UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

LUIS FELIPE DE ARAUJO ZENI

Enhancing Weakly Supervised Object Detection via Long-Term Self-Supervised Instance Mining

Thesis presented in partial fulfillment of the requirements for the degree of Doctor of Computer Science

Advisor: Prof. Dr. Claudio Rosito Jung

Porto Alegre
July 2025

CIP — CATALOGING-IN-PUBLICATION

de Araujo Zeni, Luis Felipe

Enhancing Weakly Supervised Object Detection via Long-Term Self-Supervised Instance Mining / Luis Felipe de Araujo Zeni. – Porto Alegre: PPGC da UFRGS, 2025.

113 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2025. Advisor: Claudio Rosito Jung.

1. Weakly supervised object detection. 2. Self-supervised instance mining. 3. Computer vision. 4. Machine learning. I. Jung, Claudio Rosito. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitora: Prof[a]. Marcia Barbosa
Vice-Reitor: Prof. Pedro Costa
Pró-Reitora de Pós-Graduação: Prof[a]. Claudia Wasserman
Diretor do Instituto de Informática: Prof. Luciano Paschoal Gaspary
Coordenador do PPGC: Prof. Gabriel Luca Nazar
Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro

“Life’s tragedy is that we get old too soon and wise too late.”

— BENJAMIN FRANKLIN

ACKNOWLEDGMENTS

A person played a crucial role in the success of my conclusion. He helped me transform my broad concepts into a structured dissertation plan, organised my thoughts, inspired me with insightful ideas and directions for the research, motivated me to persevere, and helped me in writing and improving my articles, among other things. I feel fortunate and proud to have had him by my side, and I am confident that he was the ideal mentor for this challenging path. I express my sincere gratitude to him for his guidance, tremendous patience, and invaluable insights. Professor Claudio, thank you very much for being my supervisor and for helping me make this a reality.

Above all, my wife deserves special thanks for her unwavering support, for her great love, for all the late nights and early mornings, for keeping me sane, and for being the source of my strength during both hardships and triumphs. She gave me two beautiful sons and the opportunity to be the father of a fantastic family that I am proud to be a part of. Aline, thanks for everything. My love for you is immense and boundless!

Parenthood has profoundly changed my perspective on my parents, and I am sincerely grateful for their love, wisdom, and for nurturing my curiosity about computers since I was a child. I also owe a debt of gratitude to my grandmother Reomaria, who significantly influenced my development. I extend my appreciation to my children, who share their unconditional love with me and inspire me to strive for improvement every day. Also, many thanks to my long-time friend Yuri, who has been a steadfast companion throughout this life journey.

I also extend my gratitude to my lab colleagues, particularly Thiago, Sergio, and Adriano, for their early and essential support in my research. I want to thank the Brazilian citizens who founded my scholarship and the university. Finally, I would like to thank all the professors, co-authors, relatives, colleagues, and friends who supported me throughout my Ph.D.

ABSTRACT

Object detection is a fundamental computer vision task that aims to localise and recognise objects within images. However, fully supervised object detection (FSOD) methods require expensive bounding box annotations, which can be time-consuming to compile for large-scale datasets. Weakly Supervised Object Detection (WSOD) presents a promising alternative by utilising only image-level category labels as supervision, thereby significantly reducing annotation costs. However, it faces two significant challenges: handling multiple instances of the same class within an image and achieving precise bounding box localisation. This dissertation addresses these challenges through three complementary approaches. First, we propose a knowledge distillation methodology combined with an adaptive refinement supervision mechanism that dynamically adjusts IoU selection thresholds and leverages knowledge from trained refiners to mitigate the harmful effects of multiple instances during training. Second, we explore domain-specific applications where instance-counting information is inherently available in weak annotations. Specifically, we develop a weakly supervised approach for license plate character detection through an Instance-Aware Self-Supervision (IASS) building process and a Number of Characters Classification (NCC) module that exploits the counting information present in license plate strings. Third, we present a novel self-supervised mechanism for discovering additional instances of the same class during training, employing a Multi-Layer Perceptron (MLP) projection network optimised with discriminative loss to maintain intra-class consistency while ensuring inter-class separation. The supervision for the MLP is made with high-quality samples from a memory bank that stores high-scoring proposals from previous iterations. We also propose an overlap suppression strategy to eliminate redundant detections. Extensive experiments on standard benchmarks (PASCAL VOC 2007, VOC 2012, and MS COCO) demonstrate the effectiveness of our approaches, achieving state-of-the-art performance with first-place rankings on VOC datasets and second-place on COCO. The dissertation contributes to the maturity of weakly supervised object detection by providing practical solutions that maintain high accuracy while substantially reducing human supervision requirements, advancing the field toward more cost-effective and scalable object detection systems.

Keywords: Weakly supervised object detection. Self-supervised instance mining. Computer vision. Machine learning.

Aprimorando a detecção de objetos fracamente supervisionada por meio de mineração de instâncias autosupervisionadas de longo prazo

RESUMO

A detecção de objetos é uma tarefa fundamental de visão computacional que visa localizar e reconhecer objetos em imagens. No entanto, métodos de detecção de objetos totalmente supervisionados exigem anotações de bounding boxes, cuja compilação pode ser demorada para conjuntos de dados de grande escala. A Detecção de Objetos Fracamente Supervisionada apresenta uma alternativa promissora ao utilizar apenas rótulos de categoria em nível de imagem como supervisão, reduzindo significativamente os custos de anotação. No entanto, ela enfrenta dois desafios significativos: lidar com múltiplas instâncias da mesma classe em uma imagem e obter uma localização precisa das bounding boxes. Esta tese aborda esses desafios por meio de três abordagens complementares. Primeiro, propomos uma metodologia de destilação de conhecimento combinada com um mecanismo de supervisão de refinamento adaptativo que ajusta dinamicamente os limiares de seleção de IoU e aproveita o conhecimento dos refinadores sendo treinados para mitigar os efeitos nocivos de múltiplas instâncias durante o treinamento. Segundo, exploramos uma aplicação específica de domínio onde a informação de número de instâncias está inerentemente presente nas anotações fracas. Especificamente, desenvolvemos uma abordagem fracamente supervisionada para detecção de caracteres de placas de veículos por meio de um processo de construção de auto-supervisão com uso do número de Instâncias e um módulo de Classificação de Número de Caracteres Presentes. Em terceiro lugar, apresentamos um novo mecanismo auto-supervisionado para descobrir instâncias adicionais da mesma classe durante o treinamento, empregando uma rede de projeção Multi-Layer Perceptron (MLP) otimizada com uma loss discriminativa para manter a consistência intra-class e garantindo a separação interclasse. A supervisão para o MLP é feita com amostras de alta qualidade de um banco de memória que armazena propostas de alta pontuação de iterações anteriores. Também propomos uma estratégia de supressão baseada na sobreposição para eliminar detecções redundantes. Experimentos extensivos em benchmarks padrão (PASCAL VOC 2007, VOC 2012 e MS COCO) demonstram a eficácia de nossas abordagens, alcançando desempenho de ponta com classificações de primeiro lugar em conjuntos de dados VOC e segundo lugar em COCO. A dissertação contribui para a maturidade da detecção de objetos fracamente supervisionados ao fornecer soluções práticas

que mantêm alta precisão ao mesmo tempo em que reduzem substancialmente os requisitos de supervisão humana, avançando o campo em direção a sistemas de detecção de objetos mais econômicos e escaláveis.

Palavras-chave: Detecção de objetos com supervisão fraca. Mineração de instancias com supervisão própria. Visão Computacional. Apredizado de maquina.

LIST OF FIGURES

Figure 1.1 Two examples of computer vision applications that could be built using an 'on the shelf' object detector. 20

Figure 1.2 Discrepancy in the number of operations required to annotate an image with bounding boxes versus with image-level category annotations. The bounding box action count is estimated based on the assumption that annotators execute a minimum of 3 error-free actions per box: two for the top-left and bottom-right spatial coordinates, and one for object category assignment. 22

Figure 1.3 The procedure for creating the self-supervision of the Online Instance Classification Refinement (OICR)'s refiner module involves selecting the highest-ranking proposal for each target class from the scoring matrix of the earlier module (dashed red). Proposals with a high IoU (solid red) are designated as positives for that class, while the remaining proposals are classified as background (yellow). 23

Figure 1.4 Illustration of typical inaccuracies in the Weakly Supervised Object Detection (WSOD) detection outcomes. The red bounding box represents a detection that encompasses multiple instances of the same category. The blue box indicates a bounding box that captures only the most descriptive part of the object, whereas the green box demonstrates a detection with inadequate bounding box alignment. Additionally, the white cat on the left was overlooked. 24

Figure 2.1 Overall VGG16 architecture. 31

Figure 2.2 Overall architecture of a Multiple Instance Learning Network (MIL)-OICR-based WSOD method. The Proposals Feature Extraction component identifies features from individual proposals by analysing their specific regions in the input image. Multiple Instance Learning (MIL) identifies the highest-scoring proposals that best represent the existing classes, guided by the classification ground truth. The refinement phase incorporates a self-supervised training strategy into WSOD, utilising instance-level supervision to enhance the overall results. 33

Figure 2.3 The update of the OICR refiner module with the bounding box regression mechanism. 37

Figure 3.1 Evolution of the detection mAP of WSOD methods over the years on dataset Pascal Visual Object Classes (VOC) 2007 test set 43

Figure 4.1 The proposed architecture and its modules. The proposals feature extraction module uses an Spatial Pyramidal Pooling (SPP) layer to extract features from proposals generated by the selective search. The multiple instance detection network module learns to select the best proposal instance and generates an image classification score. The instance refinement modules have k instances, and each one learns to refine instances from its predecessor's result. Finally, the knowledge distillation module aggregates all the knowledge learned by all the K refinements. 53

Figure 4.2 A visual comparison of the building process of a refiner's self-supervision to the "cat" class, when (a) using the standard OICR supervision scheme and (b) including the "trick" to ignore the "background" proposals with low intersection in the loss function. The red dashed box is the selected top-scoring instance that guides self-supervision. The solid red boxes are those selected that also belong to the cat class, the yellow rectangles are proposals associated with background, and the blue ones present boxes that are set to be ignored in the loss function by the "trick" in the self-supervision. 55

Figure 4.3 Effect of changing the Intersection over Union (IoU) threshold λ for instance selection. Red slashed boxes denote the top-scoring proposal, red boxes pass the threshold (selected), and yellow boxes are set to background (not selected). 56

Figure 4.4 A visual interpretation of the proposed adaptive supervision aggregation function. X-axis shows the iteration step number, and Y-axis shows the IoU with the box of the highest score. 58

Figure 4.5 Detection examples for the VOC 2007 dataset. Blue rectangles are ground-truth boxes that have at least one detection with $IoU > 0$, and yellow ones are ground-truth boxes with no detection intersection. Green boxes are correct detections ($IoU > 0.5$ with ground truth), and red boxes are wrong detections. The label in each detection box is the class label and confidence score of the detection. 62

Figure 5.1 An overview of the proposed architecture and its modules. The localisation and rectification module localises the corners of the Licence Plate (LP). These points are used to apply an affine transformation, which reduces the perspective deformations of the plate in the image. The rectified image is fed into the proposal feature extraction module, which utilises an SPP layer to extract features from proposals generated by the Selective Search algorithm. The image features extracted using the Convolutional Neural Networks (CNN) backbone are used by the number of characters classification module to estimate the number of characters present in the plate. The multiple instance detection network module learns to select the best proposal instances and generates an image classification score. The refinement modules are cascaded into a sequence of K module instances, where each one learns to refine the results using its predecessor's scoring matrix to build its self-supervision process. Finally, the knowledge distillation module combines the knowledge learned by all the refiners to enhance the results. 67

Figure 5.2 Comparison of the results of different strategies that build the refiner's self-supervision for the character class "1". Green denotes the top-scoring box(es), blue boxes are selected as also belonging to the class "1", red boxes are the background ones. Yellow boxes are set to be ignored in the loss function... 68

Figure 5.3 Some examples of the character detected in different plate layouts. Green rectangles represent proposals that have the correct character detected, and red rectangles are proposals with the wrong character. 76

Figure 6.1 Our contributions to improve each category self-supervision for OICR's refinement module. (a) MIL-OICR based techniques use the top-scoring proposal of the previous module (dashed red) to select proposals with large IoU (solid red) as positive, and all other proposals are set to the background class (yellow), including other instances of the same class. (b) Our instance discovery module is able to discover the other instances of the class and include them as positive in the creation of the supervision, but it might create more than one proposal per instance. (c) An overlap suppression strategy is used to reduce the number of positive candidates while keeping only the most prominent ones. 79
Figure 6.2 The complete high-level architecture of Long-Term Instance Mining (LTIM). Our contributions reside in the OICR refiners, where a memory bank stores past top-scoring proposals utilised to train a MLP that project features into a class-clustered embedding space. These projected features are utilized to perform the discovery of additional object instances to enhance the self-supervision building process. More details regarding each part are provided in Section 6.1.....82
Figure 6.3 The instance memory bank stores previous iterations' high-scoring proposal features. In each training iteration, it stores new top-scoring instances and removes older stored instances if the bank for a given category is full. 83
Figure 6.4 Illustration of the training procedure of each MLP^k. The features stored in the bank are recovered and fed into $\mathbf{FC}(.)$ to convert them to proposal-level features. The proposal-level features are then fed into an MLP^k for embedding generation. The projected embeddings are utilised in the clustering contrastive loss to optimise MLP^k's weights, and to calculate the next iteration embedding centroids. 84
Figure 6.5 Our method of instance mining employs embeddings generated by a MLPin conjunction with class embedding centroids to uncover new instances for each class.....86
Figure 6.6 Effects of setting the memory bank length in the final mAP of Pascal VOC 2007 test set and enabling the proposal overlap suppression technique.....90
Figure 6.7 Visualisation of the bounding boxes identified by the proposed approach in the VOC 2007 dataset. The upper row demonstrates high quality results. The second row presents results of medium quality, characterized by some omissions or incompletely detected bounding boxes. The bottom row illustrates poor quality results, with redundant bounding boxes for the same object instances or incorrect classifications.....92

LIST OF TABLES

Table 2.1 Statistics regarding the relevant datasets for training WSOD models. 38

Table 4.1 Ablation study performance (%) on the VOC 2007 in different scenarios. 61
Table 4.2 Detection performance (%) on the VOC 2007 test set. Comparison with state-of-the-art approaches. 63
Table 4.3 Localisation (Correct Localization (CorLoc)) performance (%) on the VOC 2007 trainval set. Comparison with state-of-the-art approaches. 63
Table 4.4 Detection in test set and localisation (CorLoc) in trainval set performance (%) on the VOC 2012 dataset using VGG16. 64

Table 5.1 Ablation study: Performance on the test-set of training datasets. Error 0 requires an exact match between the predicted and ground-truth strings, whereas Error 1 permits a single-character deviation. 74
Table 5.2 Ablation study: Performance on unseen datasets. Error 0 requires an exact match between the predicted and ground-truth strings, whereas Error 1 permits a single-character deviation. 74
Table 5.3 Accuracy of the proposed number of characters classification module (Number of Characters Classification (NCC)).....75
Table 5.4 Comparison with other state-of-the-art methods. † means that the method was trained on the dataset. 77

Table 6.1 Hyper-parameters used in the experiments 89
Table 6.2 Comparison of mean average precision on PASCAL VOC 2007 test set. 90
Table 6.3 Comparison of mean average precision on PASCAL VOC 2012 test set. 91
Table 6.4 Comparison of CorLoc on VOC 2007 trainval set. 91
Table 6.5 Comparison of CorLoc on VOC 2012 trainval set. 92
Table 6.6 Comparison of the state-of-the-art methods on COCO 2014 val set. 93

LIST OF ABBREVIATIONS AND ACRONYMS

AP Average Precision

BOICR Boosted Online Instance Classification Refinement

C-MIL Continuation Multiple Instance Learning

CAM Class Activation Map

CNN Convolutional Neural Networks

CorLoc Correct Localization

FC Fully Connected

FRCNN Fast R-CNN

FSOD Fully Supervised Object Detector

GAP Global Average Pooling

IoU Intersection over Union

LP Licence Plate

LPD Licence Plate Detector

LPR Licence Plate Recognition

LSTM Long Short-term Memory Network

MIL Multiple Instance Learning Network

MIDN Multiple Instance Detection Network

MLP Multi-Layer Perceptron

mAP Mean Average Precision

NMS Non-maxima Suppression

NN Neural Network

OICR Online Instance Classification Refinement

ReLU Rectified Linear Unit

RoI Region of Interest

RoIPool Region of Interest Pooling

SGD Stochastic Gradient Descent

SLV Spatial Likelihood Voting

SOTA State-of-the-Art

SPP Spatial Pyramidal Pooling

WSCD Weakly Supervised Character Detection

WSOD Weakly Supervised Object Detection

VOC Pascal Visual Object Classes

COCO Common Objects in Context

LIVIS Long Tail Instance Segmentation

OID Open Images Dataset

LTIM Long-Term Instance Mining

IASS Instance-Aware Self-Supervision

NCC Number of Characters Classification

POS Proposal Overlap Suppression

CASD Comprehensive Attention Self-Distillation

SLE Spatial Local Enhancement

LLD Loss-based Label Division

SWA Score-guided Weight Adjustment

PSOD Point Supervised Object Detection

LIST OF SYMBOLS

D_1	Dimension of...
$\boldsymbol{I}$	An input image.
$\mathcal{P}$	A collection of object proposals in bounding box format.
y	Category annotations in one-hot-encoding vector format.
C	The total number of object categories to be learnt.
$\boldsymbol{\Upsilon}$	Extracted image-level features from VGG16 backbone.
$\boldsymbol{\psi}$	Represent the features $\boldsymbol{\Upsilon}$ after being sampled with the RoI pooling.
$\boldsymbol{f}(\boldsymbol{I}, \mathcal{P})$	Function that represents the process of converting I and $\mathcal{P}$ into $\boldsymbol{\psi}$ with the backbone and RoI pooling.
$\boldsymbol{\phi}$	Proposal-level features produced by the *Proposals Feature Extraction* module.
$\mathbf{FC}(.)$	Represents a shallow MLP formed by two fully connected layers that transform $\boldsymbol{\psi}$ into a new dimensional space $\boldsymbol{\phi}$ as output.
D_0	Dimension of $\boldsymbol{\psi}$.
D_1	Dimension of $\boldsymbol{\phi}$.
P	Number of proposals bounding boxes in $\mathcal{P}$.
$\boldsymbol{\theta}_{cls}$	Classification scores produced by the classification branch of MIL module.
$\boldsymbol{\theta}_{det}$	Detection scores produced by the detection branch of MIL module.
$\mathbf{FC}_{cls}(.)$	Function that represents the Fully Connected (FC) layer that converts the features $\boldsymbol{\phi}$ into $\boldsymbol{\theta}_{cls}$ into $\boldsymbol{\theta}_{cls}$ produced by the MIL module
$\mathbf{FC}_{det}(.)$	Function that represents the FC layer that converts the features $\boldsymbol{\phi}$ into $\boldsymbol{\theta}_{det}$ produced by the MIL module.
$\boldsymbol{\theta}^0$	Represents the scoring matrix produced by the MIL module, obtained by the element-wise multiplication of $\boldsymbol{\theta}_{cls}$ with $\boldsymbol{\theta}_{det}$.
$\hat{y}_c$	Per-category image score of MIL module, obtained by summing all the proposal scores for each category. Defined at Equation (2.6).

$\mathcal{L}_{mil}$ The multi-label classification loss employed by the MIL module. Defined at Equation (2.7).

$\boldsymbol{\theta}^k$ Instance classification matrix produced by the k-th refiner in the model.

$\mathbf{FC}^k_{cls}(.)$ Function that represents the FC layer that converts the features $\boldsymbol{\phi}$ into $\boldsymbol{\theta}^k$ in the k-th refiner module.

$\bar{ts}^k_c$ Top-scoring proposals selected from $\boldsymbol{\theta}^{k-1}$ employed by the k-th refiner module to build its self-supervision. Defined at Equation (2.8).

$y^k_{i,c}$ Pseudo-labels produced by the self-supervision building process. Defined at Equation (2.9).

λ represents the IoU threshold utilised in the self-supervision building process.

$\mathcal{L}^k_{cls}$ The instance classification loss of the k-th refiner module. Defined at Equation (2.10).

$w^k_{i,c}$ weight regularizer utilised in $\mathcal{L}^k_{cls}$.

$\breve{\boldsymbol{\theta}}^k_{i,c}$ Represents the regressed bounding boxes coordinates in the k-th refiner module.

$\mathbf{FC}_{reg}(.)$ Function that represents the FC layer that converts the features $\boldsymbol{\phi}$ into $\breve{\boldsymbol{\theta}}^k_{i,c}$ in the k-th refiner module.

$\mathcal{L}^k_{reg}$ The bounding box regression loss of the k-th refiner module. Defined at Equation (2.11).

$\boldsymbol{\theta}^d$ Instance classification matrix produced by the knowledge distillation module.

$\mathbf{FC}^d_{cls}(.)$ Function that represents the FC layer that converts the features $\boldsymbol{\phi}$ into $\boldsymbol{\theta}^d$ in the knowledge distillation module.

$\hat{\boldsymbol{\theta}}^d_{sup}$ Scoring matrix produced by averaging the scoring matrix of the refiners in the model, which is utilised by the knowledge distillation module to build its self-supervision. Defined at Equation (4.1).

$\mathcal{L}_{distil}$ The instance classification loss of the knowledge distillation module. Defined at Equation (4.2).

S Total of training steps.

l_b Parameter that defines the velocity at which the curve grows in the adaptive aggregation function. Utilised at Equation (4.3).

λ_{max} defines the starting point of the adaptive "trick". Utilised to compute λ_{ign}.

λ_{ign} adaptive threshold employed to ignore proposals with low IoU with the guidance boxes (top-scoring ones) in the loss function, defined at Equation (4.4).

s_c Defines the real number of instances present in the image for class c, utilised in Chapter 2.

$\boldsymbol{\eta}$ Logits vector output from the NCC module.

T Number of output classes that the NCC module outputs.

$\boldsymbol{\theta}^F_{i,c}$ Element-wise mean from the values of the scoring matrices of the knowledge distillation module and all K refiners.

x_{min} Smaller (left) value of a bounding box.

v_i Central vertical coordinate v_i of each detected box.

μ Mean value of the central vertical coordinate of all detections.

$\boldsymbol{o}^k$ Proposal features in the clustering discovery space produced by MLP^k.

MLP^k Multi-Layer Perceptron (MLP) network present on k-th refiner, witch projects the features $\boldsymbol{\phi}$ into $\boldsymbol{o}^k$. More details on Section 6.1.

D_2 The dimension of each embedding present on $\boldsymbol{o}^k$.

$\boldsymbol{\psi}_{\bar{ts}^k_c}$ RoI-pooled features of each positive class top-scoring proposal $\bar{ts}^k_c$, which is stored in the memory bank.

$\breve{\boldsymbol{\psi}}^k \in \mathbb{R}^{N \times C}$ Denotes the *memory bank* associated with the k-th refiner.

N Max number of instances that the memory bank can record for each category c.

$\breve{\boldsymbol{o}}^k$ Defines the RoI-pooled features stored in the k-th feature bank projected in the clustering embedding space with MLP^k ($\breve{\boldsymbol{o}}^k = \text{MLP}^k(\text{FC}(\breve{\boldsymbol{\psi}}^k))$).

$\boldsymbol{\mu}^k_c$ Each cluster c centroid utilized in the discriminative loss function of each MLP^k. Details at Equation (6.1).

δ_v Distance value that defines the margin that controls the concentration of the cluster features in the discriminative loss.

$\mathcal{L}^k_{var}$ Part of the discriminative loss that penalises instances far from the cluster centre by using a margin loss. Details at Equation (6.2).

δ_d Distance value that defines the margin that controls the separation of the clusters in the discriminative loss.

$\mathcal{L}_{sep}^k$ Part of the discriminative loss that aims to repulse (separate) the clusters by forcing the cluster means to be far from each other. Details at Equation (6.3).

$\mathcal{L}_{disc}^k$ The final discriminative loss, which combines $\mathcal{L}_{var}^k$ and $\mathcal{L}_{sep}^k$. Details at Equation (6.4).

IomA$(.)$ Function that computes the Intersection-over-minimum-Area between proposals p_i^c and p_j^c. Utilised in the Proposal Overlap Suppression (POS) module, more details at Equation (6.7).

ξ Overlap threshold utilized by the POS to step positive proposals back to the background class.

CONTENTS

1 INTRODUCTION 20
1.1 Goals and Contributions 25
1.2 Hypotheses 25
1.3 Chapters Organization 26
2 BACKGROUND 27
2.1 Brief Review of Deep Neural Networks 27
2.1.1 Neural Networks Layers 28
2.1.1.1 Fully Connected Layer 28
2.1.1.2 Convolutional Layer 28
2.1.1.3 Pooling Layer 29
2.1.1.4 Region of Interest (RoI) Pooling 29
2.1.2 Activation Functions 30
2.1.3 The VGG16 pre-trained backbone 31
2.2 The Multiple Instance Learning with Online Instance Classification Refinement (MIL-OICR) framework 32
2.2.1 Data Inputs 33
2.2.2 Proposals Feature Extraction 34
2.2.3 Multiple Instance Learning 34
2.2.4 Online Instance Classification Refinement 35
2.2.5 Bounding Box Regression 36
2.3 Datasets 37
2.3.1 Pascal VOC 38
2.3.2 COCO 39
2.4 Evaluation Metrics 40
2.4.1 Detection Mean Average Precision 40
2.4.2 Correct Localisation 41
3 RELATED WORK 43
3.1 From Early Adoption of CNNs in WSOD to the MIL-OICR framework 44
3.2 Methods that proposed improvements to the MIL-OICR framework 46
3.3 Methods that Explore Activation or Attention Mechanisms 47
3.4 Methods that Mine Additional Instances during the Self-Supervision Building Process 49
3.5 Other methodologies and directions 51
4 DISTILLING SELF-SUPERVISION FROM THE REFINERS WITH AN ADAPTIVE SUPERVISION AGGREGATION FUNCTION 53
4.1 Method Overview 54
4.1.1 Knowledge Distillation 56
4.1.2 Adaptive supervision aggregation function 56
4.1.3 The global loss function 59
4.2 Experiments 59
4.2.1 Implementation Details 60
4.2.2 Ablation experiments 61
4.2.3 Comparison with the state-of-the-art 62
4.3 Conclusions 64
5 WEAKLY SUPERVISED CHARACTER DETECTION FOR LICENCE PLATE RECOGNITION TASK 66
5.1 Method Overview 67
5.1.1 Instance-Aware Self-Supervision 68

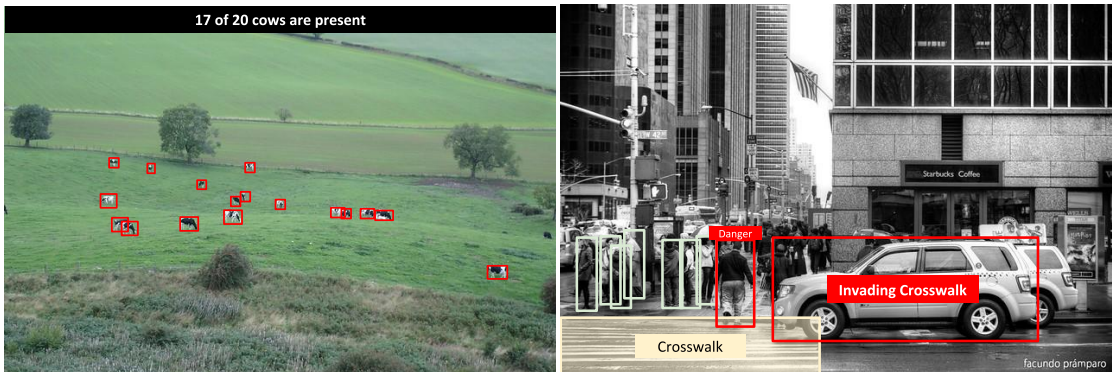
5.1.2 Recovering the Total Number of Characters 69
5.1.3 Extracting the Licence Plate string from the detections 70
5.2 Experiments 71
5.2.1 Licence Plate Datasets 71
5.2.2 Implementation Details 72
5.2.3 Ablation Experiments 73
5.2.4 Comparison with state-of-the-art 75
5.3 Conclusions 77
6 LONG-TERM INSTANCE MINING IN WEAKLY SUPERVISED OBJECT DETECTION 79
6.1 Overview of the Proposed Method 81
6.1.1 Instance Memory Bank 83
6.1.2 Clustering Discriminative Loss 83
6.1.3 Instance Mining 85
6.1.4 Proposal Overlap Suppression 86
6.2 Experiments 87
6.2.1 Datasets 87
6.2.2 Metrics 88
6.2.3 Implementation Details 88
6.2.4 Ablation Studies 89
6.2.5 Comparison with the state-of-the-art 90
6.3 Conclusions 93
7 FINAL REMARKS 95
7.1 Conclusions 95
7.2 Future Work 97
7.3 Source Code 98
7.4 Published and Submitted Papers 98
APPENDIX A — RESUMO EXPANDIDO 100
APPENDIX — REFERENCES 107

1 INTRODUCTION

The survival success of our species is directly related to our ability to identify, understand, interact with, combine, and create new objects in the physical world. Our brains can somehow learn how to localise and recognise objects with impressive precision and without much supervision. Consequently, it is not surprising that the computer vision community wishes to mimic these abilities in computers.

Object Detection is an exciting research area in computer vision that aims to reproduce the brain's ability to localise and recognise different objects. Given an input image, an object detector should be able to identify the number of instances of a known set of objects present and recover their spatial information within the image. The output of such a model can be utilised to fuel many computer vision solutions, such as (i) self-driving cars; (ii) robot vision; (iii) object counting applications; (iv) image search; (v) automatic waste sorting; (vi) security monitoring; (vii) cellular microscopy analysis, and so on. In Figure 1.1, two examples of computer vision applications based on object detection are shown.

Figure 1.1 – Two examples of computer vision applications that could be built using an 'on the shelf' object detector.



(a) Continuous animal inventory (b) Crosswalk safety

Source: Examples compiled by the author, original images are from (LIN et al., 2014a).

Historically, researchers had underestimated the complexity of the object detection task, thinking that it could be solved as a summer project (PAPERT, 1966). Unfortunately, the reality is that promising results emerged only decades later, with the combination of computer vision and machine learning algorithms (LOWE et al., 1999; VIOLA; JONES, 2004; DALAL; TRIGGS, 2005; FELZENSZWALB; MCALLESTER; RAMANAN, 2008). For a long time, the field was dominated by complex hand-crafted

local descriptors, multiple-stage hand-tuned pipelines, and discriminative classifiers. Furthermore, these initial experiments were limited to one or a few specific object classes in oversimplified scenarios. However, after Krizhevsky, Sutskever e Hinton (2012) demonstrated breakthrough results using a Convolutional Neural Networks (CNN) for image classification tasks, many researchers began utilising this kind of network in object detection pipelines. The use of CNNs eliminated the need for hand-made features, complex training pipelines, and discriminative classifiers. Since then, object detection models have achieved increasingly better results in terms of accuracy and speed (ZOU et al., 2023).

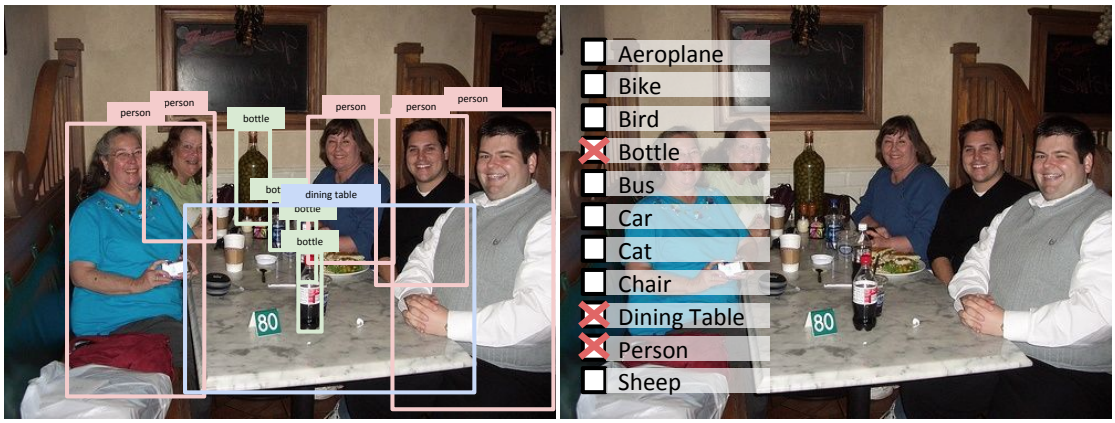
Despite the impressive results achieved by recent object detectors, a drawback of these methods is the amount of human supervision required to train such models. A Fully Supervised Object Detector (FSOD) model requires a rigorously annotated image dataset with bounding boxes and class labels. Examples of such datasets include Pascal Visual Object Classes (VOC) (EVERINGHAM et al., 2012), Common Objects in Context (COCO) (LIN et al., 2014b), Long Tail Instance Segmentation (LIVIS) (GUPTA; DOLLAR; GIRSHICK, 2019), and Open Images Dataset (OID) (KUZNETSOVA et al., 2020). Over the years, the volume of images, object categories, and bounding boxes has grown significantly in each new dataset. For example, the latest version of the VOC (EVERINGHAM et al., 2012) dataset comprises 20 categories, 11,540 images, and 27,450 bounding boxes, whereas the OID (KUZNETSOVA et al., 2020) dataset features 600 categories, nearly 1.9 million images, and 15.4 million annotated bounding boxes.

It is crucial to note that compiling a new FSOD dataset is a laborious task: As mentioned in (KUZNETSOVA et al., 2020), it takes approximately 7.4 seconds for an individual to annotate a single bounding box, which implies that a single annotator would need over 3.5 years of non-stopping work to annotate all 15.4 million bounding boxes present in OID. The cost of annotation can directly impact the viability of developing and deploying real-world object detection applications, particularly when a new problem requires building a dataset from scratch. Therefore, the development of approaches that reduce or eliminate the annotation cost associated with FSOD, while maintaining high detection accuracy, is a crucial step toward the maturity of the Object Detection field.

A potential approach to reducing annotation costs is to rely solely on image-level category labels and omit the bounding-box annotation stage when training the object detector. These annotations indicate whether target classes are present in images, and, by eliminating the need for bounding-box annotation, they substantially reduce the dataset's annotation cost. Figure 1.2 illustrates the effort required to create bounding box annota-

tions compared to the image-level category annotations. Methods that explore these weak annotations (image-level category) to train object detectors are referenced in the literature as Weakly Supervised Object Detection (WSOD). Unfortunately, as expected, the lack of localization supervision during the training process results in the detection quality of the WSOD methods being slightly worse than that of their FSOD counterparts. However, the results have improved significantly in the last decade of research (CINBIS; VERBEEK; SCHMID, 2016; TANG et al., 2017; TANG et al., 2018; ZENI; JUNG, 2020b; SEO et al., 2022; WU et al., 2024a).

Figure 1.2 – Discrepancy in the number of operations required to annotate an image with bounding boxes versus with image-level category annotations. The bounding box action count is estimated based on the assumption that annotators execute a minimum of 3 error-free actions per box: two for the top-left and bottom-right spatial coordinates, and one for object category assignment.



(a) Bounding box annotation: at least 30 actions (b) Image-level category annotation: 3 actions

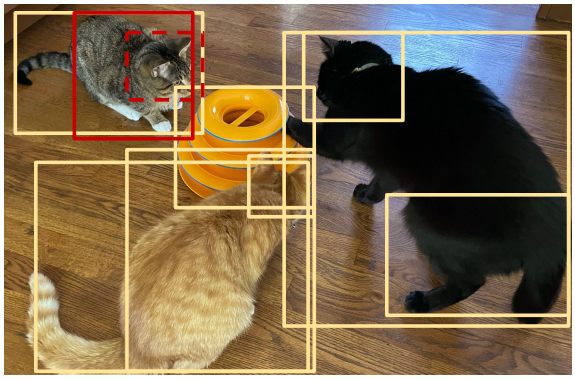
Source: Examples compiled by the author, original images are from (LIN et al., 2014a).

In the absence of bounding box annotations, most WSOD methods employ a multiple instance learning strategy to learn the spatial locations of the target classes. A region proposal method (UIJLINGS et al., 2013; ZITNICK; DOLLÁR, 2014; ARBELÁEZ et al., 2014) is utilised to generate a set of object proposals for each training image. An object proposal is a bounding box that covers specific regions in the image and may contain parts or whole objects within. The WSOD task involves categorising these proposals into target classes or as negative instances. Based on the image-level annotations, for each positive class, one can assert that at least one corresponding positive object proposal must exist in the object proposals set. Consequently, the WSOD learning process usually comprises two main stages: **Stage 1:** learning an instance classifier that assigns a score to each target category for every object proposal and **Stage 2:** employing the spatial infor-

mation of the top-ranked proposals as self-supervision to learn to identify the remaining proposals as belonging to positive classes or background (unassigned).

In the context of stage 1, Bilen e Vedaldi (2016) proposed a Multiple Instance Learning Network (MIL) module that creates a scoring matrix that assesses the relevance of each proposal for all target classes. This matrix serves two key roles: (i) it serves as a resource for selecting the top-scoring proposals, and (ii) by summing its values along the proposals' dimension, it enables the employment of a classification loss with image-level annotations as supervision. The top-scoring proposals selected from the scoring matrix of the MIL module could be used directly to train an object detector. However, this strategy tends to produce detections that are biased towards the most distinctive parts of objects, such as the faces of animals.

Figure 1.3 – The procedure for creating the self-supervision of the Online Instance Classification Refinement (OICR)'s refiner module involves selecting the highest-ranking proposal for each target class from the scoring matrix of the earlier module (dashed red). Proposals with a high IoU (solid red) are designated as positives for that class, while the remaining proposals are classified as background (yellow).

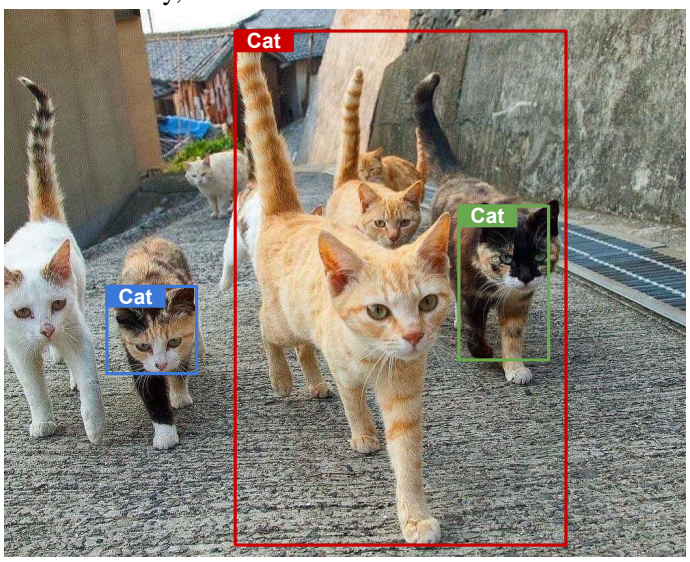


Source: Author's own illustration.

To overcome the limitations associated with MIL and perform the stage 2, Tang et al. (2017) introduced the OICR methodology. This approach is characterised by a sequence of self-supervised refiner modules that collectively enhance the detection outcomes. Each refiner builds its self-supervision on top of the top-scoring proposals selected from its predecessor's scoring matrix, except for the first one, which uses the MIL scoring matrix. As illustrated in Figure 1.3, the self-supervision procedure works by selecting proposals with a large Intersection over Union (IoU) with a top-scoring proposal and associating them with the same category to which the top-scoring proposal belongs. Proposals without a positive class are set to an additional category called *background*, which is not present in the group of classes used in the MIL module and represents proposals that do not belong to any target class. The learning objective of the refiner is then to utilise self-

supervision to differentiate proposals between the target classes and the background class, which is essentially what an object detector is expected to accomplish.

Figure 1.4 – Illustration of typical inaccuracies in the WSOD detection outcomes. The red bounding box represents a detection that encompasses multiple instances of the same category. The blue box indicates a bounding box that captures only the most descriptive part of the object, whereas the green box demonstrates a detection with inadequate bounding box alignment. Additionally, the white cat on the left was overlooked.



Source: Author's own illustration.

The combination of MIL and OICR modules constructs the essential framework that supports modern WSOD methodologies (SEO et al., 2022; WU et al., 2024a; WU et al., 2024b; TAHERKHANI; KAZEMI, 2024; FENG et al., 2024) that have significantly improved the results of WSOD in recent years. Further details of the MIL-OICR framework are provided in Chapter 2, and the State-of-the-Art (SOTA) approaches are analysed in Chapter 3. However, two significant challenges in the WSOD approaches are still not fully resolved (Figure 1.4 exemplifies the consequences of these challenges):

Dealing with multiple instances of the same class in one image: It is still common for WSOD methods to lose some instances of objects or encompass multiple of them in the same detection. Part of these problems is related to the way that OICR methodology builds self-supervision, where additional instances of the same class will be associated with the background class, which penalises the learning process (refer to the yellow rectangles in Figure 1.3).

Poor bounding box fitting: The fitting of the bounding box with the target objects in the image remains suboptimal. Sometimes, it fails to encompass the entire object,

and occasionally it exceeds the object's size. Additionally, for certain classes, it often focuses on the most distinctive aspect of the object.

1.1 Goals and Contributions

The primary objective of this Ph.D. dissertation is to develop methodologies to address the challenges described above, specifically dealing with multiple instances of the same class within an image in the context of WSOD. The very nature of the weak supervision provided for training these methods makes solving this problem a considerable challenge. It is possible to affirm that at least one instance per annotated class will be present in the image, but the annotations give no clues about how many instances of each class should be present in the images. To achieve these dissertation goals, we investigated and proposed solutions in three different forms: (i) how to reduce the harmful effects of these problems in WSOD's training process; (ii) exploiting the number of instances that is inherent present in the week annotations of some specific detection tasks; and (iii) building a self-supervised mechanism to discover and mine more instances of the same class in training time.

The key contributions of this dissertation are elaborated in detail in chapters 4, 5, and 6, where each chapter addresses different aspects of the research. Furthermore, chapter 7 serves to aggregate and synthesise the published results.

1.2 Hypotheses

This Ph.D. dissertation has as its core the following hypotheses:

— It is possible to discover and mine other instances of the same category in a self-supervised manner during the training period.

— Using information from previous training iterations can help in the mining process of additional instances.

— It is possible to mitigate the harmful effects of having multiple instances of the same category without knowing the number of object instances by creating a "feature bank".

1.3 Chapters Organization

The subsequent organisation of this dissertation proposal is outlined as follows. In Chapter 2, we elucidate the foundational background concepts vital for the reader's comprehension. Chapter 3 provides an in-depth discussion of previous studies and research efforts that are relevant to our work, offering a context for our investigation. In Chapter 4, we present the methodology published in (ZENI; JUNG, 2020b), which is designed to mitigate the adverse effects of multiple instances of the same classes within an image. We proposed a knowledge distillation module alongside an adaptive refinement supervision mechanism to improve object detection results. Subsequently, in Chapter 5, we explore the methodology published in (ZENI; JUNG, 2020a), which adapts the WSOD method proposed in Chapter 4 to the context of Weakly Supervised Character Detection (WSCD) in the application of Licence Plate Recognition (LPR). In this problem, the number of instances is inherently present in the week annotations. We propose an Instance-Aware Self-Supervision (IASS) building process and an Number of Characters Classification (NCC) module to exploit this information and improve the results. In Chapter 6, we present the methodology submitted to the IEEE Transactions on Image Processing (TIP) journal, which employs a cutting-edge self-supervised approach for discovering additional same-class instances. This method utilises an Multi-Layer Perceptron (MLP) projection network optimised with discriminative loss to maintain intraclass harmony and ensure interclass distinction. The MLP uses as supervision samples stored in a memory bank that retains the latest highest-scoring proposals from previous training iterations. Furthermore, a proposal for an overlap suppression mechanism is developed to eliminate redundant instances of the same object in the instance mining process. Ultimately, Chapter 7 presents our final synthesis of reflections, insights, conclusions, and potential directions for future research endeavours.

2 BACKGROUND

This chapter provides a brief review of the background concepts that we believe to be fundamental to the reader's understanding of the subsequent chapters. Section 2.1 presents a short review of Neural Network (NN) directed to Convolutional Neural Networks (CNN), its layers, activation functions, and the most utilised CNN (VGG16) architecture in Weakly Supervised Object Detection (WSOD) context. As supplementary material, we recommend the following references for inexperienced readers: (STEVENS; ANTIGA; VIEHMANN, 2020; GOODFELLOW; BENGIO; COURVILLE, 2016; KRIESEL, 2011; HE et al., 2016a). Subsequently, in Section 2.2 we explain the framework Multiple Instance Learning Network (MIL)-Online Instance Classification Refinement (OICR), which acts as a building structure for the current WSOD State-of-the-Art (SOTA) methods. In Section 2.3, we describe the benchmark datasets used by the WSOD methods, followed by Section 2.4, where we detail the evaluation metrics for comparing WSOD techniques on these datasets.

2.1 Brief Review of Deep Neural Networks

The goal of a NN is to approximate some desired function f^*. For example, in the image classification task, a NN defines a mapping $y = f^*(\mathbf{x}, \mathbf{w})$, where the input image $\mathbf{x}$ is mapped on the class labels $\mathbf{y}$ using the learnt parameters $\mathbf{w}$. The parameters $\mathbf{w}$ are chosen as the best approximation to the function f.

A NN is called a network because it is composed of a chain combination of many different functions, that is, $f(\mathbf{x})$ will have the form $f(\mathbf{x}) = f^{(2)}(f^{(1)}(f^{(0)}(\mathbf{x})))$, where $f^{(i)}$ is the i function (or layer) in the chain. For example, $f^{(0)}(.)$ represents the first layer of the network, and its output will feed the next layer, $f^{(1)}(.)$, and so on. During the NN training, the network will be optimised to direct $f(\mathbf{x})$ to match $f^*(\mathbf{x})$. The training data will provide a noisy approximation of $f^*(\mathbf{x})$ evaluated at different training points. Each example $\mathbf{x}$ is accompanied by the ground truth label $y \approx f^*(\mathbf{x})$. Training examples directly specify what the last layer must produce at each point $\mathbf{x}$ to approximate the values of y. The behaviour of the other layers (named hidden layers) is not directly specified in the training data. The learning algorithm must decide how to update those layer weights to produce the desired output (GOODFELLOW; BENGIO; COURVILLE, 2016). Each hidden layer of the network typically uses a vector or a matrix that defines the layer

weights, and each element of it can be interpreted analogously to a neuron.

The training procedure of a NN is performed using the back-propagation algorithm, which is often combined with an optimiser (a frequently utilised optimiser is Stochastic Gradient Descent (SGD)). Using the output of the last layer of the network, an objective function or a loss function is employed to measure how well (or similarly) the output of f^* matches the desired $\mathbf{y}$. Back-propagation works by calculating the gradients from the loss function with respect to each weight in the network using the chain rule of calculus. These gradients indicate the extent to which each weight contributes to the overall error, allowing the network to update the weights in a direction that minimises the error.

2.1.1 Neural Networks Layers

The layers of the neural network are the fundamental building blocks of a NN, and the combination of different types of layers allows the network to learn complex relationships in the data through multiple levels of abstraction.

2.1.1.1 Fully Connected Layer

A Fully Connected (FC) layer connects each unit (neuron) of the actual layer to all units in the previous layer. More precisely, given a neurone i, its output x_i is given by:

$$y_i = \sum_{j \in S_i} w_{ij} x_j + b_i, \tag{2.1}$$

where S_i are the neurons from the previous layer, and b_i is an added bias value (NIELSEN, 2015). Connecting all neurons between the two layers requires a large number of parameters and computations. That is one reason why, in image problems, it is common to first utilise convolutional layers to reduce the input size. Although they have high memory consumption, FC layers are powerful when used as classifiers, and it is common to see this kind of layer after a sequence of many Convolution Layers.

2.1.1.2 Convolutional Layer

The convolutional layer is the most commonly used layer to extract features from images in the NN context. Typically, the networks that employ them are called CNN. A

convolution is a spatial domain filter that acts directly on the signal in the spatial domain. Here we will focus on 2D signals because our work is related to images. The convolution filter applies a kernel K to the image I, generating a new image of the regions that react to the filter. The kernel is a small array with sizes m and n. Usually $m = n$, $mod(m, 2) = 1$, and $mod(n, 2) = 1$ (that is, the kernels typically have odd dimensions). Formally, the convolutional filter can be defined as:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n) \tag{2.2}$$

where i and j are an arbitrary pixel location in the I image.

The adaptation of a small Kernel in Convolutional layers leverages attractive properties. The kernel enables sparse interactions and parameter sharing, which means that we need to store fewer parameters, perform fewer computations, and thereby improve statistical efficiency. The kernel is also equivariant to translations and works with inputs of variable size (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.1.1.3 Pooling Layer

A Pooling Layer replaces the net output at a particular location with a summary statistic of the nearby outputs. For example, the max-pooling operation, one of the most utilised kinds of pooling, reports the maximum output within a rectangular neighbourhood. Other popular pooling layers include the average, the L^2 norm, and the weighted average based on the distance of the central pixel (GOODFELLOW; BENGIO; COURVILLE, 2016). The pooling layers help to make the representation approximately invariant to small translations of the input. It also helps to reduce the number of parameters in the NN by resizing the feature map to smaller dimensions. Typically, the max-pooling operation is performed within a 2 $times2$ window filter and a step size of 2. Therefore, after each max-pooling operation, the feature map size is halved compared to its original size.

2.1.1.4 Region of Interest (RoI) Pooling

The Region of Interest Pooling (RoIPool) layer is used to sample features inside a region of interest into a small feature map with a fixed spatial extent of $H \times W$ (e.g., 7×7). This process works by dividing the $h \times w$ Region of Interest (RoI) into a $H \times$

W grid of sub-windows of approximate size $\frac{h}{H} \times \frac{w}{W}$ and then applying a max-pooling operation to each sub-window. The pooling is applied independently to each feature map channel, as in the standard max pooling (GIRSHICK, 2015). This kind of pooling is attractive in the context of object detection: if we have a module that produces RoIs related to *object proposals*, with arbitrary sizes, RoIPool can be employed to normalise the dimensionality of all proposals, allowing the use of subsequent FC layers regardless of the input image/patch size. The RoIPool is a special case of the Spatial Pyramid Pooling (SPP) (HE et al., 2015) that has only one scale.

2.1.2 Activation Functions

An activation function defines whether a neuron unit should be activated (fire) or not. The purpose behind the activation function is to introduce non-linearity into the output of a given unit. One of the most used activation functions is Rectified Linear Unit (ReLU) due to its simplicity. ReLU is defined as:

$$ReLU(x) = \max(0, x). \tag{2.3}$$

Krizhevsky, Sutskever e Hinton (2012) discovered that using ReLU accelerates the convergence of stochastic gradient descent by six times and uses simpler operations (compared to sigmoid).

Another frequently utilized activation function is the Logistic Sigmoid activation function, which arises often while working with probability distributions. The Sigmoid function has the following form:

$$sigmoid(x) = \frac{1}{1 + e^{-x}}, \tag{2.4}$$

and the output range is $(0, 1)$. The function saturates when its argument is very positive or very negative, which means that the function becomes very flat and insensitive to small changes in its input (GOODFELLOW; BENGIO; COURVILLE, 2016).

Softmax is an activation function commonly used in the output of a classifier model. It is used to represent the probability distribution on n different classes. The softmax function can also be used within the model if we wish the model to choose between one of n different options for some internal variable. The softmax output is a vector of the

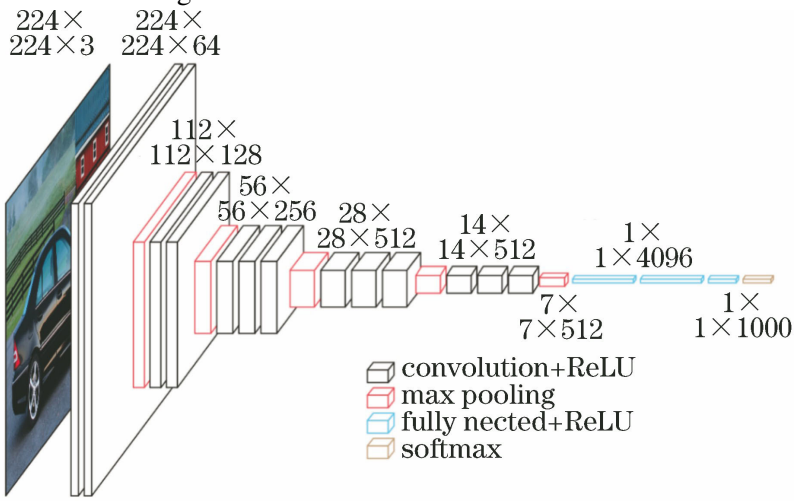
same size as the input vector. However, the values of this vector will be between 0 and 1, and the sum of the entire vector is equal to 1 (GOODFELLOW; BENGIO; COURVILLE, 2016). The softmax function is defined as follows:

$$softmax(x)_i = \frac{e^{x_i}}{\sum_{j \in x} e^{x_j}}. \tag{2.5}$$

2.1.3 The VGG16 pre-trained backbone

The VGG16 (HE et al., 2016a) is a CNN architecture, proposed by the Visual Geometry Group of the University of Oxford, which is widely used as an image feature extractor by WSOD methodologies. Its architecture is illustrated in Figure 2.1.

Figure 2.1 – Overall VGG16 architecture.



Source: researching.cn/richHtml/lop/2019/56/15/150003/img_3.jpg

The VGG16 comprises 16 layers with trainable parameters: 13 convolutional and 3 fully connected layers. Its simplicity is marked by the consistent use of 3×3 convolutional filters, with a stride of 1 and padding to maintain spatial dimensions. Periodically, a 2×2 max pooling reduces the spatial size. The network begins with 64 filters and doubles them per block up to 512. After final pooling, it generates a $7 \times 7 \times 512$ output, which is flattened into two fully connected layers of 4096 units and concludes with a softmax layer for categorising the 1000 classes. The available pre-trained weights of this model were learned using the ImageNet dataset (DENG et al., 2009).

One of the main strengths of VGG16 is its ability to learn hierarchical features by stacking small convolutional filters rather than using larger ones, which keeps the number of parameters manageable while maintaining effective receptive fields. However,

the model still contains around 138 million parameters, most of which are in the fully connected layers, making it computationally expensive and memory-intensive.

Although the VGG16 backbone seems to be outdated, as many other backbones have been proposed after it (HE et al., 2016b; SZEGEDY et al., 2016; HUANG et al., 2017; HOWARD et al., 2017; TAN; LE, 2019; RADOSAVOVIC et al., 2020), it is still widely adopted by the most recent SOTA WSOD methods (WU et al., 2023; FENG et al., 2023; YIN et al., 2023; WU et al., 2024a; WU et al., 2024b; FENG et al., 2024) primarily for consistency, stability, and fair comparison with prior work, as many seminal WSOD approaches and benchmarks were initially developed and tuned using this architecture.

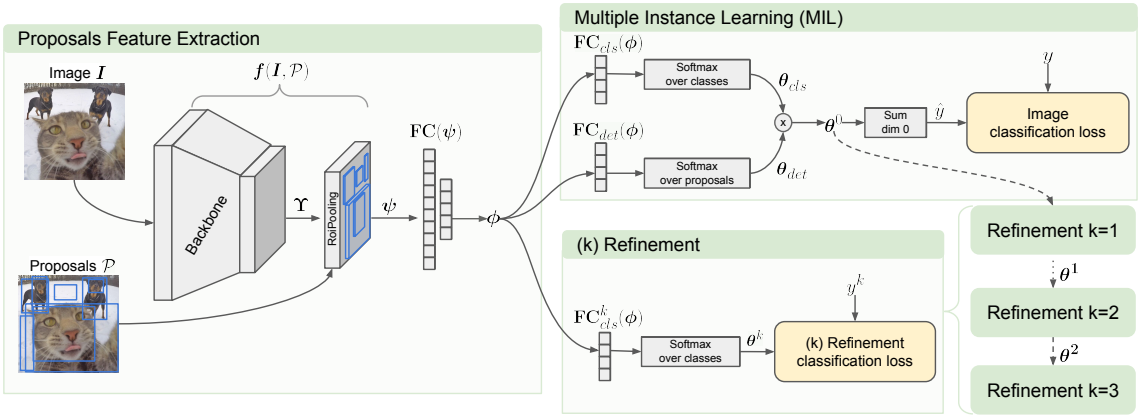
The VGG16 architecture, with a simple, sequential design, produces high-resolution feature maps that are well-suited for proposal-based and multiple-instance learning frameworks commonly used in WSOD, where fine-grained spatial information is critical for localisation under weak supervision. In addition, the computational and memory footprint is predictable and relatively modest compared to deeper modern backbones, which is advantageous given that WSOD pipelines often process thousands of region proposals per image, significantly increasing GPU memory usage. Consequently, memory constraints, especially when handling large proposal sets and batch sizes, remain a practical reason for retaining VGG16, as more complex backbones can make WSOD training prohibitively expensive without careful optimization.

2.2 The Multiple Instance Learning with Online Instance Classification Refinement (MIL-OICR) framework

As mentioned in Chapter 1, the integration of modules MIL (BILEN; VEDALDI, 2016) and OICR (TANG et al., 2017) forms the foundational structure that underpins contemporary WSOD techniques. The contributions of this dissertation, presented in Chapter 4, Chapter 5, and Chapter 6, also rely on the MIL-OICR framework. Therefore, to provide a comprehensive and centralised understanding of our contributions, this section presents a detailed overview of the operation of the two modules, as shown in Figure 2.2.

We start by defining the key inputs, followed by a detailed description of each significant module in the framework. Furthermore, we discuss the incorporation of the bounding boxes regression loss (REN et al., 2020) as a potential enhancement to the module OICR.

Figure 2.2 – Overall architecture of a MIL-OICR-based WSOD method. The Proposals Feature Extraction component identifies features from individual proposals by analysing their specific regions in the input image. Multiple Instance Learning (MIL) identifies the highest-scoring proposals that best represent the existing classes, guided by the classification ground truth. The refinement phase incorporates a self-supervised training strategy into WSOD, utilising instance-level supervision to enhance the overall results.



Source: Author's own illustration.

2.2.1 Data Inputs

The input of a WSOD approach is an RGB input image $\boldsymbol{I} \in \mathbb{R}^{3\times H\times W}$, where H represents the height and W the width of the image, and the corresponding collection of P bounding boxes object proposals $\mathcal{P} = \{p_1, p_2, \cdots, p_P\}$ generated by an unsupervised proposal generator, such as Selective Search (SS) (UIJLINGS et al., 2013) or Multiscale Combinatorial Grouping (MCG) (ARBELÁEZ et al., 2014). To create object proposals, SS starts by first over-segmenting the image into small regions and then hierarchically merging similar areas based on colour, texture, size, and shape compatibility. By exploring multiple similarity measures and scales, it efficiently produces a diverse set of candidate object bounding boxes without requiring class-specific training. In contrast, MCG adopts a more advanced strategy by unifying hierarchical segmentation and object proposal generation within a single framework. Instead of relying on hand-crafted cues such as color and texture, as in SS, MCG employs a powerful contour detector to construct a multiscale region hierarchy. Through efficient combinatorial grouping of these regions across scales, it produces high-quality object proposals and accurate segmentation masks, yielding better localization than SS by more precisely following object boundaries.

In training time, category annotations at the image level are also required, denoted by a one-hot-encoding vector $\boldsymbol{y} \in \{0, 1\}^C$, where C is the total number of object categories to learn and $y_c = 1$ indicates the presence of class c within $\boldsymbol{I}$, while $y_c = 0$ denotes its absence.

After defining the input, the goal of a WSOD model is to identify the proposals $p_i \in \mathcal{P}$ that relate to object instances of all categories c that meet the condition $y_c = 1$. This process begins with extracting the image features for every image proposal.

2.2.2 Proposals Feature Extraction

The goal of the proposal feature extraction module is to obtain fixed-size features from the image regions that correspond to each object proposal. Given an image $\boldsymbol{I}$, the convolutional layers of the VGG16 (SIMONYAN; ZISSERMAN, 2014) backbone are typically employed to extract the image-level features $\boldsymbol{\Upsilon} \in \mathbb{R}^{512 \times H' \times W'}$, where 512 is the number of features for each pixel, H' and W' are the height and width of the feature map. In the sequence, a RoI pooling (HE et al., 2015) with a 7×7 grid is used to re-sample $\boldsymbol{\Upsilon}$ uniformly within each object proposal p_i. Let $\boldsymbol{\psi} = \boldsymbol{f}(\boldsymbol{I}, \mathcal{P}) \in \mathbb{R}^{P \times D_0}$ represent the features sampled with the RoI pooling, where $\boldsymbol{\psi}_i \in \mathbb{R}^{D_0}$ stores the features of the proposal p_i, and $D_0 = 7 \times 7 \times 512 = 25,088$ is the resulting size of the 7×7 grid combined with the 512 channels of the output of the VGG last layer.

Next, a shallow MLP $\mathbf{FC}(.)$ formed by two fully connected layers transforms $\boldsymbol{\psi}$ into a new dimensional space, producing features at the proposal-level $\boldsymbol{\phi} = \mathbf{FC}(\boldsymbol{\psi}) \in \mathbb{R}^{P \times D_1}$, where $D_1 = 4,096$ is the proposal-level dimensionality. We can understand $\boldsymbol{\phi}_i$ as a unique identifier for each object proposal p_i, with the subsequent modules tasked with learning how to categorise these characteristics into relevant classes.

2.2.3 Multiple Instance Learning

The module MIL uses $\boldsymbol{y}$ as supervision to learn to select the best proposal candidate for each category c for which $y_c = 1$ will be used as self-supervision in subsequent modules. As shown in Figure 2.2, this is achieved by feeding $\boldsymbol{\phi}$ through two parallel FC layers that produce initial classification scores $\boldsymbol{\theta}_{cls} = \mathbf{FC}_{cls}(\boldsymbol{\phi}) \in (0,1)^{P \times C}$ and detection scores $\boldsymbol{\theta}_{det} = \mathbf{FC}_{det}(\boldsymbol{\phi}) \in (0,1)^{P \times C}$, respectively. The main difference between the networks is that $\boldsymbol{\theta}_{cls}$ is obtained by applying softmax over the classes (columns), while $\boldsymbol{\theta}_{det}$ is obtained by using softmax over the proposals (rows). One aims to identify the *most pertinent proposal* for each class, while the other seeks to determine the *most pertinent class* for each proposal.

By performing an element-wise multiplication of $\boldsymbol{\theta}_{cls}$ and $\boldsymbol{\theta}_{det}$, class scores are obtained for each proposal. Consequently, $\boldsymbol{\theta}^0 = \boldsymbol{\theta}_{cls} \odot \boldsymbol{\theta}_{det}$ represents the classification scores for each proposal, where $\boldsymbol{\theta}^0_{i,c}$ delineates the score for class c associated with proposal i. Here, 0 signifies the initial scoring matrix employed in the self-supervision methodology discussed in Section 2.2.4. A per-category image score $\hat{\boldsymbol{y}} \in (0,1)^C$ is computed by summing all the proposal scores for category c, i.e.,

$$\hat{y}_c = \sum_{i=1}^{P} \boldsymbol{\theta}^0_{i,c}, \tag{2.6}$$

which are fed to a multi-label classification loss $\mathcal{L}_{mil}$ using the provided ground-truth y, given by

$$\mathcal{L}_{mil} = -\sum_{c=1}^{C} y_c \log \hat{y}_c + (1 - y_c) \log(1 - \hat{y}_c). \tag{2.7}$$

Using argmax over the dimension of the proposals (i) of $\boldsymbol{\theta}^0$, the index related to the proposal with the largest contribution can be found for each category in the GT annotations. In particular, the *highest-scoring* proposal for each category c with $y_c = 1$ generally provides a reasonable indication of the localisation of the category instances. However, it tends to highlight the most discriminative part of the object (REN et al., 2020; SEO et al., 2022). The refinement modules aim to improve the object localisation in a self-supervised manner, as described next.

2.2.4 Online Instance Classification Refinement

In the refinement modules, a new learnable class $C+1$ is incorporated, known as the "background" class, designed to encompass all regional proposals that neither pertain to any $c \in \{1, \ldots, C\}$ nor adequately cover the target objects. Similarly to the classification network in the MIL module, the refinement modules perform instance classification by mapping $\boldsymbol{\phi}$ to $\boldsymbol{\theta}^k = \mathbf{FC}^k_{cls}(\boldsymbol{\phi}) \in \mathbb{R}^{P \times (C+1)}$ using a fully connected layer followed by a softmax over the class dimension, where k is the refinement level (several refinement modules can be chained).

The training of instance classification $\boldsymbol{\theta}^k$ is a self-supervised task utilising the *highest scoring proposals* $p_{ts^k_c}$ for each positive class c (where $y_c = 1$) from the prior module's score matrix $\boldsymbol{\theta}^{k-1}$ (or the MIL score matrix $\boldsymbol{\theta}^0$ when $k = 1$). The indices of the

top-scoring proposals are given by

$$\bar{ts}_c^k = \arg\max_i \boldsymbol{\theta}_{i,c}^{k-1}, \quad \text{for } c \in \{1, ..., C\} \text{ s.t. } y_c = 1, \tag{2.8}$$

and they are used to generate instance-level pseudo-labels $y_{i,c}^k \in \{0, 1\}$ for each proposal p_i related to class $c \in \{1, ..., C\}$ based on the overlap of p_i with the per-category top-scoring proposals:

$$y_{i,c}^k = \begin{cases} 1 & \text{if IoU}(p_{\bar{ts}_c^k}, p_i) \geq \lambda \text{ and } y_c = 1 \\ 0 & \text{otherwise} \end{cases}, \tag{2.9}$$

where λ represents the IoU threshold, set to 0.5. The background pseudo-labels $y_{i,C+1}^k = 1$ are assigned to instances p_i that lack adequate overlap with *every* highest scoring proposal; otherwise, $y_{i,C+1}^k = 0$. Finally, the instance classification loss is expressed as

$$\mathcal{L}_{cls}^k = \frac{1}{P} \sum_{i=1}^{P} \frac{1}{C+1} \sum_{c=1}^{C+1} w_{i,c}^k y_{i,c}^k \log \boldsymbol{\theta}_{i,c}^k, \tag{2.10}$$

where $w_{i,c}^k = \boldsymbol{\theta}_{\bar{ts}_c^k,c}^{k-1}$ is the score of the *highest scoring proposal* related to proposal p_i, which serves as a weight regularizer.

2.2.5 Bounding Box Regression

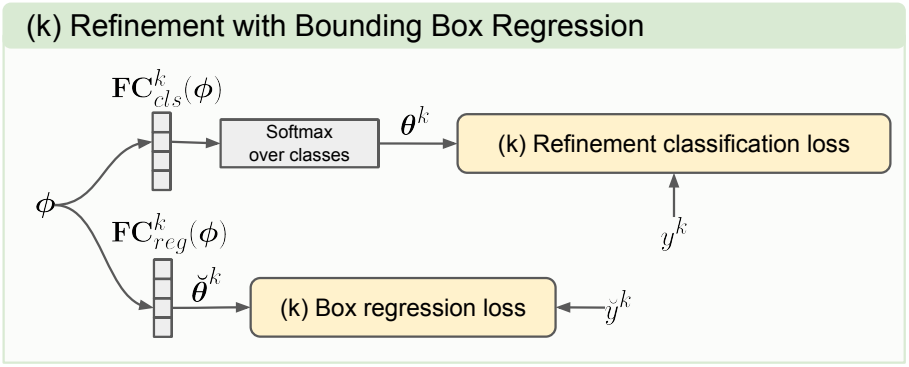
One mechanism that can be integrated into the refinement module to improve the localisation accuracy is the *bounding box regression* proposed by Ren et al. (2020). Similarly as the OICR refiner, it feeds the proposal-level features $\boldsymbol{\phi}$ to a FC layer $\mathbf{FC}_{reg}^k(\boldsymbol{\phi}) \in \mathbb{R}^{P \times C}$ to perform the regression $\breve{\boldsymbol{\theta}}_{i,c}^k$ of four bounding box coordinates for each class c. Only proposals that satisfy $y_{i,c}^k = 1$ and $c \neq C + 1$ are included in this process. The objective is to approximate the regressed coordinates of each selected proposal to the coordinates of the top-scoring proposal that made it positive in the self-supervision building process (i.e. the $p_{\bar{ts}_c^k}$ that satisfied $\text{IoU}(p_{\bar{ts}_c^k}, p_i) \geq \lambda$). The self-supervision $\breve{y}_{i,c}^k$ for this mechanism is built in a way similar to the one described in Equation (2.9). To learn the $\mathbf{FC}_{reg}$ weights, the smooth L1 loss function is used, which is commonly employed in

regression problems. The bounding box regression loss is defined as follows:

$$\mathcal{L}_{reg}^{k} = \frac{1}{P}\sum_{i=1}^{P}\frac{1}{C}\sum_{c=1}^{C} smooth_{L1}(\breve{\boldsymbol{\theta}}_{i,c}^{k}, \breve{y}_{i,c}^{k}). \tag{2.11}$$

Figure 2.3 shows a diagram of the update of the refiner module with the bounding box regression mechanism. Refer to (REN et al., 2020) for a more detailed formulation of the bounding box regression module.

Figure 2.3 – The update of the OICR refiner module with the bounding box regression mechanism.



Source: Author's own illustration.

2.3 Datasets

In this section, we will review the datasets typically used for training, evaluation, and benchmarking of WSOD models, which include Pascal Visual Object Classes (VOC) 2007 (EVERINGHAM et al., 2007), VOC 2012 (EVERINGHAM et al., 2012), and Common Objects in Context (COCO) 2014 (LIN et al., 2014a). These datasets are structured into three distinct sets of images, which are defined as follows:

Train set: images and their respective annotations that should be used to train the model;

Val set: images and their respective annotations that should be used for validation of the model;

Test set: test images with no annotation data that should be used to test the trained model.

Except for VOC 2007, the annotations of the test set are not publicly available to prevent cherry-picking well-performing models. In this case, the evaluation of the test set is performed using the evaluation server provided by each dataset. In Table 2.1, some statistics of these datasets are compiled, such as the number of images, the number of classes, the number of bounding boxes, and the mean instances per image. Although these datasets were built with the primary objective of training Fully Supervised Object Detector (FSOD) models, they satisfy the needs for WSOD methods and serve as a comparison baseline. In this context, the object annotations are converted to image-level annotations, meaning that no bounding boxes are provided to the WSOD model. The unique exception is in the evaluation of WSOD methods, which uses the same methodology as FSOD methods and requires the bounding box annotations. In the following sections, a more detailed description of each dataset is provided.

Table 2.1 – Statistics regarding the relevant datasets for training WSOD models.

Dataset	Number of Classes	Images Number	Training Images	Validation Images	Testing Images	Annotated Objects	Average Instances per Image
VOC 2007 (EVERINGHAM et al., 2007)	20	9,963	2,501	2,510	4,952	24,640	2.47
VOC 2012 (EVERINGHAM et al., 2012)	20	11,540	5,717	5,823	10,991	27,450	2.37
COCO 2014 (LIN et al., 2014a)	80	164,062	82,783	40,504	40,775	896,782[1]	7.27[1]

Source: Author's compiled data.

2.3.1 Pascal VOC

The PASCAL Visual Object Classes (VOC) datasets serve as a key benchmark in visual object recognition and detection evaluation. Originally from the PASCAL VOC challenge series, these datasets offer standardised images, annotations, and evaluation protocols to advance research in object classification, detection, and segmentation. Sourced from Flickr, these real-world images showcase a diverse range of scenes, scales, and occlusions, reflecting the natural complexities of visual perception. The 2007 and 2012 editions are commonly used in WSOD studies for method comparisons.

The VOC 2007 (EVERINGHAM et al., 2007) dataset comprises 9,963 images annotated within 20 object classes, with training, validation, and test partitions. Each object instance features a bounding box and class label, with additional details on truncation and occlusion for refined evaluation metrics. Expanding on this, the VOC 2012 (EVERINGHAM et al., 2012) collection offers 11,540 similarly annotated images and is distin-

[1]This information was derived exclusively from the training and validation sets due to a lack of available information regarding the total number of instances for the testing set.

guished by superior segmentation masks and increased scale, which benefit tasks such as semantic segmentation. Both datasets maintain identical object categories, including frequently seen classes such as “person, ” “car, ” “dog, ” “cat,” and “bicycle.”

The VOC datasets are meticulously designed to enable robust empirical evaluation, playing a crucial role in the advancement and assessment of various computer vision algorithms, ranging from classic feature-based models to contemporary deep learning methods. The datasets’ thorough annotations and standardised evaluation metrics, notably Mean Average Precision (mAP) for object detection, ensure their ongoing relevance. Their public accessibility and user-friendly nature establish them as essential assets for academic and practical machine learning in visual object recognition.

2.3.2 COCO

The Microsoft Common Objects in Context (COCO) 2014 dataset (LIN et al., 2014a) is a large-scale dataset designed to advance object recognition, segmentation, and contextual scene interpretation in computer vision. Unlike previous datasets with isolated objects on plain backgrounds, COCO 2014 focuses on contextual realism by featuring images with multiple objects in complex scenes. This characteristic makes it ideal for the development and evaluation of algorithms facing real-world visual challenges. The dataset supports various tasks, including object detection, instance and semantic segmentation, keypoint detection, and caption generation.

COCO 2014 includes approximately 164,000 images, with around 83,000 for training, 41,000 for validation, and the remaining images for testing. The dataset features over 2.5 million manually segmented instances of objects across 80 everyday categories, including humans, animals, vehicles, and household items. Each instance has a bounding box and pixel-level segmentation annotations, offering comprehensive supervision for both broad- and detailed-recognition tasks. Moreover, images are accompanied by up to five human-generated captions, facilitating functions at the interface of vision and language, like image captioning and visual question answering.

The dataset is distinguished by its focus on identifying object instances in natural settings and its inclusion of non-standard perspectives with varied poses, sizes, and occlusions of objects. It employs a meticulous annotation method to enhance labelling accuracy. Evaluation of detection and segmentation tasks is conducted using the Mean Average Precision (mAP) metric. Unlike the VOC mAP, which uses a fixed Intersection

over Union (IoU) threshold of 0.5, COCO introduces a more thorough metric. It averages mAP over several IoU thresholds from 0.50 to 0.95 in 0.05 increments, denoted as mAP@[.50:.95], offering a more detailed evaluation of detector performance with respect to localisation precision. Consequently, COCO 2014 serves as a key benchmark in computer vision, significantly contributing to the advancement and evaluation of cutting-edge models in object recognition and other areas.

2.4 Evaluation Metrics

This section examines the primary metrics, Detection mAP and Correct Localization (CorLoc), used for evaluating and comparing WSOD methodologies.

2.4.1 Detection Mean Average Precision

The mAP is a widely utilised metric to compare object detection models in a given detection dataset. Historically, the metric evolved to more stringent standards as datasets evolved. However, in a high level, the mAP is defined as the mean of the Average Precision (AP) of each class in the dataset. The AP is calculated using each class's precision/recall curve, which is obtained from the method's ranked detections. The precision measures how accurate the method's detections are, i.e. the percentage of detections that are correct. In contrast, recall measures how well the method identifies all the positive detections.

In the VOC 2007 evaluation protocol, a detection was considered correct if the predicted bounding box overlapped a ground-truth annotation with an intersection-over-union (IoU) of at least 0.5. For each class, detections were sorted by confidence score and used to construct a precision-recall curve. The AP was then computed using an 11-point interpolation scheme, where the maximum precision was sampled at fixed recall levels ranging from 0 to 1 in steps of 0.1. The final mAP score was obtained by averaging the AP values across all 20 classes. For a comprehensive description of the computation procedure, please consult the original dataset publication (EVERINGHAM et al., 2007), which documents the methodology and implementation details in depth.

The VOC 2012 evaluation protocol retained the 0.5 IoU threshold and the same per-class evaluation procedure. However, it revised the computation of the AP. Instead

of the 11-point interpolation, the AP was calculated as the area under the precision-recall curve using all available operating points. This change resulted in a more accurate and less discretised estimate of the detector's performance, as it better reflected variations in precision across the full recall range. Please refer to the dataset paper (EVERINGHAM et al., 2012) for more details on the computation.

The COCO evaluation methodology introduced a more comprehensive and challenging definition of mAP. Rather than evaluating performance at a single IoU threshold, COCO computes the AP across multiple IoU thresholds ranging from 0.5 to 0.95 in increments of 0.05. The final mAP is obtained by averaging the AP values over all classes and all IoU thresholds. This formulation penalises imprecise localisation and encourages detectors to produce tighter and more accurate bounding boxes. Additional metrics, such as AP at specific object scales, further characterise detector behaviour. For a detailed description of the evaluation methodology, see the dataset paper (LIN et al., 2014a).

These successive changes to the mAP metric reflect the increasing complexity and ambition of object detection benchmarks. As datasets grew in size and diversity, evaluation protocols evolved to provide more reliable, fine-grained, and localisation-sensitive performance measurements. Consequently, modern metrics such as COCO mAP offer a more realistic assessment of detection quality, while earlier VOC-based metrics remain useful for historical comparison and lightweight evaluation.

2.4.2 Correct Localisation

Correct Localisation (CorLoc) is an evaluation metric still commonly used in WSOD to measure a model's ability to localise objects when trained with image-level supervision only. CorLoc is defined as the percentage of images in which at least one predicted bounding box for a given class overlaps a ground-truth object of the same class by an intersection-over-union (IoU) of at least 0.5. Unlike the mAP, it evaluates localisation success at the image level rather than penalising multiple detections or ranking confidence scores. This property makes CorLoc particularly suitable for WSOD, where the objective during training is to discover and localise the most representative object instance rather than exhaustively detect all instances. As a result, CorLoc remains widely used as a complementary metric in WSOD to assess whether a method has successfully learned object

localisation under weak supervision. In other words, CorLoc is defined as:

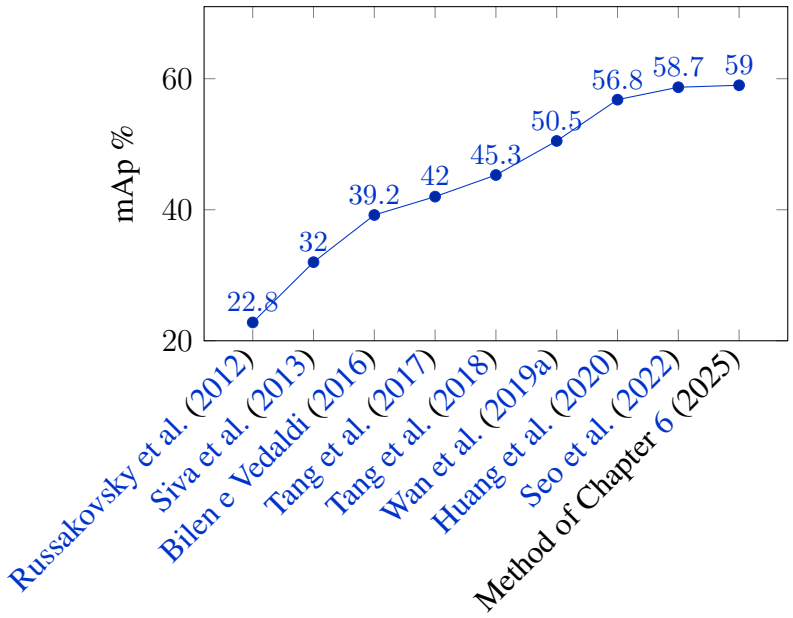
$$CorLoc = \frac{TP_{img}}{N_i}, \tag{2.12}$$

where TP_{img} is the number of images that have at least one detection with $IoU > 0.5$ and N_i is the number of images that contain instances of the target class.

3 RELATED WORK

A substantial portion of Weakly Supervised Object Detection (WSOD) research predates the Convolutional Neural Networks (CNN) adoption era (RUSSAKOVSKY et al., 2012; SIVA et al., 2013; SONG et al., 2014). However, given that current State-of-the-Art (SOTA) studies depend exclusively on features extracted with such networks, this review will focus solely on methods that include CNNs in their pipelines. In Figure 3.1 we present a graph showing the evolution over the years of the results of the WSOD methods in the Pascal Visual Object Classes (VOC) 2007 dataset. The plateau observed in the graph suggests that existing methodological approaches may have reached a performance ceiling. This apparent saturation in achievable gains provides a plausible explanation for the noticeable decline in the introduction of novel methods after 2020, as researchers may have encountered diminishing returns despite continued efforts to refine or extend the current techniques.

Figure 3.1 – Evolution of the detection mAP of WSOD methods over the years on dataset VOC 2007 test set .



Source: Author's own illustration.

3.1 From Early Adoption of CNNs in WSOD to the Multiple Instance Learning Network (MIL)-Online Instance Classification Refinement (OICR) framework

At the beginning of the adoption of CNNs for WSOD, the features extracted by the network were primarily used as additional descriptors, which were combined with other feature descriptors. An example is the work of Cinbis, Verbeek e Schmid (2016), where the CNN features are combined with Fisher Vectors as descriptors. The authors proposed a multi-fold multiple instance learning training procedure, which splits the positive instances into K training folds. Their methodology also includes an objectness refinement step to improve localisation accuracy. As the pre-trained CNN functions purely as a feature extractor, its weights were not fine-tuned to the data of interest, resulting in the use of features that are inadequately tailored to the target dataset, which in turn causes reduced accuracy.

Initial attempts of fine-tuning the CNN layers appeared in the works of Teh, Rochan e Wang (2016) and Li et al. (2016). In (TEH; ROCHAN; WANG, 2016), the authors explored *attention networks* in the context of WSOD. An attention score for each proposal is calculated using Fully Connected (FC) Layers and a softmax operation. The attention scores are combined to obtain an image-level classification vector, which feeds a logistic classification loss that guides the optimisation of the model. Detection is performed by simply selecting the object proposal with the highest attention score in the image. In contrast, (LI et al., 2016) proposes a two-stage adaptation algorithm. The first stage fine-tunes the network to collect class-specific object proposals with high precision, and the second uses high-scoring object candidates to optimise the CNN representations in such a way that object detectors are gradually learnt. A drawback of (TEH; ROCHAN; WANG, 2016) and (LI et al., 2016) is that object proposals are fed into an CNN one-by-one, because these regions have different aspect ratios. As a result, the process becomes exceedingly slow and lacks efficiency. To address this issue, He et al. (2015) proposed the Spatial Pyramidal Pooling (SPP), an ingenious and lightweight solution to the problem of extracting features from object proposals of multiple sizes in a single forward pass of the network. The SPP, also known as *RoI pooling*, was proposed in the context of Fully Supervised Object Detector (FSOD). However, it did not take long for it to be incorporated into the WSOD methodologies. More details regarding this pooling methodology are given in Section 2.1.1.4.

The approach proposed by Bilen e Vedaldi (2016) was one of the first to explore

the use of RoI pooling in the context of WSOD, making it a pioneer in end-to-end WSOD model training. However, the key contribution of this work is a two-stream module, now commonly referred to as the MIL module (some studies also refer to it as Multiple Instance Detection Network (MIDN) in their texts). The MIL module is widely utilised by the SOTA methods due to its simplicity and effectiveness in scoring the object proposals as belonging to each target category by employing a classification loss with weak annotations. More details regarding this module are provided in Section 2.2.3. The approach introduced by (BILEN; VEDALDI, 2016) represents significant progress in WSOD research. However, it suffers from the issue of part domination, wherein bounding boxes for many object categories capture only the most distinguishing characteristic of the object. This problem arises because the WSOD method lacks object localisation supervision. As there is no optimisation incentive to identify the entire object, the most descriptive part meets the classification loss criteria employed.

To address the subpart dominance issue in MIL, Tang et al. (2017) proposed a refinement module that builds its self-supervision by selecting proposals with high Intersection over Union (IoU) with the positive categories, where top-scoring proposals are selected from the MIL scoring matrix or a previous refiner. The idea behind this methodology assumes that the proposal generation method is not random and will create object proposals related to objects and their subparts. By selecting additional overlapping proposals as positive samples and including them in the model learning process, the model is gradually compelled to learn more spatial information about the target classes by these new constraints on its optimisation process. The combination of OICR with MIL is the standard framework on which modern WSOD modules are based. More details regarding the OICR module and the MIL-OICR framework are presented in Section 2.2.

As an alternative to the MIL module, Wan et al. (2019a) introduced a Continuation Multiple Instance Learning (C-MIL) methodology, which uses a series of smoothed loss functions to approximate the target loss. Additionally, the authors propose a parametric strategy for instance subset partition, which facilitates the activation of object extents. The authors claim that this smoothed process alleviates the non-convexity problem present in MIL. However, the MIL module won the “battle”, and it is chosen as the methodology for scoring the proposals guided by the week annotations in posterior methods. We believe that one of the reasons for this widespread use is that the OICR authors made the source code available. As its implementation is built on top of Python and PyTorch, while C-MIL uses Lua and Torch, it was easier for the community to adapt their methodology for later

methods.

3.2 Methods that proposed improvements to the MIL-OICR framework

After the publication of OICR Tang et al. (2017), the adoption of the MIL-OICR framework quickly became the norm between successor methods. In this section, we focus on methodologies that proposed improvements to the MIL-OICR framework but did not fall into a common category with other methods.

Gao et al. (2019) proposed using a coupled pair of MIL modules to address the problem of converging to the most discriminative parts of the object. The output of the first MIL module is used to remove some regions from the input of the second MIL module. The authors proposed a segmentation-guided proposal removal algorithm to perform this process. The idea behind it is that by removing the best detections from the first module, it opens a window to the second MIL module to learn better representations. The scoring matrices of both modules are joined (coupled) and feed the first OICR refiner, which uses it to select top-scoring proposals and build its self-supervision.

Zeng et al. (2019) introduced a training strategy for objectness distillation, which integrates bottom-up and top-down objectness in conjunction with low-level metrics and CNN confidence scores. The methodology establishes specific regression targets through an adaptive linear combination. The process enhances the region proposal learning by aligning bounding box regression with high-objectness targets, facilitating the transfer of bottom-up objectness insights into CNN through optimization.

Chen et al. (2020) proposed the Spatial Likelihood Voting (SLV) module to assist the WSOD model in converging the proposal localisation. The module is configured with a pair of FC layers. One is dedicated to the task of re-classification, while the other focuses on re-localization. The guidance for these layers is established through a process that involves averaging the classification score matrices from the OICR refiners, alongside a spatial likelihood voting algorithm. The algorithm involves selecting instances, accumulating spatial probabilities, and executing a voting step for areas with high probability likelihoods.

Xu et al. (2021) proposed a pyramidal MIL strategy, which can better tackle multiple instances of the same class in an image. The authors also include a mask-guided self-correction method to correct the initial pseudo-ground truths of MIL via a proposed self-labelling and local refinement mechanism. A limitation of this method is that it re-

lies on segmentation masks generated by a pre-trained weakly supervised segmentation method, which is not optimised for the target task and can potentially interfere with the learning process.

Yin et al. (2023) introduced a cyclic bootstrap labelling (CBL) framework to train WSOD models, which optimises the MIL module with rank information from a reliable teacher network. The framework includes a weighted ensemble teacher model, which is obtained using a weighted exponential moving average strategy to leverage the outputs of various refinement modules. In addition, the framework includes a class-specific ranking distillation algorithm that distils information from the teacher into the MIL module. The key idea is that the bootstrap strategy guides the MIL module in assigning higher scores to accurate proposals among their neighbours, thus benefiting the subsequent modules. The authors also propose including an R-CNN-like head supervised with a multi-seed R-CNN algorithm to identify accurate positive seeds for training.

Feng et al. (2023) proposed an invariant and equivariant network (IENet) framework, which aims to produce a more stable and complete WSOD model. The proposed framework simultaneously learns an invariant and an equivariant object detector by employing two learning strategies. An invariant learning structure is constructed to encourage the object detector to make consistent predictions for the labelled instances before and after arbitrary transformations during training. Additionally, an equivariance learning strategy is designed to encourage the network to perceive more holistic object patterns, potentially enabling it to be aware of the localisation quality.

Wu et al. (2024b) explored the issue of misclassification in WSOD. The scholars claim to be the first study to tackle this issue, providing its definition, analysis of causes, characteristics, and solutions. To address this issue, an adaptive misclassification correction strategy is proposed for the model and dataset. The strategy automatically captures misclassification patterns and corrects misclassified object-related proposals during training. The authors also propose a misclassification tolerance strategy that enables the classifier to make mistakes in classifying highly similar semantics, thus alleviating semantic ambiguity during inference.

3.3 Methods that Explore Activation or Attention Mechanisms

One notable aspect of CNNs is that we can visualise the activation of the extracted features by converting them to an attention map image. These maps give a very strong

clue of the spatial localisation that the model is focusing on, and some methods explore this information to improve the results of the MIL-OICR framework. Therefore, this section is dedicated to methods that explore the use of activation or attention mechanisms in their methodologies.

Diba et al. (2017) proposed a cascaded convolutional network composed of three stages. The first uses a Class Activation Map (CAM) module to create class activation maps. The second stage utilises the CAM as a supervisory signal to learn a segmentation mask of the objects of interest. The third stage employs a custom MIL module to localise the objects of interest. It utilises the object proposals selected based on the obtained segmentation masks to establish self-supervision for the remaining modules. In the same direction, Wei et al. (2018) introduced a method that extends the MIL-OICR baseline by using CAM to mine tight object boxes by exploiting segmentation confidence maps. The segmentation confidence maps are employed to evaluate the objectness scores of proposals based on two properties: purity and completeness.

Cheng et al. (2020) introduced a proposal generation method that combines the proposals generated by selective search with a Gradient-weighted Class Activation Map technique to create proposals that better encompass the target objects (i.e., better IoU with the ground truth boxes). The authors also introduce a proposal selection module, which selects confident positive proposals as much as possible while simultaneously selecting class-specific hard negatives to focus training on more discriminative negative proposals by up-weighting their losses.

Huang et al. (2020) proposed a Comprehensive Attention Self-Distillation (CASD) methodology designed to consolidate and balance feature learning among positive instances. The approach applies multiple transformations (flip and scale) to the same input image and extracts its attention maps. The attention maps are aggregated for each top-scoring proposal using the element-wise max operator, and all individual attention maps are simultaneously approximated from the aggregated attention map by using an L2 loss.

Wu et al. (2023) proposed a bottom-up aggregated attention to alleviate the issue of part dominance in WSOD. It differs from previous attention methods by incorporating multilayer features to compensate for location information without attaching learning branches or learnable parameters. The authors also proposed a high-quality proposal selection mechanism, which measures the quality of pseudo-labels based on the loss incurred during the self-supervision building phase. Wu et al. (2024a) proposed a methodology to mitigate the issue of the model focusing on the most discriminative features of the

target objects. The authors proposed a feature grouping pooling layer, called MixMax-Pool, which captures objects and image context on multiple scales. The MixMaxPool brings negligible computational overhead and does not introduce learnable parameters that affect the pre-trained model. Additionally, a Spatial Local Enhancement (SLE) module is proposed to enhance less discriminative features based on neighbourhood statistics, thereby alleviating the problem of ignoring parts of unsalient objects. In the same direction, Feng et al. (2024) addresses the ambiguity in the feature space by proposing an attention module that enhances the discriminability of proposals, thereby effectively improving the saliency of positive proposal features. Additionally, a bounding box refinement module is proposed to integrate multiple bounding box filter strategies into a unified pipeline, which significantly improves the quality of pseudo-labels while also alleviating the issue of part domination. The authors also proposed two *strongly supervised* retraining stages, named as Loss-based Label Division (LLD) and Score-guided Weight Adjustment (SWA) strategies. These strategies involve selecting partial pseudo-labels and adjusting their weights during a second stage of training.

3.4 Methods that Mine Additional Instances during the Self-Supervision Building Process

As mentioned in the introduction, the week annotations provide only the guarantee that at least one instance of the target categories is present in the image. However, the week annotations do not provide any information regarding the count information of target classes (except for a few particular cases, such as the Licence Plate Recognition (LPR) task). This limitation forces the refiner's self-supervision building process to select only the top-scoring proposal as the best candidate for the target categories, thereby excluding other additional object instances of the same category from being labelled as positive examples in the self-supervision building process. This issue presents a clear challenge to the model learning process, and developing methodologies that can identify additional instances during the self-supervision building process is essential to a better optimisation process of the WSOD methods. Therefore, this section focuses on techniques that aim to mitigate this issue or discover these additional object instances in the refiner's self-supervision building process.

Gao et al. (2018) demonstrated that the incorporation of counting annotations can enhance the performance of WSOD models based on the MIL-OICR framework, at the

cost of additional annotations (instance counts). The authors proposed a greedy graph multiple instance selection algorithm that uses the instance counts as supervision.

Tang et al. (2018) proposed a simple solution to mitigate the harmful effects of not setting the additional instances in self-supervision. The authors proposed a two-stage region proposal network that explores responses in the mid-layers of a network to generate object proposals. However, the most interesting contribution from this work is the so-called "trick". Essentially, it ignores proposals with minimal IoU within the highest-scoring proposals in the loss function, thereby reducing the optimisation effects from co-occurring same-class instances within the same image. However, ignoring the additional same-class instances in the loss is not the best optimisation choice, as the model will miss the opportunity to learn helpful information from these ignored object instances. Tang et al. (2018) introduced a proposal clustering approach, which aims to select one or more supervision boxes during training. Selecting more than one supervision box is interesting because objects typically have multiple parts and can also have various instances present in the same image. The methodology, based on k-means, employs a predefined fixed number of clusters, which limits its mining capabilities and makes it very slow.

Ren et al. (2020) presented a Multiple Instance Self-Training (MIST) approach, designed as a unified, context-focused, and instance-aware framework. The mining strategy (instance-aware) chosen by the authors is quite simple: it interactively selects a percentage of the highest-scoring proposals by employing Non-maxima Suppression (NMS). A parametric spatial dropout method, termed "Concrete DropBlock", is proposed to encourage the model to focus on context (context-focused) rather than the most discriminative parts, thus alleviating the part-domination issue. Furthermore, the authors proposed including a bounding box regression loss to OICR's refiners (more details in Section 2.2.5).

Lin et al. (2020) proposed an object instance mining methodology that can be included in the OICR self-supervision building process. The instance mining strategy utilises the top-scoring proposal of each target category to build an appearance and a spatial graph. The spatial graph is built using the IoU between each proposal and the top-scoring proposals, and the appearance graph is built based on the Euclidean distance between the feature vectors of each proposal and the top-scoring proposals. An average appearance distance threshold is computed using the proposals that have an IoU larger than a predefined threshold. Instances that have an appearance distance with the top-scoring proposal larger than the appearance distance are selected as additional positive

instances for the same target category of the top-scoring proposal in the self-supervision building process.

The previous approaches largely depend on heuristics rather than on using a learning methodology to mine additional instances. In this context, Seo et al. (2022) introduced OD-WSCL, an approach for object discovery using weakly supervised contrast learning. This method utilises a similarity network, which projects proposal features into an embedding space that facilitates instance discovery. The network is trained with a weakly supervised contrastive loss designed to bring positive samples closer and separate negatives from positives in the embedding space. The discovery of new instances is performed using a similarity threshold calculated from the top-scoring proposal and proposals that have a high IoU with it (similarly to Lin et al. (2020)). An issue with OD-WSCL is that it relies on iteration batch samples to select pseudo-ground-truth samples for training the similarity network, which requires larger batch sizes to perform effectively. In addition, its contrastive loss does not enforce that the embeddings of different categories do not overlap (are separated) in the embedding space.

3.5 Other methodologies and directions

Until mid-2025, the current time at which this dissertation is written, there are no relevant contributions to the WSOD methods based on the MIL-OICR framework. However, we start to see some research interest in exploring new methodological directions by the research community. Some of these methods are briefly discussed next.

Zhang, Wang e Ke (2025) introduced an approach that relies on one-point annotation per category to improve the performance of the detection task. In this case, this method lies in the class of methods called Point Supervised Object Detection (PSOD). The authors propose a pseudo-box generation module to generate corresponding pseudo-boxes for point annotations, which takes the annotated point as the centre to create boxes of different sizes, similar to the anchors of (REN et al., 2015). A MIL module is used to classify all of the box proposals, and they are merged with the boxes of the highest scores to get the pseudo box. The authors also propose a dense instance mining module to recall the unlabeled instances, which is based on the features of instances that are very similar in the image. However, this novel methodology does not achieve significant results in VOC and Common Objects in Context (COCO) datasets, which could justify this additional annotation cost.

In a completely different direction, Chen et al. (2025) introduced a method that relies on transformers to learn the task of Weakly Supervised Object Localization. More specifically, the authors rely on the CLIP model (RADFORD et al., 2021), which is trained on a massive dataset of image-text pairs. The authors proposed a CLIP-Driven Transformer framework for weakly supervised object localisation, which includes: a Category-aware Stimulation Module designed to embed category-specific insights into self-attention maps across transformer blocks, an Object Constraint Module to refine the object regions within the category-aware attention maps in a self-supervised manner, a Semantic Kernel Integrator to synergistically connect the previous methods and enhance category-aware, and a Semantic Boost Adapter to introduce stronger priors, leveraging the CLIP model to augment the semantic perception capabilities further. Although the methodology is interesting, it utilises datasets that are not employed in the WSOD context, which does not allow for comparisons between the methods. However, this seems to be an exciting direction of research for future WSOD methodologies.

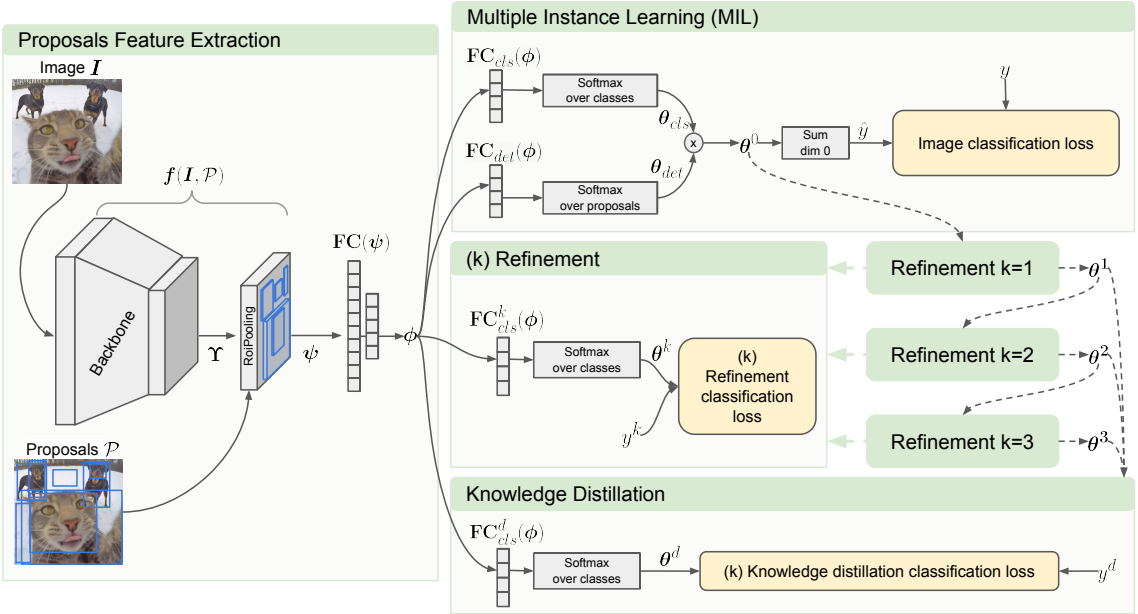
More recently, foundation segmentation models such as the Segment Anything Model (SAM) (KIRILLOV et al., 2023) and its successor SAM2 (RAVI et al., 2024) have attracted attention as a potential auxiliary tool for WSOD. SAM is trained on a large-scale dataset with diverse mask annotations and is designed to generate high-quality object masks from minimal prompts, such as points, boxes, or coarse masks. In the WSOD context, SAM has not been proposed as a standalone detection framework, but rather as a complementary component to refine object localisation, generate pseudo-labels, or improve proposal quality. Some works have explored using SAM to post-process coarse WSOD predictions, convert bounding-box hypotheses into segmentation masks, or filter region proposals based on mask consistency (ZHU et al., 2024; TAHERKHANI; KAZEMI, 2024). While this integration can significantly improve localization quality, it also introduces strong priors learned from external data, raising questions about supervision leakage and comparability with traditional WSOD methods. In the WSOD context, these models should be viewed as enabling tools that enhance proposal generation or localisation refinement rather than replacing core WSOD learning frameworks such as MIL-based approaches. While their integration often leads to notable performance gains, the strong external supervision and increased computational requirements pose challenges for fair evaluation and highlight an ongoing tension between methodological purity and practical performance in weakly supervised object detection.

4 DISTILLING SELF-SUPERVISION FROM THE REFINERS WITH AN ADAPTIVE SUPERVISION AGGREGATION FUNCTION

As discussed in Chapter 1, one problem that remains unsolved in Weakly Supervised Object Detection (WSOD) methods is the poor fitting of the detected bounding boxes, where the detections produced do not completely encompass the target categories. In addition, for certain classes, it often focuses on the most distinctive aspect of the object. To address this deficiency, in this chapter, we present two enhancements to the self-supervision process of a WSOD method based on the Multiple Instance Learning Network (MIL)-Online Instance Classification Refinement (OICR) framework.

The first improvement addresses the problem of incomplete object coverage prevalent in WSOD methods by introducing an innovative adaptive aggregation function that dynamically adjusts the Intersection over Union (IoU) selection threshold used during the creation of the refiner's self-supervision. The second improvement focuses on exploring the knowledge that the model refiners have already acquired to distil new information for self-supervision, enabling the training of an additional refiner module with the objective of "boosting" the detection outcomes.

Figure 4.1 – The proposed architecture and its modules. The proposals feature extraction module uses an Spatial Pyramidal Pooling (SPP) layer to extract features from proposals generated by the selective search. The multiple instance detection network module learns to select the best proposal instance and generates an image classification score. The instance refinement modules have k instances, and each one learns to refine instances from its predecessor's result. Finally, the knowledge distillation module aggregates all the knowledge learned by all the K refinements.



Source: Author's own illustration.

The following sections of this chapter are organised in the following manner. The specifics of the proposed methodology and its two key contributions are presented in Section 4.1. In the sequence, Section 4.2 presents in detail the results of our experiments. Finally, in Section 4.3, we synthesise our findings, offer our conclusions, and discuss potential directions for future research.

4.1 Method Overview

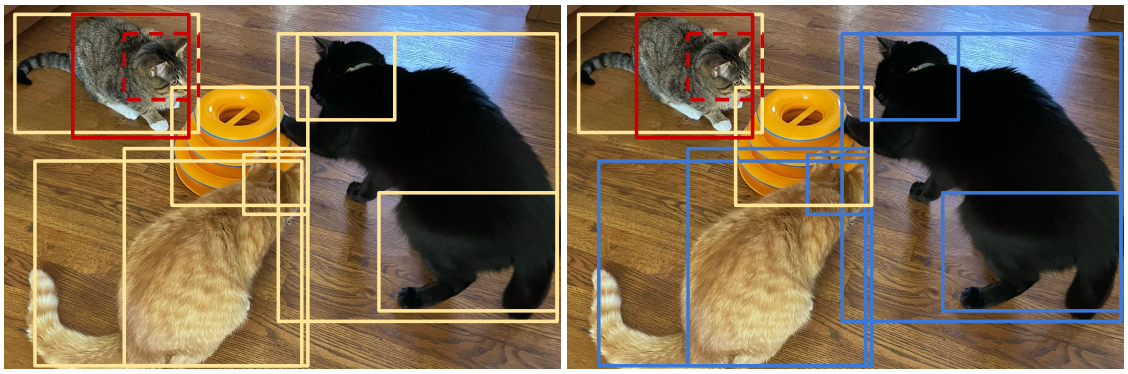
The proposed method, named *Boosted Online Instance Classification Refinement (BOICR)*, is built on top of the MIL-OICR framework described in detail in Section 2.2. To enhance comprehension of the chapter's contributions, it is advisable to consult the referenced section first. A high-level diagram of the proposed model is illustrated in Figure 4.1. It begins by extracting feature vectors at the proposal level from an input image and object proposals generated by the Selective Search (UIJLINGS et al., 2013). Next, the MIL module (BILEN; VEDALDI, 2016) is used to create the first scoring matrix in the methodology, which is optimised by a classification loss guided by the image-level annotations. Then, the first OICR (TANG et al., 2017) refiner ($k = 1$) module uses the MIL scoring matrix to build its self-supervision by using the top-scoring proposals as a starting point to select more positive instances for the target categories. The next refiners ($k > 1$) utilise the scoring matrix of the previous refiner ($k - 1$) to construct their self-supervision.

Examining the chain of refiners, one can infer that the scoring matrix of refiner $k = 1$ will be slightly different from the later refiners $k > 1$ since each refiner will learn a bit more about the problem in the sequence. The further away the refiners are in the chain, the greater this difference can be. Additionally, since the refiners are positioned in sequential order, the latter refiners do not directly impact the learning process of the previous refiners. If a refiner in the chain hallucinates, it will propagate the error to subsequent modules. Keeping this context in mind, we propose a new module, which is presented as Knowledge Distillation in the architecture diagram (Figure 4.1). This module aims to consolidate and normalise all the knowledge learnt by the K refiners. The Knowledge Distillation module employs the same layers as a standard refiner. However, it builds self-supervision by combining the scoring matrices of all K refiners in the model. It is important to note that using the term knowledge distillation in this context slightly abuses the conventional terminology, as the proposed module does not follow the clas-

sical teacher–student distillation paradigm commonly found in the literature (HINTON; VINYALS; DEAN, 2015). Instead, it refers to an internal knowledge transfer mechanism that allows the module to leverage and aggregate information from the model's refinement modules during training.

Additionally, we propose incorporating an adaptive supervision aggregation function into the refiner's self-supervision creation process. This function dynamically changes the IoU threshold used to aggregate the proposals as target categories, background, or to be ignored in the refiner's self-supervision building process. Our aggregation process follows the principle that at the beginning of training, it is better to aggregate boxes using a larger IoU threshold because the top-scoring instances are typically small and encompass a small portion of the objects of interest in the initial training iterations. To prevent an overgrowth of object-related proposals, the IoU threshold is tightened as the training phase progresses. We also included a modified adaptive version of the "trick" proposed by Tang et al. (2018), which ignores boxes with a small intersection in refinement losses.

Figure 4.2 – A visual comparison of the building process of a refiner's self-supervision to the "cat" class, when (a) using the standard OICR supervision scheme and (b) including the "trick" to ignore the "background" proposals with low intersection in the loss function. The red dashed box is the selected top-scoring instance that guides self-supervision. The solid red boxes are those selected that also belong to the cat class, the yellow rectangles are proposals associated with background, and the blue ones present boxes that are set to be ignored in the loss function by the "trick" in the self-supervision.



(a) Traditional MIL-OICR self-supervision.

(b) Including the "trick".

Source: Author's own illustration.

In Figure 4.2, we provide a comprehensive demonstration of the mechanism by which the so-called "trick" operates, particularly when several occurrences belonging to the "cat" category occur within a single training image instance. A detailed explanation of the Knowledge Distillation module is available in Section 4.1.1. Furthermore, an in-depth description of the adaptive aggregation function can be found in Section 4.1.2.

4.1.1 Knowledge Distillation

We can interpret the Knowledge Distillation module as a special kind of refiner that builds its self-supervision by using all the K refinement outputs as supervision. Similarly to the OICR's refiners, it performs instance classification by mapping ϕ to $\boldsymbol{\theta}^d = \mathbf{FC}^d_{cls}(\phi) \in \mathbb{R}^{P\times(C+1)}$ using a fully connected layer followed by a softmax over the class dimension. What differs from traditional refiners is that the Knowledge Distillation module builds its self-supervision by combining the scoring matrix of all the refiners in the model. More precisely, the scoring matrix used to construct the module's self-supervision is computed by averaging the scoring matrices $\boldsymbol{\theta}^k$ of the K refinement modules:

$$\hat{\boldsymbol{\theta}}^d_{sup} = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{\theta}^k_{i,c}. \tag{4.1}$$

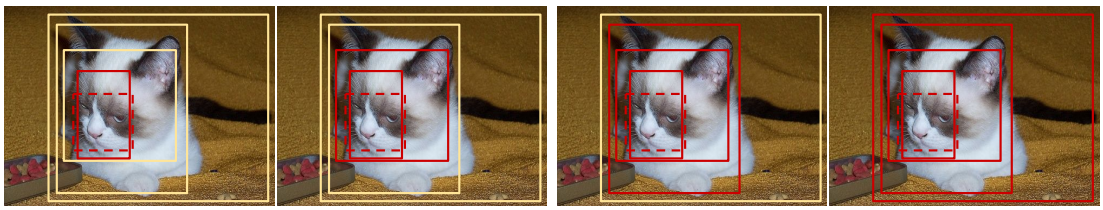
The Knowledge distillation module uses $\hat{\boldsymbol{\theta}}^d_{sup}$ to select the top-scoring proposals to guide its self-supervision building process by following the procedure that will be defined in Section 4.1.2. Its loss function L_{distil} is formulated similarly as a regular refiner:

$$\mathcal{L}_{distil} = \frac{1}{P}\sum_{i=1}^{P}\frac{1}{C+1}\sum_{c=1}^{C+1} w^d_{i,c} y^d_{i,c}\log\boldsymbol{\theta}^d_{i,c}, \tag{4.2}$$

where w^d is the weight regularize of the loss and $y^d_{i,c}$ is the self-supervision "ground truth".

4.1.2 Adaptive supervision aggregation function

Figure 4.3 – Effect of changing the IoU threshold λ for instance selection. Red slashed boxes denote the top-scoring proposal, red boxes pass the threshold (selected), and yellow boxes are set to background (not selected).



$\lambda = 0.5$ $\lambda = 0.25$ $\lambda = 0.1$ $\lambda = 0.01$

Source: Author's own illustration.

In OICR (TANG et al., 2017), the authors experimentally chose $\lambda = 0.5$ as the proposal selection IoU threshold in Equation (2.9) to create the matrices of weight regulation $w_{i,c}^k$ and supervision $y_{i,c}^k$. The interpretation of this value is that only boxes with $IoU > 0.5$ w.r.t. the best overall proposal are selected as belonging to the positive target categories c. The problem with using a fixed λ value is that, at the beginning of training, the top-scoring proposals tend to cover only a small part of the target objects, typically related to the most discriminant features of the objects (e.g., the face of a person or animal, as illustrated in Figure 4.3). Consequently, only other small boxes that have an $IoU > 0.5$ w.r.t. the top-scoring boxes are selected for the target category c. Figure 4.3 illustrates the impact of varying λ when the top-scoring proposal only covers partially the object of interest, with green boxes representing the top-scoring proposal, blue ones the proposals that align with the chosen threshold, and red the ones associated with the background class.

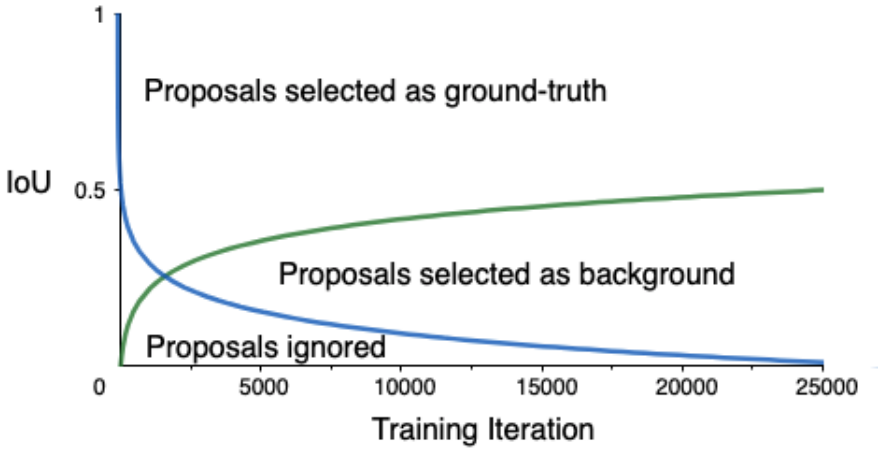
Although the motivation behind the refinement modules is to gradually improve the detection outcomes by learning broader parts of the objects, starting with a high value for λ causes each training iteration step to highlight only small boxes at the beginning of the process. In some cases, the optimisation could get stuck in small boxes during the whole training (especially for deformable objects). Relaxing λ alleviates this problem, but it also tends to include proposals with parts that are outside the boundaries of the target objects.

Instead of using a fixed value for λ, we propose using an adaptive selection function that changes the λ value as training iterations progress. This function should increase monotonically, so that more candidates are aggregated at the beginning and fewer at the end. During our experiments, we evaluated a set of different adaptive supervision aggregation functions, and the best results were achieved using the following function, also explored by Continuation Multiple Instance Learning (C-MIL) (WAN et al., 2019a) in a different context:

$$\lambda = \frac{1}{2} \frac{\log(s + l_b) - \log l_b}{\log(S + l_b) - \log l_b}, \tag{4.3}$$

where s is the current training step, S is the total of training steeps, and l_b defines the velocity that the curve grows. This function presents a smooth monotonic growth with the training step s, starting with a relatively high aggregation tolerance and gradually transitioning to stricter selection near the end of training. The logarithmic form provides a slow initial increase followed by a more pronounced growth, which empirically balances the aggregation more effectively than linear or step-based schedules.

Figure 4.4 – A visual interpretation of the proposed adaptive supervision aggregation function. X-axis shows the iteration step number, and Y-axis shows the IoU with the box of the highest score.



Source: Author's own illustration.

Another deficiency of the supervision selection approach, as given by Equation (2.9), is that when more than one instance of a class is present in the image, it will associate all other instances with the background class during the self-supervision building process (see Figure 4.2a). Since, in general, their IoU with the top-scoring instance will be small. This can be seen as a design flaw, as we do not want to optimise these positive instances as belonging to the background class in the loss function.

One solution to alleviate this problem in model optimisation is to include the so-called "trick" proposed by Tang et al. (2018), where a threshold value λ_{ign} is used to ignore boxes with a low IoU w.r.t. the top-scoring proposal of class c in the loss by setting $w_{i,c}^{k} = 0$ to the proposals p that satisfy $\text{IoU}(p_{\bar{ts}_c^k}, p_i) \leq \lambda_{ign}$. The effects of applying the trick in Figure 4.2a are displayed in Figure 4.2b, where a significant part of the proposals that belong to the other cat instances and were associated with the background class (in yellow) are set to be ignored (in blue) in the loss. Therefore, including the "trick" in the self-supervision building process helps avoid penalising almost all other instances of the target categories, and this is a better design choice.

In contrast to (TANG et al., 2018), which uses a fixed value for λ_{ign}, we propose using an adaptive threshold. Although the choice for λ_{ign} could be independent from λ, we propose a "complementary" threshold selection scheme given by

$$\lambda_{ign} = \lambda_{max} - \lambda, \tag{4.4}$$

where λ_{max} defines the starting point of the adaptive "trick".

Figure 4.4 presents a visual interpretation of how λ and λ_{ign} values change in the self-supervision process as the training evolves. Thus, we can adapt Equation (2.9) to include the trick as follows:

$$y_{i,c}^{k} = \begin{cases} 1 & \text{if IoU}(p_{\bar{ts}_c^k}, p_i) \geq \lambda \text{ and } y_c = 1 \\ 0 & \text{if } \lambda_{ign} \geq \text{IoU}(p_{\bar{ts}_c^k}, p_i) \geq \lambda \\ -1 & \text{otherwise} \end{cases}, \tag{4.5}$$

where -1 defines indices to be ignored in loss function.

4.1.3 The global loss function

As the MIL, OICR's refinement and knowledge distillation modules present individual loss functions, we combine them into a single loss function to train the model, given by:

$$L = L_{cls} + L_{distil} + \sum_{k=1}^{K} L_{cls}^{k}. \tag{4.6}$$

.

4.2 Experiments

The proposed contributions are evaluated on the datasets Pascal Visual Object Classes (VOC) 2007 (EVERINGHAM et al., 2007) and VOC 2012 (EVERINGHAM et al., 2012). At the time this method was developed (2020), evaluation on the VOC benchmarks was the standard protocol adopted by the WSOD literature, while Common Objects in Context (COCO) was not yet commonly used in this context. Moreover, the substantially larger scale and complexity of COCO, combined with the hardware limitations available at the time, made it difficult to train the model on COCO. However, in Chapter 6, with better hardware at hand, we perform evaluations on COCO 2014.

Although bounding box annotations are available in VOC 2007 and VOC 2012 datasets, during training, we only use image-level annotations, which indicate the presence or absence of each target category. In contrast, during testing, the annotated bounding boxes are used to calculate the two standard metrics in WSOD, namely, detection Mean Average Precision (mAP) (EVERINGHAM et al., 2012) and Correct Localiza-

tion (CorLoc) (DESELAERS; ALEXE; FERRARI, 2012). The former provides a measure of how well the detector learns to detect all instances in the images, while the latter indicates if the best detection is a proper match. Both metrics use VOC criteria of $IoU > 0.5$ between ground truths and predicted boxes and are described in detail in Section 2.4.

4.2.1 Implementation Details

The implementation of the model[1] and the experiments carried out were implemented using PyTorch 1.2. To extract features from the images, we employ the Convolutional Neural Networks (CNN) blocks of the VGG16 (SIMONYAN; ZISSERMAN, 2014) initialised with weights pre-trained on ImageNet (DENG et al., 2009). The last max-pooling layer is replaced by the SPP layer, and the weights of the first two Fully Connected (FC) layers in VGG16 initialise the $\mathbf{FC}(.)$ layers that extract image-level proposal features. The remaining layers are initialised using Gaussian distributions with a 0-mean and a standard deviation of 0.01. The biases are initialised with 0.

The object proposals are extracted using Selective Search (UIJLINGS et al., 2013). For data augmentation, the training images are transformed into a total of ten augmented versions. The training samples were resized on five scales corresponding to the smallest dimension ($\{480, 576, 688, 864, 1200\}$), and were horizontally flipped. The learning process was performed using the Stochastic Gradient Descent (SGD) algorithm with a momentum of 0.9, a weight decay of $5e^{-4}$, and a batch size of 2. We set $l_b = 100$ and $\lambda_{max} = 0.51$. For VOC 2007 and VOC 2012, the learning rate begins at 0.001 for the first 30K and 60K iterations, subsequently reducing to 0.0001 in the next 20K and 30K iterations, respectively.

During testing, the 10 augmented images are fed through the model and their outputs are averaged. As an additional result, as done in (TANG et al., 2017; TANG et al., 2018; WAN et al., 2019a), we also trained a Fast R-CNN (FRCNN) (GIRSHICK, 2015) object detector using the detections created by our method as supervision. The supervision is built by selecting detections with a score greater than 0.3 and by applying non-maxima suppression with a 30% IoU threshold.

[1] Source code available at github.com/luiszeni/Boosted-OICR

4.2.2 Ablation experiments

We perform ablation experiments to demonstrate the effectiveness of the proposed improvements over the baseline MIL-OICR method (TANG et al., 2017). We first study the impact of using the adaptive supervision aggregation function instead of fixed IoU thresholds for proposal mining. We show the different scenarios in Table 4.1.

Table 4.1 – Ablation study performance (%) on the VOC 2007 in different scenarios.

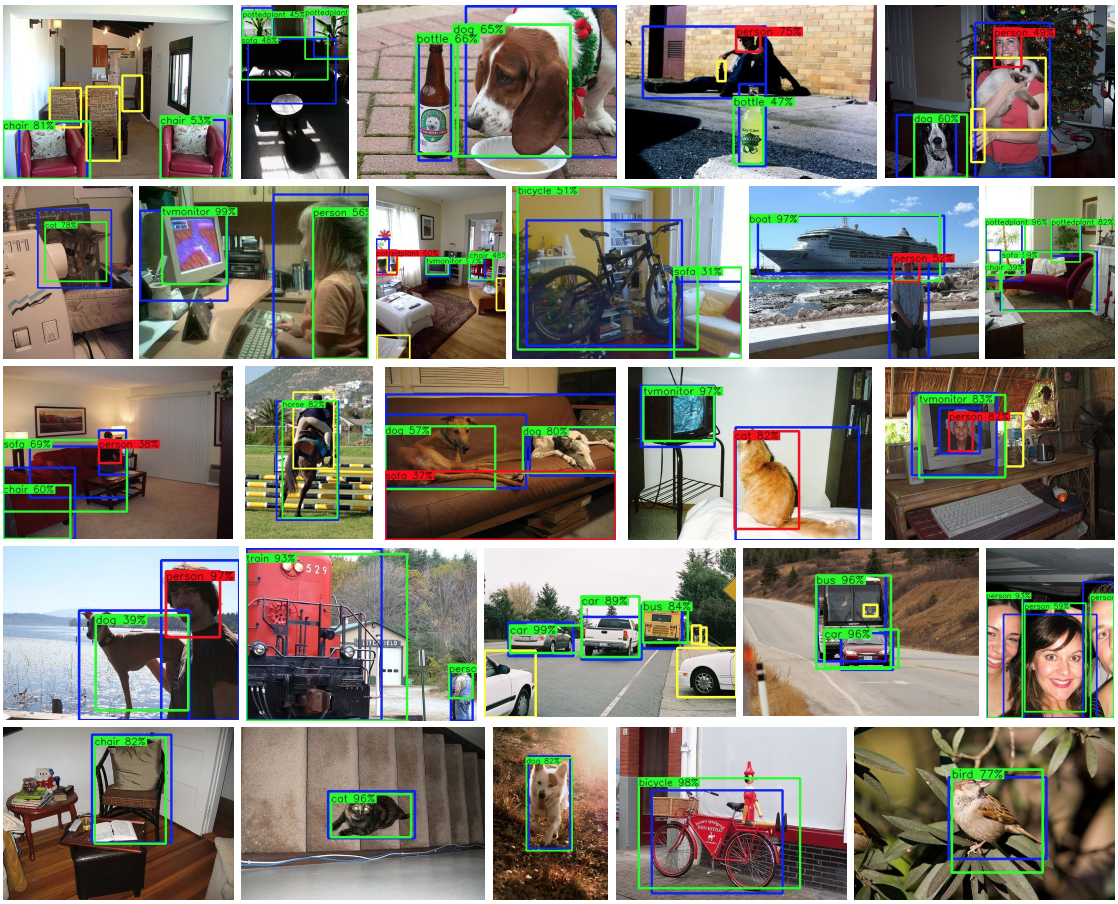
ID	K	λ	λ_{ign}	distillation	mAP
1	3	0.5	0	No	42.3
2	3	adaptive	0	No	41.6
3	3	adaptive	adaptive	No	46.6
4	3	adaptive	adaptive	Yes	**49.7**
5	4	adaptive	adaptive	No	48.1

Source: Author's compiled data.

The experiment with $ID = 1$ presents the results using the baseline model. In experiment $ID = 2$, we replace the fixed λ value with the proposed adaptive aggregation function defined in Equation (4.3). In this experiment, all boxes with $IoU < \lambda$ are set to the background class. As the experiment suggests, using the adaptive supervision aggregation function alone, without the adaptive trick, yields results slightly worse than the baseline. In contrast, in experiment $ID = 3$, we incorporate the adaptive "trick", resulting in a 4.3% mAP improvement over the baseline, indicating that the use of our adaptive supervision aggregation function can significantly enhance the detection results.

When incorporating the knowledge distillation module, the experiment with $ID = 4$ yields a 7.4% mAP increase in mAP compared to the baseline and a 3.0% mAP increase compared to $ID = 2$. Even so, one could argue that using the knowledge distillation module could produce the same result as incorporating an additional refinement module into the model. To refute this claim, the experiment with $ID = 5$ presents the results of adding one more refiner with the adaptive function. The knowledge distillation module has an advantage of 1.6% mAP over including an additional refiner, demonstrating the effectiveness of the proposed strategy in enhancing the mAP performance. For the following experiments, we select the model used in experiment $ID = 4$ to conduct subsequent comparisons.

Figure 4.5 – Detection examples for the VOC 2007 dataset. Blue rectangles are ground-truth boxes that have at least one detection with $IoU > 0$, and yellow ones are ground-truth boxes with no detection intersection. Green boxes are correct detections ($IoU > 0.5$ with ground truth), and red boxes are wrong detections. The label in each detection box is the class label and confidence score of the detection.



Source: Author's own illustration.

4.2.3 Comparison with the state-of-the-art

In this section, we compare the results of the proposed approach with the other State-of-the-Art (SOTA) methods in the datasets VOC 2007 and VOC 2012. The methodology developed in this chapter was published in 2020 (ZENI; JUNG, 2020b). Therefore, to maintain a fair comparison, the results will be evaluated against methods from the same period (in Chapter 6, we evaluated the method proposed there against the latest WSOD techniques).

Table 4.2 shows a comparison of the detection performance of our method versus SOTA in the VOC 2007 test set. The table shows that BOICR improves the original OICR paper (TANG et al., 2017) in 7.7% mAP and outperformed other approaches such as WCCN (DIBA et al., 2017) (6.9%), TS2C (WEI et al., 2018) (5.4%), WeakRPN (TANG

et al., 2018) (4.4%), PCL (TANG et al., 2018) (6.2%), and MELM (WAN et al., 2019b) (2.4%). BOICR was only inferior to C-MIL (WAN et al., 2019a) by a small value (0.8% mAP). However, BOICR presented the highest Average Precision (AP) results in 9 of the total 20 classes (`aeroplane`, `bird`, `bottle`, `bus`, `car`, `dog`, `motorbike`, `train` and `tv`). Figure 4.5 presents some detection results generated by our method. We also trained a FRCNN object detector using the detections created by our method as ground truth. It achieves a mAP of 50%, which improved the results only by 0.3%.

Table 4.2 – Detection performance (%) on the VOC 2007 test set. Comparison with state-of-the-art approaches.

Network	Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
VGG16	WSDDN (BILEN; VEDALDI, 2016)	46.4	58.3	35.5	25.9	14.0	66.7	53.0	39.2	8.9	41.8	26.6	38.6	44.7	59.0	10.8	17.3	40.7	49.6	56.9	50.8	39.2
	OICR (TANG et al., 2017)	58.0	62.4	31.1	19.4	13.0	65.1	62.2	28.4	24.8	44.7	30.6	25.3	37.8	65.5	15.7	24.1	41.7	46.9	64.3	62.6	42.0
	WCCN (DIBA et al., 2017)	49.5	60.6	38.6	29.2	16.2	70.8	56.9	42.5	10.9	44.1	29.9	42.2	47.9	64.1	13.8	23.5	45.9	54.1	60.8	54.5	42.8
	TS2C (WEI et al., 2018)	59.3	57.5	43.7	27.3	13.5	63.9	61.7	59.9	24.1	46.9	36.7	45.6	39.9	62.6	10.3	23.6	41.7	52.4	58.7	56.6	44.3
	WeakRPN (TANG et al., 2018)	57.9	70.5	37.8	5.7	21.0	66.1	69.2	59.4	3.4	57.1	**57.3**	35.2	64.2	68.6	32.8	28.6	50.8	49.5	41.1	30.0	45.3
	PCL (TANG et al., 2018)	54.4	69.0	39.3	19.2	15.7	62.9	64.4	30.0	25.1	52.5	44.4	19.6	39.3	67.7	17.8	22.9	46.6	57.5	58.6	63	43.5
	MELM (WAN et al., 2019b)	55.6	66.9	34.2	29.1	16.4	68.8	68.1	43.0	25.0	65.6	45.3	53.2	49.6	68.6	2.0	25.4	52.5	56.8	62.1	57.1	47.3
	C-MIL (WAN et al., 2019a)	62.5	58.4	49.5	32.1	19.8	70.5	66.1	63.4	20.0	60.5	52.9	53.5	57.4	68.9	8.4	24.6	51.8	**58.7**	66.7	63.5	50.5
	PG-PS (CHENG et al., 2020)	63.0	64.4	50.1	27.5	17.1	70.6	66.0	**71.1**	25.8	55.9	43.2	**62.7**	**65.9**	64.1	10.2	22.5	48.1	53.8	**72.2**	67.4	51.1
	SLV (CHEN et al., 2020)	65.6	71.4	49.0	**37.1**	24.6	69.6	70.3	70.6	30.8	63.1	36.0	61.4	65.3	68.4	12.4	**29.9**	52.4	60.0	67.6	64.5	53.5
	MIST (REN et al., 2020)	**68.8**	**77.7**	**57.0**	27.7	**28.9**	69.1	**74.5**	67.0	**32.1**	**73.2**	48.1	45.2	54.4	**73.7**	**35.0**	29.3	**64.1**	53.8	65.3	65.2	**54.9**
	Ours	68.6	62.4	55.5	27.2	21.4	**71.1**	71.6	56.7	24.7	60.3	47.4	56.1	46.4	69.2	2.7	22.9	41.5	47.7	71.1	**69.8**	49.7
FRCNN Re-train	OICR (TANG et al., 2017)	65.5	67.2	47.2	21.6	22.1	68.0	68.5	35.9	5.7	63.1	49.5	30.3	64.7	66.1	13.0	25.6	50.0	57.1	60.2	59.0	47.0
	TS2C (WEI et al., 2018)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	48.0
	PCL (TANG et al., 2018)	63.2	69.9	47.9	22.6	27.3	71.0	69.1	49.6	12.0	60.1	51.5	37.3	63.3	63.9	15.8	23.6	48.8	55.3	61.2	62.1	48.8
	WeakRPN (TANG et al., 2018)	63.0	69.7	40.8	11.6	**27.7**	70.5	**74.1**	58.5	10.0	**66.7**	**60.6**	34.7	**75.7**	**70.3**	**25.7**	**26.5**	**55.4**	56.4	55.5	54.9	50.4
	C-MIL (WAN et al., 2019a)	61.8	60.9	**56.2**	28.9	18.9	68.2	69.6	71.4	18.5	64.3	57.2	66.9	65.9	65.7	13.8	22.9	54.1	61.9	68.2	66.1	53.1
	PG-PS (CHENG et al., 2020)	59.3	66.2	55.4	**35.2**	22.3	69.7	70.2	73.8	**29.4**	63.6	47.9	**78.1**	67.9	68.2	12.2	24.9	43.2	**63.7**	**73.2**	**66.8**	**54.6**
	SLV (CHEN et al., 2020)	62.1	**72.1**	54.1	34.5	25.6	66.7	67.4	**77.2**	24.2	61.6	45.5	71.6	72.0	67.2	12.1	24.6	51.7	61.1	65.3	60.1	53.9
	Ours	**65.8**	58.6	55.0	32.4	19.5	**74.2**	71.4	70.9	19.2	54.8	46.2	67.5	57.0	65.6	1.4	16.7	40.4	53.0	69.5	61.1	50.0

Source: Author's compiled data.

Table 4.3 presents a comparison of the localisation (CorLoc) performance of our method versus SOTA in the VOC 2007 train-val set. BOICR outperformed OICR (TANG et al., 2017) (5.1%), WCCN (DIBA et al., 2017) (9.0%), TS2C (WEI et al., 2018) (4.7%), WeakRPN (TANG et al., 2018) (1.9%), PCL (TANG et al., 2018) (3.0%), MELM (WAN et al., 2019b) (4.3%), and C-MIL (WAN et al., 2019a) (0.7%). The better CorLoc result of our method in comparison with C-MIL suggests that C-MIL is a bit better at dealing with images with more than one instance (which affects the final detection mAP).

Table 4.3 – Localisation (CorLoc) performance (%) on the VOC 2007 trainval set. Comparison with state-of-the-art approaches.

Network	Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
VGG16	WSDDN (BILEN; VEDALDI, 2016)	65.1	58.8	58.5	33.1	39.8	68.3	60.2	59.6	34.8	64.5	30.5	43.0	56.8	82.4	25.5	41.6	61.5	55.9	65.9	63.7	53.5
	OICR (TANG et al., 2017)	81.7	80.4	48.7	49.5	32.8	81.7	85.4	40.1	40.6	79.5	35.7	33.7	60.5	88.8	21.8	57.9	76.3	59.9	75.3	81.4	60.6
	WCCN (DIBA et al., 2017)	83.9	72.8	64.5	44.1	40.1	65.7	82.5	58.9	33.7	72.5	25.6	53.7	67.4	77.4	26.8	49.1	68.1	27.9	64.5	55.7	56.7
	TS2C (WEI et al., 2018)	84.2	74.1	61.3	52.1	32.1	76.7	82.9	66.6	42.3	70.6	39.5	57.0	61.2	88.4	9.3	54.6	72.2	60.0	65.0	70.3	61.0
	WeakRPN (TANG et al., 2018)	77.5	81.2	55.3	19.7	44.3	80.2	86.6	69.5	10.1	**87.7**	**68.4**	52.1	84.4	91.6	**57.4**	**63.4**	77.3	58.1	57.0	53.8	63.8
	PCL (TANG et al., 2018)	79.6	**85.5**	62.2	47.9	37.0	**83.8**	83.4	43.0	38.3	80.1	50.6	30.9	57.8	90.8	27.0	58.2	75.3	**68.5**	75.7	78.9	62.7
	MELM (WAN et al., 2019b)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	61.4
	C-MIL (WAN et al., 2019a)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	65.0
	PG-PS (CHENG et al., 2020)	85.4	80.4	69.1	58.0	35.9	82.7	86.7	**82.6**	45.5	84.9	44.1	**80.2**	84.0	89.2	12.3	55.7	**79.4**	63.4	82.1	82.1	69.2
	SLV (CHEN et al., 2020)	84.6	84.3	73.3	**58.5**	**49.2**	80.2	87.0	79.4	46.8	83.6	41.8	79.3	**88.8**	90.4	19.5	59.7	**79.4**	67.7	**82.9**	**83.2**	**71.0**
	MIST (REN et al., 2020)	**87.5**	82.4	**76.0**	58.0	44.7	82.2	**87.5**	71.2	**49.1**	81.5	51.7	53.3	71.4	**92.8**	38.2	52.8	**79.4**	61.0	78.3	76.0	68.8
	Ours	86.7	73.3	72.4	55.3	46.9	83.2	**87.5**	64.5	44.6	76.7	46.4	70.9	67.0	88.0	9.6	56.4	69.1	52.4	79.8	82.8	65.7

Source: Author's compiled data.

We also compare the detection and localisation (CorLoc) performance of our method in Pascal VOC 2012 in Table 4.4. In detection BOICR improved the OICR (TANG et al., 2017) results in 8.8% mAP and outperformed other approaches such as WCCN (DIBA

et al., 2017) (8.8%), TS2C (WEI et al., 2018) (6.7%), WeakRPN (TANG et al., 2018) (5.9%), PCL (TANG et al., 2018) (6.1%), MELM (WAN et al., 2019b) (4.3%) and C-MIL (WAN et al., 2019b) (0.1%). BOICR also presents a competitive CorLoc in VOC 2012 outperforming OICR (TANG et al., 2017) (4.2%), TS2C (WEI et al., 2018) (1.9%), WeakRPN (TANG et al., 2018) (1.4%) and PCL (TANG et al., 2018) (3.1%), being inferior to C-MIL (WAN et al., 2019a) by 1.1% mAP.

Table 4.4 – Detection in test set and localisation (CorLoc) in trainval set performance (%) on the VOC 2012 dataset using VGG16.

Method	mAP	Corloc
WCCN (DIBA et al., 2017)	37.9	-
OICR (TANG et al., 2017)	37.9	62.1
TS2C (WEI et al., 2018)	40.0	64.4
WeakRPN (TANG et al., 2018)	40.8	64.9
PCL (TANG et al., 2018)	40.6	63.2
MELM (WAN et al., 2019b)	42.4	-
C-MIL (WAN et al., 2019a)	46.6	67.4
PG-PS (CHENG et al., 2020)	48.3	68.7
SLV (CHEN et al., 2020)	49.2	69.2
MIST (REN et al., 2020)	**52.1**	**70.9**
Ours	46.7	66.3

Source: Author's compiled data.

4.3 Conclusions

In this chapter, we present two improvements to "boost" the MIL-OICR framework. The initial enhancement focuses on the issue of partial object coverage, commonly encountered in WSOD techniques, by implementing a novel adaptive aggregation mechanism. This mechanism dynamically modifies the IoU selection threshold, which is a crucial step in building the refiner's self-supervised process. The subsequent enhancement focuses on exploring the knowledge that the model refiners have already acquired to distil new information for self-supervision, enabling the training of an additional refiner module with the objective of "boosting" the detection outcomes.

Both contributions were built on top of the MIL-OICR framework, and using them we were able to have an increase of 7.4 mAP over the baseline. BOICR presents competitive SOTA results on VOC 2007 dataset, being inferior only to (WAN et al., 2019a) by a small margin (0.8% mAP). In contrast, our method is top of the ranking, showing an increase of 8.8% mAP over the MIL-OICR framework and (0.1%) when compared with C-MIL (WAN et al., 2019b).

Although the methodology outlined in this chapter improves the MIL-OICR baseline outcomes, it relies on the "trick" to address the challenge posed by having multiple instances of the same class within a single image. Even though the strategy of ignoring additional instances in the loss is clever and straightforward, removing these positive instances from the model's learning optimisation eliminates the opportunity for the model to learn more about the target classes with them. It is not hard to see that this is not the best choice when the problem is viewed from an optimisation perspective. In the subsequent chapters, we investigate the process of mining extra instances in problems where the number of object instances is inherent in weak annotations, as well as strategies for discovering additional instances in problems where no information is provided regarding the number of instances.

5 WEAKLY SUPERVISED CHARACTER DETECTION FOR LICENCE PLATE RECOGNITION TASK

A Licence Plate (LP) is a sign on road vehicles that displays its registration code (a string of characters). The main objective of an Licence Plate Recognition (LPR) method is to output the correct LP string. Existing approaches can be split into two main groups: techniques that depend on costly bound box character annotations as supervision (SILVA; JUNG, 2017; SILVA; JUNG, 2018; SILVA; JUNG, 2020; LAROCA et al., 2018; Laroca et al., 2019; BJÖRKLUND et al., 2019), and those that use only the plate string as supervision (CHENG et al., 2017; WANG et al., 2020; BAEK et al., 2019).

Methods that receive the bounding box as supervision tend to have better adaptation to different plate layouts and countries because they learn the appearance of each character separately. However, annotating bounding boxes is a costly process, which can make it unfeasible to train a model that works in multiple regions and contexts. In contrast, methods that use only the string as supervision learn a coarse localisation of the characters from the annotation string, tend not to generalise well to new plate layouts, and face problems when the plate contains two rows of characters (as in motorcycle LPs).

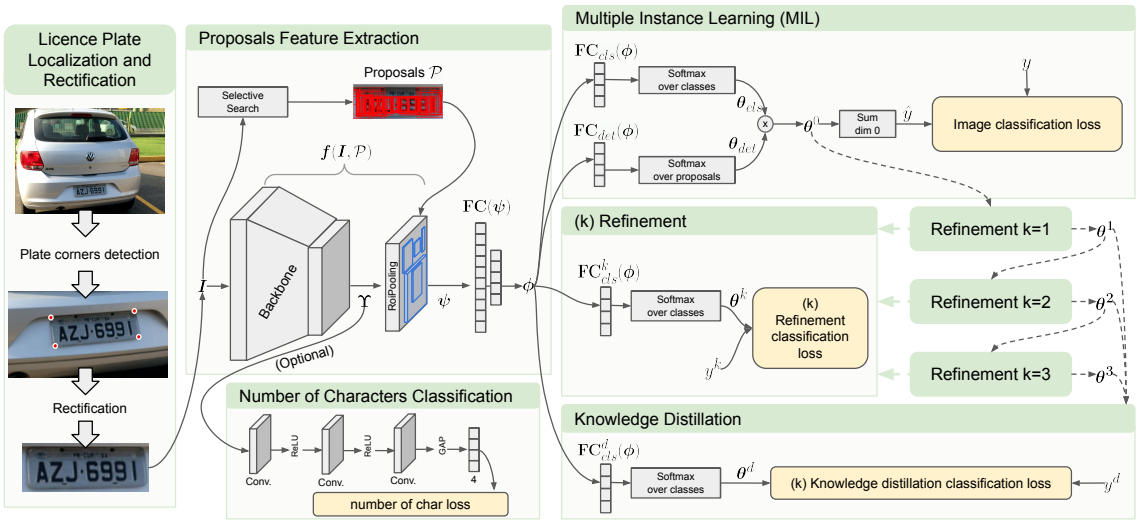
One special characteristic of LPR annotations is that the number of instances is given in the annotated string, for example, given the plate annotation string "AIA-3244", we can easily infer that there are two characters "A", two characters "4", and one instance for the "I", "2" and "3". In this context, if we were able to explore the number of character instances to train a Weakly Supervised Object Detection (WSOD) model, it could join the two advantageous parts of the methods existing in the literature. It will learn the LPR task using only string annotations, and, by employing the WSOD methodology, it can also provide spatial information (bounding box) regarding each character in the LP.

This chapter investigates the use of a WSOD approach for the LPR task, supporting an analysis of the outcomes when the LPR task is addressed as a Weakly Supervised Character Detection (WSCD). The following sections are organised in the following manner. The details of the proposed methodology and its contributions are presented in Section 5.1. In Section 5.2, we present the detailed results of our experiments, which were conducted using LP datasets from various regions, including Brazil, China, Taiwan, the United States, and the European Union. Finally, in Section 5.3, we synthesise our findings, offer our conclusions, and discuss potential directions for future research.

5.1 Method Overview

We present an overview of the proposed architecture and its modules in Figure 5.1. The input to our method is a "rectified" LP image that corresponds to (or emulates) a fronto-parallel view. In scenarios where it is already captured in a frontal pose, any approach based on bounding boxes can be used. However, some methods can detect the four corners of the plate (such as (XU et al., 2018; SILVA; JUNG, 2018; BJÖRKLUND et al., 2019)), so that a planar homography can be used to warp the detected LP to an approximately fronto-parallel view. To locate the four corners of the plate, we use the approach presented in (SILVA; JUNG, 2021), which is suitable for vehicle and motorcycle plates.

Figure 5.1 – An overview of the proposed architecture and its modules. The localisation and rectification module localises the corners of the LP. These points are used to apply an affine transformation, which reduces the perspective deformations of the plate in the image. The rectified image is fed into the proposal feature extraction module, which utilises an Spatial Pyramidal Pooling (SPP) layer to extract features from proposals generated by the Selective Search algorithm. The image features extracted using the Convolutional Neural Networks (CNN) backbone are used by the number of characters classification module to estimate the number of characters present in the plate. The multiple instance detection network module learns to select the best proposal instances and generates an image classification score. The refinement modules are cascaded into a sequence of K module instances, where each one learns to refine the results using its predecessor's scoring matrix to build its self-supervision process. Finally, the knowledge distillation module combines the knowledge learned by all the refiners to enhance the results.



Source: Author's own illustration.

Object proposals are extracted from the rectified images using selective search (UIJLINGS et al., 2013), making it possible to train the WSOD model described in Chapter 4 to learn the spatial location of the characters using only the plate string as supervision. As

the annotations give us the number of instances for free, we exploit this information by proposing two additional modifications to the WSOD model: (i) a novel instance-aware approach that takes advantage of number of instances to extract additional object instances during the refiner's self-supervision building process and (ii) a new module to estimate the number of characters present in the LP, which is fed by the backbone image features.

5.1.1 Instance-Aware Self-Supervision

To take advantage of the information regarding the number of instances of each category in an image, we propose some modifications to Equation (4.5) to accommodate this valuable information in the refiner's self-supervision building process. We refer to this process as Instance-Aware Self-Supervision (IASS). The overall concept is shown in Figure 5.2 for the character class "1." In this scenario, traditional Multiple Instance Learning Network (MIL)-Online Instance Classification Refinement (OICR) will penalise the other instances by setting them to the background class, and the Boosted Online Instance Classification Refinement (BOICR) disregards them in loss calculations. However, the instance-aware method introduced in this chapter designates both '1's as positives, as well as any overlapping proposal that passes the selection criteria.

Figure 5.2 – Comparison of the results of different strategies that build the refiner's self-supervision for the character class "1". Green denotes the top-scoring box(es), blue boxes are selected as also belonging to the class "1", red boxes are the background ones. Yellow boxes are set to be ignored in the loss function.



Traditional MIL-OICR | BOICR | Instance Aware

Source: Author's own illustration.

Consider that s_c is the number of instances for a given class c. If $s_c = 1$, we follow the same supervision process as in Equation (4.5), where proposals that have an Intersection over Union (IoU) with the top-scoring proposal greater than a threshold are selected as positive to category c. In the case where multiple occurrences of a category are present ($s_c > 1$), our objective is to have, in self-supervision, at least one positive proposal that overlaps with each instance of class c. Thus, we employ an iterative top-scoring

selection strategy, which starts by applying Non-maxima Suppression (NMS) using the proposals' bounding boxes p_i and their related class c score from the previous refiner's scores matrix $\boldsymbol{\theta}_{i,c}^{k-1}$. Applying the NMS will help avoid selecting multiple top-scoring proposals for the same character instance. And, since its results are already ordered, we can assume that the first s_c indexes $\bar{ts}_{c,1}^{k}, \bar{ts}_{c,2}^{k}, \cdots, \bar{ts}_{c,s_c}^{k}$ on NMS result are the indexes of the most probable top-scoring proposal that overlap with each instance of the category c. Using the indices, Equation (4.5) can be reformulated to accommodate multiple top-scoring proposals per class:

$$y_{i,c}^{k} = \begin{cases} 1 & \text{if IoU}(p_{\bar{ts}_{c,j}^{k}}, p_i) \geq \lambda \text{ and } y_c = 1 \\ 0 & \text{if } \lambda_{ign} \geq \text{IoU}(p_{\bar{ts}_{c,j}^{k}}, p_i) \geq \lambda \\ -1 & \text{otherwise} \end{cases}, \tag{5.1}$$

where $j \in [1, s_c]$ is the index of one of the selected top-scoring proposals.

5.1.2 Recovering the Total Number of Characters

In contrast to training time, during inference, the WSCD method has no information regarding the length of the LP's string (i.e., the total number of characters). During our experiments, we found that using a detection threshold alone can, in some cases, result in either more or fewer character detections than the actual number present on the plates. The number of characters in an LP typically depends on the country/region. For some areas (as in the case of Brazil), the number of characters is fixed. However, European and North American LPs present a variable number of characters, which should be taken into account in inference time. Therefore, having the total number of characters is very valuable information for accurately extracting the correct number of character detections on plates from different regions.

In this context, we propose the Number of Characters Classification (NCC) module, which is trained to recover this information during the inference time. Although this task could be tackled as a regression problem, the number of characters in an LP does not vary significantly (typically from 4 to 8). Hence, we adopt a classification-based approach. The objective of this module is to produce, as a result, the number of characters N^c in a given plate image. In this way, we can use it to select N^c top-scoring character detections in the post-processing part.

As shown in Figure 5.1, the module receives as input the image-level features Υ

extracted with the VGG16 backbone, which are then fed into two convolutional layers with 512 filters each and a kernel size of 3×3. After each convolutional layer, we add one Rectified Linear Unit (ReLU) activation function. In the sequence, a convolutional layer with T channels and a 1×1 kernel (without padding) is applied, where T denotes the number of output classes for character-number classification. For example, class t_1 corresponds to plates with four characters, t_2 to images with five characters, and so on. Lastly, a Global Average Pooling (GAP) layer followed by a softmax is used to generate a T dimensional score vector $\boldsymbol{\eta}$ for the number of characters in the LP. As this classification task is simple, we choose to use convolutional layers in place of Fully Connected (FC) layers, which require fewer parameters in the model, which means having a minor impact on the memory consumption of the model.

The ground-truth labels to train the module are built using the week annotation of the plate string, given by $y_t = 1$ if $c = \#\text{plate_string}$. For the loss function L_{num_char}, we choose the standard cross-entropy loss, which is widely used for classification problems, and include it in the global loss defined in Equation (4.6) as follows:

$$L = L_{wsod} + L_{num_char}, \tag{5.2}$$

where L_{wsod} represent the output of the loss defined in Equation (4.6).

5.1.3 Extracting the Licence Plate string from the detections

To extract the final LP string, we first combine the output of all the K refinements plus the distillation module, i.e., we compute the mean scores between them

$$\boldsymbol{\theta}^F_{i,c} = \frac{1}{K+1}(\boldsymbol{\theta}^d_{i,c} + \sum_{k=1}^{K} \boldsymbol{\theta}^k_{i,c}), \tag{5.3}$$

restricting the analysis to $c \in \{1, 2, \cdots, C\}$ (we discard the background class introduced in the refinement step). We then select the indices of the top-scoring class for each proposal

$$b_i = \arg\max_c \boldsymbol{\theta}^F_{i,c}, \tag{5.4}$$

and apply NMS to discard highly overlapping results, resulting in a set of pruned detection proposals b'_i, sorted in decreasing order w.r.t. the detection score. Next, we obtain the

predicted number of characters N from the vector $\boldsymbol{\eta}$ as

$$N = \arg\max_{n} \boldsymbol{\eta}_n, \tag{5.5}$$

and use it to select the N top-scoring proposals b'_i for $i = 1, 2, \cdots, N$. Note that if the number of characters in the LP is known *a priori* (for example, if we analyse only Brazilian LPs, we know that the LP will present $N = 7$ characters), then the prediction of the number of characters can be ignored. Finally, we sort the retrieved boxes in increasing order with respect to the horizontal coordinate x_{min} of the top left corner of the image and generate the LP string by concatenating the class predictions of the boxes.

To handle LPs that present two rows of characters (such as motorcycle LPs in Brazil), we first identify the detections related to each row. We first retrieve the central vertical coordinate v_i of each detected box and compute the mean value μ of v_i in a robust manner using an α-trimmed mean, which ignores the lowest and highest values (which may be related to outliers). The characters presenting v_i smaller than μ are then assigned to the first row, and the others to the second row. For each row, detections are scanned from left to right (as described before).

5.2 Experiments

We conducted our experiments using datasets with LPs from different regions, namely: Brazil, China, Taiwan, the United States, and the European Union. We chose to use only a subset of these regions for training our method and testing it using all regions. This allows us to investigate how well our method can generalise to unseen LP regions. We start by describing each dataset, its characteristics, and the modifications we made. Next, we performed an ablation experiment to compare the effects of our contributions to the MIL-OICR framework when applied to the LPR problem. Finally, we compare our model with other state-of-the-art methods.

5.2.1 Licence Plate Datasets

We used five different datasets in our experiments. Three of them were used for training and testing (AOLP, Fazenda, and CCPD), while the other two (OpenALPR and UFPR-ALPR) were used solely for testing the model. Due to the LP fonts, we used pairs

‘I-‘1’ and ‘0’-‘O’ as a single label each, yielding 34 classes for character detection. A brief description of each data set is provided below.

CCPD (XU et al., 2018) contains Chinese plates. We used 100,000 images for training and the 20,000 images from the validation set for testing.

AOLP (HSU; CHEN; CHUNG, 2012) contains Taiwanese plates. We used 1,162 images for training and 776 images for testing.

Fazenda is a private dataset of Brazilian [1] LPs, we used 34,584 images for training and 20,000 images for testing.

OpenALPR[2] contains 112 Brazilian, 173 American, and 107 European plates, which were used only for testing. As other datasets already include Brazilian plates, we retained only the American and European plates for this dataset.

UFPR-ALPR(Laroca et al., 2018) have Brazilian plates. We selected only the motorcycle subset for testing, which resulted in 783 images after running the Licence Plate Detector (LPD) method.

As mentioned before, we run the LPD method described in (SILVA; JUNG, 2021) to extract the four LP corners (except for CCPD, which already provides such annotation) and rectify them to a fronto-parallel view using a planar homography, keeping a 3×1 aspect ratio for cars and 1.17×1 for motorcycles. This process is utilised in both the training and testing phases of our method.

5.2.2 Implementation Details

All experiments were performed using PyTorch 1.2, using VGG16 (SIMONYAN; ZISSERMAN, 2014) pre-trained on ImageNet (DENG et al., 2009) as a backbone. The object proposals of the rectified images were extracted using Selective Search (UIJLINGS et al., 2013), and we filter out proposals that present a width larger than the height to remove false candidates. For the module that estimates the number of characters in the LP, we used five classes (4 to 8 characters).

For data augmentation, the rectified images were resized to five scales 480, 576, 688, 864, and 1200 with respect to the smallest dimension, keeping the aspect ratio of

[1] When we performed the experiments and released this study, in 2020, the implementation of the new Brazilian LP layout had just begun. Consequently, data sets featuring these “new” plates were non-existent, which is why we did not include these plates in our experiments.

[2] Available at <https://github.com/openalpr/benchmarks.>

the original image. We also included other image enhancements, such as colour changes (in the HSV colour space), morphological greyscale dilations or erosions (which thicken or thin the LP characters), blur, and noise during the training process. We chose the Stochastic Gradient Descent (SGD) algorithm with momentum 0.9, weight decay $5e^{-4}$ and batch size set to 4. In the adaptive aggregation function, we set $l_b = 100$ and $\lambda_{max} = 0.51$. The learning rate is set to 0.001 for the first 30K iterations and then decreased to 0.0001 in the following 20K iterations. At test time, the five augmented images are passed through the model, and the output is averaged.

5.2.3 Ablation Experiments

We perform two ablation experiments, the first is to evaluate the effectiveness of the proposed improvements (IASS and NCC)compared to the baseline MIL-OICR (TANG et al., 2017) and BOICR (Chapter 4), and the second is to evaluate the capacity of the NCC module in recovering the correct number of characters.

To compare the proposed improvements with the MIL-OICR baseline and the BOICR across all the datasets, we trained two versions of it: by including only the Instance-Aware Self-Supervision (BOICR + IASS), and by including both the Instance-Aware Self-Supervision and the Number of Characters Classification module (BOICR + IASS + NCC). To enhance readability, the results are divided into two tables. Table 5.1 presents the results in the datasets where the train-set was used to train the models, and Table 5.2 presents the results in datasets that were not used (unseen) in the training. For each dataset, we evaluated each method with three detection thresholds: '0.5' and '0.6', which means that proposals with a score lower than these thresholds are pruned after the NMS step. The '-1' threshold means that we use the counting from the ground-truth to select the top N proposals after NMS, which is an exception to BOICR + IASS + NCC version, where we recover N with the NCC module. We also evaluated the accuracy in terms of the edit error between the detected and ground-truth strings. Error 0 requires an exact string match, while error 1 allows a tolerance of one incorrect character. This "error" assessment helps determine whether a method's poor performance on a dataset is due to minor character errors or multiple discrepancies.

The MIL-OICR baseline consistently underperforms compared to BOICR across all datasets examined and exhibits even worse results in unseen datasets, suggesting its inadequacy for the LP recognition task. When including the IASS in the BOICR, we

Table 5.1 – Ablation study: Performance on the test-set of training datasets. Error 0 requires an exact match between the predicted and ground-truth strings, whereas Error 1 permits a single-character deviation.

Error	Method	AOLPR			Fazenda			CCPD val 20k		
		-1.0	0.5	0.6	-1.0	0.5	0.6	-1.0	0.5	0.6
0	MIL-OICR	98.20%	69.72%	65.46%	93.86%	42.27%	38.71%	99.48%	69.02%	67.39%
	BOICR	98.20%	97.55%	97.55%	**94.79%**	**93.41%**	92.76%	**99.85%**	**99.43%**	99.51%
	BOICR+IASS	**98.71%**	**98.20%**	**98.07%**	94.58%	93.23%	**92.78%**	99.81%	**99.43%**	**99.52%**
	BOICR+IASS+NCC	98.07%	97.81%	97.42%	94.55%	93.12%	92.50%	99.82%	99.13%	99.25%
1	MIL-OICR	99.48%	97.04%	95.62%	98.41%	82.75%	80.01%	99.76%	95.44%	94.88%
	BOICR	**100%**	**99.87%**	**99.87%**	**98.93%**	98.82%	98.56%	99.97%	99.96%	**99.96%**
	BOICR+IASS	99.87%	99.74%	99.74%	98.66%	**98.90%**	**98.75%**	99.96%	99.96%	**99.96%**
	BOICR+IASS+NCC	99.87%	99.74%	99.74%	98.87%	98.83%	98.58%	**99.98%**	**99.97%**	99.95%

Source: Author's compiled data.

Table 5.2 – Ablation study: Performance on unseen datasets. Error 0 requires an exact match between the predicted and ground-truth strings, whereas Error 1 permits a single-character deviation.

Error	Method	OpenALPR US			OpenALPR EU			UFPR Mbike		
		-1.0	0.5	0.6	-1.0	0.5	0.6	-1.0	0.5	0.6
0	MIL-OICR	72.83%	48.55%	45.09%	41.12%	10.28%	9.35%	78.93%	45.08%	38.83%
	BOICR	83.82%	**76.88%**	**75.72%**	87.85%	73.83%	77.57%	87.10%	**74.58%**	**75.22%**
	BOICR+IASS	**84.97%**	72.83%	74.57%	**98.13%**	**82.24%**	**83.18%**	86.08%	70.37%	73.31%
	BOICR+IASS+NCC	53.76%	72.25%	71.68%	5.61%	68.22%	75.70%	**86.97%**	**74.58%**	72.29%
1	MIL-OICR	89.60%	82.66%	77.46%	63.55%	48.60%	41.12%	92.46%	85.31%	80.33%
	BOICR	**98.27%**	96.53%	97.11%	97.20%	95.33%	96.26%	**97.32%**	**96.30%**	**95.53%**
	BOICR+IASS	95.95%	97.69%	97.69%	**99.07%**	**98.13%**	99.07%	96.04%	95.27%	95.02%
	BOICR+IASS+NCC	93.06%	**99.42%**	**98.27%**	96.26%	97.20%	97.20%	96.42%	95.79%	94.25%

Source: Author's compiled data.

observe an incremental improvement with an error of 0 in the datasets AOLPR, CCPD, and OpenALPR EU. However, it yields slightly inferior results [3] in the OpenALPR US and UFPR Mbike datasets. In the Fazenda and CCPD datasets, the results are similar. By incorporating the NCC module (BOICR + IASS + NCC), our methodology yields results that are remarkably similar to those obtained using ground-truth counts in the seen datasets. In contrast, for the OpenALPR US and OpenALPR EU datasets, the results are suboptimal when evaluated with "error 0". However, with "error 1", the results are better, which makes us believe that the module is not recovering the correct number of characters for these datasets. We further investigate this problem by checking the accuracy of the NCC module.

In Table 5.3, we present the classification accuracy of the NCC module in cor-

[3]For the datasets in question, for the thresholds of 0.5 and 0.6, the version BOICR + IASS + NCC outperforms the BOICR model. As we do not use the NCC with these thresholds, we can consider this version to be practically the same as the BOICR + IASS. We believe that these inferior results can be attributed to the inherently nondeterministic nature of training Deep Neural Networks.

Table 5.3 – Accuracy of the proposed number of characters classification module (NCC).

OpenALPR US	OpenALPR EU	AOLP	Fazenda	CCPD	UFPR Mbike
64.16%	5.61%	99.61%	99.99%	100.00%	99.74%

Source: Author's compiled data.

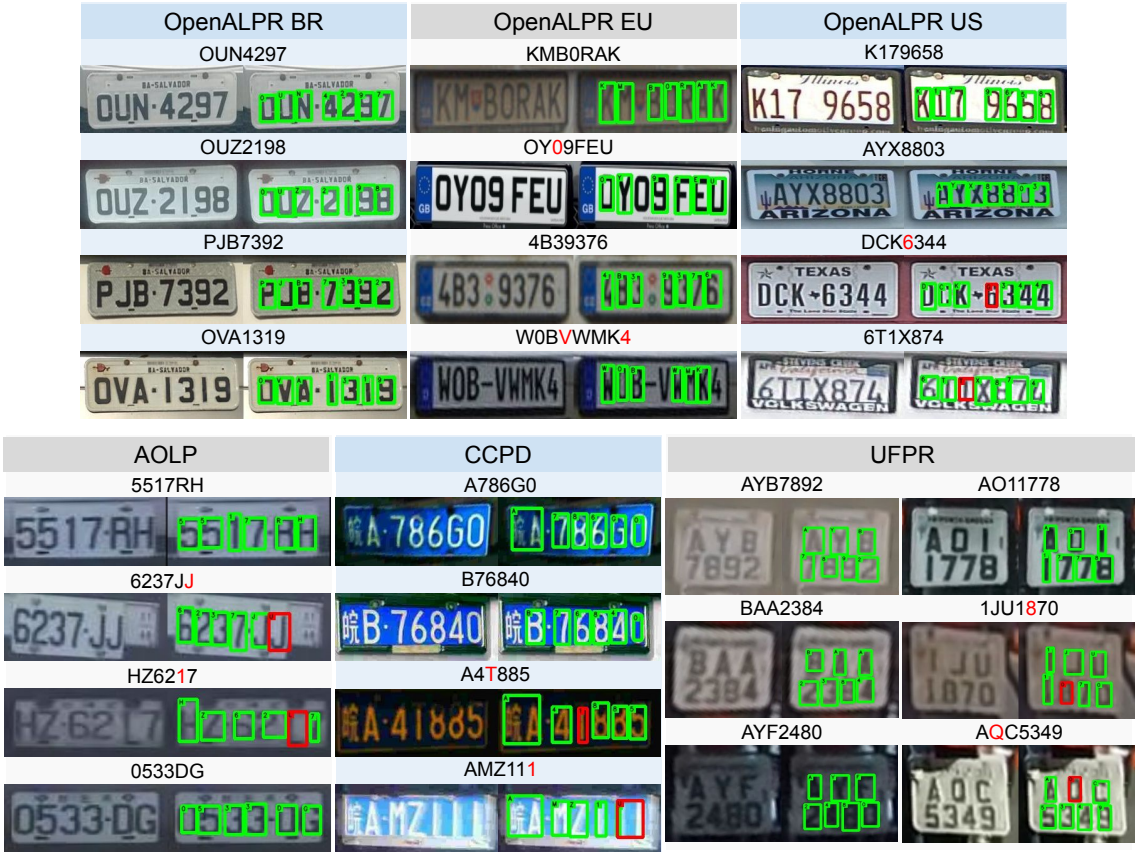
rectly classifying the number of characters. The results are very good for the datasets present in the training set, as well as for the UFPR Mbike, which was not included in the training data and has a different layout compared to the traditional Brazilian LPs. However, the results for OpenALPR US were worse, and for OpenALPR EU, the results were very poor ($\sim 5.6\%$). We believe that the explanation for this problem is that the European Union LPs does not have a fixed number of characters, varying the counts from region to region. Our current hypothesis is that the NCC module is primarily capturing a statistical association between each country-specific LP layout and the overall length of the sequence, rather than truly performing an explicit count of the individual characters. It seems that it would be better to approach the counting estimation as a regression problem instead of a classification one. However, it is not our primary research objective to have a model that performs well with unseen LP layouts, and the results in the trained datasets (**CCPD**, (XU et al., 2018), **AOLP** (HSU; CHEN; CHUNG, 2012) and **Fazenda**) indicate that the NCC module can be very useful in inference time.

We present some detection results of the BOICR + IASS + NCC with different plate layouts in Figure 5.3. The detected bounding boxes exhibit satisfactory fitting with the characters, indicating that our methodology is indeed capable of learning character spatial locations solely using the plate string as supervision.

5.2.4 Comparison with state-of-the-art

We compare our method with three other state-of-the-art methods for both generic text recognition and LPR, and results are shown in Table 5.4. The technique called "pre-trained STR" is a generic pre-trained text recognition approach presented in (BAEK et al., 2019), which performs poorly across all datasets. We believe that this behaviour is related to the Long Short-term Memory Network (LSTM) memorising the sequence of characters from the datasets where it was trained. We also re-trained this model with our dataset, which considerably improves the accuracy in these datasets. However, the

Figure 5.3 – Some examples of the character detected in different plate layouts. Green rectangles represent proposals that have the correct character detected, and red rectangles are proposals with the wrong character.



Source: Author's own illustration.

generalisation abilities in unseen datasets are disappointing, and it is not able to handle motorcycle LPs (characters in two rows). Quoting the authors of (BAEK et al., 2019), "Dataset Really Matters in STR".

We also evaluate fully supervised methods (SILVA; JUNG, 2018) and (Laroca et al., 2019) in all data sets, which are based on object detectors and trained using character bounding box annotations. In (SILVA; JUNG, 2018), LPs from different regions (Brazilian, Taiwanese, European) were used to train the model with the help of synthetic images. In (Laroca et al., 2019), the authors also explore LPs from different regions (Brazilian, Taiwanese, English, Chinese) and data augmentation to increase variability. Our method outperforms all other methods in the UFPR dataset and has very competitive results in the AOLPR, Fazenda, and CCPD datasets. As discussed previously, the classification of the number of characters is problematic for Open ALPR EU and the US, which compromises our final results for these datasets. However, our results in Table 5.2 (BOICR + IASS)

Table 5.4 – Comparison with other state-of-the-art methods. † means that the method was trained on the dataset.

Method	Error	OpenALPR BR	OpenALPR US	OpenALPR EU	AOLPR	Fazenda	CCPD val 20k	UFPR Mbike
LPR of (SILVA; JUNG, 2017)	Error 0	98.21% †	69.36% †	85.98% †	94.20% †	73.85%	58.45%	0.26%
	Error 1	100% †	94.80% †	98.13% †	99.61%†	93.74%	89.66%	0.26%
LPR of (Laroca et al., 2019)	Error 0	98.21% †	94.80% †	99.07% †	99.10% †	86.57%	68.28% †	84.93% †
	Error 1	100% †	99.42% †	100% †	99.87% †	97.66%	94.69% †	95.66% †
pretrained STR (BAEK et al., 2019)	Error 0	4.46%	0.57%	0%	31.82%	0.33%	0.15%	0%
	Error 1	9.82%	3.46%	9.34%	54.12%	1.71%	34.58%	0%
retrained STR (BAEK et al., 2019)	Error 0	98.21%	34.68%	4.67%	97.68% †	97.23% †	99.86% †	0.26%
	Error 1	100%	69.36%	28.97%	99.09% †	99.84% †	99.98% †	12.13%
BOICR th=0.6	Error 0	100%	75.72%	77.57%	97.55% †	92.76% †	99.51% †	75.22%
	Error 1	100%	97.11%	96.26%	99.87% †	98.56% †	99.96% †	95.53%

Source: Author's compiled data.

show that if the number of characters is known for these datasets, our results are competitive as well. It is worth emphasising that the approaches proposed in (SILVA; JUNG, 2018) and (Laroca et al., 2019) require detailed supervision in the form of bounding box annotations for every individual character present on the LP during training. In contrast, our method, similarly to (BAEK et al., 2019), relies solely on the full license plate string as supervisory information, thereby eliminating the need for character-level localisation labels and reducing the annotation burden.

5.3 Conclusions

This chapter presents a novel weakly supervised approach for character detection in the context of license plate recognition. The primary contributions include the development of the Instance-Aware Self-Supervision (IASS) building process, which explores the inherent counting information present in weak annotation strings using an iterative top-scoring selection strategy. Complementing this, the proposed Number of Characters Classification (NCC) module provides a practical solution to estimate the total number of characters during inference time. The results demonstrate that weakly supervised learning can be successfully applied in the LP recognition domain, offering a practical solution that significantly reduces annotation costs while maintaining competitive performance comparable to that of fully supervised alternatives. The proposed IASS and NCC modules represent valuable methodological contributions.

The experimental evaluation conducted in five diverse datasets from different geographical regions, including Brazil, China, Taiwan, the United States, and the European Union, provided comprehensive validation of the proposed methodology. The results consistently demonstrated the superiority of the proposed approach over the MIL-OICR base-

line, and BOICR. The BOICR + IASS + NCC configuration achieves remarkably high detection quality, similar to methods that use bounding boxes as supervision in the tested datasets. The technique demonstrated robust generalisation capabilities, particularly evident in its strong performance on the UFPR Motorcycle dataset. This dataset was not included in the training time, and the LPs display a different layout (two rows of characters and different aspect ratio). While the NCC module achieved excellent classification accuracy approaching 100% for the datasets included in the training set, the evaluation also revealed essential limitations when dealing with European plates that exhibit highly variable character counts in different regions, highlighting areas for future investigation.

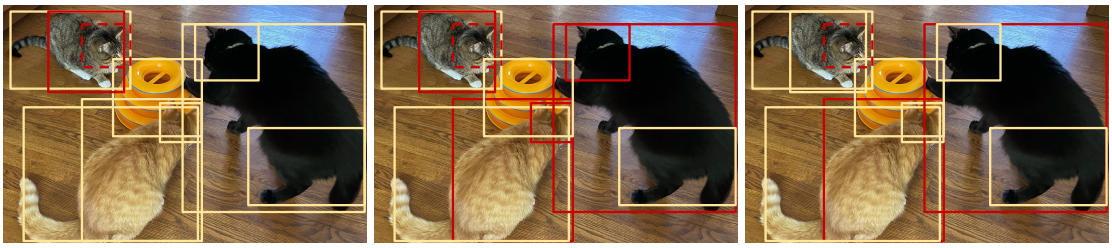
Looking towards future research directions, several promising avenues emerge from the findings presented in this chapter. The limitations observed with the NCC module suggest that regression-based approaches to character counting might provide a better generalisation to unseen plate formats with diverse character count distributions. Another alternative could be to use the proposal-image features instead of the image-level features. Another direction to explore is testing how our methodology handles the new Brazilian LPs layout, which began being used in 2020.

(SILVA; JUNG, 2017)

Although in some particular tasks we could have the same-class object counting embedded in the annotations, that is not the standard case, and developing a generic methodology to include more instances in the refiner's self-supervision building process can potentially improve the WSOD methods detection results. Another important aspect to highlight is that our methodology is not optimised for runtime. Consequently, when compared to Fully Supervised Object Detector (FSOD)-based approaches, our method requires substantially more time to generate predictions. On the other hand, by eliminating the need to annotate each character bounding box manually, our approach significantly reduces annotation overhead. In this context, the predictions produced by our method can be leveraged as a form of weak or surrogate supervision to train fast detection architectures such as (SILVA; JUNG, 2017), which are specifically designed for real-time or near real-time operation but typically demand extensive and costly manual annotation. In the next chapter, we propose a methodology to mine these additional instances without requiring additional supervision.

6 LONG-TERM INSTANCE MINING IN WEAKLY SUPERVISED OBJECT DETECTION

Figure 6.1 – Our contributions to improve each category self-supervision for Online Instance Classification Refinement (OICR)'s refinement module. (a) Multiple Instance Learning Network (MIL)-OICR based techniques use the top-scoring proposal of the previous module (dashed red) to select proposals with large IoU (solid red) as positive, and all other proposals are set to the background class (yellow), including other instances of the same class. (b) Our instance discovery module is able to discover the other instances of the class and include them as positive in the creation of the supervision, but it might create more than one proposal per instance. (c) An overlap suppression strategy is used to reduce the number of positive candidates while keeping only the most prominent ones.



(a) Traditional MIL-OICR self-supervision.

(b) Example of output of our discovery module.

(c) Example of our proposal overlap suppression algorithm.

Source: Author's own illustration.

This chapter faces the challenge of identifying more instances of the same object within a single image using a MIL-OICR-based Weakly Supervised Object Detection (WSOD) approach during the self-supervision building process. The issue of traditional OICR refiner self-supervision is exemplified in Figure 6.1a, where the top-scoring proposal localised in one of the cats (marked with a dashed red box) is used as an "anchor" to identify additional positive proposals by evaluating the degree of overlap with it (displayed in a solid red box). The remaining low-overlapping proposals (shown in yellow boxes) are assigned to the background class. The problem with this methodology is that the other two cats will be labelled as the background and not as the right target class to which they belong. The erroneous assignment of these additional object instances brings confusion to the model's optimisation, which could lead the model to *learn to omit some instances* due to the loss function imposing penalties on these instances during the optimisation process. Not including these additional instances as positive also negatively affects the model's ability to generalise for the specified category. For example, the excluded pair of cats exhibits divergent visual traits, such as size, pose, and fur pattern. Therefore, the model will be denied the opportunity to learn these characteristics from these instances.

To address the issue of *missing instances* during the creation of the refiner's self-supervision, we propose a novel instance mining module designed to discover multiple instances of objects of the same category in a given image. During training, our methodology stores the top-scoring proposals for each target category in a memory bank. In each training iteration, the bank is updated with new instances and older instances are removed. An interpretation of this mechanism is that it serves as a *long-term memory repository* containing the *best* (top-scoring) examples of each target category that the refiner has encountered in previous training iterations. The samples stored in the bank are employed as learning guidance for an Multi-Layer Perceptron (MLP) network, which is trained using a novel discriminative loss function in the WSOD context.

The learning target of the MLP network is to project features that belong to the same category into similar embeddings enclosed within a spherical subspace. The discriminative loss enforces that instances of the same class are enclosed inside the same sphere and that spheres of different classes do not overlap by using a minimal acceptable margin between them. Two parameters govern this process: one specifies the sphere's maximum diameter, and the other determines the inter-sphere margin. Essentially, it learns to cluster intraclass features while distancing interclass features.

The restrictions imposed in the learning process of the spherical clustering space allow the use of the MLP's projected features to perform the discovery of additional same-class object instances, which can be included as positive instances in the self-supervision building phase. The discovery process works by projecting all proposal-level features into the spherical clustering space with the MLP. As the centroid of each sphere is known, the diameter of the sphere can be used as a threshold to select positive instances of each category. The results of using this instance mining methodology on the same image in Figure 6.1a are depicted in Figure 6.1b. While the instance mining successfully identifies positive proposals on the other cat instances (boxes that changed from yellow to red), it may introduce a new issue of having an excessive number of positive instances, implying that several subparts of the target classes may be incorrectly included as positives, which can also adversely affect the learning process.

In the context of having an excessive number of positive instances after the discovery step, we also proposed an overlap suppression strategy that focuses on instance disambiguation. It has the objective of reducing the issue of having multiple proposals contained within the same object instance, as illustrated in Figure 6.1c, where some of the positive proposals are reassigned to the background class. The key contributions of this

chapter are summarised in the list below.

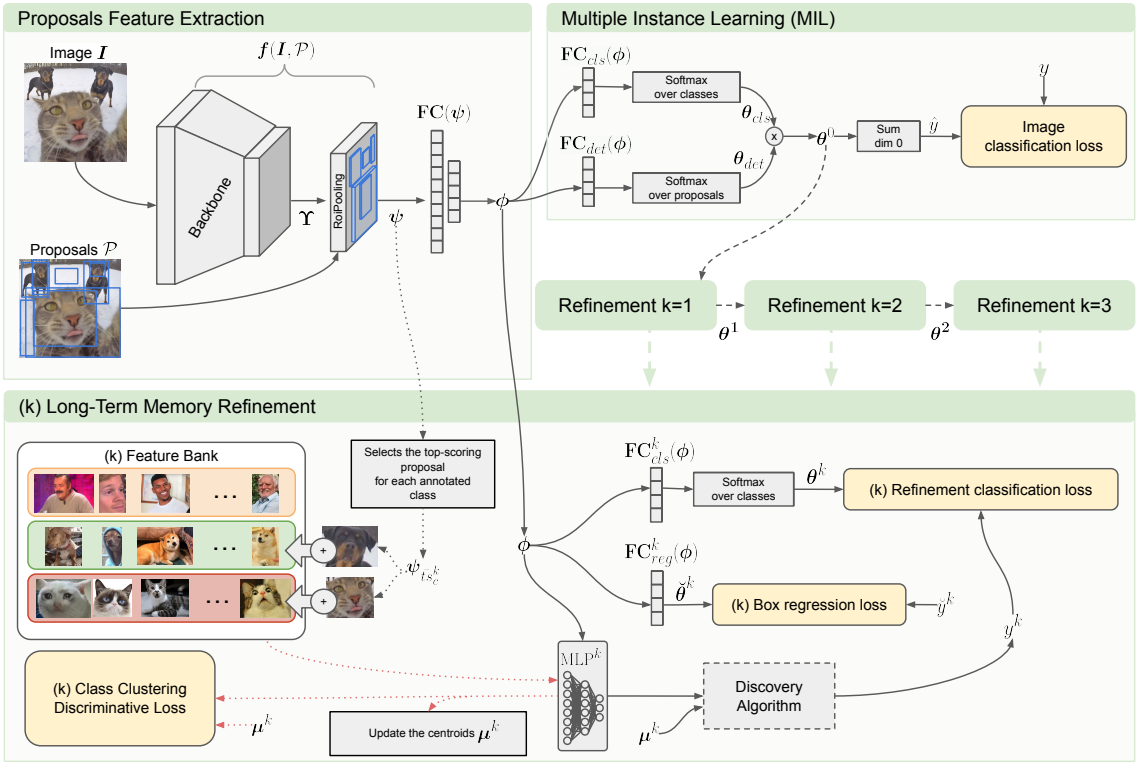
- Introduction of a projection MLPtrained with a discriminative loss function that ensures intraclass coherence and interclass differentiation.
- Introduction of a memory bank that records a collection of the most recent top-scoring proposals from previous training iterations, which are used as supervision to train the MLP.
- Development of a proposal overlapping suppression strategy to prune some discovered instances that belong to the same object.

Our method, named Long-Term Instance Mining (LTIM), achieves the best AP scores for the datasets Pascal Visual Object Classes (VOC) 2007 and VOC 2012 and ranks second for AP50 in the dataset Common Objects in Context (COCO) 2014, while also demanding fewer hardware resources compared to other WSOD instance mining strategies, which means that our methodology can produce comparable or superior results in environments with limited hardware resources (the hardware employed in this chapter is detailed in Section 6.2.3), which are often found in small organisations or developing countries. The following sections of this chapter are organised in the following manner. In Section 6.1, we elaborate on the specifics of our proposed methodology. Next, in Section 6.2, we detail the results of our experiments. Finally, in Section 6.3, we synthesise our findings and offer our conclusions.

6.1 Overview of the Proposed Method

One significant limitation of most MIL-OICR-based WSOD methods is the reliance on only top-scoring proposals to build self-supervision of the refinement modules. This approach may hinder the accurate detection of categories that include multiple instances within an image. Inspired by OD-WSCL (SEO et al., 2022), in each refiner k we include a shallow MLP^k that projects the proposal-level features $\boldsymbol{\phi} \in \mathbb{R}^{D_1}$ into proposal embeddings $\boldsymbol{o}^k = \text{MLP}^k(\boldsymbol{\phi}) \in \mathbb{R}^{P \times D_2}$, where $D_2 < D_1$ is the dimension of the embeddings. The central concept is that these embeddings help in the identification of new instances for inclusion in the self-supervision building processor of each refiner. One key difference of our methodology is that each refiner has its own MLP^k and discovery module, while OD-WSCL has only one discovery model for all refiners. We posit that this

Figure 6.2 – The complete high-level architecture of LTIM. Our contributions reside in the OICR refiners, where a memory bank stores past top-scoring proposals utilised to train a MLP that project features into a class-clustered embedding space. These projected features are utilized to perform the discovery of additional object instances to enhance the self-supervision building process. More details regarding each part are provided in Section 6.1.



Source: Author's own illustration.

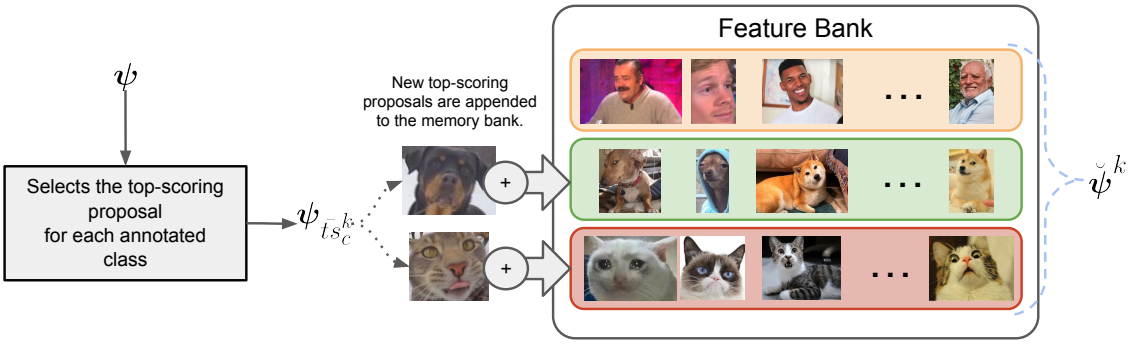
method enhances specialisation in the discovery process and integrates more effectively with each refiner's expertise.

The key advance we present is a loss function designed to train the MLP^k, which uses a *memory bank* for *long-term instance learning*. During the training phase, top-scoring proposals from target classes are retained, and a discriminative loss is used to learn class-specific embeddings that are restricted to predefined areas within the embedding space. When a new image is fed to the training process, the embeddings allow us to discover new instances of annotated categories that will be used as supervision. Furthermore, we present a *proposal overlap suppression* methodology that aims to prevent the selection of multiple proposals from the same instance. The complete high-level architecture of the proposed method is lustrated in Figure 6.2 diagram, and each part of it is detailed in the following sections.

6.1.1 Instance Memory Bank

During training, we retrieve the RoI-pooled features of each positive class top-scoring proposal $\boldsymbol{\psi}_{\bar{ts}_c^k}$ at each refinement stage k, and store them in a *instance memory bank*. Figure 6.3 presents a visual representation of the addition of new features to the *memory bank*. The rationale for storing $\boldsymbol{\psi}$ instead of $\boldsymbol{\phi}$ is that the memory bank is filled dynamically during training, and each sample is stored at a different time. Since the weights of the network are constantly updated and the backbone changes more slowly than $\mathbf{FC}(.)$ due to backpropagation, we retain more "stable" features in the bank.

Figure 6.3 – The instance memory bank stores previous iterations' high-scoring proposal features. In each training iteration, it stores new top-scoring instances and removes older stored instances if the bank for a given category is full.



Source: Author's own illustration.

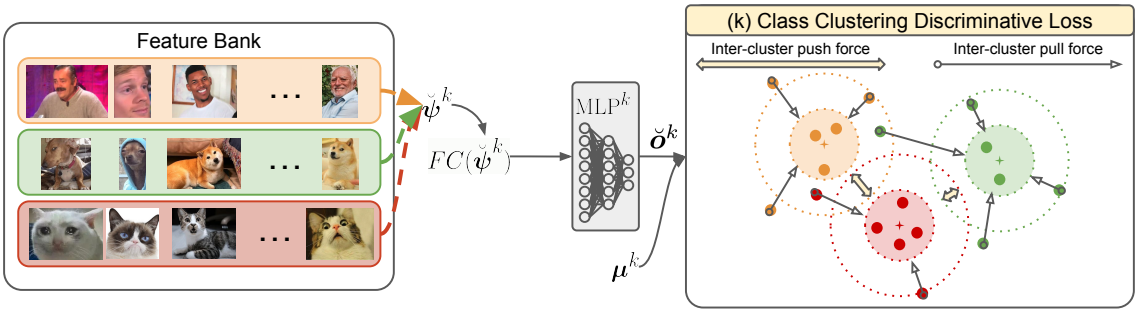
Let $\breve{\boldsymbol{\psi}}^k \in \mathbb{R}^{N \times C}$ denote the *memory bank* associated with the refiner k, where N is the maximum number of instances that can be memorised for each category. Here, $\breve{\boldsymbol{\psi}}_{n,c}^k \in \mathbb{R}^{D_0}$ corresponds to the nth stored feature of class c. The bank starts empty and, when the bank capacity fills for a given category, it is updated in a First In, First Out (FIFO) manner. The samples stored in the bank are used as supervision to train the MLP^k module. However, the MLP^k module requires features in the same format as $\boldsymbol{\phi}$ and the features stored in the bank are in $\boldsymbol{\psi}$ format. Therefore, when features of the bank are requested, they are first passed through an updated version of $\mathbf{FC}(.)$.

6.1.2 Clustering Discriminative Loss

In contrast to OD-WSCL (SEO et al., 2022), which is highly dependent on the batch size to obtain sufficient instances for training the MLPwith its contrastive loss, our long-term instance mining strategy utilises the samples stored in the memory bank, allow-

ing training with reduced batch sizes without compromising the learning process of the instance discovery module. Figure 6.4 presents a high-level diagram of the training process of each MLP^k, which uses the features stored in the memory bank and the clustering discriminative loss.

Figure 6.4 – Illustration of the training procedure of each MLP^k. The features stored in the bank are recovered and fed into **FC**$(.)$ to convert them to proposal-level features. The proposal-level features are then fed into an MLP^k for embedding generation. The projected embeddings are utilised in the clustering contrastive loss to optimise MLP^k's weights, and to calculate the next iteration embedding centroids.



Source: Author's own illustration.

The stored features $\breve{\psi}^k$ are used to train MLP^k by exploring a discriminative loss function, inspired by (BRABANDERE; NEVEN; GOOL, 2017), that enforces intraclass consistency and interclass repulsion. In each training iteration, we project the stored features into the embedding space $\breve{o}^k = \text{MLP}^k(\text{FC}(\breve{\psi}^k) \in \mathbb{R}^{N \times D_2}$ and compute the embedding centroids.

$$\boldsymbol{\mu}_c^k = \frac{1}{N_c} \sum_{n}^{N_c} \breve{\boldsymbol{o}}_{n,c}^k \tag{6.1}$$

for each class c, where $N_c \leq N$ is the number of instances stored for category c (as training advances, we tend to have $N_c = N, \forall c$).

The first part of the loss penalises instances far from the cluster centre by using a margin loss, given by

$$\mathcal{L}_{var}^k = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N} \sum_{n=1}^{N} \text{ReLU}\left(\left\|\boldsymbol{\mu}_c^k - \breve{\boldsymbol{o}}_{n,c}^k\right\| - \delta_v\right)^2, \tag{6.2}$$

where $\delta_v > 0$ is the margin that controls the concentration of the cluster. This loss aims to generate features that present a distance to the centre of the corresponding cluster smaller than δ_v. The second part of the loss aims to repulse (separate) the clusters by forcing the

cluster means to be far from each other, given by

$$\mathcal{L}_{sep}^{k} = \frac{1}{C(C-1)} \sum_{i=1}^{C} \sum_{\substack{j=1 \\ j \neq i}}^{C} \text{ReLU}\left(2\delta_d - \left\|\boldsymbol{\mu}_i^k - \boldsymbol{\mu}_j^k\right\|\right)^2, \tag{6.3}$$

where δ_d is the separation margin. The final discriminative loss is given by

$$\mathcal{L}_{disc}^{k} = \mathcal{L}_{var}^{k} + \mathcal{L}_{sep}^{k}, \tag{6.4}$$

and its minimum value $\mathcal{L}_{disc}^{k} = 0$ is achieved when all intra-cluster distances are at most δ_v, and all inter-cluster centroid distances are at least $2\delta_d$. If this ideal situation is achieved and $\delta_d > \delta_v$, we can guarantee that if $\|\breve{\boldsymbol{o}}_{n,c}^{k} - \boldsymbol{\mu}_c^k\| < \delta_v$, and then feature $\breve{\boldsymbol{o}}_{n,c}^{k}$ is closer to $\boldsymbol{\mu}_c^k$ than any other cluster centre, which can be used to mine new instances (we suggest $\delta_d = 3\delta_v$ to obtain an extra margin, as indicated in Table 6.1).

The final loss of each refinement module k is a combination of the baseline loss functions (as described in Chapter (2.2.4), in Equation (2.10) and in Equation (2.11)) and the clustering discriminate loss, given by

$$\mathcal{L}^{k} = \mathcal{L}_{cls}^{k} + \mathcal{L}_{reg}^{k} + \tau \mathcal{L}_{disc}^{k}, \tag{6.5}$$

where $\tau > 0$ is a hyper-parameter.

6.1.3 Instance Mining

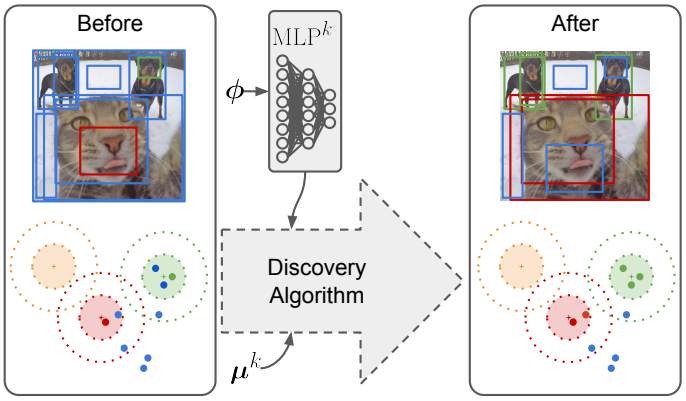
To extract new instances to be included in the supervision, we compute the embeddings $\boldsymbol{o}^k = \text{MLP}^k(\boldsymbol{\phi})$ for all proposals p_i of a training image $\boldsymbol{I}$ and compare them with the cluster centres $\bar{\boldsymbol{\mu}}_c^k$, for all $y_c = 1$. If they are sufficiently close, the corresponding proposal is labelled as a new positive instance. This process is achieved by updating the pseudo-label computation in Equation (2.9) through

$$y_{i,c}^{k} = \begin{cases} 1 & \text{if } \|\boldsymbol{o}_i^k - \bar{\boldsymbol{\mu}}_c^k\| < \delta_v, \\ 0 & \text{otherwise} \end{cases}, \tag{6.6}$$

and refine the background-related pseudo-labels accordingly.

The instance discovery process is illustrated in Figure 6.5, while Figure 6.1b

Figure 6.5 – Our method of instance mining employs embeddings generated by a MLPin conjunction with class embedding centroids to uncover new instances for each class.



Source: Author's own illustration.

shows an example of the supervision generated by it compared to a MIL-OICR-based model (Figure 6.1a). Although our strategy can mine proposals related to other instances of the same category (cat), sometimes it can also mine proposals that belong to sub-parts of the target objects. This multitude of proposals for each instance might negatively impact the model's training by possibly promoting the detection of small and large boxes, which can either emphasise part domination or proposals that contain significant portions of the background. To mitigate this problem, we introduce a *Proposal Overlap Suppression (POS)* strategy, described in the next section.

6.1.4 Proposal Overlap Suppression

The instance mining procedure is expected to capture other instances of the same class present in the image. However, as illustrated in Figure 6.1b, it could also mine more than one proposal related to the same object instance, and these proposals could be sub-parts of the object or larger than the object's real boundaries. To deal with these cases, we propose a simple suppression strategy to avoid duplicate proposals from the same instance by also considering the spatial relationship of the proposal and not only the features, as done in Section 6.1.3.

Let us consider the subset of *positive* pseudo-labels provided by Equation (6.6) for a given class c, i.e., the set of proposals p_i^c for which $y_{i,c}^k = 1$, denoted by $\mathcal{P}_c$. For each pair of proposals $p_i^c, p_j^c \in \mathcal{P}_c$ with $p_i^c \neq p_j^c$, we use the Intersection-over-minimum-Area (IomA) given by

$$\text{IomA}(p_i^c, p_j^c) = \frac{\left|p_i^c \cap p_j^c\right|}{\min\{\left|p_i^c\right|, \left|p_j^c\right|\}}. \tag{6.7}$$

Note that $\text{IomA}(p_i^c, p_j^c)$ reaches the maximum value of one if and only if one proposal is contained in the other, which possibly indicates duplicate proposals for the same instance. To detect these situations, we introduce a large overlap threshold ξ (set empirically to 0.9) and suppress the lowest-scoring proposal through

$$y_{i,c}^k = \begin{cases} 1 & \text{if } \text{IomA}(p_i^c, p_j^c) > \xi \text{ and } \boldsymbol{\theta}_{i,c}^k \geq \boldsymbol{\theta}_{j,c}^k \\ 0 & \text{otherwise} \end{cases} . \tag{6.8}$$

The results of POS applied to the proposals shown in Figure 6.1b are illustrated in Figure 6.1c, where we can observe that our strategy can reduce the selection of duplicate proposals for the cats at the bottom and to the right of the image.

6.2 Experiments

In this section, we present the ablation experiments and compare our results with the WSOD State-of-the-Art (SOTA) methods present in the literature.

6.2.1 Datasets

Like most WSOD approaches, we conducted assessments using the datasets VOC 2007 (EVERINGHAM et al., 2007), VOC 2012 (EVERINGHAM et al., 2012), and COCO 2014 (LIN et al., 2014b) datasets. As customary in WSOD methodology, in both VOC datasets, we utilised train + val sets to train the model, consisting of 5,011 images for VOC 2007 and 11,540 images for VOC 2012. To evaluate the results, we used the respective test sets, comprising 4,951 and 10,991 images, respectively. Furthermore, our method was evaluated on COCO 2014, training it on 82,783 images and using the validation set of 40,504 images as the test set (the same methodology employed in other WSOD SOTA methods for COCO). The COCO dataset was frequently overlooked in WSOD studies due to its higher complexity: It contains considerably more instances per image (around seven versus two in VOC) and a greater number of categories (80 versus 20 in VOC).

6.2.2 Metrics

We use the conventional metrics commonly used by the WSOD methods. For VOC 2007 and VOC 2012, the mean Average Precision (mAP) and CorLoc (correct location) are calculated, both with the standard IoU threshold of 0.5. For COCO, we measure AP (average precision over 10 IoU thresholds ranging from 0.5 to 0.95 in increments of 0.05) and AP50 (average precision at the 0.5 IoU threshold alone). It is essential to note that AP metrics are significantly influenced by noise or imprecise detections/annotations (MURRUGARRA-LLERENA; JUNG, 2025), particularly when stricter IoU thresholds are employed. Hence, we believe that the conventional threshold of 0.5 is more suitable to evaluate WSOD methods.

6.2.3 Implementation Details

We implemented our model in PyTorch and used an Nvidia RTX 3090 with 24GB of VRAM for all experiments. We utilised the proposals generated by MCG (ARBELÁEZ et al., 2014), which yielded better results than those of the Selective Search (SS) (UIJLINGS et al., 2013), as reported by Seo et al. (2022). The MIL-OICR baseline that we employed is described as follows. We use the Convolutional Neural Networks (CNN) layers of the VGG16 (SIMONYAN; ZISSERMAN, 2014) backbone pre-trained on ImageNet (DENG et al., 2009). After the convolutional layers, a RoI pooling layer replaces the global average pooling layer. Next, two Fully Connected (FC) layers in the same format as the first two FC layers of VGG16 are included. We also use the values from these layers as initialisation values for the FC layers. A MIL (BILEN; VEDALDI, 2016) module is included to optimise classification loss and to provide the first scoring matrix that feeds a chain of OICR's refiner modules (TANG et al., 2017) enhanced with bounding box regression (REN et al., 2020). In addition to the MIL-OICR baseline, we incorporate the proposed strategies for long-term instance mining and proposal overlap suppression, which comprise our approach LTIM. Table 6.1 presents the hyperparameter values utilised in the experiments for each dataset. Some of these parameter values were defined based on the findings of the ablation experiments conducted in the next section.

Table 6.1 – Hyper-parameters used in the experiments

Dataset	VOC 2007	VOC 2012	COCO 2014
Batch Size	8	8	6
Training Iterations	30,000	60,000	173,333
Optimizer	SGD		
Base Learning Rate	10^{-2}		
Weight Decay	10^{-4}		
Training Resolutions	480, 576, 688, 864, 1000		480, 576, 688, 864, 1000, 1200
Testing Resolutions	480, 576, 688, 864, 1000, 1200		
τ	0.03		0.01
K	3		
N	64		
D_2	128		
δ_v	0.05		
δ_d	0.15		

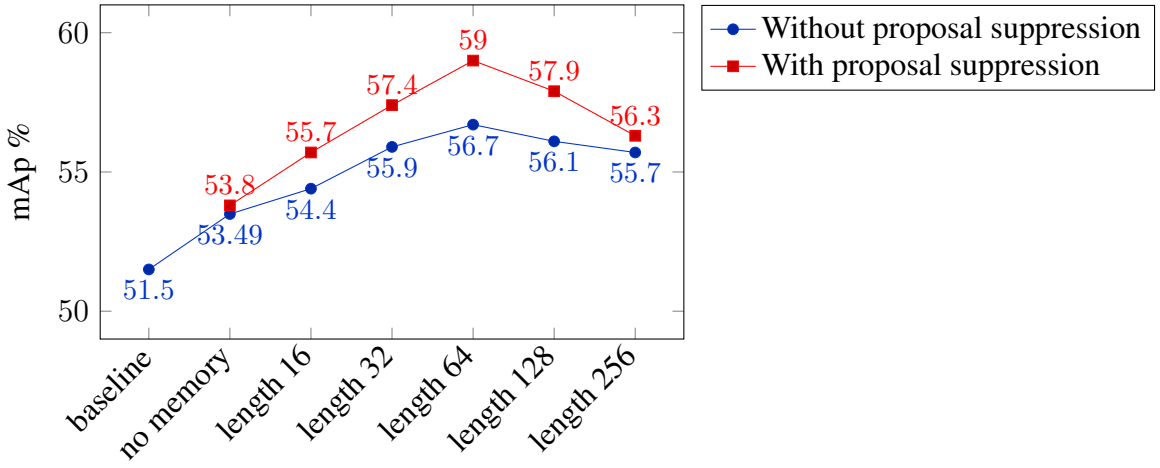
Source: Author's compiled data.

6.2.4 Ablation Studies

In this section, we investigate how changes to the memory bank length and the inclusion of the proposal overlap suppression technique affect the outcomes on the VOC 2007 test set. We present the results in Figure 6.6, where *"baseline"* refers to the MIL-OICR baseline, a version of our method without any discovery techniques, which achieves 51.5% mAP. The *"no memory"* refers to the *baseline* plus the discovery method applied at batch-level (i.e., only with samples present in the current batch), which improves mAP to 53.5%, and, by adding the POS approach to it, increases the results to 53.8%.

Including the discovery process by itself enhances the baseline mAP by ≈ 2 percentage points. However, integrating the memory bank amplifies this improvement even further, with optimal performance achieved at a length of 64, reaching 56.7% mAP without POS and 59.0% with it. This represents a significant improvement of 5.2 and 7.5 percentage points over the baseline, respectively. Interestingly, increasing the memory length beyond 64 (to 128 or 256) leads to a slight performance degradation, suggesting that there is an optimal maximum bank size for this task. Furthermore, the addition of POS technique consistently yields improvements across all memory lengths, with gains ranging from 0.3 to 2.3 percentage points. Although our method is built on top of the MIL-OICR baseline, it is possible to include our methodology in other WSOD baselines, which could be a topic for future research.

Figure 6.6 – Effects of setting the memory bank length in the final mAP of Pascal VOC 2007 test set and enabling the proposal overlap suppression technique.



Source: Author's own illustration.

6.2.5 Comparison with the state-of-the-art

Table 6.2 – Comparison of mean average precision on PASCAL VOC 2007 test set.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
OICR (TANG et al., 2017)	58.0	62.4	31.1	19.4	13.0	65.1	62.2	28.4	24.8	44.7	30.6	25.3	37.8	65.5	15.7	24.1	41.7	46.9	64.3	62.6	42.0
PCL (TANG et al., 2018)	54.4	69.0	39.3	19.2	15.7	62.9	64.4	30.0	25.1	52.5	44.4	19.6	39.3	67.7	17.8	22.9	46.6	57.5	58.6	63.0	43.5
C-MIL (WAN et al., 2019a)	62.5	58.4	49.5	32.1	19.8	70.5	66.1	63.4	20.0	60.5	**52.9**	53.5	57.4	68.9	8.4	24.6	51.8	58.7	66.7	63.5	50.5
BOICR (ZENI; JUNG, 2020b)	68.6	62.4	55.5	27.2	21.4	71.1	71.6	56.7	24.7	60.3	47.4	56.1	46.4	69.2	2.7	22.9	41.5	47.7	71.1	69.8	49.7
WSOD^2 (ZENG et al., 2019)	65.1	64.8	57.2	**39.2**	24.3	69.8	66.2	61.0	29.8	64.6	42.5	60.1	71.2	70.7	21.9	28.1	58.6	59.7	52.2	64.8	53.6
OIM (LIN et al., 2020)	55.6	67.0	45.8	27.9	21.1	69.0	68.3	70.5	21.3	60.2	40.3	54.5	56.5	70.1	12.5	25.0	52.9	55.2	65.0	63.7	50.1
SLV (CHEN et al., 2020)	65.6	71.4	49.0	37.1	24.6	69.6	70.3	70.6	30.8	63.1	36.0	61.4	65.3	68.4	12.4	29.9	52.4	60.0	67.6	64.5	53.5
MIST (REN et al., 2020)	68.8	77.7	57.0	27.7	28.9	69.1	74.5	67.0	32.1	73.2	48.1	45.2	54.4	73.7	**35.0**	29.3	**64.1**	53.8	65.3	65.2	54.9
CASD (HUANG et al., 2020)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	56.8
EFSL (WU et al., 2024a)	62.2	76.3	44.1	37.5	31.6	70.1	74.7	72.0	31.8	66.9	51.0	**66.1**	65.7	72.0	13.0	24.5	50.3	63.9	63.2	55.8	54.6
BUAA_PALoss (WU et al., 2023)	66.1	80.1	41.3	30.1	28.5	75.3	72.0	76.2	33.5	69.7	48.6	62.1	60.2	73.1	16.7	26.8	54.1	60.4	70.8	63.3	55.4
LDD-SWA (FENG et al., 2024)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	55.6
MCC-MCT (WU et al., 2024b)	66.1	75.0	56.8	31.9	27.1	74.0	71.7	72.6	31.8	68.4	52.4	62.0	63.4	72.3	16.9	**31.2**	57.2	63.7	72.4	58.1	56.3
CBL (YIN et al., 2023)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	57.4
IENet (FENG et al., 2023)	60.6	77.3	**64.5**	27.5	34.5	73.9	76.0	**80.4**	**33.9**	70.7	50.8	58.7	68.2	72.2	18.4	27.2	58.6	**66.4**	71.8	67.7	58.0
OD-WSCL (SEO et al., 2022)	**69.2**	**81.5**	56.4	28.5	30.5	**77.6**	**79.1**	71.6	13.0	70.8	48.8	56.9	74.9	78.4	34.9	27.6	61.4	65.4	**74.4**	**73.4**	58.7
LTIM (Ours)	66.5	79.1	55.1	31.3	**35.6**	73.0	77.7	75.4	31.8	**76.3**	49.4	51.8	74.8	78.3	27.8	30.2	63.9	63.7	72.6	64.8	**59.0**

Source: Author's compiled data.

The first experiment explored the VOC datasets (2007 and 2012). The VOC 2007 dataset provides public access to the labels of the test set, which can be used for parameter tweaking. On the other hand, the VOC 2012 test set maintains hidden annotations, requiring the submission of detections to an evaluation server to acquire results, thus ensuring a truly fair comparison of competing methods. Tables 6.2 and 6.3 present the AP metrics of the leading techniques in the VOC 2007 and VOC 2012 datasets, respectively, with the best results for each class shown in bold and the second best underlined (we keep this notation for all subsequent tables). Our approach presents the

Table 6.3 – Comparison of mean average precision on PASCAL VOC 2012 test set.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
OICR (TANG et al., 2017)	67.7	61.2	41.5	25.6	22.2	54.6	49.7	25.4	19.9	47.0	18.1	26.0	38.9	67.7	2.0	22.6	41.1	34.3	37.9	55.3	37.9
PCL (TANG et al., 2018)	63.4	64.2	44.2	25.6	26.4	54.5	55.1	30.5	11.6	51.0	15.8	39.4	55.9	70.7	8.2	26.3	46.9	41.3	44.1	57.7	41.6
C-MIL (WAN et al., 2019a)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	46.7
BOICR (ZENI; JUNG, 2020b)	73.0	67.6	51.7	29.8	31.8	60.0	59.6	33.1	18.9	60.6	22.6	46.8	60.2	73.8	3.9	26.6	51.6	40.6	60.2	61.2	46.7
WSOD^2 (ZENG et al., 2019)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	47.2
OIM (LIN et al., 2020)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	45.3
SLV (CHEN et al., 2020)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	49.2
MIST (REN et al., 2020)	**78.3**	73.9	56.5	30.4	37.4	64.2	59.3	60.3	26.6	66.8	25.0	55.0	61.8	79.3	14.5	30.3	61.5	40.7	56.4	63.5	52.1
CASD (HUANG et al., 2020)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	53.6
EFSL (WU et al., 2024a)	74.5	75.3	57.4	32.2	34.4	60.3	58.4	65.6	23.6	64.2	31.4	51.5	67.7	75.9	16.1	26.5	57.4	36.0	38.7	61.3	50.4
BUAA_PALoss (WU et al., 2023)	74.5	75.4	60.4	29.7	31.2	64.5	57.9	77.4	25.1	64.5	25.1	72.9	68.4	74.7	**25.7**	28.7	51.3	36.5	59.6	56.6	53.1
LDD-SWA (FENG et al., 2024)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	50.7
MCC-MCT (WU et al., 2024b)	75.6	**76.3**	56.6	32.1	35.7	63.6	64.1	73.7	26.8	65.4	31.4	69.8	66.6	77.9	9.0	28.1	57.2	37.4	63.0	57.7	53.4
CBL (YIN et al., 2023)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	53.5
IENet (FENG et al., 2023)	72.0	75.1	**65.6**	32.1	38.8	63.5	61.9	**82.1**	27.9	66.7	29.8	68.7	61.8	76.2	19.7	32.7	59.5	39.8	**67.9**	56.8	54.9
OD-WSCL (SEO et al., 2022)	74.7	61.3	32.9	**40.0**	**64.6**	59.8	**68.1**	26.3	**67.5**	23.0	**67.1**	62.8	**80.6**	**80.6**	17.3	34.1	63.4	44.4	66.2	**64.9**	54.6
LTIM (Ours)	75.1	75.8	58.1	34.1	38.2	**66.7**	61.5	82.0	24.6	**70.8**	22.6	**77.8**	73.8	80.5	21.1	**35.7**	**66.6**	**58.4**	65.1	56.2	**57.2**

Source: Author's compiled data.

best results for both datasets. In the case of VOC 2007, it surpasses the second-best approach (OD-WSCL) by 0.3 AP, which is also an instance mining-based method and requires much more expensive hardware to achieve such results. For VOC 2012 (<http://host.robots.ox.ac.uk/anonymous/HCQYFV.html>), our results were even more impressive, surpassing the second-best approach (IENet) by 2.3 AP points.

Table 6.4 – Comparison of CorLoc on VOC 2007 trainval set.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	CorLoc
OICR (TANG et al., 2017)	81.7	80.4	48.7	49.5	32.8	81.7	85.4	40.1	40.6	79.5	35.7	33.7	60.5	88.8	21.8	57.9	76.3	59.9	75.3	81.4	60.6
PCL (TANG et al., 2018)	79.6	85.5	62.2	47.9	37.0	83.8	83.4	43.0	38.3	80.1	50.6	30.9	57.8	90.8	27.0	58.2	75.3	68.5	75.7	78.9	62.7
C-MIL (WAN et al., 2019a)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	65.0
BOICR (ZENI; JUNG, 2020b)	86.7	73.3	72.4	55.3	46.9	83.2	87.5	64.5	44.6	76.7	46.4	70.9	67.0	88.0	9.6	56.4	69.1	52.4	79.8	82.8	65.7
WSOD^2 (ZENG et al., 2019)	87.1	80.0	74.8	**60.1**	36.6	79.2	83.8	70.6	43.5	88.4	46.0	74.7	87.4	90.8	**44.2**	52.4	81.4	61.8	67.7	79.9	69.5
OIM (LIN et al., 2020)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	67.2
SLV (CHEN et al., 2020)	84.6	84.3	73.3	58.5	49.2	80.2	87.0	79.4	46.8	83.6	41.8	**79.3**	88.8	90.4	19.5	59.7	79.4	67.7	82.9	83.2	71.0
MIST (REN et al., 2020)	**87.5**	82.4	76.0	58.0	44.7	82.2	87.5	71.2	49.1	81.5	51.7	53.3	71.4	92.8	38.2	52.8	79.4	61.0	78.3	76.0	68.8
CASD (HUANG et al., 2020)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	70.4
EFSL (WU et al., 2024a)	82.5	87.1	74.5	55.3	58.0	82.7	90.0	86.3	48.3	87.0	53.2	79.1	87.4	94.0	23.4	56.8	68.0	68.5	74.9	60.2	70.9
BUAA_PALoss (WU et al., 2023)	84.8	**90.4**	61.7	54.2	50.2	**87.8**	86.0	84.1	50.0	85.9	49.7	73.1	80.6	**95.0**	28.1	62.9	76.3	60.7	83.9	79.1	71.2
LDD-SWA (FENG et al., 2024)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	71.2
MCC-MCT (WU et al., 2024b)	85.8	85.5	70.5	55.2	40.2	86.8	90.4	71.8	49.5	86.9	52.7	72.2	81.8	93.2	25.2	**65.0**	82.4	66.0	84.1	75.5	71.0
CBL (YIN et al., 2023)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	71.8
IENet (FENG et al., 2023)	80.4	87.8	**80.2**	46.3	**64.5**	84.8	**90.7**	**90.7**	50.0	86.3	51.7	68.1	79.9	93.2	32.1	61.5	78.4	69.9	**84.8**	**83.9**	**73.3**
OD-WSCL (SEO et al., 2022)	86.3	87.8	74.5	47.3	43.9	85.8	84.6	78.2	49.1	83.6	49.4	61.6	74.5	92.4	42.2	46.9	80.4	62.1	82.9	82.8	69.8
LTIM (Ours)	81.5	89.2	74.1	58.0	49.6	86.9	87.6	81.9	44.4	**92.9**	**56.7**	64.7	**88.9**	90.1	36.0	55.9	**90.8**	**71.3**	78.0	80.0	72.9

Source: Author's compiled data.

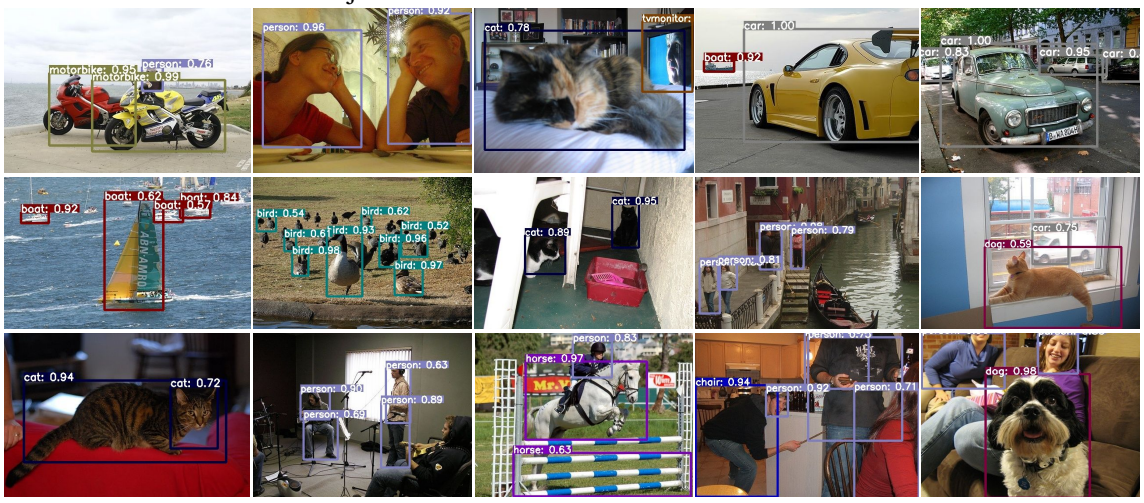
We also include a comparative analysis of the CorLoc performance of our technique against other leading methods for VOC 2007 in Table 6.4 and for VOC 2012 in Table 6.5. It should be noted that while CorLoc has been a long-standing metric for comparing WSOD methods, it does not account for multiple instances during evaluation, which limits its effectiveness for a comprehensive comparison regarding the presence of multiple object instances. Despite this observation, our method achieves a second-place ranking in CorLoc for VOC 2007, and the best CorLoc value for VOC 2012.

Table 6.5 – Comparison of CorLoc on VOC 2012 trainval set.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	CorLoc
OICR (TANG et al., 2017)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	62.1
PCL (TANG et al., 2018)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	63.2
BOICR (ZENI; JUNG, 2020b)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	66.3
WSOD^2 (ZENG et al., 2019)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	71.9
OIM (LIN et al., 2020)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	67.1
SLV (CHEN et al., 2020)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	69.2
MIST (REN et al., 2020)	91.7	85.6	71.7	56.6	55.6	88.6	77.3	63.4	53.6	90.0	51.6	62.6	79.3	94.2	32.7	58.8	90.5	57.7	70.9	**85.7**	70.9
CASD (HUANG et al., 2020)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
EFSL (WU et al., 2024a)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BUAA_PALoss (WU et al., 2023)	90.2	89.9	75.8	48.5	56.2	89.6	84.5	82.9	50.9	86.5	59.6	76.8	82.6	**95.2**	35.0	64.3	80.9	66.5	83.0	81.9	74.0
LDD-SWA (FENG et al., 2024)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	71.9
MCC-MCT (WU et al., 2024b)	**91.8**	**90.7**	71.3	53.2	58.4	90.2	**91.4**	**88.5**	53.2	88.7	**65.7**	74.5	78.1	94.6	20.6	65.2	87.9	67.5	**83.7**	72.7	74.4
CBL (YIN et al., 2023)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
IENet (FENG et al., 2023)	88.0	88.6	**80.0**	58.9	**59.0**	**90.7**	80.7	76.9	**58.1**	82.5	56.0	65.6	75.4	93.3	34.8	**67.6**	88.3	47.6	82.0	75.6	72.5
OD-WSCL (SEO et al., 2022)	88.2	88.3	75.0	59.7	58.9	89.3	73.2	57.8	53.4	88.0	48.7	67.5	78.3	94.0	34.8	61.6	91.7	59.4	70.9	84.4	71.2
LTIM (Ours)	88.2	87.2	76.5	**62.6**	54.4	90.0	77.0	86.4	50.0	90.0	43.8	**81.3**	**88.2**	93.8	**35.5**	63.2	**92.0**	**74.8**	77.3	79.3	**74.6**

Source: Author's compiled data.

Figure 6.7 – Visualisation of the bounding boxes identified by the proposed approach in the VOC 2007 dataset. The upper row demonstrates high quality results. The second row presents results of medium quality, characterized by some omissions or incompletely detected bounding boxes. The bottom row illustrates poor quality results, with redundant bounding boxes for the same object instances or incorrect classifications.



Source: Author's own illustration.

Figure 6.5 presents a visualisation of the detection results of our method applied to the VOC 2007 test set. The top row illustrates perfect or near-perfect results, where all instances were detected with the correct category and suitable bounding boxes. The middle row demonstrates some errors in LTIM, including missing instances, lower-quality bounding boxes, and incorrect detection classifications. Finally, the bottom row shows results where the proposal overlap suppression did not work well (more than one bounding box per instance or bounding boxes that focus on salient parts of objects, such as faces).

As a final comparison experiment, Table 6.6 presents the AP (AP50:95) and AP50 results for the challenging COCO 2014 dataset. Our approach achieves an AP50 of 27.9%, ranking second among all the compared methods. The results for the AP metrics are

Table 6.6 – Comparison of the state-of-the-art methods on COCO 2014 val set.

Method	**AP**	**AP50**
PCL (TANG et al., 2018)	8.5	19.4
WSOD^2 (ZENG et al., 2019)	10.8	22.7
MIST (REN et al., 2020)	11.4	24.3
CASD (HUANG et al., 2020)	12.8	26.4
EFSL (WU et al., 2024a)	10.4	22.8
BUAA_PALoss (WU et al., 2023)	12.4	26.3
MCC-MCT (WU et al., 2024b)	13.0	26.8
CBL (YIN et al., 2023)	13.6	27.6
IENet (FENG et al., 2023)	**14.2**	**29.1**
OD-WSCL (SEO et al., 2022)	<u>13.7</u>	27.7
LTIM (Ours)	12.9	<u>27.9</u>

Source: Author's compiled data.

lower, indicating more errors when tighter IoU thresholds are used (as discussed earlier, this decay can be strongly related to noisy annotations, as shown in (MURRUGARRA-LLERENA; JUNG, 2025)).

The increase in COCO object classes also affected the amount of vRAM needed for training, which limited the maximum batch size to six images (compared to eight in VOC experiments). Although our memory bank delivered comparable COCO results, reducing iterations by two samples will inevitably impact optimisation results. Despite this, our AP50 score remained remarkably high, only surpassed by IENet, highlighting the substantial efficacy of our approach in object detection with standard IoU thresholds.

6.3 Conclusions

This Chapter presented a novel approach for WSOD, focusing on long-term instance discovery and suppression of overlapping proposals. Our main contribution is the use of a discriminative loss function that promotes within-class consistency of instances and intra-class separation, enabling the use of a simple distance-based criterion for mining new instances during training. The set of instances is stored in a memory bank that maintains a repository of the highest-scoring proposals from previous training phases, reducing the projection network's reliance on batch dependency. Additionally, we introduce an overlap suppression strategy to eliminate multiple instances of the same object that are detected. Ablation studies confirmed the impact of our design choices: The memory bank

significantly improves performance compared to the baseline, and the proposal overlap suppression technique further contributes to detection accuracy.

Our experimental findings highlight the success of the proposed method. In the dataset VOC 2007, our method achieved state-of-the-art performance in mean average precision (mAP), outperforming previous methods. The results were even more remarkable on the VOC 2012 dataset, where we exceeded the closest competitor by 2.3% mAP. For the more challengingCOCO dataset, our approach achieved the second-best AP50 value among all compared methods.

7 FINAL REMARKS

7.1 Conclusions

In this dissertation, we developed methodologies to address the challenge of dealing with multiple instances of the same class within an image in the context of Weakly Supervised Object Detection (WSOD). From the weak annotations, we can guarantee that at least one object instance per annotated class will be present in the image. However, the annotations do not provide information about the real counts of object instances for each category in a given image. In this way, we proposed and explored solutions in three different contexts: (i) how to reduce the harmful effects of these problem in WSOD's training process; (ii) exploiting the number of instances that is inherent present in the week annotations of some specific detection tasks (such as Licence Plate Recognition (LPR)); and (iii) building a self-supervised long-term mechanism to discover and mine more instances of the same class in training time.

The methodology presented in Chapter 4 focused on mitigating the adverse effects of these additional instances on the model's optimisation process and the partial covering of the objects. To achieve this, we propose two improvements to enhance the Multiple Instance Learning Network (MIL)-Online Instance Classification Refinement (OICR) framework. The initial enhancement implements a novel adaptive aggregation mechanism that dynamically modifies the Intersection over Union (IoU) selection thresholds to aggregate positive proposals and ignore potential negative ones during the refiner's self-supervised building process. The subsequent enhancement focuses on exploring the knowledge that the model refiners have already acquired to distil new information for self-supervision, enabling the training of an additional refiner module with the objective of "boosting" the detection outcomes. Both improvements enabled our model to be very competitive against the State-of-the-Art (SOTA) methods of the same period. The results of this chapter support our third hypothesis, indicating that it is possible to mitigate the harmful effects of having multiple instances of the same category in the same image.

Next, in Chapter 5, we focused on a problem where information regarding object counts is provided for free in the week annotations and explored this information to our advantage. In this case, we presented a novel weakly supervised approach for character detection in the context of license plate recognition. The primary contributions

include the development of the Instance-Aware Self-Supervision (Instance-Aware Self-Supervision (IASS)) building process, which explores the inherent counting information present in weak annotation strings using an iterative top-scoring selection strategy. Complementing this, the proposed Number of Characters Classification (Number of Characters Classification (NCC)) module provides a practical solution to estimate the total number of characters during inference time. The results demonstrate that weakly supervised learning can be successfully applied in the Licence Plate (LP) recognition domain, offering a practical solution that significantly reduces annotation costs while maintaining competitive performance comparable to that of fully supervised alternatives of the same period. The results of this chapter support our belief that it is possible to use the available counting information inherent in the week annotations of specific tasks to our advantage.

Finally, Chapter 6 presents the most significant contribution of this dissertation: a novel approach to WSOD, focusing on the discovery (mining) of long-term instances and the suppression of overlapping proposals. Our main contribution is the use of a discriminative loss function to train a MLPnetwork that projects the features of the proposals into an embedding space that allows for the discovery of additional instances for the target classes. The loss function enforces within-class consistency and intra-class separation, and the way we build it enables the use of the same threshold value for mining new instances during training. To reduce MLPsupervision dependence on the batch size and to be able to optimise the embedding space with all target classes, we proposed a memory bank that stores the top-scoring proposals from previous iterations. Additionally, we introduced an overlap suppression strategy to prune cases where multiple proposals are detected on the same object instance. Our ablation studies confirmed the impact of our design choices: The memory bank significantly improves performance compared to the baseline, and the proposal overlap suppression technique further contributes to detection accuracy. Also, our method is very competitive with methods of the same period, being top of the rank for the Pascal Visual Object Classes (VOC) 2007 and VOC 2012 datasets, and being the second-best AP50 in the Common Objects in Context (COCO) dataset. The results of this chapter support our first and second hypotheses, indicating that it is possible to discover additional instances of the same category and that information from previous training iterations can help in the mining process.

7.2 Future Work

Although there has been a considerable increase in the results of WSOD methods in the last decade of research, the results are still considerably worse than those of their Fully Supervised Object Detector (FSOD) counterparts, meaning that there is still a long road to get it slightly lower. In the following, we list some directions for future work in WSOD:

- We proposed storing the features ψ (features after the backbone pass) in the feature bank for computational reasons. As the backbone also changes its weights as training advances, it would be more interesting to store the pixels of the RoI of each top-scoring proposal instead. In this way, we avoid using older versions of ψ, which can potentially improve the stability of the model optimisation. However, as each proposal has different dimensions, we would need to: (i) pass each RoI individually into the model, which would make the training process a lot slower, or (ii) develop a new methodology to make this process more computationally efficient. Therefore, developing an efficient proposal's ROI-forwarding methodology could be very beneficial to the model.

- We employ a straightforward proposal overlap suppression algorithm to prune the excessive number of positive proposals after the discovery step. However, a learning approach could make this process more efficient. One possible solution is to train an MLP-based module that learns to differentiate between proposals that belong to the same object instance and those that do not.

- Although we proposed an adaptive aggregation function and a knowledge distillation module, due to time constraints, we were unable to integrate them into our long-term instance mining model. It could be interesting to incorporate them with the mining approach to boost the results.

- Exploring the attention maps in the mining process of our methodology also seems to be a very prominent direction of future research. Based on the results of other methods in the literature, exploring this information can also help to improve the results.

- One problem that we observed mainly in the COCO dataset is that when one object class overlaps with another object, the model tends not to make detections that

overlap, making both detections cover less than necessary for both classes. This phenomenon is widespread among the "person" class, which appears in many situations interacting with other object classes, such as skateboards, benches, snowboards, bottles, and so on. We believe that this could be related to the way pseudo-labels are selected for self-supervision, and that there may be better ways to mitigate this effect in model optimisation.

- One direction that is not explored is to employ Transformers in the context of WSOD. There are very interesting Transformers backbones trained without any supervision that can extract rich image features, such as Dino (CARON et al., 2021) and its variants. The interesting part of these backbones is that when we visualise their activation maps, it seems that they learn the spatial information regarding where the objects are located without any supervision. Unfortunately, Transformers require access to expensive GPU hardware, which we did not have during the development of this dissertation.

- Another interesting direction of investigation is to replace object proposals in the bounding box representation with an oriented bounding box format, as we did in (MURRUGARRA-LLERENA et al., 2024), but for the WSOD context. To the best of our knowledge, this direction has never been explored before.

7.3 Source Code

We made the source code with the implementation of Chapter 4 and Chapter 6 available in the following GitHub repositories:

Chapter 4: https://github.com/luiszeni/Boosted-OICR

Chapter 6: https://github.com/luiszeni/LTIM

7.4 Published and Submitted Papers

The papers produced during the development of this dissertation are listed below.

Papers directly related to this dissertation:

1. Zeni, Luis Felipe, and Claudio R. Jung. "Distilling Knowledge from Refinement in Multiple Instance Detection Networks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops 2020.

2. Zeni, Luis Felipe, and Claudio Jung. "Weakly Supervised Character Detection for License Plate Recognition". 33st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) 2020.

3. Zeni, Luis Felipe, and Claudio Jung. "Weakly Supervised Object Detection with Long-Term Instance Mining". Submitted to IEEE Transactions on Image Processing as a Regular Paper in May 2025.

Papers related to Object Detection with no direct relation with this dissertation:

1. Zeni, Luis Felipe, and Claudio Jung. "Real-Time Gender Detection in the Wild Using Deep Neural Networks." 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) 2018.

2. Murrugarra-Llerena, Jeffri. Kristen, Lucas. Zeni, Luis Felipe, and Claudio Jung. "Probabilistic Intersection-Over-Union for Training and Evaluation of Oriented Object Detectors," in IEEE Transactions on Image Processing, vol. 33, pp. 671-681, 2024.

APPENDIX A — RESUMO EXPANDIDO

A Detecção de Objetos é uma área de pesquisa em visão computacional que visa reproduzir a capacidade do cérebro humano em localizar e reconhecer diferentes objetos. Dada uma imagem de entrada, um detector de objetos deve ser capaz de identificar o número de instâncias de um conjunto conhecido de objetos e de recuperar suas informações espaciais dentro da imagem. Após Krizhevsky, Sutskever e Hinton (2012) demonstrar resultados impressionantes com o uso de uma rede neural convolutiva (CNN) para a tarefa de classificação de imagens, muitos pesquisadores começaram a utilizar esse tipo de rede em pipelines de detecção de objetos e, desde então, modelos de detecção de objetos têm alcançado resultados cada vez melhores em termos de precisão e velocidade (ZOU et al., 2023).

Apesar dos resultados impressionantes alcançados por detectores de objetos recentes, uma desvantagem desses métodos é a quantidade de supervisão humana necessária para treinar tais modelos. Um modelo Detector de Objetos Totalmente Supervisionado requer um conjunto de dados de imagens rigorosamente anotado com bounding boxes (caixas delimitadoras) e rótulos de classe. É crucial notar que compilar um novo conjunto de dados pra treinar um modelo Totalmente Supervisionado é uma tarefa trabalhosa que pode impactar diretamente a viabilidade de desenvolvimento e implantação de aplicações que dependem da detecção de objetos.

Uma abordagem potencial para reduzir custos de anotação é utilizar anotações de categoria no nível da imagem, anotações indicam apenas se classes de interesse estão presentes nas imagens, e omitir o estágio de anotação da bounding box para treinar o detector de objetos. Métodos que exploram essas anotações "fracas" são referenciados na literatura como Detectores de Objetos com Supervisão Fraca (DOSF) e, como esperado, a falta de supervisão de localização durante o processo de treinamento impacta na qualidade de detecção deste métodos.

Na ausência de anotações de caixa delimitadora, a maioria dos métodos Fracamente Supervisionados emprega uma estratégia de aprendizado de múltiplas instâncias para aprender as localizações espaciais das classes alvo. Um método de geração regiões (UIJLINGS et al., 2013; ZITNICK; DOLLÁR, 2014; ARBELÁEZ et al., 2014) é utilizado para gerar um conjunto de propostas de objetos para cada imagem de treinamento. A tarefa do modelo envolve categorizar essas propostas em classes alvo ou como instâncias negativas. Baseado nas anotações no nível da imagem, para cada classe pos-

itiva, pode-se afirmar que pelo menos uma proposta de objeto positiva correspondente deve existir no conjunto de propostas de objetos. Consequentemente, o processo de aprendizado de um modelo DOSF geralmente compreende dois estágios principais: **Estágio 1:** aprender um classificador de instância que atribui uma pontuação a cada categoria alvo para cada proposta de objeto e **Estágio 2:** empregar a informação espacial das propostas com maior pontuação como auto-supervisão para aprender a identificar as propostas restantes como pertencentes a classes positivas ou fundo (não atribuídas).

Nos últimos anos os resultados dos detectores com supervisão fraca melhoraram significativamente (SEO et al., 2022; WU et al., 2024a; WU et al., 2024b; TAHERKHANI; KAZEMI, 2024; FENG et al., 2024). No entanto, dois desafios continuam não totalmente resolvidos:

Lidar com múltiplas instâncias da mesma classe em uma imagem: Ainda é comum que os modelos percam algumas instâncias de objetos ou cobre múltiplas delas na mesma detecção. Parte desses problemas está relacionada à forma como a auto-supervisão é construída, onde instâncias adicionais da mesma classe usualmente são associadas à classe de fundo, o que penaliza o processo de aprendizado.

Ajuste errado da caixa delimitadora: O ajuste da caixa delimitadora com os objetos alvo na imagem permanece subótimo. Às vezes, falha em cobrir todo o objeto, e ocasionalmente excede o tamanho do objeto. Além disso, para certas classes, frequentemente foca no aspecto mais distintivo do objeto.

O objetivo principal desta tese de doutorado é desenvolver metodologias para abordar os desafios descritos acima, especificamente lidando com múltiplas instâncias da mesma classe dentro de uma imagem no contexto de DOSF. As principais contribuições desta tese são elaboradas na língua Inglesa em detalhes nos capítulos 4, 5, e 6, onde cada capítulo aborda diferentes aspectos da pesquisa. A seguir, é apresentado um resumo das contribuições de cada capítulo na língua portuguesa.

Capítulo 4 - Destilando Auto-Supervisão dos Refinadores com uma Função de Agregação de Supervisão Adaptativa

O Capítulo aborda o problema de ajuste inadequado e traz uma solução paliativa para o problema de múltiplas instâncias de mesma classe na mesma imagem. São apresentadas duas melhorias para tanto. O método proposto, denominado *BOICR*, é construído

sobre o framework MIL-OICR e começa extraindo vetores de características no nível da proposta a partir de uma imagem de entrada e propostas de objetos geradas pelo Selective Search (UIJLINGS et al., 2013). Em seguida, o módulo MIL (BILEN; VEDALDI, 2016) é usado para criar a primeira matriz de pontuação na metodologia, que é otimizada por uma função de perda de classificação guiada pelas anotações no nível da imagem. Então, o primeiro módulo refinador OICR (TANG et al., 2017) ($k = 1$) usa a matriz de pontuação MIL para construir sua auto-supervisão usando as propostas com maior pontuação como ponto de partida para selecionar mais instâncias positivas para as categorias de interesse. Os refinadores seguintes ($k > 1$) utilizam a matriz de pontuação do refinador anterior ($k - 1$) para construir sua auto-supervisão.

Ao examinar a sequência de refinadores, pode-se inferir que a matriz de pontuação do refinador $k = 1$ será ligeiramente diferente dos refinadores posteriores $k > 1$, uma vez que cada refinador aprenderá um pouco mais sobre o problema na sequência. Quanto mais distantes os refinadores estão na cadeia, maior pode ser essa diferença. Além disso, como os refinadores estão posicionados em ordem sequencial, os refinadores posteriores não impactam diretamente no processo de aprendizado dos refinadores anteriores e se um refinador na cadeia alucina, ele propagará o erro para os módulos subsequentes. Mantendo esse contexto em mente, propomos um novo módulo, que é apresentado como Destilação de Conhecimento a partir dos refinadores. Este módulo tem como objetivo consolidar e normalizar todo o conhecimento aprendido pelos K refinadores. O módulo de Destilação emprega as mesmas camadas que um refinador padrão. No entanto, ele constrói sua auto-supervisão combinando a matriz de pontuação de todos os K refinadores presentes no modelo.

Além disso, propomos incorporar uma função de agregação de supervisão adaptativa no processo de criação de auto-supervisão do refinador. Esta função altera dinamicamente o limiar de IoU usado para agregar as propostas como categorias alvo, fundo, ou para serem ignoradas no processo de construção de auto-supervisão do refinador. Nosso processo de agregação segue o princípio de que no início do treinamento, é melhor agregar caixas usando um limiar de IoU maior porque as instâncias com maior pontuação são tipicamente pequenas e abrangem uma pequena porção dos objetos de interesse nas iterações iniciais de treinamento. Para prevenir um crescimento excessivo de propostas relacionadas a objetos, o limiar de IoU é reduzido conforme as iterações de treinamento progridem. Também incluímos uma versão adaptativa modificada do “truque” proposto por Tang et al. (2018), que ignora caixas com pequena interseção nas perdas de refina-

mento.

Nos experimentos, avaliamos o impacto que ambas as contribuições tem nos resultados do modelo sobre a arquitetura base MIL-OICR e, ao combinar ambas, obtemos um aumento de 7.4 mAP sobre o modelo base. Nosso modelo BOICR apresenta resultados competitivos com o estado da arte do mesmo período em que foi publicado. Atingindo segundo lugar no dataset VOC 2007, sendo inferior apenas ao método de Wan et al. (2019a) por uma margem de 0.8% mAP. Em contraste, o método proposto fica no topo do ranking do dataset VOC 2012, um dataset muito mais rigoroso, o qual impossibilita o ajuste dos resultados com múltiplos retreinos, pois o conjunto de dados de teste é fechado e só se pode obter os resultados de mAP enviando as detecções para o dataset de avaliação que limita o número de submissões.

Capítulo 5 - Detecção de Caracteres Fracamente Supervisionada para Tarefa de Reconhecimento de Placas de Veículo

O Capítulo apresenta a aplicação do modelo proposto no Capítulo 4 para a tarefa de reconhecimento de caracteres em placas veiculares. Uma característica especial das anotações desta tarefa é que o número de instâncias de cada classe é dado "gratuitamente" na anotação fraca, por exemplo, dada a string de anotação da placa "AIA-3244", podemos facilmente inferir que há dois caracteres "A", dois caracteres "4", e uma instância para o "I", "2" e "3".

A entrada para o método proposto é uma imagem de placa veicular “retificada” que corresponde a (ou emula) uma vista fronto-paralela da placa. Em cenários onde já é capturada em uma pose frontal, qualquer abordagem baseada em caixas delimitadoras pode ser usada. No entanto, alguns métodos podem detectar os quatro cantos da placa (como (XU et al., 2018; SILVA; JUNG, 2018; BJÖRKLUND et al., 2019)), de modo que uma homografia planar pode ser usada para ajustar o recorte da placa detectada para uma vista aproximadamente fronto-paralela. Para localizar os quatro cantos da placa, usamos a abordagem apresentada em (SILVA; JUNG, 2021), que é adequada para placas de veículos e motocicletas.

Propostas de objetos são extraídas das imagens retificadas usando Selective Search (UIJLINGS et al., 2013), tornando possível treinar o modelo proposto Capítulo 4 para aprender a localização espacial dos caracteres usando apenas a string da placa como supervisão. Como as anotações nos dão o número de instâncias gratuitamente, explo-

ramos essa informação propondo duas modificações adicionais ao modelo: (i) uma nova abordagem que aproveita a informação de número de instâncias para extrair instâncias adicionais da mesma classe durante o processo de construção da auto-supervisão do refinador e (ii) um novo módulo para estimar o número de caracteres presentes na placa, que tem como objetivo ter suas predições utilizadas durante a inferência do modelo para melhoria dos resultados.

A avaliação experimental é conduzida em cinco conjuntos de dados de placas veiculares de diversas regiões geográficas, incluindo Brasil, China, Taiwan, Estados Unidos e União Europeia, forneceu validação abrangente da metodologia proposta. Os resultados demonstraram a superioridade da abordagem proposta sobre os modelos de base MIL-OICR e BOICR. Ao adicionar nossas duas contribuições, o modelo alcança uma qualidade de detecção notavelmente alta, semelhante a métodos que usam caixas delimitadoras como supervisão nos conjuntos de dados testados. A técnica demonstrou boas capacidades de generalização, particularmente evidente no desempenho no conjunto de dados UFPR Motorcycle. Este conjunto de dados não foi incluído no tempo de treinamento, e as placas apresentam um layout diferente (duas fileiras de caracteres e proporção diferente). Embora o módulo de estimativa de caracteres tenha alcançado excelente precisão de classificação aproximando-se de 100% para os conjuntos de dados incluídos no conjunto de treinamento, a avaliação também revelou limitações essenciais ao lidar com placas europeias que exibem contagens de caracteres altamente variáveis em diferentes regiões, destacando áreas para investigação futura.

Capítulo 6 - Mineração de Instâncias de Longo Prazo em Detecção de Objetos Fracamente Supervisionada

O capítulo enfrenta o desafio de identificar múltiplas instâncias de objetos de mesma classe na mesma imagem durante o processo de construção de auto-supervisão. No caso, a criação tradicional da auto-supervisão de um refinador é unicamente baseada na proposta de maior score de uma dada classe e propostas que tem um IoU maior que um limiar. As demais propostas são associadas a classe de "background" (não pertencente a nenhuma categoria de interesse). Esta abordagem tem o problema de que, se existirem, as demais instancias desta classe serão associadas ao "background". A atribuição errônea dessas instâncias traz confusão à otimização do modelo, o que poderia levar o modelo a *aprender a omitir algumas instâncias* devido à função de perda impor penalidades nes-

sas instâncias durante o processo de otimização. Não incluir essas instâncias adicionais como positivas também afeta negativamente a capacidade do modelo de generalizar para a categoria especificada, pois o mesmo perde a oportunidade de aprender as características visuais destas outras instâncias, como tamanho, pose e padrões de textura.

Para abordar a questão de *instâncias perdidas* durante a criação da auto-supervisão do refinador, propomos um módulo de mineração de instâncias que projeta as feições das propostas em um novo espaço de embedings, o qual é projetado para descobrir instâncias adicionais de objetos. Durante o treinamento, nossa metodologia armazena as propostas de maior pontuação para cada categoria alvo em um banco de memória. Em cada iteração de treinamento, o banco é atualizado com novas instâncias e instâncias mais antigas são removidas. Uma interpretação deste mecanismo é que ele serve como um *repositório de memória de longo prazo* contendo os *melhores* (maior pontuação) exemplos de cada categoria de interesse que o refinador encontrou em iterações de treinamento anteriores. As amostras armazenadas no banco são empregadas como amostras de supervisão para aprendizado para uma rede MLP, que é treinada usando uma função de perda discriminativa inovadora no contexto de DOFS.

O objetivo de aprendizado da rede MLP é projetar características que pertencem à mesma categoria em embeddings similares contidos dentro de um subespaço esférico. A função de perda discriminativa impõe que instâncias da mesma classe sejam contidas dentro da mesma esfera e que esferas de classes diferentes não se sobreponham usando uma margem mínima aceitável entre elas. Dois parâmetros governam este processo: um especifica o diâmetro máximo da esfera, e o outro determina a margem entre esferas. Essencialmente, a rede deve aprender a agrupar as feições intra-classe enquanto distancia feições inter-classe.

As restrições impostas no processo de aprendizado do espaço de agrupamento esférico permitem o uso das feições projetadas pela MLP para realizar a descoberta de instâncias de objetos adicionais, que podem ser incluídas como instâncias positivas na fase de construção de auto-supervisão. O processo de descoberta inicia projetando todas as feições das propostas neste espaço de agrupamento esférico com o uso da MLP. Como o centroide de cada esfera é conhecido, o diâmetro da esfera pode ser usado como um limiar para selecionar instâncias positivas de cada categoria. Em nossos experimentos notamos que, enquanto a mineração de instâncias adicionais identifica com sucesso propostas positivas de outras instâncias, ela pode introduzir um o problema de termos um número excessivo de instâncias positivas, implicando que várias subpartes das classes

alvo podem ser erroneamente incluídas como positivas, o que também pode afetar adversamente o processo de aprendizado.

No contexto de ter um número excessivo de instâncias positivas após o passo de descoberta, também propusemos uma estratégia de supressão baseada na sobreposição das instancias. Ela tem o objetivo de reduzir essa questão de ter inúmeras propostas contidas dentro da mesma instância de objeto. As principais contribuições deste capítulo são resumidas na lista abaixo.

- Introdução de uma MLP de projeção treinada com uma função de perda discriminativa que garante coerência intra-classe e diferenciação inter-classe.

- Introdução de um banco de memória que registra uma coleção das propostas de maior pontuação mais recentes de iterações de treinamento anteriores, que são usadas como supervisão para treinar a MLP.

- Desenvolvimento de uma estratégia de supressão de sobreposição de propostas para remover algumas instâncias descobertas que pertencem ao mesmo objeto.

O método proposto, nomeado LTIM, alcança as melhores pontuações de AP para os conjuntos de dados VOC 2007 e VOC 2012 e ocupa o segundo lugar para AP50 no conjunto de dados COCO 2014, enquanto também demanda menos recursos de hardware comparado a outras estratégias de mineração de instâncias.

APPENDIX — REFERENCES

ARBELÁEZ, P. et al. Multiscale combinatorial grouping. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2014. p. 328–335.

BAEK, J. et al. What is wrong with scene text recognition model comparisons? dataset and model analysis. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 4715–4723.

BILEN, H.; VEDALDI, A. Weakly Supervised Deep Detection Networks. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, v. 2016-Decem, p. 2846–2854, 2016.

BJÖRKLUND, T. et al. Robust license plate recognition using neural networks trained on synthetic images. **Pattern Recognition**, v. 93, p. 134–146, 2019. ISSN 00313203.

BRABANDERE, B. D.; NEVEN, D.; GOOL, L. V. Semantic instance segmentation with a discriminative loss function. **arXiv preprint arXiv:1708.02551**, 2017.

CARON, M. et al. Emerging properties in self-supervised vision transformers. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2021. p. 9650–9660.

CHEN, Z. et al. SLV: Spatial likelihood voting for weakly supervised object detection. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2020. p. 12995–13004.

CHEN, Z. et al. Clip-driven transformer for weakly supervised object localization. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, 2025.

CHENG, G. et al. High-quality proposals for weakly supervised object detection. **IEEE Transactions on Image Processing**, IEEE, v. 29, p. 5794–5804, 2020.

CHENG, Z. et al. Focusing attention: Towards accurate text recognition in natural images. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2017. p. 5076–5084.

CINBIS, R. G.; VERBEEK, J.; SCHMID, C. Weakly supervised object localization with multi-fold multiple instance learning. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 39, n. 1, p. 189–203, 2016.

DALAL, N.; TRIGGS, B. Histograms of Oriented Gradients for Human Detection. In: SCHMID, C.; SOATTO, S.; TOMASI, C. (Ed.). **International Conference on Computer Vision & Pattern Recognition (CVPR '05)**. San Diego, United States: IEEE Computer Society, 2005. v. 1, p. 886–893. Disponível em: <https://hal.inria.fr/inria-00548512>.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 248–255.

DESELAERS, T.; ALEXE, B.; FERRARI, V. Weakly supervised localization and learning with generic knowledge. **International journal of computer vision**, Springer, v. 100, n. 3, p. 275–293, 2012.

DIBA, A. et al. Weakly supervised cascaded convolutional networks. **Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017**, v. 2017-Janua, p. 5131–5139, 2017. ISSN 1063-6919.

EVERINGHAM, M. et al. **The PASCAL Visual Object Classes Challenge 2007 (VOC2007)**. 2007. Http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

EVERINGHAM, M. et al. **The PASCAL Visual Object Classes Challenge 2012 (VOC2012)**. 2012. Http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

FELZENSZWALB, P.; MCALLESTER, D.; RAMANAN, D. A discriminatively trained, multiscale, deformable part model. In: IEEE. **2008 IEEE conference on computer vision and pattern recognition**. [S.l.], 2008. p. 1–8.

FENG, X. et al. Learning an invariant and equivariant network for weakly supervised object detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 45, n. 10, p. 11977–11992, 2023.

FENG, Y. et al. Refining and reweighting pseudo labels for weakly supervised object detection. **Neurocomputing**, Elsevier, v. 577, p. 127387, 2024.

GAO, M. et al. C-wsl: Count-guided weakly supervised localization. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 152–168.

GAO, Y. et al. C-midn: Coupled multiple instance detection network with segmentation guidance for weakly supervised object detection. In: **Proceedings of the IEEE/CVF international conference on computer vision**. [S.l.: s.n.], 2019. p. 9834–9843.

GIRSHICK, R. Fast r-cnn. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 1440–1448.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. MIT Press, 2016. (Adaptive Computation and Machine Learning series). ISBN 9780262035613. Disponível em: <https://books.google.com.br/books?id=Np9SDQAAQBAJ>.

GUPTA, A.; DOLLAR, P.; GIRSHICK, R. LVIS: A dataset for large vocabulary instance segmentation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019.

HE, K. et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 37, n. 9, p. 1904–1916, 2015.

HE, K. et al. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.

HE, K. et al. Deep residual learning for image recognition. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016.

HINTON, G.; VINYALS, O.; DEAN, J. Distilling the knowledge in a neural network. **arXiv preprint arXiv:1503.02531**, 2015.

HOWARD, A. G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017.

HSU, G.-S.; CHEN, J.-C.; CHUNG, Y.-Z. Application-oriented license plate recognition. **IEEE transactions on vehicular technology**, 2012.

HUANG, G. et al. Densely connected convolutional networks. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2017.

HUANG, Z. et al. Comprehensive attention self-distillation for weakly-supervised object detection. **Advances in neural information processing systems**, v. 33, p. 16797–16807, 2020.

KIRILLOV, A. et al. Segment anything. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [S.l.: s.n.], 2023. p. 4015–4026.

KRIESEL, D. A brief introduction to neural networks. dkriesel. com. 2011.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 1097–1105.

KUZNETSOVA, A. et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. **International journal of computer vision**, Springer, v. 128, n. 7, p. 1956–1981, 2020.

LAROCA, R. et al. A robust real-time automatic license plate recognition based on the YOLO detector. **CoRR**, abs/1802.09567, 2018.

Laroca, R. et al. A robust real-time automatic license plate recognition based on the YOLO detector. In: **International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2018. p. 1–10. ISSN 2161-4407.

Laroca, R. et al. An efficient and layout-independent automatic license plate recognition system based on the YOLO detector. **arXiv preprint**, arXiv:1909.01754, p. 1–14, 2019.

LI, D. et al. Weakly Supervised Object Localization with Progressive Domain Adaptation. **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, p. 3512–3520, 2016. ISSN 10636919.

LIN, C. et al. Object instance mining for weakly supervised object detection. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2020. v. 34, p. 11482–11489.

LIN, T. et al. Microsoft COCO: common objects in context. **CoRR**, abs/1405.0312, 2014. Disponível em: <http://arxiv.org/abs/1405.0312>.

LIN, T.-Y. et al. Microsoft COCO: Common objects in context. In: **ECCV**. [S.l.: s.n.], 2014. p. 740–755.

LOWE, D. G. et al. Object recognition from local scale-invariant features. In: **iccv**. [S.l.: s.n.], 1999. v. 99, n. 2, p. 1150–1157.

MURRUGARRA-LLERENA, J.; JUNG, C. R. Noise-aware evaluation of object detectors. In: IEEE. **2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)**. [S.l.], 2025. p. 9322–9331.

MURRUGARRA-LLERENA, J. et al. Probabilistic intersection-over-union for training and evaluation of oriented object detectors. **IEEE Transactions on Image Processing**, v. 33, p. 671–681, 2024.

NIELSEN, M. A. **Neural networks and deep learning**. [S.l.]: Determination Press USA, 2015.

PAPERT, S. A. The summer vision project. 1966.

RADFORD, A. et al. Learning transferable visual models from natural language supervision. In: PMLR. **International conference on machine learning**. [S.l.], 2021. p. 8748–8763.

RADOSAVOVIC, I. et al. Designing network design spaces. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2020.

RAVI, N. et al. Sam 2: Segment anything in images and videos. **arXiv preprint arXiv:2408.00714**, 2024.

REN, S. et al. Faster r-cnn: Towards real-time object detection with region proposal networks. **arXiv preprint arXiv:1506.01497**, 2015.

REN, Z. et al. Instance-aware, context-focused, and memory-efficient weakly supervised object detection. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2020. p. 10598–10607.

RUSSAKOVSKY, O. et al. Object-centric spatial pooling for image classification. **Computer Vision–ECCV 2012**, Springer, p. 1–15, 2012.

SEO, J. et al. Object discovery via contrastive learning for weakly supervised object detection. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2022. p. 312–329.

SILVA, S. M.; JUNG, C. R. Real-time brazilian license plate detection and recognition using deep convolutional neural networks. In: **2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.: s.n.], 2017. p. 55–62.

SILVA, S. M.; JUNG, C. R. License plate detection and recognition in unconstrained scenarios. In: **2018 European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 580–596.

SILVA, S. M.; JUNG, C. R. Real-time license plate detection and recognition using deep convolutional neural networks. **Journal of Visual Communication and Image Representation**, Elsevier, p. 102773, 2020.

SILVA, S. M.; JUNG, C. R. A flexible approach for automatic license plate recognition in unconstrained scenarios. **IEEE Transactions on Intelligent Transportation Systems**, 2021. Accepted for publication.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.

SIVA, P. et al. Looking beyond the image: Unsupervised learning for object saliency and detection. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2013. p. 3238–3245.

SONG, H. O. et al. Weakly-supervised discovery of visual pattern configurations. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2014. p. 1637–1645.

STEVENS, E.; ANTIGA, L.; VIEHMANN, T. **Deep Learning with PyTorch**. Manning Publications, 2020. ISBN 9781617295263. Disponível em: <https://books.google.com.tr/books?id=89BlwwEACAAJ>.

SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016.

TAHERKHANI, F.; KAZEMI, E. Instance-dependent noise refinement in segment anything model for weakly supervised object detection. In: **Proceedings of the Asian Conference on Computer Vision**. [S.l.: s.n.], 2024. p. 4505–4525.

TAN, M.; LE, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. In: **International Conference on Machine Learning (ICML)**. [S.l.: s.n.], 2019. p. 6105–6114.

TANG, P. et al. Pcl: Proposal cluster learning for weakly supervised object detection. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, 2018.

TANG, P. et al. Multiple instance detection network with online instance classifier refinement. **Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017**, v. 2017-January, p. 3059–3067, 2017.

TANG, P. et al. Weakly supervised region proposal network and object detection. In: **Proceedings of the European conference on computer vision (ECCV)**. [S.l.: s.n.], 2018. p. 352–368.

TEH, E. W.; ROCHAN, M.; WANG, Y. Attention networks for weakly supervised object localization. In: **British Machine Vision Conference**. [S.l.: s.n.], 2016. p. 1–11.

UIJLINGS, J. R. et al. Selective search for object recognition. **International journal of computer vision**, Springer, v. 104, p. 154–171, 2013.

VIOLA, P.; JONES, M. J. Robust real-time face detection. **International journal of computer vision**, Springer, v. 57, n. 2, p. 137–154, 2004.

WAN, F. et al. C-MIL: Continuation Multiple Instance Learning for Weakly Supervised Object Detection. **The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, v. 1, p. 2199–2208, 2019.

WAN, F. et al. Min-Entropy Latent Model for Weakly Supervised Object Detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, p. 1–1, 2019. ISSN 0162-8828.

WANG, T. et al. Decoupled attention network for text recognition. In: **AAAI**. [S.l.: s.n.], 2020. p. 12216–12224.

WEI, Y. et al. Ts2c: Tight box mining with surrounding segmentation context for weakly supervised object detection. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 434–450.

WU, Z. et al. Selecting high-quality proposals for weakly supervised object detection with bottom-up aggregated attention and phase-aware loss. **IEEE Transactions on Image Processing**, v. 32, p. 682–693, 2023.

WU, Z. et al. Enhanced spatial feature learning for weakly supervised object detection. **IEEE Transactions on Neural Networks and Learning Systems**, v. 35, n. 1, p. 961–972, 2024.

WU, Z. et al. Misclassification in weakly supervised object detection. **IEEE Transactions on Image Processing**, v. 33, p. 3413–3427, 2024.

XU, Y. et al. Pyramidal multiple instance detection network with mask guided self-correction for weakly supervised object detection. **IEEE Transactions on Image Processing**, IEEE, v. 30, p. 3029–3040, 2021.

XU, Z. et al. Towards end-to-end license plate detection and recognition: A large dataset and baseline. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 255–271.

YIN, Y. et al. Cyclic-bootstrap labeling for weakly supervised object detection. In: **Proceedings of the IEEE/CVF International Conference on Computer Vision**. [S.l.: s.n.], 2023. p. 7008–7018.

ZENG, Z. et al. Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In: **Proceedings of the IEEE/CVF international conference on computer vision**. [S.l.: s.n.], 2019. p. 8292–8300.

ZENI, L. F.; JUNG, C. Weakly supervised character detection for license plate recognition. In: **2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**. [S.l.: s.n.], 2020. p. 218–225.

ZENI, L. F.; JUNG, C. R. Distilling knowledge from refinement in multiple instance detection networks. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops**. [S.l.: s.n.], 2020. p. 768–769.

ZHANG, S.; WANG, Z.; KE, W. One point is all you need for weakly supervised object detection. **Pattern Recognition**, Elsevier, v. 159, p. 111087, 2025.

ZHU, L. et al. Weaksam: Segment anything meets weakly-supervised instance-level recognition. In: **Proceedings of the 32nd ACM international conference on multimedia**. [S.l.: s.n.], 2024. p. 7947–7956.

ZITNICK, C. L.; DOLLÁR, P. Edge boxes: Locating object proposals from edges. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2014. p. 391–405.

ZOU, Z. et al. Object detection in 20 years: A survey. **Proceedings of the IEEE**, v. 111, n. 3, p. 257–276, 2023.