CORRELATION LOSS: ENFORCING CORRELATION BETWEEN CLASSIFICATION AND LOCALIZATION IN OBJECT DETECTION

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

BY

FEHMİ KAHRAMAN

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

AUGUST 2022

Approval of the thesis:

CORRELATION LOSS: ENFORCING CORRELATION BETWEEN CLASSIFICATION AND LOCALIZATION IN OBJECT DETECTION

submitted by **FEHMİ KAHRAMAN** 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** _____

Assoc. Prof. Dr. Sinan Kalkan
Supervisor, **Computer Engineering, METU** _____

Assist. Prof. Dr. Emre Akbaş
Co-supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Assist. Prof. Dr. Hande Alemdar
Computer Engineering, METU _____

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

Assist. Prof. Dr. Cemil Zalluhoğlu
Computer Engineering, Hacettepe University _____

Date: 18.08.2022

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: Fehmi Kahraman

Signature :

ABSTRACT

CORRELATION LOSS: ENFORCING CORRELATION BETWEEN CLASSIFICATION AND LOCALIZATION IN OBJECT DETECTION

Kahraman, Fehmi
M.S., Department of Computer Engineering
Supervisor: Assoc. Prof. Dr. Sinan Kalkan
Co-Supervisor: Assist. Prof. Dr. Emre Akbaş

August 2022, 42 pages

Object detectors are conventionally trained by a weighted sum of classification and localization losses. Recent studies (e.g., predicting IoU with an auxiliary head, Generalized Focal Loss, Rank & Sort Loss) have shown that forcing these two loss terms to interact with each other in non-conventional ways creates a useful inductive bias and improves performance. Inspired by these works, we focus on the correlation between classification and localization and make two main contributions in this thesis: (i) We provide an analysis about the effects of correlation between classification and localization tasks in object detectors. We identify why correlation affects the performance of various NMS-based and NMS-free detectors, and we devise performance measures to evaluate the effect of correlation and use them to analyze common detectors. (ii) Motivated by our observations, e.g., that NMS-free detectors can also benefit from correlation, we propose Correlation Loss, a novel plug-in loss function that improves the performance of various object detectors by directly optimizing correlation coefficients: E.g., Correlation Loss on Sparse R-CNN, an NMS-free method, yields 1.6 AP gain on COCO dataset. Our best model on Sparse R-CNN reaches 51.0 AP

without test-time augmentation on COCO test-dev, reaching state-of-the-art.

Keywords: Object detection, correlation, classification and localization, loss function

ÖZ

KORELASYON KAYIP FONKSİYONU: NESNE TESPİTİNDE SINIFLANDIRMA İLE KONUMLANDIRMA ARASINDAKİ KORELASYONU ARTIRMAK

Kahraman, Fehmi
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Sinan Kalkan
Ortak Tez Yöneticisi: Dr. Öğr. Üyesi. Emre Akbaş

Ağustos 2022, 42 sayfa

Nesne tespit edicileri, geleneksel olarak, sınıflandırma ve konumlandırma kayıplarının ağırlıklı toplamı ile eğitilir. Son zamanlardaki çalışmalar (örneğin, auxiliary head ile IoU'yu tahmin etme, Generalized Focal Loss, Rank & Sort Loss), bu iki kayıp terimini geleneksel olmayan yollarla birbirleriyle etkileşime girmeye zorlamanın yararlı bir model varsayımı oluşturduğunu ve iyileştirmeler yaptığını göstermiştir. Bu çalışmalardan esinlenerek, bu tezde sınıflandırma ve konumlandırma arasındaki korelasyona odaklanıyoruz ve iki ana katkıda bulunmayı amaçlıyoruz: (i) Nesne tespit edicilerde sınıflandırma ve konumlandırma görevleri arasındaki korelasyonun etkileri hakkında bir analiz sunuyoruz. Korelasyonun çeşitli Maksimum-Olmayanı-Bastırma(NMS) tabanlı ve NMS içermeyen nesne tespit edicilerin performansını neden etkilediğini ortaya koyuyor, korelasyonun etkisini değerlendirmek için performans ölçümleri tasarlıyor ve bunları yaygın kullanılan nesne tespit edicileri analiz etmek için kullanıyoruz. (ii) NMS içermeyen nesne tespit edicilerin de korelasyondan yararlanabileceği gibi analizleri içeren analizlerimiz sonucunda korelasyon katsayı-

larını doğrudan optimize ederek çeşitli nesne tespit edicilerin performansını artıran yeni bir tak-çalıştır kayıp fonksiyonu olan Korelasyon Kayıp Fonkisyonunu (Correlation Loss) öneriyoruz: Örneğin, Korelasyon Kayıp Fonksiyonu, NMS içermeyen bir yöntem olan Sparse R-CNN ile COCO veri setinde 1.6 AP puan iyileştirme sağlamaktadır. En iyi modelimiz Sparse R-CNN test safhasında veri artırması yöntemini kullanmadan COCO tes-dev veri setinde 51.0 AP puanı ile diğer tüm modellerin seviyesine ulaşmıştır.

Anahtar Kelimeler: Nesne tespiti, korelasyon, sınıflandırma ve konumlandırma, kayıp fonksiyonu

This thesis is dedicated to my lovely daughter, Nil İpek.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisors, Dr. Sinan Kalkan and Dr. Emre Akbaş, for their invaluable advice and continuous support. Their guidance supported and encouraged me all the time during my research.

I am also thankful to the members of my thesis committee, Dr. Hande Alemdar and Dr. Cemil Zalluhoğlu for their insightful comments and time.

I want to thank my colleague Kemal Öksüz for his technical support, advice, and positive attitude.

The numerical calculations reported in this thesis are mainly performed on the resources of TÜBİTAK ULAKBİM High Performance and Grid Computing Center (TRUBA). Therefore, I thank this organization for the support. This work was partially supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) (under grant 120E494).

Finally, I would like to express my gratitude to my wife, daughter, and parents. Without the understanding and encouragement of my beloved wife, Gökçin, over the past few years, it would be impossible for me to complete my study.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

AP	Average Precision
NMS	Non-Maximum-Suppression
GT	Ground Truth
RS	Rank & Sort
FL	Focal Loss
QFL	Quality Focal Loss
IoU	Intersection-over-Union
LRP	Localisation Recall Precision
CNN	Convolutional Neural Networks
BB	Bounding Box
FN	False Negative
FP	False Positive
TP	True Positive
SOTA	State-of-the-art
wrt.	with respect to

CHAPTER 1

INTRODUCTION

Object detection, a fundamental computer vision task, involves drawing a bounding box around each object (object localization) and proposing the correct class label for each object (object classification) in a given image (see Figure 1.1). Thanks to advances in deep convolutional neural networks, significant progress has been made in object detection [4, 6, 11, 12, 13, 14, 15, 16, 17, 18, 19] in recent years.

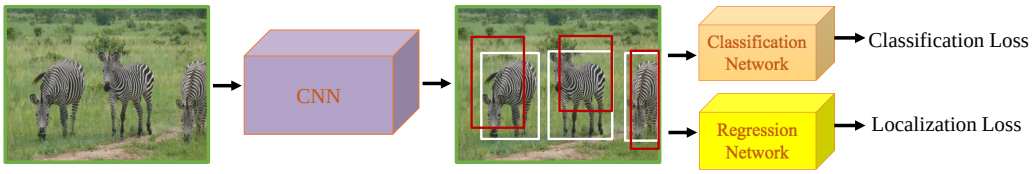


Figure 1.1: Object detection pipeline.

1.1 Motivation and Problem Definition

Most object detectors optimize a weighted sum of classification ($\mathcal{L}_{cls}$) and localization losses ($\mathcal{L}_{loc}$) during training. Owing to their classification and localization (box regression) tasks, object detection methods have an overall objective loss [10] defined as:

$$\mathcal{L}_{OD} = \sum_{k \in K} \sum_{t \in T} \lambda_t^k \mathcal{L}_t^k, \tag{1.1}$$

which combines the loss functions $\mathcal{L}_t^k$ for task t on stage k, weighted by λ.

Results from recent work suggest that performance improves when these two loss functions ($\mathcal{L}_{cls}$ and $\mathcal{L}_{loc}$) are forced to interact with each other in non-conventional ways. Object detectors typically output a bounding box together with an associated class ID and a classification score for each detected object. During training, loss functions try to maximize both the classification score (pertaining to the correct class) and its localization quality (as measured by Intersection-over-Union (IoU) with the ground-truth box). Instead of training classification and localization tasks independently (Figure 1.2(a)), several methods have been recently proposed to improve the correlation between classification scores and IoUs between these tasks.

For example, training an auxiliary head to regress the localization qualities of the positive examples, e.g. centerness, IoU or mask-IoU, has proven useful [1, 2, 6, 7] (Figure 1.2(b)). Other methods remove such auxiliary heads and aim directly to enforce correlation in the classification or localization task during training. Among these methods, Average LRP Loss [8] mainly weighs the examples in the localization task by ranking them with respect to (wrt.) their classification scores (Figure 1.2(c)). Using localization quality as an additional supervision signal for classification has been more commonly adopted (Figure 1.2(d)) [9, 10, 20, 21] in two main ways:

1. Score-based approaches aim to regress the localization qualities [9, 21, 22] in the classification score.

2. Ranking-based approaches enforce the classifier to rank the confidence scores with respect to the localization qualities [10, 20].

Despite this growing literature [1, 2, 6, 7, 8, 9, 10, 20, 21, 23], the effect of correlation on object detectors has not been thoroughly studied. In this thesis, we first identify that correlation affects the performance of object detectors at two levels:

- *Image-level correlation*, the correlation between the classification scores and localization qualities of the detections in a single image before post-processing, which is important to promote NMS performance

- *Class-level correlation*, the correlation over the entire dataset for each class after post-processing, which is related to the COCO-style Average Precision

(AP), the de facto performance measure of object detection.

Moreover, we quantitatively define correlation at each level to enable analyses on how well an object detector captures correlation (e.g.β_{cls} in Figure 1.3(a)). Then, we provide an analysis on both levels of correlation and draw important observations using common models. Finally, to better exploit correlation, we introduce a more direct mechanism to enforce correlation: *Correlation Loss*, a simple plug-in and detector-independent method (Figure 1.2(e)), improving performance for a wide range of object detectors including NMS-free detectors, aligning with our analysis (Figure 1.3(b)). Similar to the novel loss functions [9, 10, 21], our Correlation Loss also boosts the performance without an auxiliary head, but different from them, it is a simple plug-in technique that can easily be incorporated into any object detector, whether NMS-based or NMS-free.

1.2 Contributions and Novelties

In this thesis, we propose the following main contributions:

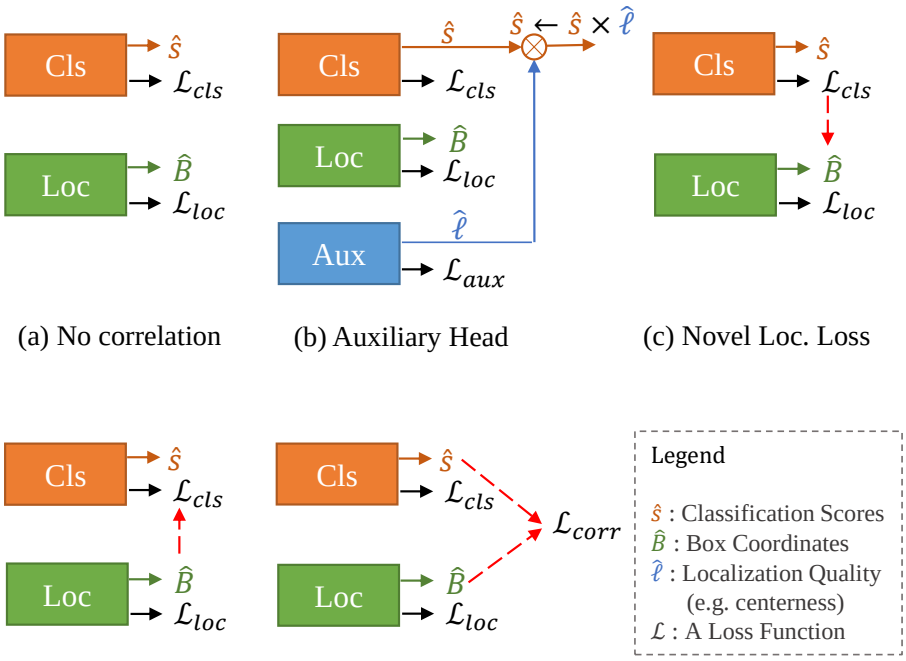
- We identify how correlation affects NMS-based and NMS-free detectors, and design appropriate performance measures to evaluate a detector with respect to its correlation performance.

- We analyze the effects of correlation at different levels on various object detectors.

- We propose Correlation Loss as a plug-in loss function to optimize correlation explicitly. Thanks to its simplicity, our loss function can be easily incorporated into a diverse set of object detectors and improves the performance of e.g., Sparse R-CNN up to 1.6 AP and $2.0\mathrm{AP}_{75}$, suggesting, for the first time, that NMS-free detectors can also benefit from correlation. Our best method reaches 51.0 AP, reaching state-of-the art.

1.3 The Outline of the Thesis

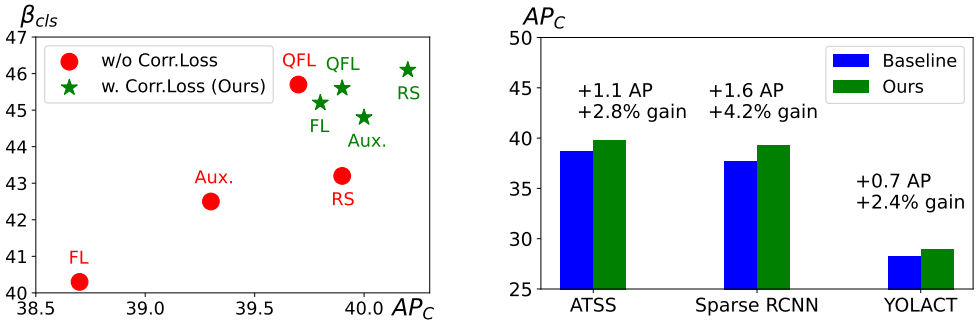
This thesis consists of six chapters. Chapter 2 presents the background and studies related to NMS-based and NMS-free object detectors and provides a summary of three different correlation coefficients. In Chapter 3, we analyze the effects of correlation on object detectors. Chapter 4 introduces a plug-in loss function, Correlation Loss. In Chapter 5, we introduce our experimental results. Finally, Chapter 6 concludes this thesis.



(a) No correlation (b) Auxiliary Head (c) Novel Loc. Loss

(d) Novel Cls. Loss (e) Correlation Loss (Ours)

Figure 1.2: Different ways of handling the classification and localization tasks from the perspective of correlation. **(a)** Conventional case of optimizing the two tasks independently (e.g., [3, 4, 5]). **(b)** An additional auxiliary head predicts centerness [1, 6] or IoU [2, 7], which introduces additional learnable parameters. **(c)** Novel loss functions replace the standard localization loss [8] or **(d)** classification loss [9, 10] by more complicated ones to leverage correlation. **(e)** Our Correlation Loss explicitly optimizes a correlation coefficient. It is a simple, plug-in loss function which does not introduce additional parameters. Black and colored arrows respectively denote the loss functions (i.e., during training) & the network outputs (i.e., during inference).



(a) Detection vs. Correlation

(b) Effect of our Corr. Loss

Figure 1.3: **(a)** Detection performance, measured by COCO-style AP ($\mathrm{AP_C}$) vs. correlation quality, measured by class-level correlation (β_{cls} - see Section **??** for details). The methods proposed to improve the correlation between classification and localization tasks also improve $\mathrm{AP_C}$. Compare using auxiliary head, QFL, RS Loss with the baseline ATSS only using Focal Loss (FL – all in red dots) to see the positive correlation between $\mathrm{AP_C}$ and β_{cls}. Our Correlation Loss as a plug-in loss function explicitly optimizes a correlation coefficient and improves the detection performance ($\mathrm{AP_C}$) over different settings of ATSS (i.e. using FL, auxiliary head, QFL, RS Loss) consistently owing to increasing β_{cls}, validating our hypothesis (compare green stars with red dots). **(b)** Our Correlation Loss as a plug-in loss function optimizes a correlation coefficient and improves (i) ATSS [1] without auxiliary head, an NMS-based detector by $1.1\mathrm{AP_C}$, (ii) Sparse R-CNN, an NMS-free detector by $1.6\mathrm{AP_C}$ and (iii) YOLACT, an instance segmentation method by $0.7\mathrm{AP_C}$.

CHAPTER 2

BACKGROUND AND RELATED WORK

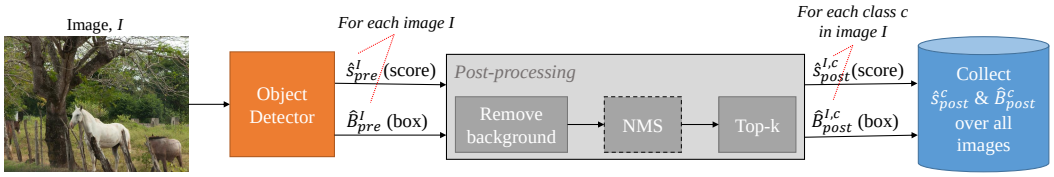


Figure 2.1: Object detection pipeline and notation. Given an input image, I, NMS-based detectors yield raw detections before post-processing, each of which has a predicted bounding box (BB) and an array of confidence scores over ground truth classes. We denote the confidence scores and the predicted bounding boxes pertaining to the *positive detections*, i.e., the detections matching with ground truth objects during training, by $\hat{s}^I_{pre}$ and $\hat{B}^I_{pre}$ respectively. In order to obtain final detections, raw detections are post-processed in three steps: (i) Detections with low confidence scores, i.e., background, are removed, (ii) duplicates are eliminated by NMS, and (iii) top-k scoring detections are kept. As for these final detections, we denote the confidence scores and bounding boxes of *true positive detections* for class c in a single image I by $\hat{s}^{I,c}_{post}$ and $\hat{B}^{I,c}_{post}$ respectively, and over the entire dataset by $\hat{s}^c_{post}$ and $\hat{B}^c_{post}$. As for NMS-free detectors; NMS, dashed gray box in post-processing, is excluded, hence post-processing is lighter.

2.1 Object Detection Pipeline.

We group object detectors considering their usage of NMS (see Figure 2.1 for an overview and notation).

2.1.1 NMS-based Detectors

To detect all objects with different scales, locations and aspect ratios; most methods [1, 4, 6, 13, 14, 24, 25, 26] employ a large number of object hypotheses (e.g., anchors, points), which are labeled as positive (also known as foreground) or negative (also known as background) [1, 27, 28, 29] during training, based on whether/how they match ground truth boxes. In this setting, there is no restriction to prevent an object to be predicted by multiple object hypotheses, causing duplicates. Accordingly, during inference, NMS picks the detection with the largest confidence score among the detections that overlap more than a predetermined IoU threshold to avoid duplicate detections.

2.1.2 NMS-free Detectors

An emerging research direction in the community is to remove the need for doing NMS, simplifying detection pipeline. They [5, 30, 31, 32, 33, 34] achieve this by ensuring a one-to-one matching between the ground truths and detections, which supervises the detector to avoid duplicates in the first place.

2.2 Methods Enforcing Correlation

One common way to ensure correlation is to use an additional auxiliary head, supervised by the localization quality of a detection such as centerness [1, 6], IoU [2, 7], mask IoU [35] or uncertainty [36], during training. Unlike these methods focusing on loss function design, TOOD [37] mainly focuses on the detection head design considering correlation.

Centerness. FCOS [6] presents centerness branch (see Figure 2.2) to suppress the low-quality detected bounding boxes produced by the locations far from the center of an object. [1] proposes the adaptive training sample selection, which automatically divides positive and negative training samples according to the statistical characteristics of the object.

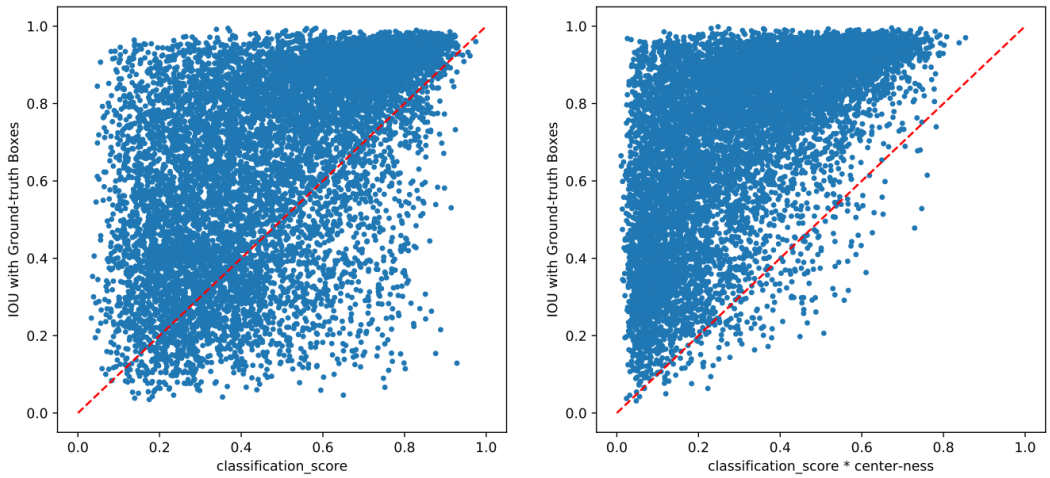


Figure 2.2: The plot on the left side is without centerness, and the plot on the right is with centerness. A dot in the figure indicates a detected bounding box. The dashed red line is the line $y = x$. As shown in the figure (right), the low-quality boxes which are below the line $y = x$ are pushed to above the line. It shows that the scores of low-quality boxes are reduced significantly. Figure is from [6].

IoU. [7] predicts the IoU between detected bounding boxes and their corresponding ground-truth boxes, acquiring localization confidence. [2] combines the score of classification and localization qualities serving as a box selection metric in non-maximum suppression.

During inference, the predictions of the auxiliary head are then combined with those of the classifier to improve detection performance. Recent methods show that the auxiliary head can be removed in two ways:

1. The regressor can prioritize the positive examples [8].

2. The classifier can directly be supervised to prioritize detections with confidence scores. This is ensured either by regressing the localization qualities by the classifier [9, 21] or by training the classifier to rank confidence scores [10, 20] with respect to the localization qualities.

2.3 Correlation Coefficients

Correlation coefficients measure the strength and direction of the "relation" between two sets of data values, $X = \{x_1, ..., x_N\}$ and $Y = \{y_1, ..., y_N\}$. Different relations are evaluated by different correlation coefficients. All correlation coefficients have a range of $[-1, +1]$ where positive/negative correlation corresponds to increasing/decreasing relation, while 0 implies no correlation between X and Y.

2.3.1 Pearson Correlation Coefficient

Pearson correlation coefficient, denoted by $\alpha(\cdot, \cdot)$, measures the linear relationship between the sets. The closeness of points (see Figure 2.3) determines the correlation.The slope of the line formed by the data points does not indicate the Pearson Correlation. Pearson correlation coefficient is defined as:

$$\alpha(X, Y) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}, \tag{2.1}$$

where x_i and y_i are the i^{th} element of X and Y respectively. $\bar{x}$ and $\bar{y}$ are the mean of X and Y. N is sample size.

2.3.2 Spearman Correlation Coefficient

Spearman correlation coefficient, $\beta(\cdot, \cdot)$, corresponds to the ranking relationship. It is obtained by ranking the values of the two data sets (X and Y) and then calculating the Pearson on the resulting ranks. Spearman correlation coefficient is defined as:

$$\beta(X, Y) = \alpha(R(X), R(Y)) = \frac{cov(R(X), R(Y))}{\sigma_{R(X)}\sigma_{R(Y)}}, \tag{2.2}$$

where $R(X)$ and $R(Y)$ are the rank values of the data sets, cov is the covariance and σ is the standard deviation.

2.3.3 Concordance Correlation coefficient

Concordance correlation coefficient, $\gamma(\cdot, \cdot)$, is more strict than Pearson correlation coefficient, measuring the similarity of the values and maximized when $x_i = y_i$ for

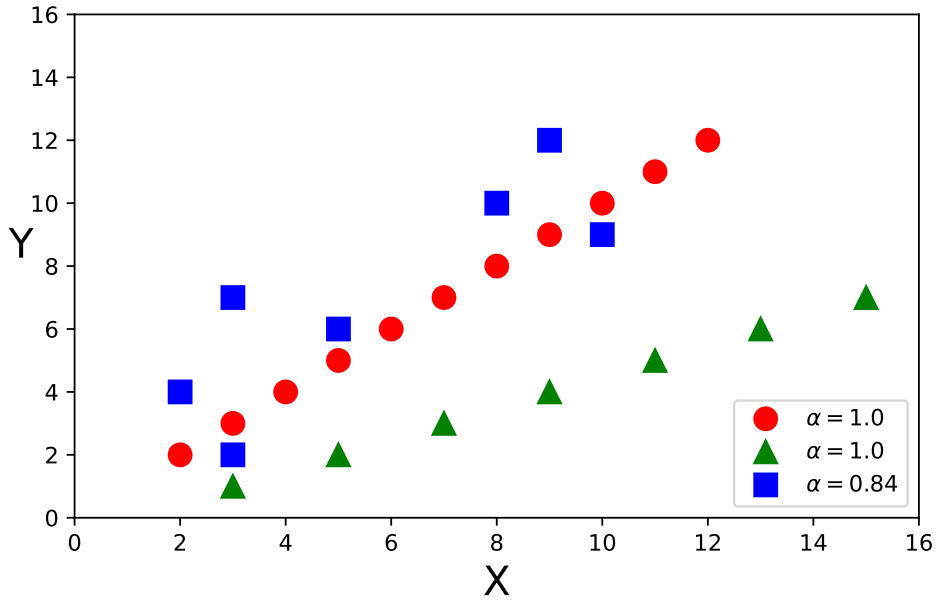


Figure 2.3: Pearson Correlation Coefficient. $\alpha(X, Y)$ is the same for red circles and green triangles but different for the blue squares. Changing the scale of measurement does not change the value of the correlation (red and green).

all $i \in 1, ..., N$. The concordance correlation line (see Figure 2.4) passes through the origin[38]. Concordance correlation coefficient is defined as:

$$\gamma(X, Y) = \frac{2\alpha(X, Y)\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2 + (\bar{x} - \bar{y})^2}, \tag{2.3}$$

where $\alpha(X, Y)$ is Pearson correlation coefficient. $\bar{x}$ and $\bar{y}$ are the mean of X and Y respectively, and σ_x^2 σ_y^2 are corresponding variances.

2.4 Comparative Summary

In this thesis, we comprehensively identify, measure, and analyze the effect of explicitly correlating classification and localization in object detectors. Unlike other methods that also enforce correlation, some of which are tested only on a single architecture [6, 7, 35], we propose a simple solution (see Figure 1.2) by directly optimizing the correlation coefficient, which is auxiliary-head free and easily applicable to *all* object detectors, whether NMS-based or NMS-free (see Table 2.1). Also, we are the first to work on NMS-free detectors in this context.

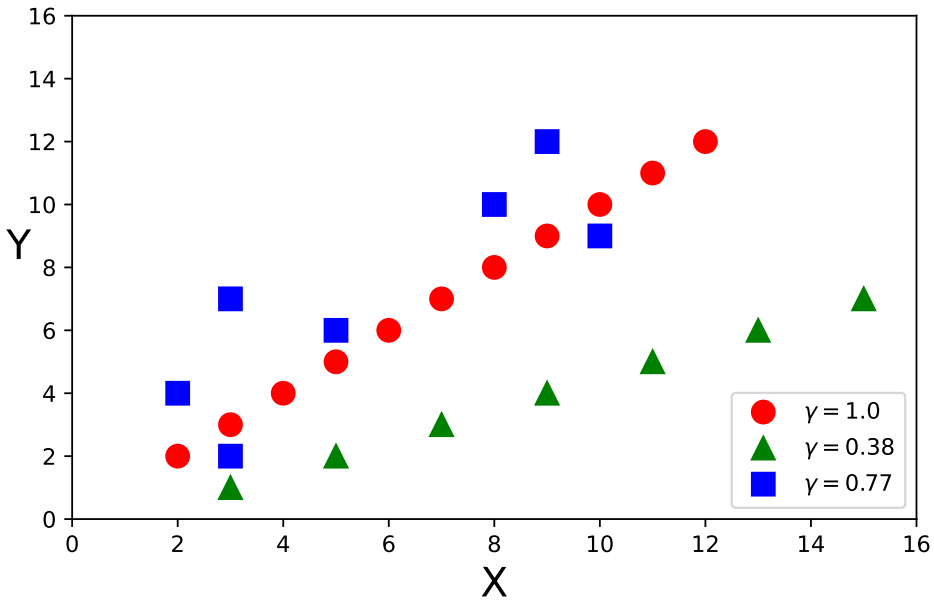


Figure 2.4: Concordance Correlation Coefficient. $\gamma(X, Y)$ is 1.0 for red circles because it passes through the origin. Unlike Pearson correlation coefficient, changing the scale of measurement changes the value of the Concordance correlation coefficient (red and green).

Table 2.1: Comparison of methods enforcing correlation. While existing methods can introduce additional learnable parameters, may not be incorporated into a model easily, or have not been applied to different types of detectors, our proposed Correlation Loss does not have such deficiencies.

Method	**Additional learnable parameters?**	**Plug-in?**	**Applied to NMS-based detectors?**	**Applied to NMS-free detectors?**
Using Aux Head	✗	✓	✓	✗
Quality Focal Loss	✓	✗	✓	✗
aLRP Loss	✓	✗	✓	✗
RS Loss	✓	✗	✓	✗
Correlation Loss (**Ours**)	✓	✓	✓	✓

CHAPTER 3

EFFECTS OF CORRELATION ON OBJECT DETECTORS

This section presents why maximizing correlation is important for object detectors, introduces performance measures to evaluate object detectors with respect to correlation (Section 3.1), and provides an analysis on common methods designed for improving correlation (Section 3.2).

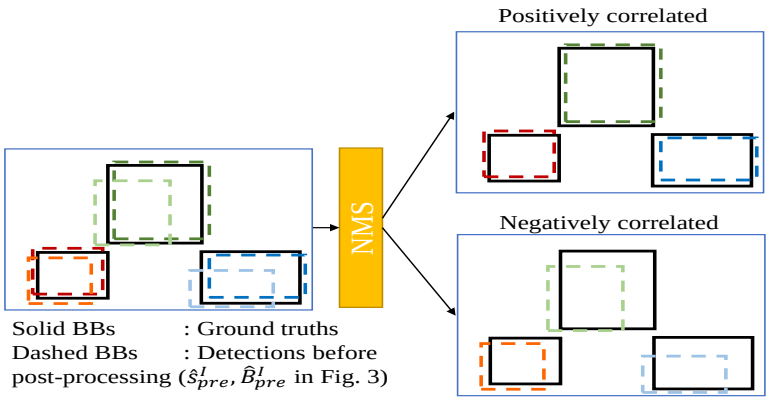
3.1 How Correlation Affects Object Detectors: Definition and Performance Measures

Object detectors are affected by correlation at two different levels: Image-level correlation and class-level correlation – see also Figure 3.1.

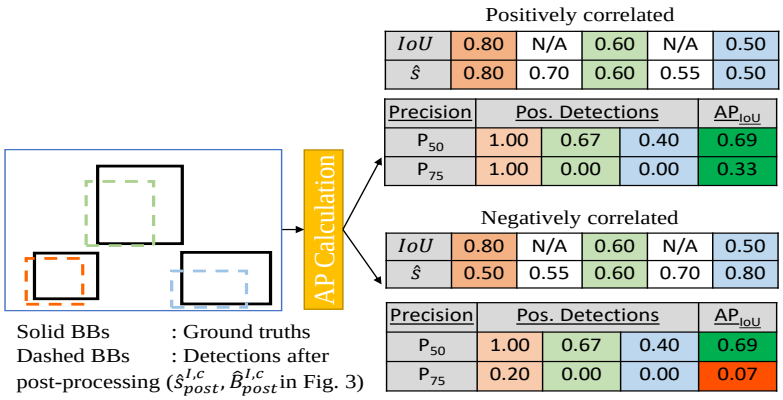
3.1.1 Image-level Correlation

This level of correlation corresponds to the correlation between the classification scores and localization qualities of the detections in a single image before post-processing, and accordingly, we measure it by the Spearman correlation coefficient[1], $\beta(\cdot,\cdot)$, averaged over images. Denoting the set of images to be evaluated by $\mathcal{I}$ and IoUs between the bounding boxes of the positive detections ($\hat{B}^I_{pre}$, Figure 2.1) and their associated ground truths by IoU^I_{pre}, the performance in terms of image-level correlation is measured as follows:

[1] While analyzing object detectors in terms of correlation, we employ Spearman correlation coefficient, $\beta(\cdot,\cdot)$, to measure the relation between the ranks of the values (i.e., scores and IoUs) instead of the values themselves, and aim to isolate the correlation performance from the localization and classification performances.



(a) Image-level Correlation for better NMS



(b) Class-level Correlation for better AP

Figure 3.1: How correlation affects object detection performance. **(a)** Image-level correlation. Given detections before post-processing, NMS benefits from positive image-level correlation, thereby yielding detections with better localization qualities. Compare the localization qualities of detections in "positively correlated" (i.e., when the dark-colored ones have larger score) and "negatively correlated" (i.e., when the light-colored ones have larger score) outputs after NMS. **(b)** Class-level correlation. Given detections after post-processing, APs with larger IoUs and COCO-style AP benefit from positive class-level correlation (compare $\mathrm{AP}_{\mathrm{IoU}}$ columns in "positively correlated" and "negatively correlated" outputs after AP Calculation to see lower AP_{75} for the "negatively correlated" output in the red cell). $\mathrm{P}_{\mathrm{IoU}}$: Precision computed on a detection using the threshold IoU, True positives are color-coded in tables and input, white cells: false positives, and hence their IoU is not available, N/A.

$$\beta_{img} = \frac{1}{|\mathcal{I}|} \sum_{I \in \mathcal{I}} \beta(\mathrm{IoU}^{I}_{pre}, \hat{s}^{I}_{pre}). \tag{3.1}$$

Maximizing image-level correlation is important for NMS-based detectors since NMS aims to suppress duplicates, i.e., to keep only a single detection for each ground truth when there is more than one. More particularly among overlapping detections (e.g., dark and light green detections in the detector output image in Figure 3.1(a)), NMS picks the one with the larger score, and hence, if there is positive correlation between the confidence scores and localization qualities of those overlapping detections, then the one with the best localization quality (e.g., dark green detection in the same example in Figure 3.1(a)) will survive and detection performance will increase.

3.1.2 Class-level Correlation

This level of correlation indicates the correlation between the classification scores and localization qualities of the detections obtained after post-processing for each class. Since class-level correlation is related to COCO-style Average Precision, $\mathrm{AP_C}$, we average $\beta(\cdot, \cdot)$ over classes to be consistent with the computation of $\mathrm{AP_C}$:

$$\beta_{cls} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \beta(\mathrm{IoU}^{c}_{post}, \hat{s}^{c}_{post}), \tag{3.2}$$

where $\mathcal{C}$ is the set of classes in the dataset and IoU^{c}_{post} is the set IoUs of BBs of true positives for class c ($\hat{B}^{c}_{post}$, Figure 2.1).

Class-level correlation affects the performance of all detectors since it is directly related to $\mathrm{AP_C}$, the performance measure itself. To be more specific, $\mathrm{AP_C}$ for a single class is defined as the average of APs computed over 10 different localization quality thresholds, $\mathrm{IoU} \in \{0.50, 0.55, ..., 0.95\}$, validating the true positives. For a specific threshold IoU, the detections are first sorted with respect to the classification scores, and then precision and recall pairs are calculated on each detection. Using these pairs, a precision-recall (PR) curve is obtained, and the area under the PR curve corresponds to the single AP value, $\mathrm{AP_{IoU}}$. When the correlation between classification and localization is maximized among true positives, larger precision values are obtained on

the same detections in larger IoU values (e.g. P_{75} of orange detection is 1.00 and 0.20 with positive and negative correlation respectively in Figure 3.1(b)).

3.2 Analysis of Object Detectors in Terms of Correlation

This section presents our analysis on how correlation affects object detectors.

See Section 5.1 for dataset and implementation details.

Table 3.1: Evaluation of NMS-based detectors in terms of image-level correlation. See Equation 3.1 for β_{img}. $\mathrm{AP}^{+1}_{\mathrm{IoU}}$ and $\mathrm{AP}^{-1}_{\mathrm{IoU}}$ refer to the upper & lower bound APs (see analysis setup for details). The values are in %. Our β_{img} captures correlation consistently, e.g. that (i) Focal Loss is improved by ctr. head and QFL and (ii) AP Loss is improved by aLRP Loss and RS Loss wrt. β_{img}. Also, there is still room for improvement for object detectors wrt. β_{img} with a range between 27.2% and 33.8%.

	Performance				Modify ranking of scores					
Method	AP_{C}	AP_{50}	AP_{75}	β_{img}	AP^{-1}_{C}	AP^{-1}_{50}	AP^{-1}_{75}	AP^{+1}_{C}	AP^{+1}_{50}	AP^{+1}_{75}
Not Enforcing Correlation										
ATSS w. AP Loss [3]	38.1	58.2	41.0	27.2	24.9	53.2	19.2	57.0	72.4	62.2
ATSS w. Focal Loss [4]	38.7	57.6	41.5	27.3	25.6	51.8	21.1	55.8	70.6	60.5
Using Aux. Head										
ATSS w. ctr. head [1]	39.3	57.5	42.6	28.7	22.4	45.9	19.0	40.6	56.8	41.8
Using Novel Loss										
ATSS w. aLRP Loss [8]	37.7	57.4	39.9	33.8	22.7	48.8	17.5	54.2	70.4	58.7
ATSS w. QFL [9]	39.7	58.1	42.7	33.2	25.7	51.1	21.9	55.8	70.9	60.6
ATSS w. RS Loss [10]	39.9	58.9	42.6	30.9	26.2	53.9	21.3	57.1	71.8	62.1

3.2.1 Analysis Setup

We conduct experiments to analyze the effects of the image-level (Table 3.1) and class-level (Table 3.2) correlations. For both analyses, we compare three sets of methods, all of which are incorporated into the common ATSS baseline [1] (see Chapter 2 for a discussion of these methods):

Table 3.2: Evaluation of object detectors wrt. class-level correlation. See Equation 3.2 for β_{cls}. $\mathrm{AP}^{+1}_{\mathrm{IoU}}$ and $\mathrm{AP}^{-1}_{\mathrm{IoU}}$ denote upper & lower bound APs (analysis setup for details). The values are in %. NMS-free detectors can also benefit from class-level correlation (compare $\mathrm{AP}^{+1}_{\mathrm{C}}$ with AP_{C} for Sparse R-CNN and DETR), and as in β_{img} (c.f. Table 3.1 and its caption), β_{cls} measures the correlation consistently.

	Performance				Modify ranking of scores					
Method	AP_{C}	AP_{50}	AP_{75}	β_{cls}	AP^{-1}_{C}	AP^{-1}_{50}	AP^{-1}_{75}	AP^{+1}_{C}	AP^{+1}_{50}	AP^{+1}_{75}
Not Enforcing Correlation										
- NMS-free Detectors										
Sparse R-CNN [5]	37.7	55.8	40.5	37.5	30.1	55.8	28.9	48.6	55.8	52.7
DETR [30]	40.1	60.6	42.0	47.0	32.9	60.6	30.6	51.9	60.6	55.8
- NMS-based Detectors										
ATSS w. AP Loss [3]	38.1	58.2	41.0	39.4	30.0	58.2	26.6	48.5	58.2	54.0
ATSS w. Focal Loss [4]	38.7	57.6	41.5	40.3	30.2	57.6	27.3	48.7	57.6	53.6
Using Aux. Head										
ATSS w. ctr. head [1]	39.3	57.4	42.5	42.5	30.2	57.4	27.6	48.7	57.4	53.5
Using Novel Loss										
ATSS w. aLRP Loss [8]	37.7	57.4	39.9	42.0	29.1	57.4	25.0	47.8	57.4	52.7
ATSS w. QFL [9]	39.7	58.1	42.7	45.7	30.6	58.1	27.7	49.1	58.1	53.9
ATSS w. RS Loss [10]	39.9	58.9	42.6	43.2	31.1	58.9	28.1	49.8	58.9	54.8

- AP Loss and Focal Loss as methods not enforcing correlation,
- Using an auxiliary head to enforce correlation,
- Quality Focal Loss (QFL) [9], aLRP Loss [8] and Rank & Sort Loss [10] as recent loss functions enforcing correlation.

In our class-level analysis, we also employ NMS-free methods to demonstrate the effects of correlation on that approach.

We compare the methods based on:

1. Their AP-based performance,
2. Our proposed evaluation measures for correlation (Equations 3.1 and 3.2),

3. Lower/upper bounds, $\mathrm{AP}_{\mathrm{C}}^{+1}/\mathrm{AP}_{\mathrm{C}}^{-1}$, obtained by modifying the ranking of the confidence scores pertaining to the GT classes of the positive detections to minimize/maximize Equation 3.1 in Table 3.1 and Equation 3.2 in Table 3.2.

More particularly, in Table 3.1, given $\hat{s}^I_{pre}$ and $\hat{B}^I_{pre}$ (Figure 2.1), we collect the ground truth class probabilities of positive detections and change their ranking in $\hat{s}^I_{pre}$ within an image following the ranking order of IoUs (computed using $\hat{B}^I_{pre}$), and in Table 3.2, we do the same operation class-wise for true positives given $\hat{s}^c_{post}$ and $\hat{B}^c_{post}$ (Figure 2.1). To decouple other types of errors as much as possible; in Table 3.1, we *do not modify* the scores of the negative detections (i.e., that match with background), the predicted BBs and the scores of non-GT classes of the positive detections, and in Table 3.2, we *do not modify* the scores of the false positives and the predicted BBs of the true positives. Note that achieving the upper bound for image-level correlation also corresponds to perfectly minimizing RS Loss.

3.2.2 Observations

Based on Tables 3.1 and 3.2, we observe the following:

(1) Our proposed performance measures in Equations 3.1 and 3.2 can measure the improvements in correlation consistently.

In Tables 3.1 and 3.2, (i) aLRP Loss [8] and RS Loss [10] are proposed to improve AP Loss baseline and (ii) using auxiliary head [1] and QFL [9] are proposed to improve Focal Loss baseline. In both tables, the proposed methods are shown to improve their baselines in terms of β_{img} and β_{cls}, suggesting that our measures can consistently evaluate image-level and class-level correlations respectively.

(2) NMS-free detectors can also potentially benefit from correlation.

All detectors, including NMS-free ones, can benefit from class-level correlation (compare AP_{C} and AP_{C}^{+1} to see ~ 10 points gap in Table 3.2). However, existing methods do not enforce this correlation on NMS-free detectors.

(3) Existing methods enforcing correlation have still a large room for improvement.

Considering that $\beta_{img} \in [27.2\%, 33.8\%]$ (Table 3.1) and $\beta_{cls} \in [37.5\%, 47.0\%]$ (Table 3.2), there is still room for improvement to reach the perfect correlation.

(4) While significantly important, improving correlation may not always imply performance improvement.

For example, aLRP Loss [8] in Table 3.1 has the largest correlation but the lowest $\mathrm{AP_C}$. Such a situation can arise, for example, when a method does not have good localization performance. In the extreme case, assume a detector yields perfect β_{img}, image-level ranking correlation, but the IoUs of all positive examples are less than 0.50 implying no TP at all. Hence, boosting the correlation, while simultaneously preserving a good performance in each branch, is critical.

CHAPTER 4

CORRELATION LOSS: A NOVEL LOSS FUNCTION FOR OBJECT DETECTION

Correlation Loss is a simple plug-in loss function to improve correlation of classification and localization tasks. Correlation Loss is unique in that it can be easily incorporated into any object detector, whether NMS-based or NMS-free (see Observation (2) - Section 3.2.2), and improves performance without affecting the model size, inference time and with negligible effect on training time.

4.1 Definition of Correlation Loss

Given an object detector with loss function $\mathcal{L}_{OD}$, our Correlation Loss ($\mathcal{L}_{corr}$) is simply added using a weighting hyper-parameter λ_{corr}:

$$\mathcal{L}_{OD} + \lambda_{corr}\mathcal{L}_{corr}. \tag{4.1}$$

$\mathcal{L}_{corr}$ is the Correlation Loss defined as:

$$\mathcal{L}_{corr} = 1 - \rho(\hat{\mathrm{IoU}}, \hat{\mathrm{s}}), \tag{4.2}$$

where $\rho(\cdot, \cdot)$ is a correlation coefficient; $\hat{\mathrm{s}}$ and $\hat{\mathrm{IoU}}$ are the confidence scores of the ground truth class and Intersection-over-Unions of the predicted BBs respectively, both of which pertaining to the positive examples in the batch.

4.2 Practical Usage

To avoid promoting trivial cases with high correlation but low performance (Observation (4) - Sec. 3.2.2), similar to QFL [9] and RS Loss [10], we only use the gradi-

ents of $\mathcal{L}_{corr}$ with respect to classification score, i.e., we backpropagate the gradients through only the classifier. We mainly adopt two different correlation coefficients for $\rho(\cdot, \cdot)$ and obtain two versions of Correlation Loss:

- *Concordance Loss*, defined as the Correlation Loss when Concordance correlation coefficient is optimized ($\rho(\cdot, \cdot) = \gamma(\cdot, \cdot)$), which aims to match the confidence scores with IoUs. Concordance Loss is differentiable.
- *Spearman Loss* as Correlation Loss when Spearman correlation coefficient is optimized ($\rho(\cdot, \cdot) = \beta(\cdot, \cdot)$), thereby enforcing the ranking of the classification scores considering IoUs. To tackle the non-differentiability of ranking operation while computing Spearman Loss, we leverage the differentiable sorting operation from Blondel et al. [39].

When applying our Correlation Loss to NMS-free methods, which use an iterative multi-stage loss function, we incorporate $\mathcal{L}_{corr}$ to every stage.

We also adopt Pearson correlation coefficient for $\rho(\cdot, \cdot)$ and obtain Pearson Loss as Correlation Loss when Pearson correlation coefficient is optimized ($\rho(\cdot, \cdot) = \alpha(\cdot, \cdot)$). Pearson Loss is differentiable. We observe that while Pearson Loss has similar performance with Concordance Loss on ATSS [1] and Spearman Loss on Sparse R-CNN [5], it does not outperform the other two in both of the cases (see Table 5.6 in Chapter 5).

It is generally sufficient to search over $\{0.1, 0.2, 0.3.0.4, 0.5\}$ to tune λ_{corr} for all the models we trained on COCO dataset [40] (see more details in Section 5.5). Outside of this range, performance drops.

CHAPTER 5

EXPERIMENTS

We evaluate our Correlation Loss on the COCO dataset [40] with five object detectors of different types:

- Sparse R-CNN [5], which is NMS-free,
- RetinaNet [4], which is anchor-based,
- FoveaBox [24], which is anchor-free,
- ATSS [1], which is anchor-based with a centerness head,
- PAA [2], which is an anchor-based recent strong baseline.

We also employ Correlation Loss on an instance segmentation method, YOLACT [41].

5.1 Dataset and Implementation Details

For the rest of the thesis:

- We employ the widely-used and public COCO dataset [40] licensed under Creative Commons Attribution 4.0.
- We train the models on *trainval35K* (115K images), test on *minival* (5k images) and compare with SOTA on *test-dev* (20k images) sets.

- We mainly report AP-based performance measures, and also use Optimal LRP (oLRP) [42, 43], β_{img} (Equation 3.1) and β_{cls} (Equation 3.2) to provide more insights.
- Our implementation is based on the mmdetection framework [44] with Pytorch.
- Unless otherwise explicitly specified, we keep the standard configuration of the models and use a ResNet-50 backbone with FPN [45].
- Unless otherwise explicitly specified, we train models for $1\times$ (12 epochs) training schedule for comparison.
- We use distributed training. We train models on 4 GPUs (A100 or V100 type GPUs in an internal cluster) with 4 images on each GPU (16 batch size).
- All PyTorch-style pretrained backbones on ImageNet [46] are from PyTorch [47] model zoo.

5.2 Comparison with Methods Not Enforcing Correlation

We train the aforementioned five object detectors and the instance segmentation method (Tables 5.1, 5.2 and 5.3) with and without our Correlation Loss (as Concordance Loss or Spearman Loss).

5.2.1 NMS-based Detectors

The results in Table 5.1 suggest $\sim 1.0\mathrm{AP_C}$ gains on NMS-based detectors:

- Spearman Loss ($\lambda_{corr} = 0.1$) improves RetinaNet by $1.0\mathrm{AP_C}$ and oLRP,
- Concordance Loss ($\lambda_{corr} = 0.2$) enhances FoveaBox [24], which is anchor-free, by $0.7\mathrm{AP_C}$,
- Concordance Loss ($\lambda_{corr} = 0.3$) improves ATSS [1] ($\lambda_{corr} = 0.3$) and PAA [2] by $\sim 1\mathrm{AP_C}$ and $\sim 1\mathrm{oLRP}$.

Table 5.1: Comparison on NMS-based and NMS-free detectors not considering correlation. Accordingly, we remove auxiliary heads from ATSS [1] and PAA [2] for fair comparison (see Table 5.4 for comparison with auxiliary heads and novel loss functions). We use ResNet-50 backbone and train the models for 12 epochs. Simply incorporating our Correlation Loss provides (i) $\sim 1\mathrm{AP_C}$ improvement for NMS-based detectors consistently and (ii) $\sim 1.5\mathrm{AP_C}$ on the NMS-free Sparse R-CNN.

Method	$\mathrm{AP_C} \uparrow$	$\mathrm{AP_{50}} \uparrow$	$\mathrm{AP_{75}} \uparrow$	$\mathrm{oLRP} \downarrow$	$\mathrm{oLRP_{Loc}} \downarrow$	$\mathrm{oLRP_{FP}} \downarrow$	$\mathrm{oLRP_{FN}} \downarrow$
Retina Net [4]	36.5	55.4	39.1	70.7	16.8	32.0	**48.1**
w. Conc.Corr (Ours)	37.0	**55.7**	39.7	70.2	16.3	**30.8**	49.3
w. Spear.Corr (Ours)	**37.5**	55.4	**40.5**	**69.7**	**16.0**	31.3	48.4
Fovea Box [24]	36.4	56.5	38.6	70.2	17.0	30.2	**47.2**
w. Conc.Corr (Ours)	**37.1**	**56.4**	**39.6**	**69.7**	16.6	**28.6**	48.1
w. Spear.Corr (Ours)	37.0	55.6	39.3	70.0	**16.3**	31.0	47.9
ATSS [1]	38.7	57.6	41.5	69.0	16.0	**29.1**	47.0
w. Conc.Corr (Ours)	**39.8**	**57.9**	**43.2**	**68.2**	15.4	**29.1**	46.9
w. Spear.Corr (Ours)	39.3	56.6	42.5	68.7	**15.2**	31.2	**46.7**
PAA [2]	39.9	57.3	43.4	68.6	15.0	30.4	47.0
w. Conc.Corr (Ours)	**40.7**	**58.8**	**44.3**	**67.7**	15.2	**28.5**	**46.3**
w. Spear.Corr (Ours)	40.4	58.0	43.7	67.8	**14.9**	29.5	46.6
Sparse R-CNN [5]	37.7	55.8	40.5	69.5	16.0	28.7	48.6
w. Conc.Corr (Ours)	38.9	**57.2**	41.8	**68.1**	15.7	27.7	**47.2**
w. Spear.Corr (Ours)	**39.3**	56.7	**42.5**	68.3	**15.3**	**27.1**	48.4

5.2.2 NMS-free Detectors

Our results in Table 5.1 suggest that Sparse R-CNN, an NMS-free method, can also benefit from our Correlation Loss:

- Both Concordance ($\lambda_{corr} = 0.3$) and Spearman Losses ($\lambda_{corr} = 0.2$) improve baseline performance,
- Spearman Loss improves $\mathrm{AP_C}$ significantly by up to 1.6,
- As hypothesized, the gains are owing to APs with larger IoUs, e.g., $\mathrm{AP_{75}}$ improves by up to 2.0,

Table 5.2: Comparison with a stronger setting of Sparse R-CNN with 36 epochs training, 300 proposals, multi-scale training and random cropping.

Method	AP	AP_{50}	AP_{75}
Sparse R-CNN [5]	45.0	64.1	48.9
w. Conc.Corr (Ours)	45.5	**64.4**	49.7
w. Spear.Corr (Ours)	**46.1**	64.0	**50.4**

- Gains can be also observed in a stronger setting of Sparse R-CNN with 36 epochs training, 300 proposals, multi-scale training and random cropping following Sun et al. [5] (Table 5.2).

5.2.3 Instance Segmentation

To analyze the generalizability of Correlation Loss, we trained an instance segmentation method, YOLACT, in its standard setting [41] with Correlation Loss. We observed 0.7 mask AP gain using Spearman Loss ($\lambda_{corr} = 0.5$ - Table 5.3), implying 1.7% relative gain.

Table 5.3: Comparison with YOLACT.

Method	AP_C^{mask}	AP_{50}^{mask}	AP_{75}^{mask}
YOLACT [41]	28.3	47.8	28.8
w. Conc.Corr (Ours)	28.8	**48.3**	29.6
w. Spear.Corr (Ours)	**29.0**	48.3	**30.0**

5.3 Comparison with Methods Enforcing Correlation

Table 5.4 compares Correlation Loss with using an auxiliary head [1], QFL [9] and RS Loss [10] on the common ATSS baseline [1] with respect to detection and correlation performances:

5.3.1 Comparison wrt. Detection Performance

Reaching $39.8\mathrm{AP_C}$ without an auxiliary head, Concordance Loss (as validated in Table 5.1 for ATSS) outperforms using an auxiliary head, which introduces additional learnable parameters (39.8 vs $39.3\mathrm{AP_C}$), and reaches on-par performance with the recently proposed, relatively complicated loss functions, QFL [9] and RS Loss [10]. Besides, thanks to its simple usage, Concordance Loss is complementary to the existing methods: It yields $40.0\mathrm{AP_C}$ with an auxiliary head (+0.7 $\mathrm{AP_C}$ gain) and $40.2\mathrm{AP_C}$ with RS Loss (+0.3 $\mathrm{AP_C}$ gain), outperforming all methods without introducing additional learnable parameters.

Table 5.4: Comparison with methods enforcing correlation. Correlation Loss (i) reaches similar results with existing methods on ATSS, (ii) is complementary to those methods thanks to its simple design and (iii) once combined with RS Loss, outperforms compared methods.

Aux. [1]	QFL [9]	RS Loss[10]	Corr. Loss (Ours)	$\mathrm{AP_C}\uparrow$	$\mathrm{AP_{50}}\uparrow$	$\mathrm{AP_{75}}\uparrow$	oLRP $\downarrow$	$\beta_{img}\uparrow$	$\beta_{cls}\uparrow$
				38.7	57.6	41.5	68.9	27.3	40.3
✓				39.3	57.5	42.6	68.6	28.7	42.5
	✓			39.7	58.1	42.7	68.0	33.2	45.7
		✓		39.9	**58.9**	42.6	67.9	30.9	43.2
			✓	39.8	57.6	43.1	68.2	31.6	45.2
✓			✓	40.0	58.0	43.3	68.0	31.1	44.8
	✓		✓	39.9	58.2	43.2	**67.7**	**34.6**	45.6
		✓	✓	**40.2**	58.6	**43.5**	67.9	33.6	**46.1**

5.3.2 Comparison wrt. Correlation Performance

Comparison with respect to correlation (using our proposed performance measures, β_{img} in Equation 3.1 and β_{cls} in Equation 3.2) shows:

- Concordance Loss improves baseline correlation performance significantly, enhancing β_{img} (from 27.3% to 31.6%) and β_{cls} (from 40.3% to 45.2%) both by $\sim 5\%$,

- Concordance Loss outperforms all methods wrt. β_{img} and β_{cls} once combined with QFL and RS Loss respectively.

This set of results confirms that Concordance Loss improves correlation between classification and localization tasks in both image-level and class-level.

5.4 Comparison with SOTA

Here, we prefer Sparse R-CNN owing to its competitive detection performance and our large gains. We train our "Corr-Sparse R-CNN" for 36 epochs using DCNv2 [48] and multiscale training by randomly resizing the shorter size within [480, 960] similar to compared methods, e.g. VFNet [21], GFLv2 [22] and RS Loss [10]. Table 5.5 presents the results on COCO test-dev [40]. We note the following:

5.4.1 NMS-based Methods

On the common ResNet-101-DCN backbone and with similar data augmentation, our Corr-Sparse R-CNN yields $49.6\mathrm{AP_C}$ at 13.7 fps (on a V100 GPU) outperforming recent NMS-based methods, all of which also enforce correlation, e.g.,

- RS-R-CNN [10] by $1.8\mathrm{AP_C}$,
- GFLv2 [22] by more than $1\mathrm{AP_C}$,
- VFNet [21] in terms of not only $\mathrm{AP_C}$ but also efficiency (with 12.6 fps on a V100 GPU).

On ResNeXt-101-DCN, our Corr-Sparse R-CNN provides $51.0\mathrm{AP_C}$ at 6.8 fps, surpassing all methods including RS-Mask R-CNN+ ($50.2\mathrm{AP_C}$ at 6.4 fps), which additionally exploits ground truth masks and Carafe FPN [49].

5.4.2 NMS-free Methods

Our Corr-Sparse R-CNN outperforms;

Table 5.5: Comparison with SOTA on COCO *test-dev*. Without bells and whistles, our Corr-Sparse R-CNN outperforms all recent (i) NMS-based methods, all of which also enforce correlation, and (ii) NMS-free methods by a notable margin ($\sim 1\mathrm{AP_C}$ compared to its nearest counterparts). Results are obtained from papers.

	Method	Backbone	Epochs	$\mathrm{AP_C}$	$\mathrm{AP_{50}}$	$\mathrm{AP_{75}}$	$\mathrm{AP_S}$	$\mathrm{AP_M}$	$\mathrm{AP_L}$	Venue
NMS-based	ATSS [1]	ResNet-101-DCN	24	46.3	64.7	50.4	27.7	49.8	58.4	CVPR 2020
	GFL [9]	ResNet-101-DCN	24	47.3	66.3	51.4	28.0	51.1	59.2	NeurIPS 2020
	RS-R-CNN [10]	ResNet-101-DCN	36	47.8	68.0	51.8	28.5	51.1	61.6	ICCV 2021
	GFLv2 [22]	ResNet-101-DCN	24	48.3	66.5	52.8	28.8	51.9	60.7	CVPR 2021
	aLRP Loss [8]	ResNeXt-101-DCN	100	48.9	69.3	52.5	30.8	51.5	62.1	NeurIPS 2020
	VFNet [21]	ResNet-101-DCN	24	49.2	67.5	53.7	29.7	52.6	62.4	CVPR 2021
	DW [51]	ResNet-101-DCN	24	49.3	67.6	53.3	29.2	52.2	63.5	CVPR 2022
	TOOD [37]	ResNet-101-DCN	24	49.6	67.4	54.1	30.5	52.7	62.4	ICCV 2021
	RS-Mask R-CNN+ [10]	ResNeXt-101-DCN	36	50.2	**70.3**	54.8	**31.5**	53.5	63.9	ICCV 2021
NMS-free	TSP R-CNN [50]	ResNet-101-DCN	96	47.4	66.7	51.9	29.0	49.7	59.1	ICCV 2021
	Sparse R-CNN [5]	ResNeXt-101-DCN	36	48.9	68.3	53.4	29.9	50.9	62.4	CVPR 2021
	Dynamic DETR [31]	ResNeXt-101-DCN	36	49.3	68.4	53.6	30.3	51.6	62.5	ICCV 2021
	Deformable DETR [34]	ResNeXt-101-DCN	50	50.1	69.7	54.6	30.6	52.8	64.7	ICLR 2021
Ours	Corr-Sparse R-CNN	ResNet-101-DCN	36	49.6	67.8	54.1	29.2	52.3	64.9	
	Corr-Sparse R-CNN	ResNeXt-101-DCN	36	**51.0**	69.2	**55.7**	31.1	**53.7**	**66.3**	

- TSP R-CNN [50] by more than $2\mathrm{AP_C}$ on ResNet-101-DCN with significantly less training,
- Sparse R-CNN [5] by $\sim 2\mathrm{AP_C}$
- Deformable DETR [34], a recent strong NMS-free method, by $\sim 1\mathrm{AP_C}$ on ResNeXt-101-DCN.

5.5 Ablation & Hyper-parameter Analyses

5.5.1 Optimizing Different Correlation Coefficients

Spearman Loss yields better localization performance , i.e. the lowest localization error with respect to $\mathrm{oLRP_{Loc}}$ in all experiments (Table 5.1) while it rarely yields

the best $\mathrm{oLRP}_{\mathrm{FP}}$ or $\mathrm{oLRP}_{\mathrm{FN}}$, implying its contribution to classification to be weaker than Concordance Loss.

We tried optimizing Pearson correlation coefficient as well and observed that while it has similar performance with concordance correlation coefficient on ATSS and Spearman correlation coefficient on Sparse R-CNN, it does not outperform the other two in both of the cases (Table 5.6). Considering the similarities of Spearman and concordance correlation coefficients in terms of scoring the relation of the values, we preferred concordance correlation coefficient over Spearman correlation coefficient due to the fact that concordance correlation coefficient enforces the scores to be equal to the IoUs imposing a tighter constraint than pearson correlation coefficient.

Table 5.6: Effect of using Pearson correlation coefficient.

Method	AP_{C}	AP_{50}	AP_{75}
ATSS w/o aux head	38.7	57.6	41.5
w. Pearson Corr	39.4	56.6	42.7
w. Conc.Corr	**39.8**	**57.9**	**43.2**
w. Spear.Corr	39.3	56.6	42.5
Sparse R-CNN	37.7	55.9	40.5
w. Pearson Corr	**39.3**	56.6	42.2
w. Conc.Corr	38.9	**57.2**	41.8
w. Spear.Corr	**39.3**	56.7	**42.5**

5.5.2 Sensitivity to λ_{corr}

We found it generally sufficient to search over $\{0.1, 0.2, 0.3.0.4, 0.5\}$ to tune λ_{corr} for all the models we trained.

In Table 5.7, we see that (i) $\lambda_{corr} = 0.2$ provides the best performance overall, (ii) the performance is not very sensitive to λ_{corr} and (iii) a grid search over $\{0.1, 0.2, 0.3.0.4, 0.5\}$ is sufficient (outside of this range, performance drops).

Table 5.7: Grid search to tune λ_{corr} on different models.

λ_{corr}	0.0	0.1	0.2	0.3	0.4	0.5
ATSS w/o aux. head + Concordance Loss	38.7	38.8	39.3	**39.8**	39.7	39.7
ATSS w. RS Loss + Concordance Loss	39.9	39.8	**40.2**	40.1	39.8	39.5
Sparse R-CNN + Spearman Loss	37.7	38.7	**39.3**	39.1	39.0	38.1
YOLACT + Concordance Loss	28.3	28.6	28.8	28.8	**29.0**	28.8

5.5.3 Effect on Training Time

The training speed is measure with seconds/iteration. The lower, the better. Using Spearman or Correlation Loss to train Sparse R-CNN, computing the loss for 6 times each iteration, increases iteration time 0.50 seconds to 0.51 seconds on V100 GPUs, suggesting a negligible additional overhead.

5.5.4 Effect on Backpropagating Different Heads

On Sparse R-CNN, we observed that the performance degrades when we backpropagate either only localization head (37.5 AP) or both heads (38.9 AP). Hence, we preferred backpropagating the gradients only through the classification head (39.3 AP).

5.5.5 Effect on Different Classes

Our Correlation Loss on Sparse R-CNN improves 64 of the 80 classes, implying that its effect is not limited to a small subset of the classes (See Table 5.8).

Table 5.8: Effect on Classes. Correlation Loss on Sparse R-CNN improves 64 of the 80 classes.

Class Name	mAP (Sparse R-CNN)	mAP (Corr Sparse R-CNN)
person	49.7	51.3
motorcycle	38.6	39.5
train	61.4	61.1
traffic light	22.4	23.0
parking meter	45.0	45.6
cat	68.3	72.3
sheep	46.7	46.1
bear	72.4	75.3
backpack	13.3	14.9
tie	27.0	28.5
skis	21.4	21.8
kite	37.1	39.4
skateboard	50.1	49.2
bottle	30.2	33.7
fork	29.6	28.3
bowl	36.9	38.6
sandwich	34.2	35.3
carrot	16.5	20.3
donut	40.2	40.8
couch	42.7	41.8
dining table	27.2	28.4
laptop	54.9	56.9
keyboard	46.4	44.6
oven	30.9	32.7
refrigerator	52.6	54.8
vase	33.4	33.0
hair drier	16.1	2.4

Continued on next page

Table 5.8 (continued)

Class Name	mAP (Sparse R-CNN)	mAP (Corr Sparse R-CNN)
bicycle	25.1	25.6
airplane	65.8	66.4
truck	32.4	33.1
fire hydrant	63.2	62.4
bench	22.1	21.5
dog	64.3	64.3
cow	54.6	56.5
zebra	64.4	66.2
umbrella	31.1	34.2
suitcase	30.3	35.0
snowboard	33.9	34.8
baseball bat	24.4	28.7
surfboard	30.6	37.0
wine glass	28.5	29.1
knife	11.5	14.2
banana	18.5	19.2
orange	26.7	30.5
hot dog	29.2	34.0
cake	29.7	31.8
potted plant	21.6	22.7
toilet	56.9	58.4
mouse	55.8	58.7
cell phone	30.0	33.0
toaster	22.1	37.7
book	8.6	9.9
scissors	31.2	29.1
toothbrush	15.4	23.9
car	37.2	39.1
bus	61.7	62.9

Continued on next page

Table 5.8 (continued)

Class Name	mAP (Sparse R-CNN)	mAP (Corr Sparse R-CNN)
boat	21.5	23.4
stop sign	64.0	63.1
bird	32.1	33.4
horse	55.1	58.6
elephant	63.8	64.9
giraffe	64.7	64.7
handbag	10.2	12.3
frisbee	64.2	65.1
sports ball	45.7	46.2
baseball glove	34.7	35.5
tennis racket	44.5	46.1
cup	35.8	38.2
spoon	12.0	11.7
apple	14.7	18.1
broccoli	18.2	20.3
pizza	51.0	50.2
chair	20.5	22.1
bed	45.4	48.6
tv	53.2	54.8
remote	22.1	25.5
microwave	48.2	54.1
sink	30.1	33.7
clock	48.4	51.7
teddy bear	43.2	44.7

CHAPTER 6

CONCLUSION

6.1 Summary

In this thesis, we studied deep object detectors from the perspective of correlation between the classification and localization branches. For this end, we first provided an analysis on the importance and the effect of correlation on different stages of an object detector. Then, motivated by the results of this analysis, we introduced a novel plug-in loss function for improving the correlation between the classification and localization branches.

As our first contribution, we analysed the effects of correlation between classification and localization tasks in object detectors. With this analysis, we identified why correlation affects the performance of various NMS-based and NMS-free detectors. Furthermore, we devised performance measures to evaluate the effect of correlation and use them to analyze common detectors.

As our second contribution, motivated by our observations from the analysis, we proposed Correlation Loss, a novel plug-in loss function that improves the performance of various object detectors by directly optimizing correlation coefficients. Our extensive experiments on six detectors show that Correlation Loss consistently improves the detection and correlation performances and reaches state of the art results.

6.2 Limitations and Future Work

In this work, while we emphasized the importance of correlation for detection performance, Observation (4) (Section 3.2.2) suggests that the localization and the clas-

sification performances should be preserved while optimizing correlation; hence correlation may not be the sole objective of the training. On a related note, following prior work [9, 10], we supervised only the classifier considering the localization performance (and not the other way around), which may be a limitation since correlation is not enforced on both tasks.

REFERENCES

[1] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[2] K. Kim and H. S. Lee, "Probabilistic anchor assignment with iou prediction for object detection," in *The European Conference on Computer Vision (ECCV)*, 2020.

[3] K. Chen, W. Lin, J. li, J. See, J. Wang, and J. Zou, "Ap-loss for accurate one-stage object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 1–1, 2020.

[4] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 42, no. 2, pp. 318–327, 2020.

[5] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, and P. Luo, "SparseR-CNN: End-to-end object detection with learnable proposals," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[7] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *The European Conference on Computer Vision (ECCV)*, 2018.

[8] K. Oksuz, B. C. Cam, E. Akbas, and S. Kalkan, "A ranking-based, balanced

loss function unifying classification and localisation in object detection," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[9] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[10] K. Oksuz, B. C. Cam, E. Akbas, and S. Kalkan, "Rank & sort loss for object detection and instance segmentation," in *The International Conference on Computer Vision (ICCV)*, 2021.

[11] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] R. Girshick, "Fast R-CNN," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[13] 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 (TPAMI)*, vol. 39, no. 6, pp. 1137–1149, 2017.

[14] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[15] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[18] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *The European Conference on Computer Vision (ECCV)*, 2016.

[20] J. Liu, D. Li, R. Zheng, L. Tian, and Y. Shan, "Rankdetnet: Delving into ranking constraints for object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 264–273, June 2021.

[21] H. Zhang, Y. Wang, F. Dayoub, and N. Sünderhauf, "Varifocalnet: An iou-aware dense object detector," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[22] X. Li, W. Wang, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss v2: Learning reliable localization quality estimation for dense object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[23] J. Choi, D. Chun, H. Kim, and H.-J. Lee, "Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[24] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi, "Foveabox: Beyound anchor-based object detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 7389–7398, 2020.

[25] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *The European Conference on Computer Vision (ECCV)*, 2018.

[26] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Generating positive bounding boxes for balanced training of object detectors," in *IEEE Winter Applications on Computer Vision (WACV)*, 2020.

[27] K. Oksuz, B. C. Cam, F. Kahraman, Z. S. Baltaci, S. Kalkan, and E. Akbas, "Mask-aware iou for anchor assignment in real-time instance segmentation," in *The British Machine Vision Conference (BMVC)*, 2021.

[28] X. Zhang, F. Wan, C. Liu, R. Ji, and Q. Ye, "Freeanchor: Learning to match anchors for visual object detection," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[29] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "Imbalance problems in object detection: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 1–1, 2020.

[30] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision (ECCV)*, 2020.

[31] X. Dai, Y. Chen, J. Yang, P. Zhang, L. Yuan, and L. Zhang, "Dynamic detr: End-to-end object detection with dynamic attention," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[32] B. Roh, J. Shin, W. Shin, and S. Kim, "Sparse detr: Efficient end-to-end object detection with learnable sparsity," in *The International Conference on Learning Representations (ICLR)*, 2022.

[33] P. Sun, Y. Jiang, E. Xie, W. Shao, Z. Yuan, C. Wang, and P. Luo, "What makes for end-to-end object detection?," in *International Conference on Machine Learning (ICML)*, 2021.

[34] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable {detr}: Deformable transformers for end-to-end object detection," in *International Conference on Learning Representations (ICLR)*, 2021.

[35] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[36] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[37] C. Feng, Y. Zhong, Y. Gao, M. R. Scott, and W. Huang, "Tood: Task-aligned one-stage object detection," in *The International Conference on Computer Vision (ICCV)*, 2021.

[38] L. I. Lin, "A concordance correlation coefficient to evaluate reproducibility," *Biometrics*, vol. 45, pp. 255–268, 1989.

[39] M. Blondel, O. Teboul, Q. Berthet, and J. Djolonga, "Fast differentiable sorting and ranking," in *International Conference on Machine Learning (ICML)*, 2020.

[40] 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 *The European Conference on Computer Vision (ECCV)*, 2014.

[41] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[42] K. Oksuz, B. C. Cam, E. Akbas, and S. Kalkan, "Localization recall precision (LRP): A new performance metric for object detection," in *The European Conference on Computer Vision (ECCV)*, 2018.

[43] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, "One metric to measure them all: Localisation recall precision (lrp) for evaluating visual detection tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[44] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv*, vol. 1906.07155, 2019.

[45] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[47] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito,

M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[48] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[49] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, "Carafe: Content-aware reassembly of features," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[50] Z. Sun, S. Cao, Y. Yang, and K. M. Kitani, "Rethinking transformer-based set prediction for object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[51] S. Li, C. He, R. Li, and L. Zhang, "A dual weighting label assignment scheme for object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.