring of detection 2, while the same procedure is applied for all detections in the same scene.

### 1.2 Contributions

The following items are the main contributions of this work:

- We propose a context model as a post processing step for object detection to improve performance.
- Our model utilizes spatial and scale-based relationships between objects in the same scene and can be plugged into any object detector.
- We evaluated our model on MS COCO and showed that it improves average precision (AP) of baseline by %1.8. Moreover, we analyze different false positive types obtained by both baseline model and our model.

### 1.3 Thesis Outline

The rest of this thesis is organized as follows.

In Chapter 2, we review the current state-of-the-art one-stage and two-stage detectors. Also, methods proposed to integrate explicit context to detectors in literature are discussed.

Chapter 3 describes how we use bounding boxes and scores of detections obtained from predictions of a detector to utilize context information embedded in images for object detection task.

Chapter 4 presents experimental results of our method, compares our results with the base object detector, analyzes false positive predictions of both our method and base detector. Additionally, we present qualitative successful and failure cases obtained using our context model.

Chapter 5 provides a brief summary and discussion.

# CHAPTER 2

# BACKGROUND AND RELATED WORK

In this chapter, advances in object detection task are presented. Different approaches to object detection in the literature are overviewed. Finally, methods that integrate contextual information to detectors are reviewed and compared.

## 2.1 Object Detection Methods

Object detection is the task of proposing object locations and map that locations to a predefined set of object classes in an image. For each candidate object location in an image, most detectors predict object proposals that consist of 4+C values where C is the number of classes; (x,y) coordinates, width, height of proposal and confidence score for all classes. Two approaches have been used for object detection task; one-stage and two-stage object detection. Both of these approaches try to refine initial candidate object locations by applying a regression network in order not to be obliged to predict exact ground truth object locations precisely in the beginning. However, two approaches basically diverge in the way of preparing initial candidate object locations. Two-stage detectors make use of various methods for region proposal such as Selective Search (SS)[3] to determine candidate object locations and then refine coordinates of those locations to approximate ground truth boxes. Region proposal phase narrows down candidate object locations to a small number. For example, Faster R-CNN [4] achieves state-of-the-art performance with only 300 proposals per image. Since object detector's performance depends on the quality of region proposals, any object location should not be missed as much as possible on region proposal step while filtering out majority of negative locations. One-stage detectors start prediction generally with a lot more number of predefined proposals compared to two-stage detectors, e.g. RetinaNet classifies ~100K proposals per image that are covering different scales and aspect ratios and regress them closer to the ground truth boxes. Since one-stage detectors use predefined proposals, there is no intermediate step such as region proposal. Therefore, one-stage detectors are mostly faster than two-stage detectors but with lower mean average precision (mAP) because of the limitations arise from predefined proposals.

Both of the approaches firstly extract rich feature representations from images. To extract those features, they make use of well known CNN architectures such as VGG-16. This CNN part is called the backbone network and it is mostly pretrained on a simpler task such as image classification and then fine-tuning is applied for object detection task. After training the backbone network for the classification task, this network is truncated before classification layers and an auxiliary network structure is added instead. Then, the new network is trained to regress proposals and predict class scores for proposals. The following section explains some of the currently popular methods of two approaches in more detail. Then, methods integrate context information on existing detectors are discussed.

### 2.1.1 Two-stage Object Detectors

Modern object detectors mostly follow a two-stage approach that first generates candidate object proposals in the first stage, then classifies and refines proposals by adjusting their coordinates in the second stage. Because of the intermediate steps such as region proposal and subsequent feature resampling, two-stage detectors are usually slower but show better detection performance compared to one-stage object detectors [5].

R-CNN[6], Fast R-CNN[7] and Faster R-CNN are the representative methods of two-stage approach.

R-CNN follows 3 steps to train a detector. First step is to generate class-independent region proposals by Selective Search. Second step does feature resampling by warping each region proposal into fixed size images and extracting fixed length feature vectors from those images by a large CNN. Finally, for each feature vector extracted in the second step, two sibling outputs are generated by fully connected layers; one is to predict classes of proposals, the other is to regress proposals that were generated by Selective Search in the first step.

Although R-CNN achieves great accuracy, it is slow in training and testing since it processes same CNN for each proposal individually. Also, generated features for each proposals are written to disk which requires lots of storage. Moreover, R-CNN can not be trained end-to-end which may lead to instabilities while training. To handle these drawbacks of R-CNN, Girshick proposed Fast R-CNN. Fast R-CNN preserves most of R-CNN structure other than the feature extraction step for proposals. Instead of processing CNN for each region of interest (RoI) obtained by selective search, Fast R-CNN processes whole image and patches are extracted from whole image features. Then, those patches are warped into fixed size features by RoI pooling. Since there are thousands of RoIs, this modification over R-CNN reduces training time by 3x.

Selective Search method runs on CPU and it is the main performance drawback of Fast R-CNN. Also, since accuracy of Fast R-CNN highly depends on the proposals generated by SS, too many proposals are generated in the first step not to miss any object position with different scales and aspect ratios. Faster R-CNN replaces SS method of Fast R-CNN with an internal convolutional neural network that is Region Proposal Network (RPN). RPN shares same convolutional features with detection framework, thus proposes a nearly cost free solution to extract ROIs. RPN works by running sliding windows over the convolutional feature map output by the last shared convolutional layer. For each sliding window location, k region proposals are predicted where k denotes different scales and aspect ratios. These region proposals are called anchors. Anchors are centered at the related sliding windows' center and anchors centered at same locations have different scale or aspect ratio. After sliding window, two sibling fully connected layer is applied; reg layer and cls layer. Reg layer outputs proposed box regressions, cls layer outputs objectness score of that anchor. Thus, for each location in sliding window, k*(4+2) outputs generated. Then, the classifier of Faster R-CNN only looks at the anchors having objectness score over a threshold. By replacing SS with RPN, Faster R-CNN derives performance gain over Fast R-CNN while achieving more accuracy.

### 2.1.2 One-stage Object Detectors

One-stage detectors are faster compared to two-stage detectors, but often with lower mAP. Since one-stage detectors do not have region proposal and use a single network for both classification and regression, they often achieve real time performance. Since speed is important for real time applications, several approaches are suggested in this area.

In You Look Only Once (YOLO)[8], Redmon et al. use a single CNN network which is trained on the entire image to predict all objects in an image simultaneously. They divide image into SxS grid cells. Each cell contains B bounding boxes. A bounding box consists of 5 predictions; (x,y) coordinates, width, height and confidence score. If a cell does not contain any objects, confidence scores of boxes inside that cell should be zero; otherwise, the confidence score should be intersection over union (IoU) between predicted box and the ground truth box. Class scores are predicted for each grid cell which means all B bounding boxes in the same cell have the same class predictions. Probability of an object of a category present in a bounding box is calculated by multiplying the confidence of the box with the class scores of the related grid cell. Consequently, they use a single network which uses entire image features to output SxSx(Bx5+C) shaped tensor where C is the number of object categories. Since YOLO uses a single network, it is extremely fast and has comparable mAP results with most of the two-stage object detectors. Also, they demonstrate that YOLO is highly generalizable, so it can be trained on real world images and results still will be fairly accurate. Since YOLO uses global image features, they claim it is able to encode contextual information implicitly. However, YOLO constraints with spatial limitations due to the limited number of bounding boxes and grid cells. YOLO may not detect nearby objects or object groups since a grid cell can contain only one object class.

Liu et al. propose Single Shot Multibox Detector (SSD)[9] that is also a single network like YOLO but starts with predefined bounding boxes called priors (similar to anchor boxes in Faster R-CNN) as prediction. SSD trains a single network to produce labels to priors and regress the priors closer to the ground truth boxes. SSD uses VGG-16 as backbone network. Fully connected layers are removed from the VGG-16 structure and some convolutional layers are added producing feature maps of different sizes and depth. Each feature map has multiple priors at different spatial locations, scales and aspect ratios as reference point to the ground truth boxes. For each prior, a discrete class probability vector and continuous regression values are predicted using related feature map. This way, feature maps are used to detect objects of different sizes and scales; earlier layers are used to detect smaller objects while last layers are used to detect larger objects. YOLO is limited since aspect ratios of predefined grid cells are fixed. SSD improves that by allowing more aspect ratios. Another improvement over YOLO is that SSD uses more convolutional layers for different scales of ground truth boxes that helps to detect objects at multiple scales better. When S is 7 and B is 2 in above YOLO calculation, YOLO predicts 7x7x2 = 98 confidence score for each class while SSD makes prediction for over 10K priors. Most of these priors are not a match to a ground truth box since most images contain only a few objects. This results in class imbalance problem on classification task for one-stage detectors. To solve class imbalance, SSD uses Hard Negative Mining.

To eliminate class imbalance emerges in one-stage detectors, Lin et al. proposes RetinaNet that uses focal loss. They reveal that the foreground-background class imbalance problem in one-stage detectors is the reason behind the lower accuracy against two-stage detectors. RetinaNet works on ~100k predefined anchors and most of these anchors are background i.e, do not match to any ground truth object. Unlike many other works like Huber Loss [10] focusing on eliminating or down weighting outlier data in loss calculation, RetinaNet down weights easy examples by modifying cross entropy loss in order to prevent easy examples dominating loss calculation so that classifier can focus on hard examples. Focal loss is based on standard cross entropy loss for binary classification. Standard cross entropy loss definition is given in Equation 1.

$$CE(p,y) = \begin{cases} -log(p) & \text{if } y = 1. \\ -log(1-p) & \text{otherwise.} \end{cases} \tag{1}$$

Where $y \in \{\pm 1\}$ is the ground truth label, and $p \in [0,1]$ is the estimated class probability by the model. With this loss calculation, even easily classified examples with high confidence have nontrivial effect on loss value. When there is too much easy examples, they dominate the loss calculation. To prevent easy examples from overwhelming the classifier subnetwork, RetinaNet modifies cross entropy loss with a modulating factor. Modified loss is given in Equation 2.

$$FL(p, y) = \begin{cases} -(1-p)^{\gamma} log(p) & \text{if } y = 1. \\ -p^{\gamma} log(1-p) & \text{otherwise.} \end{cases} \tag{2}$$

This way, if an example is classified correctly with high confidence, modulating factor approaches to zero and have smaller effect on loss, while the effect of misclassified examples increases since the modulating factor approaches to 1 as the confidence increases.

Other than focal loss, RetinaNet has a simple structure. As the backbone network, RetinaNet builds Feature Pyramid Network (FPN) [11] on the top of ResNet architecture. This way, a rich multi-scale feature pyramid is extracted from an image. By using multiple levels of the pyramid, they can better detect objects at different scales. After extracting features using FPN, RetinaNet regresses and classifies predefined proposals using two sibling subnetworks. Different than other one-stage detectors, they propose focal loss on classification task to handle huge class imbalance problem.

### 2.1.3 Context Information

Most two-stage object detectors consider proposals individually without making inference about contextual information explicitly. Detectors based on CNNs, such as YOLO, implicitly make use of context information since the receptive fields of neurons grow with depth, and covers the entire image in the last layers.

To analyze the effect of visual context for data augmentation in scene understanding, Dvornik et al. [12] propose a context driven data augmentation method. Basically, they augment training images by blending objects in existing scenes and then train a common CNN based detector, Faster R-CNN, on augmented dataset. Object instances to blend are extracted from images using the segmentation annotations of the same dataset. Unlike Dwibedi et al.'s work [13] where random object instances are copied and pasted to random locations, they blend object instances at locations consistent with context. To blend objects consistent with context, they don't paste random object instances at random locations, but train an explicit context model. They generate multiple subimages with different scales and aspect ratios from each image of dataset where a subimage fully encloses a subregion that is either a ground truth bounding box or a background region. The subregion content from the subimage is masked out. The context model is trained to predict the class of the object in the masked out area using the features of surrounding region in the subimage. After the context model is trained, it is used to blend objects consistent with context at random locations of images with different scales and aspect ratios. After blending artefacts, standard object detection training is applied on the augmented dataset. In order to prevent the detector to detect blended objects instead of the initial objects, several blending methods are used such as gaussian or linear blur. The authors showed improved results on the augmented dataset compared to results of original dataset which indicates that implicit utilization of context is available in CNN based detectors. Also, they compare their results with the work by Dwibedi et al. [13], and experimentally show that random placement of objects may hurt the detector performance while placement consistent with context increases the performance.

To make context explicitly available, several approaches are proposed recently. Current studies on context try to make use of context information mostly by integrating with the modern object detectors after the region proposal step. Given the RoI pooled features of each proposal, they use non-local features such as features of other proposals, features of whole scene or features of predefined regions as complementary information along with the local features. This new contextually rich features are used to classify and regress proposals instead of features that are extracted locally. Thus, recent studies on contextual information exploit object relationships, scene context or context around query proposal. Figure 2.1 presents the processing pipeline of current context models.

Chen and Gupta [14] integrate an external spatial memory module on Fast R-CNN to model object-to-object relationships. Fast R-CNN detects each object in parallel so there is no dependency between objects. In this work, Chen and Gupta model a

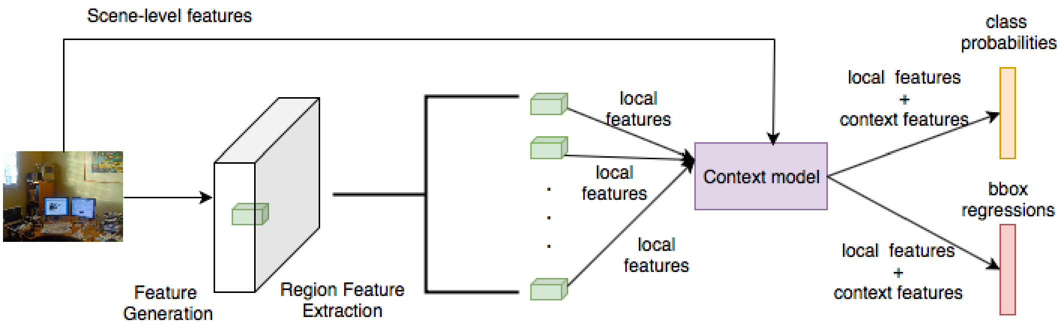


Figure 2.1: **Processing pipeline of object detection using context models.** Current context models are integrated into the object detection pipeline typically after the region proposal stage. They complement local features of proposals with the features extracted from whole scene, from predefined regions, or from other proposals in the same scene.

sequential structure so that the detection of an object depends on detection of other objects. To achieve that, a spatial memory is used. The memory is an image-like 2D structure that is initialized as empty. To update the state of memory, features of each proposal obtained by Fast R-CNN are used in order. They start with the proposal with the highest foreground confidence score and continue with other proposals in order by descending confidence scores. Each proposal updates the region of memory that corresponds to its projection in the image spatially. To detect first object that is the object having highest foreground confidence score, Fast R-CNN is used. The first object updates the state of the memory module, and the next object is detected using both memory module features and the Fast R-CNN module. The same iteration is applied until a predefined maximum number of object instances are detected.

Bell et al. [15] use RNNs to extract contextual information from an image. They use Fast-RCNN as baseline detector and "conv5" layer of VGG16 is fed to a model called Four-directional IRNN to compute context features outside of a RoI. In Four-directional IRNN, they place four RNNs that move in left, rigth, up and down across the image. They use a modified version of IRNN that works like an accumulator. To extract context features outside of a RoI both global and local, they stack 2 Four-directional IRNN as they claim that after the second IRNN, all output cells depend on all input cells. Inside the RoI, they use multiple layers of VGG16 to extract fixed size features at different scales. Then, the inside features concatenated with the outside context features and the resulting features are used for class prediction and bounding box regression.

Hu et. al [16] propose relation networks that integrates into a modern object detector after RoI pooling where features of possible object locations are calculated. Relation networks combine the features of the proposals obtained by a FC layer after RoI pooling with messages coming from other proposals. Multiple relations are defined between each proposal so each proposal sends multiple messages to other proposals. The messages are calculated using the visual features of source proposal. In order to integrate messages from multiple proposals, relation score between each proposal is calculated using the visual features and geometrical features of proposal pairs. Relation scores between proposal pairs are used to measure the effect of proposals on each other. Each message is multiplied with the relation score between proposals and new features of proposals are calculated using these messages coming from other proposals in the same scene. Thus, new features of a proposal consist of not only the visual features of the region it covers but also features from other proposals related geometrically and visually. The new features of proposals are used to predict object classes of proposals and bounding box regressions.

Liu et al. [17] exploit both scene-level and instance-level context. They model Structure Inference Network (SIN), an object detector that integrates a context method into a typical detection framework which is Faster-R-CNN for this work. To predict label of an object, unlike Faster-R-CNN, they use not only features from that object but also features from both other objects and scene. To combine features from other objects and scene with the features of the object in question, they build a graph structure for each individual image where objects are nodes and relationships between objects are edges. To adapt graph structure into a neural network, they use Gated Recurrent Units (GRU) in a novel way. Visual features are extracted for whole scene and for each object proposal detected by Faster R-CNN, and spatial features are extracted for each pair of proposals. Visual features are extracted for each object by a FC layer applied after RPN. Spatial features are encoded for each edge between objects as spatial relationship such as width and height ratio, or distance between pairs of proposals. Since some proposals have stronger relationships compared to others, e.g. a mouse is more important to a keyboard, a scalar value is calculated using visual features and spatial relationships of node pairs to estimate the effect of one node on another node. Then, visual features of each proposal are multiplied by that scalar before being passed as a message to other side of the edge to integrate a single message by max pooling between all messages coming from all nodes. The integrated message is used as an input to a GRU that takes features of the proposal in question as initial state. Another GRU also takes proposal features as initial state but the input is the whole scene features. Then the outputs of these 2 sets of GRUs are combined together to update the node state. Finally, instead of only own visual features of each node, integrated node features are used to predict object category and to refine proposal locations.

Chu and Cai [18] propose an ensemble of object detector that tries to improve performance of Faster R-CNN by exploiting scene-level and instance-level context. To obtain proposals based on local appearance, they use predictions of the baseline object detector. Then, they refine scores of proposals based on object relationships and global scene context. Like our method, they define layouts to encode spatial relationship between two objects:

- If two proposals do not have intersection, then the spatial relationships of them can be classified into far, up, down, left and right.
- Otherwise, they can be defined as inside, outside, up, down, left and right.

They use these layouts to estimate the log probability of an object with category label $i$ appears with an object category $j$ for a given layout which is learned from the statistic summary of the training set. To encode global scene context, they train a CNN on scene categorization task on Places2 dataset and the last layer of the CNN is used to represent scene context. They combine proposals based on local appearance with relationships between objects and global appearance of image to model a fully connected conditional random field (CRF) which is formulated in Equation 3.

$$E(X) = \sum_{i=1}^{N} \phi_{\mathrm{u}}(x_{\mathrm{i}}) + \omega_{\mathrm{p}} \sum_{i=1}^{N} \phi_{\mathrm{p}}(x_{\mathrm{i}}, x_{\mathrm{j}}, r) + \omega_{\mathrm{g}} \sum_{i=1}^{N} \phi_{\mathrm{g}}(x_{\mathrm{i}}) \tag{3}$$

where $\phi_{\mathrm{u}}(x_{\mathrm{i}})$ is called unary potentials and are the prediction scores of base detector for proposal $i$ of image $x$. $\phi_{\mathrm{p}}(x_{\mathrm{i}}, x_{\mathrm{j}}, r)$ is called the pairwise potentials and are estimation of log probabilities that represents object relationships mentioned before. $\phi_{\mathrm{g}}(x_{\mathrm{i}})$

is called global potentials and represents the global context as mentioned before. $N$ is the number of proposals while $\omega_{\mathrm{p}}$ and $\omega_{\mathrm{g}}$ are the weights of the pairwise and global potentials respectively.

Li et al. [19] propose a method that utilizes both local context surrounding a proposal and global context. They claim that surrounding of an object (e.g., "road") and discriminative parts of objects (e.g., "wheel") help to infer the category class of an object (e.g., "car"). Global context also provides useful clues, e.g., objects such as person, road, or another car are usually co-occur with a target object of car. However, not all global regions are helpful for classifying objects, so they try to extract only positive global context by applying an attention model. To incorporate both local and global context to detection pipeline, they add two subnetworks after feature map extraction of an image; one is to collect local information around object proposals, the other one is to select useful information from global image to help classifying proposals. The first subnetwork is not much different from original detection pipeline; instead of using only proposal features, the bounding box proposals are scaled with three predefined factors and feature representations of these new bounding boxes are concatenated with the original features. After concatenation, dimension reduction is applied by a convolution operation to match the original detection pipeline feature shape and the reduced features are fed to FC layers. The second subnetwork first pools the image feature map to a fixed size representation and then a recurrent attention model is applied to detect useful regions from global view. Then, attention map is fed into FC layers. Finally, the result of first network is used for bounding box regression and the results of two subnetworks are concatenated to predict object classes.

Chen et al.[20] propose a context refinement algorithm that augments the existing refinement procedure of two-stage object detectors. The algorithm can be interpolated in any two-stage object detector after the existing refinement procedure. The first step in their algorithm is to select surrounding regions of a proposal that may carry useful context. To select useful context regions, they consider IoU between proposal and other regions. Only regions having IoU greater than a threshold and label prediction same as the proposal are selected. Second step is to aggregate features of selected regions to form a unified representation based on an adaptive weighting strategy; visual features of each selected region are multiplied by a weight obtained by multiplying confidence scores of the region with the IoU value between region and the proposal. Then, context visual features are summed and this sum is normalized by the sum of the weight values. Lastly, context refinement is applied with a FC layer based on unified context feature representation and the proposal's own visual features.

Kim et al. [21] propose a context model that uses manually picked regions as context unlike previously mentioned methods that include other proposal regions as context. Since most of the two-stage object detectors classify objects only looking the features of the proposals, misaligned proposals may lead to incorrect classification. To prevent detection difficulties arise from misaligned object proposals, Boundary Aware Network (BAN) enhance detection accuracy by exploiting additional visual information embedded in boundary regions of object proposals. They define 10 boundary contexts from three types of boundary contexts; side, vertex, in/out-boundary context. The RoIs for side context are centered left, right, up and down sides of the proposal. The RoIs for vertex context are centered at each vertex of the proposal. The RoIs for the in and out-boundary contexts are defined as a half size region and double size region centered at the center of the proposal. Unlike previously mentioned methods, they do not aggregate context features with the proposals features; instead they train 11 subnetworks corresponding to 10 context regions and the original proposal. To classify and align each object proposal integrating with the baseline model, they aggregate results from 10 different subnetworks corresponding to boundary context regions and a subnetwork from the original proposal. This way, they not only integrate missing parts due to localization, but also exploit contextual relations between close objects such as person on a horse.

Arbel et al. [22] also propose a model that uses a subset of proposals as context instead of using all proposals. They also use proposals obtained by Fast R-CNN and refine their scores with the help of other proposals. Unlike previously mentioned approaches that integrate to detection pipeline after RoI pooling, they update only scores of proposals calculated by detector using scores of other proposals that are visually similar. They select proposals visually similar to target proposal as contextual regions. To select visually similar proposals, they calculate a distance value between proposals by using their color histogram and texture. Then the proposals having lower distance are selected. After visually similar proposals called supporters are selected for each target proposal, the score of the target proposal is updated using its own score, the scores of the supporter proposals, and the visual distance calculated in previous step between the target proposal and the supporter by using minimum mean square error linear estimator (MMSE).

Bozcan et al. [23] extend Boltzmann Machines (BM) and propose Triway BM to utilize context in scene modeling task. They incorporate objects and spatial relationships between objects for 4 scene modelling tasks; relation estimation between objects, finding missing objects in a scene, finding objects contradicting with the context in a scene and generating new scenes given objects or relations between objects.

Table 2.1 summarizes the reviewed work above.

Table 2.1: **Comparison of different context models.** Type 1 utilizes scene-level context or context around the proposal. Type 2 utilizes relationships between objects. The last row of the table lists our proposed method to put it into context within the related work.

| **Study** | **Type 1** | **Type 2** | **Selected context regions** | **Context model & feature unification** |
|---|---|---|---|---|
| [14] | ✗ | ✓ | other proposals | A spatial memory module is proposed. Objects are detected in sequential manner and each detected object updates the state of the memory. Subsequent objects are better detected with the help of the memory module. |
| [15] | ✓ | ✗ | Whole scene | Context features outside of proposal region are obtained by applying Four directional IRNN on conv5 layer of VGG16. Proposal features are concatenated with context features before prediction. |

*Continued on next page*

Table 2.1 – *Continued from previous page*

| **Study** | **Type 1** | **Type 2** | **Selected context regions** | **Context model & feature unification** |
|---|---|---|---|---|
| [16] | ✗ | ✓ | Other proposals | Multiple relation types between objects are defined so multiple context features obtained from other proposals using linear transformation. Each context feature received from other proposals with same relation type are summed, and the resulting vector for each relation type is concatenated. The result is summed with the original proposal features. |
| [17] | ✓ | ✓ | Whole scene and other proposals | Features of other proposals are multiplied with a scalar calculated considering the spatial and semantic relationships with the query proposal. These features are input to a GRU where the initial state is set to the features of query proposal. Another GRU with same logic is applied where the input is scene-level features. Output state of 2 GRUs are summed. |
| [18] | ✓ | ✓ | Whole scene & score predictions of other proposals | A conditional random field model is used to include the effect of other proposals and scene-level context. |

*Continued on next page*

Table 2.1 – *Continued from previous page*

| **Study** | **Type 1** | **Type 2** | **Selected context regions** | **Context model & feature unification** |
|---|---|---|---|---|
| [19] | ✓ | ✓ | Surrounding regions of proposal and the regions selected by attention network from global view | Global context regions are unified by pooling; local context regions are unified by concatenation. The results of global and local context regions are concatenated to make final classification. |
| [20] | ✗ | ✓ | Other proposals in image having same label predictions and a IoU greater than a threshold with query proposals | Visual features of context proposals are multiplied by a scalar obtained by multiplying IoU between context proposal and query proposal with the confidence score of context proposal. |
| [21] | ✓ | ✗ | 10 manually picked regions around the query proposal | Features of manually picked regions and proposal in question are used in different subnetworks separately. The results of subnetworks are aggregated by applying learnable weights for final prediction. |
| [22] | ✗ | ✓ | Proposals visually similar to the target proposal | Confidence scores of target proposal and other visually similar proposals, and visual distance between them are used to re-score target proposal by MMSE. |

*Continued on next page*

Table 2.1 – *Continued from previous page*

| **Study** | **Type 1** | **Type 2** | **Selected context regions** | **Context model & feature unification** |
|---|---|---|---|---|
| Our work | ✗ | ✓ | Confidence scores of other detections | Confidence scores of other detections are used to generate spatial and scale-based region summaries. Region summaries and scores of query detection are fed through an MLP network to re-score the query detection. |

# CHAPTER 3

# CONTEXT RE-SCORING AS POST PROCESSING

In this chapter, we explain our proposed method which utilizes context information to improve performance of object detectors along with the neural network model details.

## 3.1 Context Information

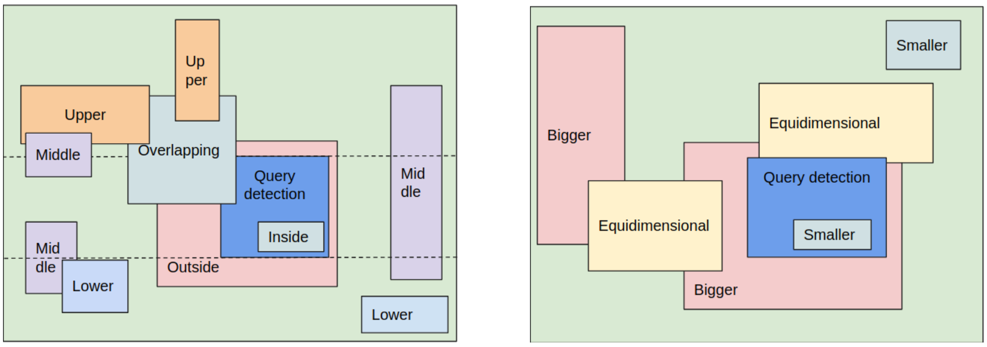
Some object categories in images tend to co-occur frequently; such as mouse and keyboard. Also, objects have spatial relationships between each other; for example, monitor objects tend to appear above of mouse objects or tie objects tend to be on person objects. Humans are able to make use of these kinds of contextual relationships between objects while interpreting an image. The performance of an object detector can be enhanced by exploiting object relationships in the same manner.

Object detector models process visual features and output an arbitrary number of detections for each image. A detection is given by a bounding box and scores for each category in the dataset. Each bounding box consists of 4 predictions; x, y, w, h. The (x, y) coordinates represent the upper left corner or the center of the bounding box where the (w, h) represent the width and height of the box. Category scores are estimated probability scores between [0,1] defined for each category and represent the confidence of the model for that bounding box to belong to that specific category. Our model runs on these results predicted by any object detector.

The aim of our model is to improve object detection accuracy by exploiting context information embedded in images. We benefit from two types of relationships between objects based on relative locations and scales of objects. By using spatial and scale based relationships, for example, our model may predict that a relatively bigger object next to a mouse object is a keyboard. Unlike many other models that use non-local visual features from the same scene as context, visual features of the scene are not considered in our model. Our method can be seen as a post-processing step on predictions of any object detector. An object detector is run on the input image, we keep the highest scoring 100 detections. Each prediction is re-scored using the scores of other detections in the same scene. In the rest of this section, we use the expression "query detection" to refer to the detection to be re-scored and "context detections" to refer to all other detections in the same image.

To establish relationships between detections, we have to tackle detections in different locations and scales. To simplify this problem and encode spatial and semantic relationships between objects, we define multiple regions based on relative locations and scales of detections. Specifically, we use 9 regions; 6 spatial: upper, lower, middle, overlapping, inside, outside; and 3 scale-related: bigger, smaller, equidimensional. Since spatially horizontal (i.e. left and right) relationships between objects vary from one image to another, i.e., a mouse object may appear at both sides of a keyboard object, they are not included in regions and ignored. The definitions of regions are given below:

- Upper: Context detections whose lower edge is above the upper edge of the query detection.
- Lower: Context detections whose upper edge is below the lower edge of the query detection.
- Middle: Context detections falling in the area between upper and lower.
- Overlapping: Context detections whose IoU with the query detection is greater than 0.
- Inside: Context detections residing inside of the query detection.
- Outside: Context detections surrounding the query detection.
- Bigger: Context detections whose area is larger than the query detection with a margin of x1.5



(a) Spatial regions.

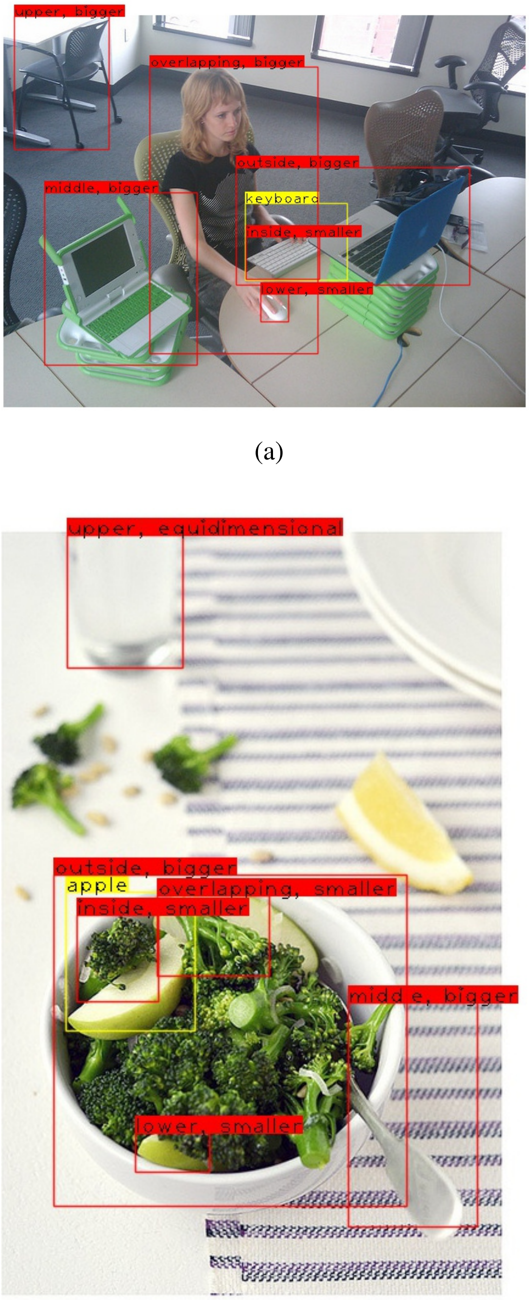
(b) Scale-based regions

Figure 3.1: **Visualization of spatial and scale-based contextual regions.** Each context detection is assigned to one spatial region and one scale-based region based on its relative location and scale with respect to the query detection.

- Smaller: Context detections whose area is smaller than the query detection with a margin of x1.5
- Equidimensional: Context detections whose area is between bigger and smaller.

Figure 3.1 demonstrates region assignments of context detections based on their relative locations and sizes with respect to the query detection. These regions are defined relative to the query detection and they are used to construct a context summary around the query detection using other detections as context in the same image. Each context detection is included in 2 regions; one of the 6 spatial regions and one of the 3 scale-related. Sample objects for these regions are given in Figure 3.2. One example is demonstrated for each spatial region while zero or more objects for each region may exist for a detection in any image. Query detections are denoted by yellow boxes while context detections are denoted by red boxes. In the first scene, region categories of some objects are shown with respect to the keyboard object while in the second scene, region categories of objects are shown with respect to the apple object. Score predictions of objects denoted by red boxes and other objects that exist in the same scene but are not denoted in the figure are used to calculate region summaries with respect to the objects in yellow bounding boxes.

We propose three models that use previously defined 9 regions; Multi layer perceptron



(a)

(b)

Figure 3.2: **Visualization of spatial and scale-based contextual regions.** The query detection is marked with yellow borders, while the contextual detections (with respect to the query detection) are shown with red borders.

(MLP), Gated MLP and Pairwise MLP. The structure of these models are visualized in figures 3.3, 3.4 and 3.5, respectively. Our proposed models are explained in Section 3.1.1, 3.1.2 and 3.1.3.

### 3.1.1 The MLP Model

For the MLP and the Gated MLP models, we extract region summaries by using scores of a subset of detections that spatially or dimensionally fall into the same region with respect to the query detection. More specifically, to extract summaries of a region, we evaluate maximum class scores for each object category using the detections reside in that specific region. For example, detections to the upper side to query detection are found and maximum class scores are evaluated using only those detections' scores for the 'Upper' region. The same is applied for all regions. This way, we obtain a unified context summary representations for each detection using class scores of surrounding detections for multiple regions. After calculating summaries for all regions, we obtain a tensor shaped $(NumberOfRegions * NumberOfClasses)$ for each detection. $NumberOfRegions$ is the number of regions explained before and 9 in our case. $NumberOfClasses$ is the number of object categories exist in detection benchmark (For COCO benchmark, it is 80). Since this tensor holds maximum category scores for each region, all values are between 0 and 1. The same region extraction step is applied for each detection in the same scene. For example, if keyboard and laptop objects detected in a region correspond to upper region of a mouse detection by RetinaNet, the score indices correspond to upper keyboard and upper laptop of mouse detection will be close to 1 while the score indices correspond to down mouse of both laptop and keyboard detections will be close to 1. Thus, we obtain a tensor shaped $(NumberofDetections,\ NumberOfRegions * NumberOfClasses)$ where $NumberOfDetections$ is the count of the detections predicted by baseline object detector for an image which is 100 in our case. This tensor is fed to a neural network to re-score each detection benefiting from the instance-level relationships. For the MLP model, we input query detection's scores concatenating with the 9 regions summary scores to a 3-layer MLP. This network predicts new class scores for the query detection benefiting from the relationships between instance summaries for regions and the query detection.

### 3.1.2 The Gated MLP Model

The MLP model is trained to learn relationships between objects. However, some object categories in dataset may not be convenient to extract contextual relationships as other categories. Therefore, to make the MLP model more robust to errors in contextual inference, we propose the Gated MLP model that is built on the MLP model. The Gated MLP model attaches a sigmoid gate to the MLP model that aggregates predictions of the MLP model with the predictions of baseline object detector. It consists of two MLP branches where the first branch shares the same structure with the MLP model and evaluates new category scores for each detection using region summaries. Since the results of the first network are logits, it is required to transform the score predictions of the base detector to logits before aggregation. The input to the second network is only the scores of query detection. The second network learns to transform the class scores that are between 0 and 1 to logits and scale these logits to match the scales of the results of the first network that makes contextual inference. The results of two networks are aggregated using a sigmoid gate. The sigmoid gate has different value for each object category so categories that are hard to model in context may preserve category predictions of RetinaNet.

$$s^{'}_{x,i} = f([s_{x,i}, s_{region1,i}, s_{region2,i}, ..., s_{region9,i}]) \tag{1}$$

$$s^{'}_{x,i} = \sigma(\omega) * f_1(s_{x,i}) + (1 - \sigma(\omega)) * f_2([s_{x,i}, s_{region1,i}, s_{region2,i}, ..., s_{region9,i}]) \tag{2}$$

Equation 1 and 2 formulates the MLP model and the Gated MLP model respectively where $s_{x,i}$ and $s^{'}_{x,i}$ are the score predictions of detection $i$ in image $x$ obtained from RetinaNet and our methods respectively. $s_{region,i}$ variables are the region summaries with respect to detection $i$ obtained by using maximum scores of detections falling in specific regions. $f$, $f_1$ and $f_2$ represent the MLP networks. $\omega$ is a learnable parameter having shape of the number of classes of the dataset that enables weighted aggregation of RetinaNet predictions with the context predictions using sigmoid gate $\sigma$.

### 3.1.3 The Pairwise MLP Model

The Pairwise MLP model differs from the first two models since instead of using region summaries to extract relationships between objects, it calculates the effect of each detection on the query detection individually by modelling pairwise relationships. To calculate the effect of a detection on another detection, a binary vector that is in the shape of number of regions is generated. Each index of the binary vector corresponds to one of the regions. To encode relationship between two objects, the indices that correspond to region of the binary vector are set to 1 while other indices are set to 0. For instance, if a detection is 'Upper Bigger' to the query detection, the index that corresponds to upper region and the index that corresponds to bigger region of the binary vector are set to 1. The generated binary vector, the scores of the query detection and the scores of context detection are concatenated. The concatenated tensor is fed to a 3-layer MLP. To calculate the final score for a detection, same procedure is applied using each context detection, and the final score for the query detection is obtained by summation of all results. Thus, the final score of each detection is calculated using the pairwise relationships between the query detection and all other detections in the same scene. The formulation of the Pairwise MLP model is given in Equation 3 where $s_{\text{x,i}}$ is the score predictions of detection $i$ and $s_{\text{x,j}}$ is the score predictions of detection $j$ for image $x$ obtained from RetinaNet. $R_{\text{i,j}}$ is the binary vector generated using the relation between detection $i$ and detection $j$. $f$ represents the MLP network.

$$s'_{\text{x,i}} = \sum_{j \neq i} f([s_{\text{x,i}}, s_{\text{x,j}}, R_{\text{i,j}}]) \tag{3}$$

## 3.2 Methods of Analyzing Results

Using our models, we can refine false positive predictions of the base detector by refining class scores with the help of the surrounding predictions. There are multiple types of false positives in object detection task. We can classify predictions as true positive (TP) or one of the false positive (FP) types inspired by the study of [24]. According to that study, detections can be classified based on the following context:

- Correct (TP): Predictions having a maximum IoU >= 0.5 with a ground truth object and sharing the same class with that object.
- Similar (FP): Predicted class is in the same super-category with the ground truth and IoU >= 0.1
- Background (FP): IoU < 0.1
- Localization (FP): Correct class and 0.1 <= IoU < 0.5. Also duplicate detections that matching the same ground truth object are classified as localization error.
- Other (FP): Class is wrong and IoU >= 0.1

Our models try to correct false positives while enhancing correctly classified detections. Our models may potentially improve all the false positive categories except the localization errors arise from low IoU between detections and the ground truth objects since we do not regress the bounding boxes and only refine the scores of detections obtained by another object detector.

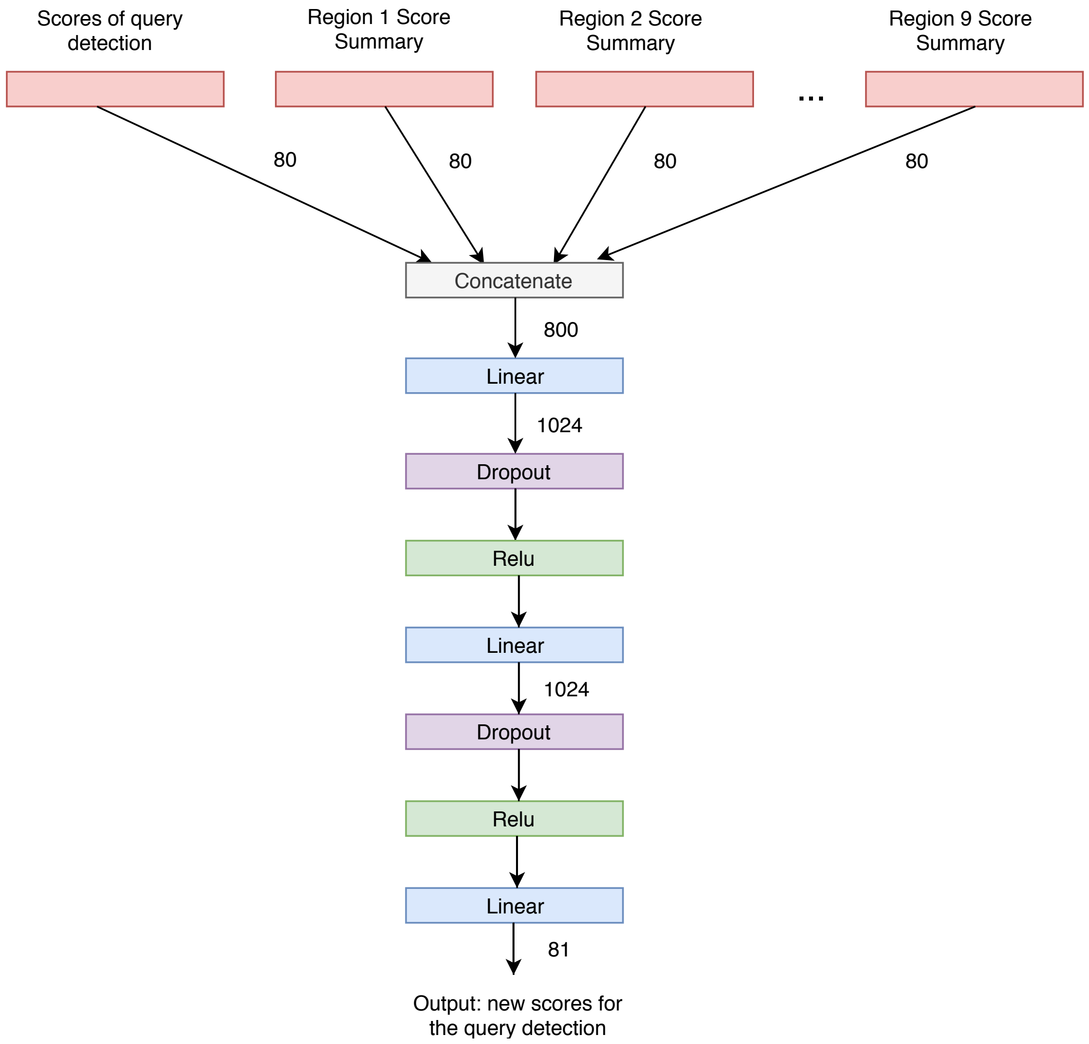


Figure 3.3: **Network architecture of the MLP model.**

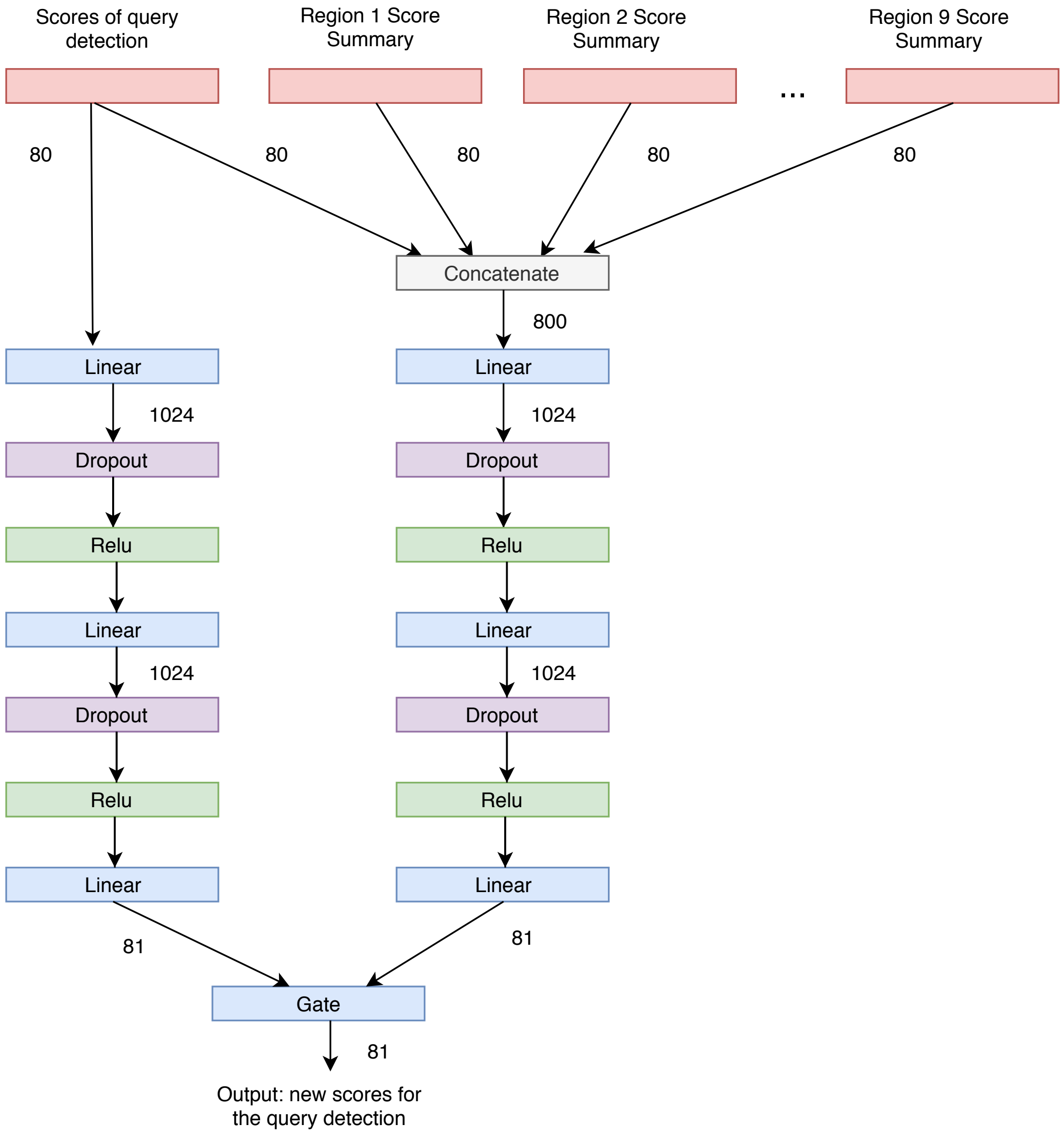


Figure 3.4: **Network architecture of the Gated MLP model.**

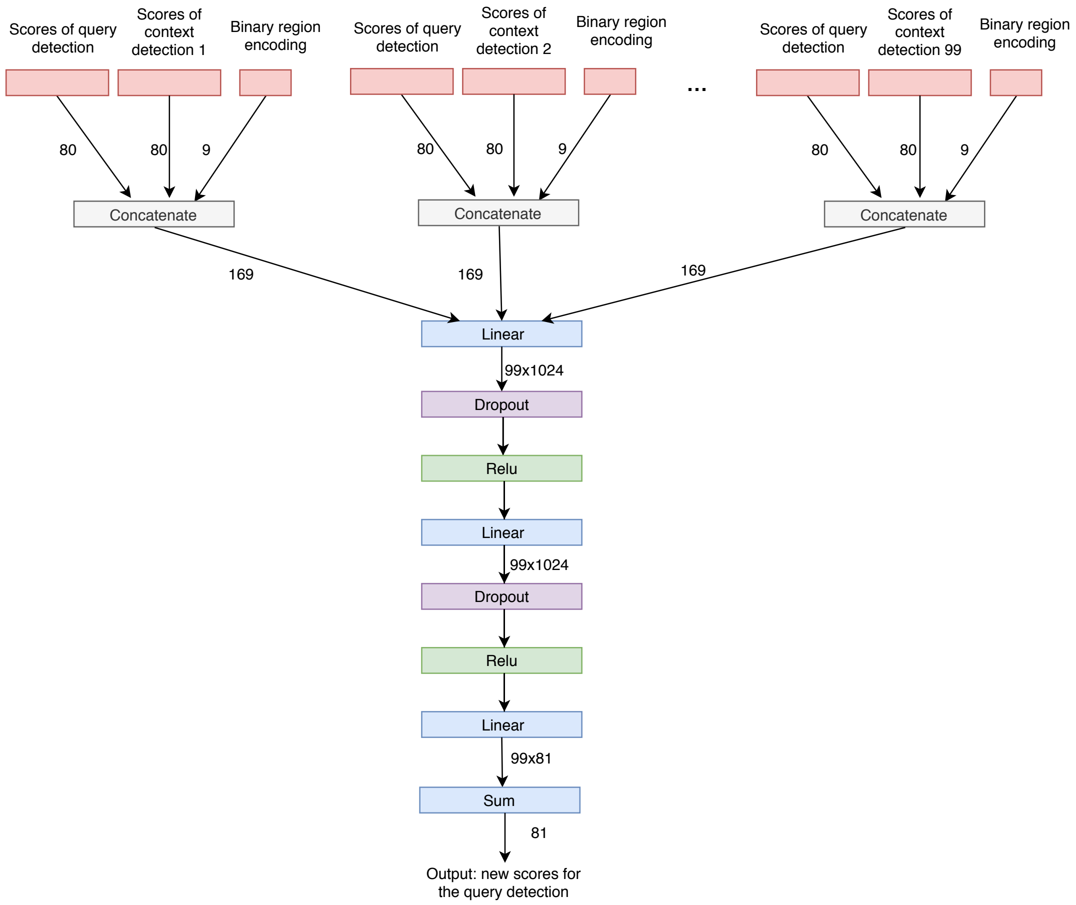

Scores of query detection
Scores of context detection 1
Binary region encoding
Scores of query detection
Scores of context detection 2
Binary region encoding
...
Scores of query detection
Scores of context detection 99
Binary region encoding
80
80
9
80
80
9
80
80
9
Concatenate
Concatenate
Concatenate
169
169
169
Linear
99x1024
Dropout
Relu
Linear
99x1024
Dropout
Relu
Linear
99x81
Sum
81
Output: new scores for the query detection

Figure 3.5: **Network architecture of the Pairwise MLP model.**

# CHAPTER 4

# EXPERIMENTS

In this chapter, we describe the dataset and evaluation metric used in experiments. The effect of context in our proposed methods is presented. Moreover, obtained results are compared against the results of the baseline method (i.e. the base object detector without our context model).

## 4.1 The Dataset

COCO [1] is an object detection, segmentation and captioning dataset. In our experiments, we use COCO for object detection task. We use the “train2017” split for training and the “val2017” split for validation. COCO contains instances of 80 object categories where the train2017 split contains 118K images while the val2017 split contains 5K images and 36K annotations. We use COCO dataset since it is more suitable for a context model compared to other datasets due to the following reasons:

- COCO consists of everyday scenes of common objects in natural context. Thus, intra-class variation is high and scenes contain ambiguous object instances that are hard to detect without using context information.
- The average sizes of objects are smaller for COCO compared to those in other popular object detection datasets such as Pascal VOC[25] and ImageNet[26]. As small objects are harder to detect, they require more contextual information than other objects.
- Compared to Pascal VOC and ImageNet, COCO contains more object instances per image that indicates more context information is available.

- While ImageNet and Pascal VOC contain less than 2 categories per image in average, COCO contains 3.5 categories per image.

## 4.2 Evaluation Metric

As evaluation metric, mean average precision (mAP) is used. mAP is evaluated by first evaluating average precision (AP) for each category existing in the detection dataset and then taking the average of AP values. In the rest of this thesis, AP and mAP are used interchangeably where the distinction can be made from context. To evaluate AP for each object category, each detection is labeled as one of the following categories:

- True Positive (TP): Detections having IoU >= threshold with at least one of the ground truth boxes and share the same label.

- False Positive (FP): Background detections that are having IoU < threshold with all of the ground truth boxes or misclassified detections.

To categorize detections as TP or FP, IoU is used. IoU is a measure to evaluate the overlap between predicted bounding boxes and ground truth bounding boxes. IoU is calculated by dividing intersection between two bounding boxes to union of those boxes. IoU value is always between 0 and 1; 0 means there is no intersection and 1 means two boxes are perfectly aligned. We need to specify a threshold value for IoU to decide if a predicted box matches to a ground truth box. COCO API evaluates AP for multiple values of IoU thresholds; such as $AP^{50}$ means the threshold is set to 0.5. After detections are classified as TP or FP, detections are sorted by their confidence scores in descending order and a score threshold is applied to eliminate detections having low confidence. If a ground truth object matches with multiple detections, the first detection is accepted as TP and the others are categorized as FP. Then, precision/recall (PR) curve is sampled by calculating the precision for each unique recall value where the x axis is recall and the y axis is the precision. AP is defined as the area under PR curve. Precision and recall equations are given in

Equation 1 and Equation 2 respectively.

$$Precision = \frac{NumberofTruePositives}{NumberofAllDetections} \tag{1}$$

$$Recall = \frac{NumberofTruePositives}{NumberofAllGroundTruth} \tag{2}$$

Precision measures the accuracy of the predictions while recall measures the percentage of the ground truth objects that are matched. If redundant predictions are made to increase recall, precision value will decrease since the quality of predictions will decrease. If quality of detections is increased by setting an ultrahigh score threshold value, then precision will be high but recall will decrease. Thus, there is always a trade-off between precision and recall. Recall monotonically increases as new detections are appended while precision may increase or decrease which causes zigzags in the PR curve. To smooth the zigzag pattern, precision at each recall level $r$ is replaced with the maximum precision value of any recall level $r' > r$. After the PR curve is smoothed, AP is evaluated by calculating the area under the smoothed curve.

Although AP is the most widely used performance evaluation metric for object detection, it may return same results for different PR curves. To understand the differences between different PR curves, further analysis is required by inspecting PR curves. Also, AP do not directly measure localization accuracy that is how tightly the detections are intersecting with the ground-truth objects. Therefore, we also evaluate localization recall precision [27] (LRP) error on detections obtained by both our models and base object detector. LRP error consists of three components that are related to localization, false positive rate and false negative rate. Optimal LRP corresponds to the minimum achievable LRP error that represents the best achievable configuration by an object detector.

### 4.3 Detections

Keras implementation of RetinaNet is available on github[1]. The implementation uses ResNet-50 as backbone network and a 800 pixel train and validation image

[1] `https://github.com/fizyr/keras-retinanet`

scale. RetinaNet predicts regression values and confidence scores for each predefined bounding box as output. Then, deduplication and a score threshold are applied to eliminate duplicate detections and boxes having low confidence scores. Non maximum suppression (NMS) is used for deduplication. NMS calculates IoU between each pair of bounding boxes in the same scene and if IoU between two detections is over a threshold value, the detection with the low confidence score is removed. Applying deduplication and score threshold results in arbitrary number of detections for each image. Our context models are applied as post processing step on detections of RetinaNet. To fix the number of detections for each image, we remove score threshold and take top 100 scored detections instead after NMS is applied. To compare our methods with the baseline that is RetinaNet, we evaluate AP for both baseline and our methods. AP for baseline is evaluated applying a 0.05 threshold on top 100 detections of baseline.

To be able to train and validate our model, it is required to label each bounding box obtained from RetinaNet as background or as one of the 80 categories of COCO dataset. We follow the same labeling strategy that is used in the RetinaNet work [2]. Specifically, labels of detections having 0.5 or greater IoU with a ground truth object are set to the label of that ground truth object. Detections having maximum IoU of less than 0.4 with ground truth objects are labeled as background. Other detections having IoU between 0.4-0.5 are ignored during training to ensure stability.

### 4.4 Experimental Setup

For training, we use the train2017 split of COCO benchmark which is validated using the val2017 split. Our network is trained with the Adam optimizer [28]. In order to tune learning rate parameter, we used values of $[10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}]$ and best results are achieved with $10^{-5}$. Dropout [29] rate of 0.3 and ReLU activation [30] is applied between fully-connected layers. Cross entropy loss for multiclass classification is used as the objective function. Mini-batch size is set to 100 since there are 100 detections for each image.

## 4.5 Preliminary Work

Since our contextual inference to re-score a detection is based on the scores of other detections, errors of the base detector may mislead our model. To eliminate such cases and measure the limits of our model, we first evaluate our method with the ground-truth labels of context detections. In other words, to re-score a query detection, we use the ground-truth class labels of the context detections (i.e. all the other detections except the query detection) in the same image. This evaluation using the ground-truth labels for the context would show how much our method can increase mAP results at most. Therefore, we evaluate the MLP model using the ground-truth labels of the context detections and **obtain a 4.4% increase on AP$^{50}$ over the baseline results.** The results are listed in Table 4.1 for both baseline and our model. These results reveal that object to object relationships can be used to improve performance of an object detector. Since annotations for test2017 split of COCO are not available, tests using labels of context detections are evaluated only on the val2017 split.

Table 4.1: **Improvement obtained by our MLP context model when ground-truth labels are used for the context detections.** Results are shown on the COCO val2017 dataset.

| Method | AP | $AP^{50}$ | $AP^{75}$ | $AP^{small}$ | $AP^{medium}$ | $AP^{large}$ |
|---|---|---|---|---|---|---|
| Baseline (RetinaNet) | 34.7 | 53.7 | 36.9 | 18.9 | 37.7 | 46.6 |
| Baseline +<br>MLP with Labels | 36.6 | 58.1 | 38.4 | 21.9 | 40.0 | 47.5 |

## 4.6 Experimental Results

In this section, we compare our results against results of the baseline detector that is RetinaNet. For each model, AP results for different IoU thresholds are listed. $AP^{50}$ and $AP^{75}$ correspond to AP calculated using IoU threshold of 0.5 and 0.75 while overall AP is calculated by averaging AP results for 10 IoU thresholds between 0.5 and 0.95. Also, AP for small, medium and large objects are demonstrated separately. Small objects are considered as the objects having area smaller than $32^2$ while medium objects are the ones having area between $32^2$ and $96^2$. Objects with area larger than $96^2$ are classified as large. Obtained results are reviewed to understand the contribution of our model. Train and validation loss graphs over epochs for our models are shown in Figure 4.1.

AP results obtained by our models and the baseline model on the val2017 split of COCO are listed in Table 4.2. Gated MLP improves AP results for all AP categories while the other models improve AP for all categories except the AP for large objects which is expected since it is harder to detect small objects for modern CNN based object detectors while large objects are classified easily.

Table 4.2: **Results of different models on MS COCO val2017.**

| Method | AP | $\text{AP}^{50}$ | $\text{AP}^{75}$ | $\text{AP}^{small}$ | $\text{AP}^{medium}$ | $\text{AP}^{large}$ |
|---|---|---|---|---|---|---|
| Baseline (RetinaNet) | 34.7 | 53.7 | 36.9 | 18.9 | 37.7 | 46.6 |
| Baseline + MLP | 35.1 | 55.3 | 37.0 | 19.5 | 38.5 | 46.5 |
| Baseline + Gated MLP | **35.3** | **55.5** | **37.3** | 19.4 | **38.7** | **47.2** |
| Baseline + Pairwise MLP | 34.8 | 54.9 | 36.8 | **19.7** | 38.0 | 46.4 |

Mean optimal LRP (moLRP) errors of detections obtained by our models and the base object detector on the val2017 split of COCO are listed in Table 4.3. The MLP model and the Gated MLP model decrease moLRP error for all components except for the component related to localization. The increase in localization component of moLRP is expected since the number of mislocalized detections of base detector increase as our models correct the false positive detections that are not tightly intersecting with the ground-truth objects.

For each object category, the results obtained by baseline detector and Gated MLP model on test2017 split of COCO are compared in table 4.4. Results for small and medium objects improved by Gated MLP while for large objects results do not change. AP for almost all object categories are increased except for 4 categories; person, dog, zebra, and toaster. The results of these categories are slightly lower than the

Table 4.3: **LRP errors obtained by different models on MS COCO val2017.**

| Method | moLRP | moLRP$_{IoU}$ | moLRP$_{FP}$ | moLRP$_{FN}$ |
|---|---|---|---|---|
| Baseline (RetinaNet) | 0.7106 | **0.1703** | 0.2856 | 0.5069 |
| Baseline + MLP | 0.7052 | 0.1726 | 0.2714 | 0.4980 |
| Baseline + Gated MLP | **0.7039** | 0.1737 | **0.2713** | 0.4914 |
| Baseline + Pairwise MLP | 0.7086 | 0.1748 | 0.2952 | **0.4903** |

baseline. The biggest improvement is obtained for toothbrush category whose AP is improved from 18.5% to 24.7%.

Table 4.4: **Class by class detection results on MS COCO test2017 split.** $AP^{50}$ scores are given for class by class comparison. RetinaNet results are obtained by applying a score threshold of 0.05 over top 100 scored detections. Our results are obtained by applying a score threshold of $5 \cdot 10^{-4}$.

| | **Baseline (RetinaNet)** | **Baseline + Gated MLP** |
|---|---|---|
| AP | 34.8 | **35.3** |
| AP$^{50}$ | 54.0 | **55.8** |
| AP$^{75}$ | 37.4 | **37.5** |
| AP$^{small}$ | 18.0 | **18.7** |
| AP$^{medium}$ | 37.3 | **37.9** |
| AP$^{large}$ | 45.0 | 45.0 |
| person | **76.9** | 76.8 |
| bicycle | 53.6 | **54.9** |
| car | 61.2 | **61.6** |
| motorcycle | 65.4 | **66.6** |
| airplane | 76.0 | **77.2** |
| bus | 81.0 | **81.2** |

Continued on next page

**Table 4.4 – continued from previous page**

| | **Baseline (RetinaNet)** | **Baseline + Gated MLP** |
|---|---|---|
| train | 79.9 | **81.0** |
| truck | 48.7 | **48.9** |
| boat | 41.6 | **43.6** |
| traffic light | 46.7 | **48.4** |
| fire hydrant | 74.7 | **75.9** |
| stop sign | 76.1 | **77.1** |
| parking meter | 55.5 | **56.7** |
| bench | 32.9 | **33.9** |
| bird | 51.5 | **54.6** |
| cat | 83.0 | **83.4** |
| dog | **74.6** | 74.2 |
| horse | 76.7 | **79.2** |
| sheep | 71.0 | **74.6** |
| cow | 69.3 | **74.7** |
| elephant | 87.4 | **89.1** |
| bear | 88.7 | **89.7** |
| zebra | **87.3** | 87.2 |
| giraffe | 89.2 | **89.9** |
| backpack | 34.0 | **34.4** |
| umbrella | 57.0 | **58.3** |
| handbag | 25.7 | **26.7** |
| tie | 46.0 | **48.0** |
| suitcase | 51.0 | **55.4** |
| frisbee | 65.8 | **69.2** |
| skis | 29.8 | **32.1** |
| snowboard | 36.8 | **38.6** |
| sports ball | 50.6 | **53.2** |
| kite | 55.8 | **60.5** |
| baseball bat | 48.1 | **53.1** |

Continued on next page

**Table 4.4 – continued from previous page**

| | **Baseline (RetinaNet)** | **Baseline + Gated MLP** |
|---|---|---|
| baseball glove | 56.4 | **59.4** |
| skateboard | 66.0 | **68.9** |
| surfboard | 47.2 | **50.7** |
| tennis racket | 73.6 | **76.5** |
| bottle | 51.6 | **52.6** |
| wine glass | 55.2 | **56.1** |
| cup | 52.5 | **53.6** |
| fork | 34.8 | **36.0** |
| knife | 22.5 | **24.1** |
| spoon | 17.2 | **18.9** |
| bowl | 51.0 | **51.7** |
| banana | 40.1 | **40.5** |
| apple | 30.7 | **31.9** |
| sandwich | 49.6 | **52.2** |
| orange | 39.9 | **40.8** |
| broccoli | 50.0 | **50.7** |
| carrot | 31.1 | **32.4** |
| hot dog | 35.7 | **39.5** |
| pizza | 70.9 | **72.8** |
| donut | 61.2 | **64.9** |
| cake | 39.3 | **41.8** |
| chair | 41.4 | **42.7** |
| couch | 54.0 | **55.4** |
| potted plant | 40.9 | **41.7** |
| bed | 63.3 | **63.6** |
| dining table | 42.2 | **44.3** |
| toilet | 75.0 | **76.3** |
| tv | 71.6 | **72.8** |
| laptop | 74.7 | **76.1** |

Continued on next page

**Table 4.4 – continued from previous page**

| | **Baseline (RetinaNet)** | **Baseline + Gated MLP** |
|---|---|---|
| mouse | 67.9 | **71.4** |
| remote | 43.7 | **47.8** |
| keyboard | 63.9 | **67.4** |
| cell phone | 42.6 | **43.9** |
| microwave | 72.7 | **73.7** |
| oven | 55.2 | **56.5** |
| toaster | **12.5** | 11.4 |
| sink | 49.0 | **50.6** |
| refrigerator | 62.4 | **64.0** |
| book | 19.8 | **21.7** |
| clock | 70.4 | **70.7** |
| vase | 51.8 | **54.2** |
| scissors | 35.7 | **36.7** |
| teddy bear | 60.5 | **62.1** |
| hair drier | 2.8 | **4.7** |
| toothbrush | 18.5 | **24.7** |

The RetinaNet implementation we use in experiments achieves a mAP of 53.7 on the val2017 split and 54.0 on test-dev2017 split of COCO benchmark with an IoU threshold of 0.5. The train2017 split of COCO contains 118K images, the val2017 split contains 5K images and 36K annotations where RetinaNet predicts ~600K detections for the val2017 split after applying NMS and a score threshold of 0.05. Since RetinaNet returns almost 17 times more detections than the number of annotations, most of these detections are false positives. There are multiple types of false positives in the object detection task that are explained in Section 3.1. We need to analyze the distribution of errors for RetinaNet predictions and our predictions to fully understand the contribution of our model by comparing results. To analyze false positives of RetinaNet, we use top 100 scored detections for each image since our models run on those detections. We use the val2017 split results for false positive analysis since annotations are not publicly available for test2017 split. Figure 4.2 visualizes the distribution of false positives and true positives of RetinaNet detections on the val2017 split. Analyzing false positives helps us to understand the impact of different false positive types on results. For each category, top $N$ scored detections are used where $N$ is the number of ground truth objects for that category.

Figure 4.2 reveals that most of the detections are classified correctly by RetinaNet, while there is a considerable amount of false positives. We can improve mAP of the detector by refining category scores and eliminating these false positives. In our models, we can refine all the false positive categories except the localization errors arising from low IoU with the ground truth object since our method does not modify the bounding boxes and only refine the scores of pre-predicted boxes. Charts in Figure 4.2 show that our Gated MLP method increases the percentage of correctly classified detections while decreasing false positives categorized as similar, other and background that are the false positives due to misclassification. Percentage of localization errors where detections are classified correctly but misaligned, increases with our method. Correcting only misclassified detections increases the percentage of TP detections while correcting both mislabeled and mislocalized detections increases the percentage of localization errors. Therefore, increase in localization errors is expected of our models.

Figure 4.3 visualizes a series of precision recall (PR) curves for both RetinaNet implementations where each PR curve is guaranteed to be strictly higher than the previous as the evaluation setting becomes more permissive. The definitions of these curves[2] are given below for self-readability of this thesis:

- C75: PR at IoU=.75 (AP at strict IoU), area under curve corresponds to $AP^{IoU=.75}$ metric.
- C50: PR at IoU=.50 (AP at PASCAL IoU), area under curve corresponds to $AP^{IoU=.50}$ metric.
- Loc: PR at IoU=.10 (localization errors ignored, but not duplicate detections). All remaining settings use IoU=.1.
- Sim: PR after super-category false positives (fps) are removed. Specifically, any matches to objects with a different class label but that belong to the same super-category don't count as either a fp (or tp). Sim is computed by setting all objects in the same super-category

[2] http://cocodataset.org/#detection-eval

to have the same class label as the class in question and setting their ignore flag to 1. Note that person is a singleton super-category so its Sim result is identical to Loc.

- Oth: PR after all class confusions are removed. Similar to Sim, except now if a detection matches any other object it is no longer a fp (or tp). Oth is computed by setting all other objects to have the same class label as the class in question and setting their ignore flag to 1.
- BG: PR after all background (and class confusion) fps are removed. For a single category, BG is a step function that is 1 until max recall is reached then drops to 0 (the curve is smoother after averaging across categories).
- FN: PR after all remaining errors are removed (trivially AP=1).

These plots demonstrate how mAP will increase as we eliminate false positives of detection method. They are drawn considering all detections different from the charts in Figure 4.2 where top N scored detections are considered. Plots demonstrate that if false positives of type similar are removed from RetinaNet detections, a 2.3% improvement on mAP can be achieved while if false positives categorized as other are removed a 3.2% improvement can be obtained. Like charts in Figure 4.2, plots also show that our method improves mAP results by decreasing false positives of categories similar and other. Also, plots reveal that effect of localization errors is increased by our model as expected.

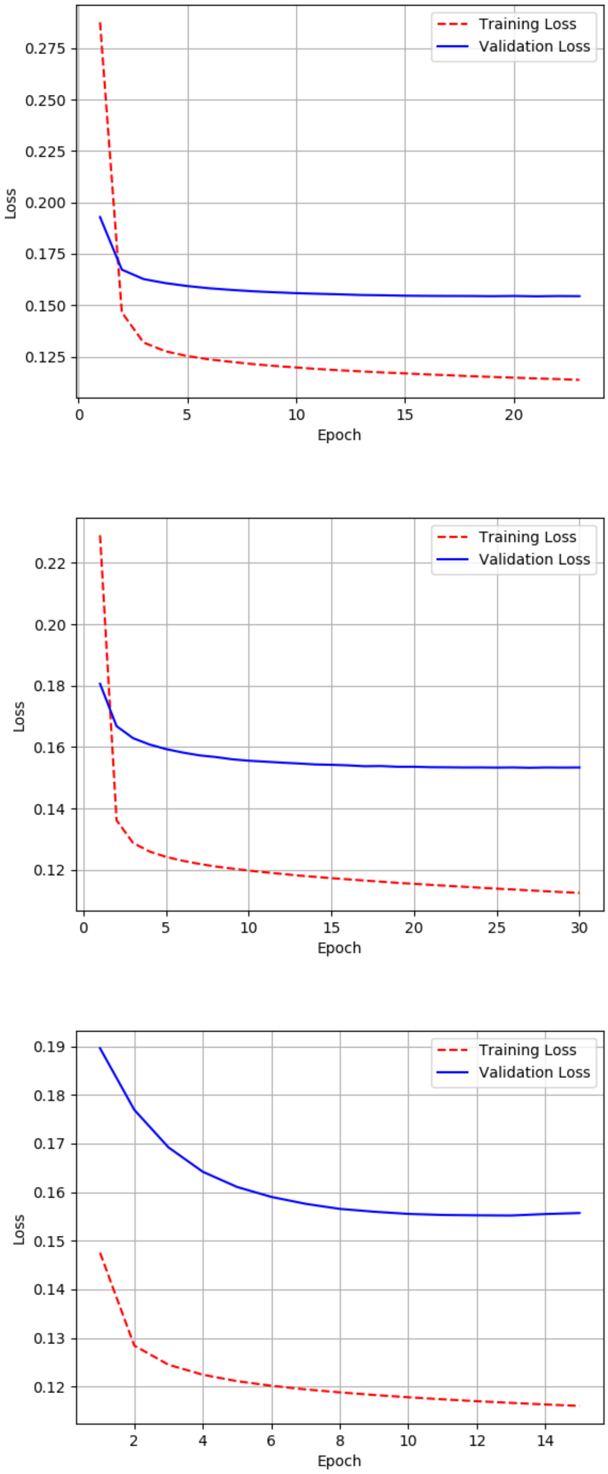
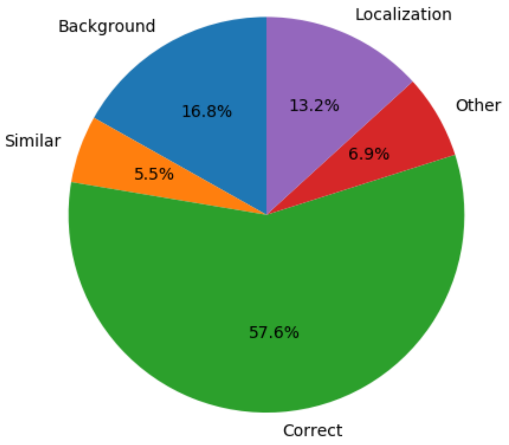
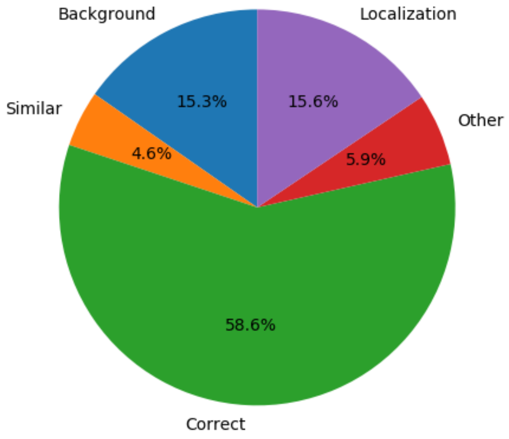


Figure 4.1: **Train and validation loss graphs.** Loss graphs of MLP, Gated MLP and Pairwise MLP respectively with parameters learning rate=1e-5, batch size=100, dropout=0.3

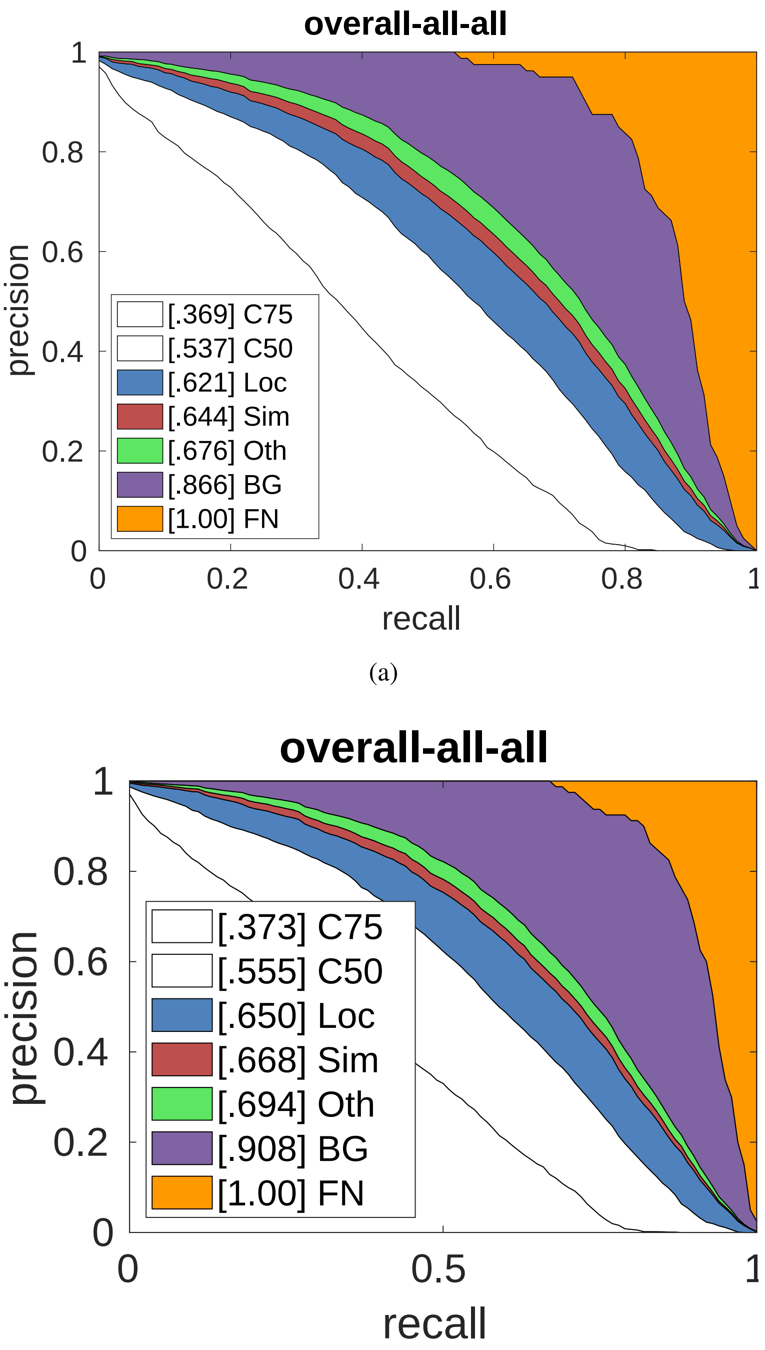


(a)



(b)

Figure 4.2: **Error analysis charts.** These charts visualize percentage of different error types in the top N scored predictions of RetinaNet and Gated MLP for all categories. N parameter is selected as the number of ground truth objects in each category. (a) RetinaNet results. (b) Gated MLP results



(a)

(b)

Figure 4.3: **Error analysis plots.** These plots visualize overall precision-recall curve of results averaged over all categories. Results are obtained on the val2017 split of COCO. (a) RetinaNet results. (b) Gated MLP results

## 4.7 Ablation Study

In our experiments, we use 9 regions jointly where 6 of them are spatial and 3 of them are scale-related. To examine the effect of the spatial and scale-related regions separately, we evaluate our MLP model using only spatial regions or only scale-related regions. Also, we evaluated our MLP network by replacing cross entropy loss for multi-class classification with cross entropy loss for binary classification. The results are given in table 4.5.

The Gated MLP model has two branches both of which are MLP based networks. The first network re-scores detections by utilizing context information. The result of the first network is logits for detections obtained by contextual inference. To make the predictions of first network more robust to errors, the result is aggregated with results of RetinaNet using a sigmoid gate. To aggregate logits obtained by the first network with the class probabilities obtained by RetinaNet, it is required to convert the class probabilities to logits. For that purpose, a second MLP network is used. Both of the sibling networks consist of 3 layer MLP. Since the second network could be in a simpler structure, we evaluate the Gated MLP model by removing 2 layers from the second network; only 1 FC layer is used. Another model we evaluate for the second network is to subtract a learnable $mu$ vector from the scores of query detection and multiply the result with a $sigma$ vector. The results are given in table 4.6.

Table 4.5: **AP results for different structures of the MLP model.** Results obtained by using only spatial regions and results obtained by only scale-related regions are compared. Also, results by cross entropy loss for binary classification are listed.

| Method | AP | $AP^{50}$ | $AP^{75}$ | $AP^{small}$ | $AP^{medium}$ | $AP^{large}$ |
|---|---|---|---|---|---|---|
| Baseline + MLP | **35.1** | **55.3** | 37.0 | 19.5 | **38.5** | 46.5 |
| Only Spatial Regions | **35.1** | 55.2 | **37.1** | 19.5 | 38.3 | 46.5 |
| Only Scale Regions | 35.0 | 55.1 | 37.0 | 19.5 | 38.4 | 46.3 |
| Binary Classification | 35.0 | 55.2 | 37.0 | 19.5 | 38.3 | **46.7** |

Table 4.6: **AP results for different structures of the Gated MLP model.**

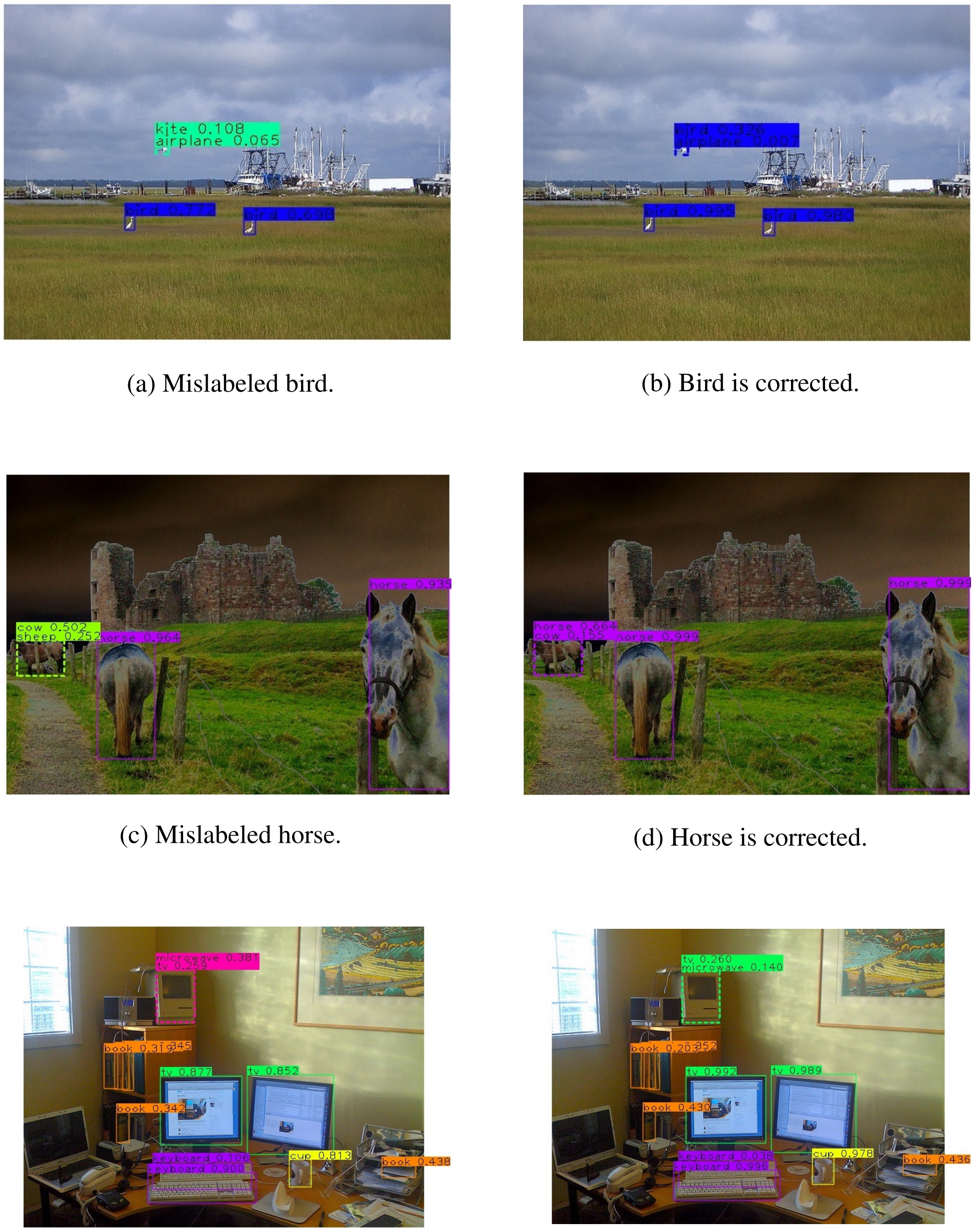
| Method | AP | AP$^{50}$ | AP$^{75}$ | AP$^{small}$ | AP$^{medium}$ | AP$^{large}$ |
|---|---|---|---|---|---|---|
| Baseline + Gated MLP | **35.3** | **55.5** | **37.3** | 19.4 | **38.7** | **47.2** |
| Single FC layer | 35.0 | 55.1 | 37.0 | 19.4 | 38.2 | 46.7 |
| mu-sigma | 35.1 | 55.2 | 37.0 | 19.4 | 38.5 | 46.4 |

### 4.8 Qualitative Results

In this section, we demonstrate mislabeled examples by base object detector that are corrected by Gated MLP and examples that are mislabeled by Gated MLP although they are labeled correctly by baseline. Figure 4.4 demonstrates detections corrected by our model. In the first case, the bird in the sky is mislabeled as kite by RetinaNet while our method correctly labels this small object. Also scores of other bird instances in the same scene are increased by our method. For the next pair in the figure, our method corrects the horse object that is mislabeled by RetinaNet as cow. In the final pair of scenes microwave is scored higher than tv for tv object by RetinaNet, while our method decreases the score for microwave.

Figure 4.5 demonstrates the background regions that are mislabeled by RetinaNet as detection while our method removes these background detections. In the first case a background region is classified as suitcase by RetinaNet based on the visual appearance while our method removes the detection based on the detections in the same scene. For the other cases, our method removes hair drier, broccoli and mouse detections of RetinaNet based on object to object context information.

Figure 4.6 demonstrates detections that are labeled correctly by baseline but mislabeled by Gated MLP. In the first case a book is labeled as keyboard. A tennis racket is mislabeled as bicycle and a teddy bear is mislabeled as vase by Gated MLP although they are labeled correctly by base object detector.



(a) Mislabeled bird.

(b) Bird is corrected.

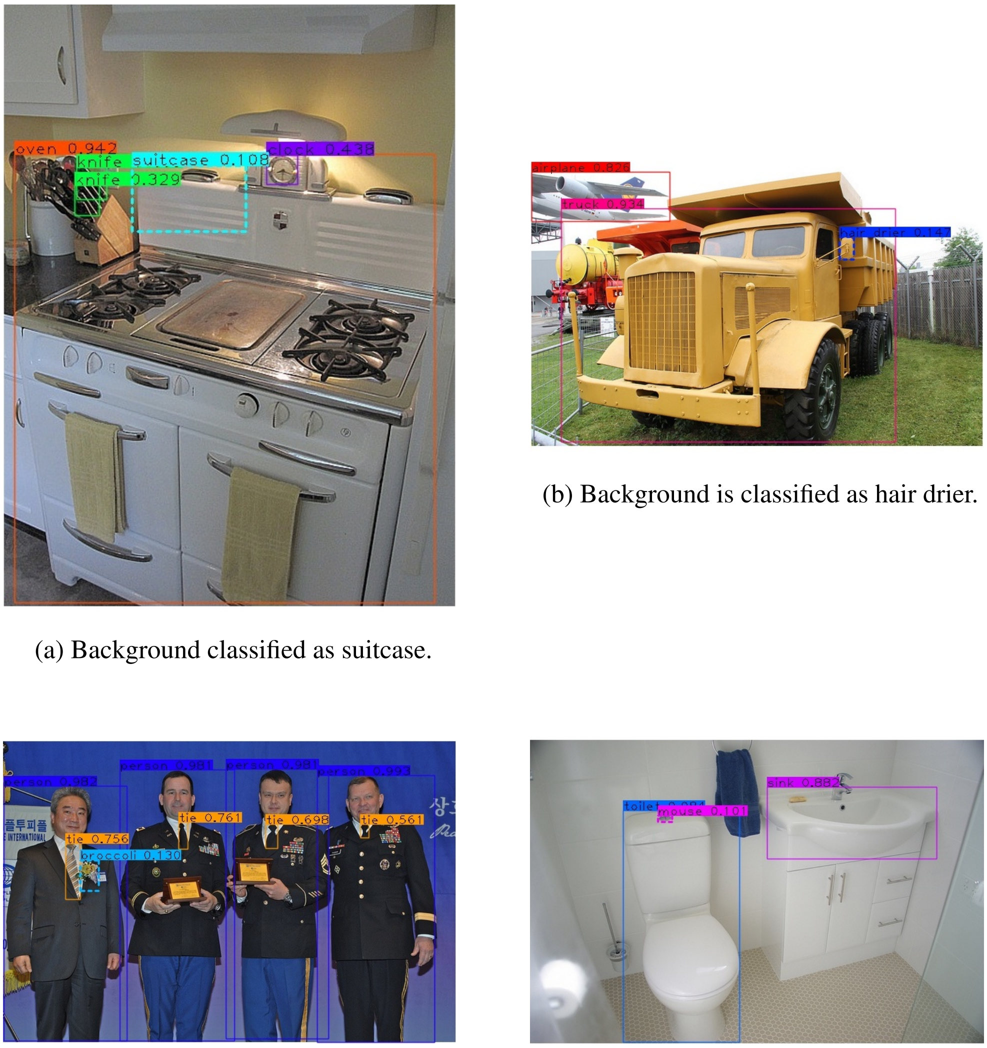
(c) Mislabeled horse.

(d) Horse is corrected.

(e) Tv labeled as microwave.

(f) Confidence score of microwave is decreased.

Figure 4.4: **Qualitative results of baseline vs Gated MLP on MS COCO.** In every pair, left is based on baseline, right is based on Gated MLP. Detections drawn with dashed-line boundaries are mislabeled by baseline and corrected by Gated MLP while detections drawn with solid line boundaries are labeled correctly. Top 2 class confidence scores for corrected detections are provided for both methods.



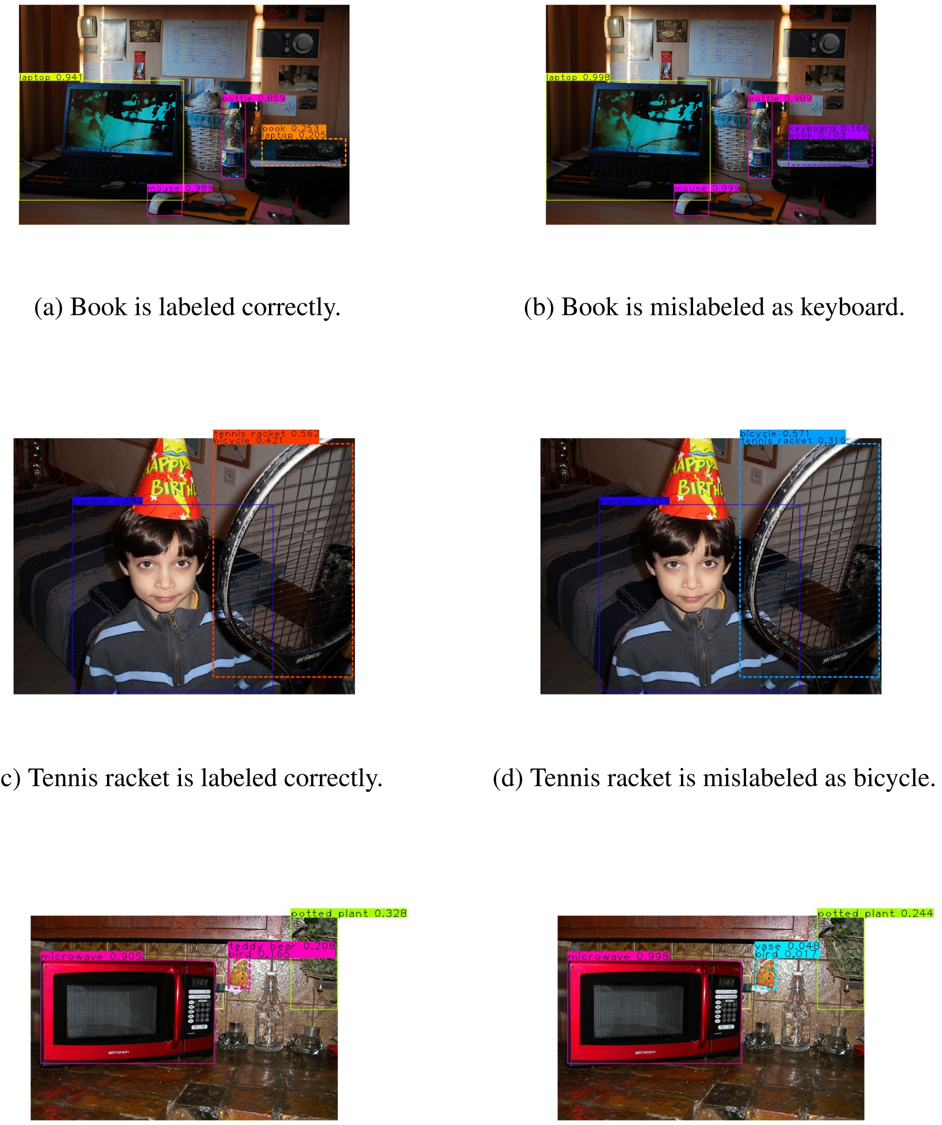
(a) Background classified as suitcase.

(b) Background is classified as hair drier.

(c) Background is classified as broccoli.

(d) Background is classified as mouse.

Figure 4.5: **Qualitative results of baseline vs Gated MLP on MS COCO.** Background regions labeled as objects by baseline method are corrected by Gated MLP. Background detections of baseline model that are removed by Gated MLP are drawn with dashed-line boundaries while detections labeled correctly by baseline are drawn with solid line boundaries.



(a) Book is labeled correctly.

(b) Book is mislabeled as keyboard.

(c) Tennis racket is labeled correctly.

(d) Tennis racket is mislabeled as bicycle.

(e) Teddy bear is labeled correctly.

(f) Teddy bear is mislabeled as vase.

Figure 4.6: **Qualitative results of baseline vs Gated MLP on MS COCO.** In every pair, left is based on baseline, right is based on Gated MLP. Detections drawn with dashed-line boundaries are mislabeled by Gated MLP although they are labeled correctly by baseline while detections drawn with solid line boundaries are labeled correctly. Top 2 class confidence scores for falsified detections are provided for both methods.

# CHAPTER 5

# CONCLUSION

In this thesis, we propose a contextual method that works on predictions of any common object detector to improve the object detection performance. Obtaining detections predicted by any object detector, our contextual method uses MLP based models to utilize the spatial and scale-based relationships between detections and refine misclassified detections. Our method does not propose new candidate object locations, only re-scores existing detections by applying a post-processing step on them.

Experiments using RetinaNet predictions on the COCO dataset show that our models decrease the percentage of false positives obtained by the base detector, and improve the mAP results. We review the false positive distribution of predictions of the baseline detector and one of our models and observe that our model decreases the percentage of false positives categorized as *similar* and *other*. Percentage of localization errors increases with our model since we do not regress bounding boxes; only refine the class predictions and as we correct the mislocalized detections, localization errors increase. We evaluated one of our models using the ground-truth labels of detections except for the query detection, and observe a 4.4% increase on mAP over the baseline results shows that context information is available in images that can be used to improve results of object detectors further.

We evaluated our models only on predictions of RetinaNet detector but they are applicable to any object detector. In this work, we evaluate our model after obtaining the predictions of the detector on training, validation and test datasets but it can be integrated to the detection pipeline to train and evaluate the model end to end with the object detector. We leave this as future work.

# REFERENCES

[1] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014.

[2] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[3] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, 2013.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[5] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014.

[7] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[8] J. S. D. R. G. A. F. Redmon, "(YOLO) You Only Look Once," in *CVPR proceedings*, 2016.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[10] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning: data mining, inference, and prediction, Springer Series in Statistics," 2009.

[11] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[12] N. Dvornik, J. Mairal, and C. Schmid, "On the Importance of Visual Context for Data Augmentation in Scene Understanding," *CoRR*, vol. abs/1809.0, sep 2018.

[13] D. Dwibedi, I. Misra, and M. Hebert, "Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[14] X. Chen and A. Gupta, "Spatial Memory for Context Reasoning in Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[15] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016.

[16] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation Networks for Object Detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018.

[17] Y. Liu, R. Wang, S. Shan, and X. Chen, "Structure Inference Net: Object Detection Using Scene-Level Context and Instance-Level Relationships," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. abs/1807.0, 2018.

[18] W. Chu and D. Cai, "Deep feature based contextual model for object detection," *Neurocomputing*, 2018.

[19] J. Li, Y. Wei, X. Liang, J. Dong, T. Xu, J. Feng, and S. Yan, “Attentive contexts for object detection,” *IEEE Transactions on Multimedia*, 2017.

[20] Z. Chen, S. Huang, and D. Tao, “Context refinement for object detection,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.

[21] Y. Kim, T. Kim, B.-N. Kang, J. Kim, and D. Kim, “{BAN:} Focusing on Boundary Context for Object Detection,” *CoRR*, vol. abs/1811.0, 2018.

[22] N. Arbel, T. Avraham, and M. Lindenbaum, “Inner-Scene Similarities as a Contextual Cue for Object Detection,” *CoRR*, vol. abs/1707.0, 2017.

[23] I. Bozcan, Y. Oymak, I. Z. Alemdar, and S. Kalkan, “What is (missing or wrong) in the scene? {A} Hybrid Deep Boltzmann Machine For Contextualized Scene Modeling,” *CoRR*, vol. abs/1710.05664, 2017.

[24] D. Hoiem, Y. Chodpathumwan, and Q. Dai, “Diagnosing error in object detectors,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012.

[25] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *International Journal of Computer Vision*, 2010.

[26] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[27] K. Oksuz, B. Cam, E. Akbas, and S. Kalkan, “Localization Recall Precision (LRP): A New Performance Metric for Object Detection,” in *European Conference on Computer Vision (ECCV)*, 2018.

[28] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic gradient descent,” in *ICLR: International Conference on Learning Representations*, 2015.

[29] I. Sutskever, G. Hinton, A. Krizhevsky, and R. R. Salakhutdinov, “Dropout : A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, 2014.

[30] V. Nair and G. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” in *Proceedings of the 27th International Conference on Machine Learning*, 2010.