Universitat
de les Illes Balears

DOCTORAL THESIS
2021

NOVEL DEEP LEARNING-BASED IDENTIFICATION METHODS FOR ACCURATE, ORIENTATION-AWARE VISUAL DETECTION WITH APPLICATION TO INSPECTION AND QUALITY CONTROL

KAI YAO

Universitat
de les Illes Balears

DOCTORAL THESIS
2021

Doctoral Programme of Computer Science

NOVEL DEEP LEARNING-BASED IDENTIFICATION METHODS FOR ACCURATE, ORIENTATION-AWARE VISUAL DETECTION WITH APPLICATION TO INSPECTION AND QUALITY CONTROL

KAI YAO

Thesis Supervisor: Dr. Alberto Ortiz Rodríguez
Co-Supervisor: Dr. Francisco Bonnín Pascual

Doctor by the Universitat de les Illes Balears

Statement of Authorship

This thesis has been submitted to the *Escola de Doctorat*, *Universitat de les Illes Balears*, in fulfilment of the requirements for the degree of *Doctor por la Universitat de les Illes Balears*. I hereby declare that, except where specific reference is made to the work of others, the content of this dissertation is entirely my own work, describes my own research and has not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

KAI YAO

Palma de Mallorca, November, 2021

Funding

The work reported in this thesis was supported by project IMABIA (Ref. PROCOE/4/2017, Govern Balear, 50% funded by Programa Operatiu FEDER 2014-2020 de les Illes Balears), by the EU-H2020 projects BUGWRIGHT2 (GA 871260) and ROBINS (GA 779776), and by project FUZZYMAR (Ref. PGC2018-095709-B-C21, MCIU/AEI/FEDER, UE).

Supervisor Agreement

Alberto Ortiz Rodríguez, Ph.D. in Computer Science and Associate Professor at the *Department of Mathematics and Computer Science*, *Universitat de les Illes Balears*. Francisco Bonnín Pascual, Ph.D. in Computer Science and Lecturer at the *Department of Mathematics and Computer Science*, *Universitat de les Illes Balears.*

ATTEST THAT:

this dissertation, titled *Novel deep learning-based indentification methods for accurate, orientation-aware visual detection with application to inspection and quality control* and submitted by KAI YAO for obtaining the degree of *Doctor por la Universitat de les Illes Balears*, was carried out under our supervision and contains enough contributions to be considered as a doctoral thesis.

Dr. Alberto Ortiz Rodríguez Dr. Francisco Bonnín Pascual

Palma de Mallorca, November, 2021

Abstract

Machine vision systems have emerged as a superior alternative to human labor in industrial applications, and are still being developed on the way to an agile and flexible industry. With its capability to achieve high accuracy while ensuring high throughput on the production line, machine vision systems have also helped to adopt inspection and quality control processes efficiently. Over the past few years, machine vision systems making use of Deep Learning (DL) methodologies have been reported to be able to achieve high performance, producing consistent and accurate detections in various vision tasks by means of Deep Convolutional Neural Networks (DCNN).

In this dissertation we approach target detection/recognition problems from two different points of view, namely bounding boxes regression and semantic segmentation using DCNNs, and validate them by means of two industry-related applications connected with visual inspection and quality control. Both solutions try to produce efficient detections, either by design through regression of rotated bounding boxes, or by means of individual pixel labelling, the way how semantic segmentation adapts, by nature, to the shape of the target.

In the first part of this thesis, a two-stage solution for object recognition is proposed. In this work, a Feature Pyramid Architecture based on the Single Shot Multi-box Detector is developed to infer unrotated bounding boxes. Subsequently, a lightweight regression network is designed to provide the rotated bounding boxes-based detections on the basis of the resulting unrotated bounding boxes. In the second part of this thesis, pixel-level classification solutions using full and weak annotations are developed. Regarding the fully supervised solution, a fully convolutional network is trained using different loss functions, what aims at solving the detection problem for small area targets. As for the weakly supervised semantic segmentation approach, a novel loss function is proposed to counteract the effects of weak annotations. At last, we test several simple strategies to combine the unoriented bounding boxes detection and semantic segmentation approaches in order to get better performance compared with applying the two methods independently. All methods mentioned before are evaluated using datasets from the two vision tasks considered in this dissertation. The results obtained allow us to confirm the competitive performance achieved by the methods developed.

Resumen

Los sistemas de visión artificial han surgido como alternativas competitivas en aplicaciones industriales, siendo potenciadas actualmente en el camino hacia una industria ágil y flexible. Con su capacidad para lograr una alta precisión al tiempo que garantiza un alto rendimiento en la línea de producción, los sistemas de visión artificial también han ayudado a adoptar procesos de inspección y control de calidad de manera eficiente. En los últimos años, se ha informado que los sistemas de visión artificial que utilizan metodologías de aprendizaje profundo (DL) pueden lograr un alto rendimiento, produciendo detecciones consistentes y precisas en diversas tareas de visión por computdor mediante redes neuronales convolucionales profundas (DCNN).

En esta tesis, abordamos los problemas de detección / reconocimiento de objetivos desde dos puntos de vista diferentes: regresión de cajas circundantes (bounding boxes) y segmentación semántica, ambos utilizando DCNN. Ambos enfoques se validan mediante dos aplicaciones conectadas con la industria, relacionadas con la inspección visual y el control de calidad. Ambas soluciones intentan producir detecciones eficientes, ya sea por diseño a través de la regresión de cajas circundantes rotadas, o por medio del etiquetado de píxeles individuales, la forma a través de la que la segmentación semántica se puede adaptar, por naturaleza, a la forma de los objetos.

En la primera parte de esta tesis, se propone una solución para el reconocimiento de objetos en dos etapas. En este trabajo, se desarrolla una arquitectura piramidal basada en el método Single-Shot multi-box Detector, con el objetivo de inferir cajas circundantes no rotadas. Posteriormente, desarrollamos una red de regresión sencilla para inferir las cajas rotadas sobre la base de las cajas no rotadas resultantes de la primera etapa. En la segunda parte de esta tesis, desarrollamos soluciones de segmentación a nivel de píxel que utilizan anotaciones completas y débiles. En cuanto a la solución totalmente supervisada, entrenamos una red totalmente convolucional utilizando diferentes funciones de pérdida, con el objetivo de resolver el problema de detección de objetivos de área pequeña. En cuanto al enfoque de segmentación semántica débilmente supervisada, proponemos una función de pérdida novedosa para contrarrestar los efectos de las anotaciones débiles. Por último, probamos varias estrategias simples para combinar la detección de cajas circundantes no orientados con enfoques de segmentación semántica con el fin de obtener mejor rendimiento en comparación con la aplicación de los dos métodos de forma independiente. Todos los métodos mencionados anteriormente se evalúan utilizando datasets de las dos tareas de visión consideradas en esta tesis. Los

resultados obtenidos nos permiten confirmar un rendimiento competitivo por parte de los métodos desarrollados.

Resum

Els sistemes de visió artificial han sorgit com alternatives competitives en aplicacions industrials, sent potenciades actualment en el camí cap a una indústria àgil i flexible. Amb la seva capacitat per aconseguir una alta precisió al temps que garanteix un alt rendiment en la línia de producció, els sistemes de visió artificial també han ajudat a adoptar processos d'inspecció i control de qualitat de manera eficient. En els últims anys, s'ha informat que els sistemes de visió artificial que utilitzen metodologies d'aprenentatge profund (DL) poden aconseguir un alt rendiment, produint deteccions consistents i precises en diverses tasques de visió per computador mitjançant xarxes neuronals convolucionals profundes (DCNN).

En aquesta tesi, abordem els problemes de detecció / reconeixement d'objectius des de dos punts de vista diferents: regressió de caixes circumdants (bounding boxes) i segmentació semàntica, ambdues utilitzant DCNN. Tots dos enfocaments es validen mitjançant dues aplicacions connectades amb la indústria, relacionades amb la inspecció visual i el control de qualitat. Les dues solucions intenten produir deteccions eficients, ja sigui per disseny a través de la regressió de caixes circumdants rotades, o per mitjà de l'etiquetatge de píxels individuals, la forma mitjançant la qual la segmentació semàntica es pot adaptar, per naturalesa, a la forma dels objectes.

A la primera part d'aquesta tesi, es proposa una solució per al reconeixement d'objectes en dues etapes. En aquest treball, es desenvolupa una arquitectura piramidal basada en el mètode Single-Shot multi-box Detector, amb l'objectiu d'inferir caixes circumdants no rotades. Posteriorment, desenvolupam una xarxa de regressió senzilla per inferir les caixes rotades sobre la base de les caixes no rotades resultants de la primera etapa. A la segona part d'aquesta tesi, desenvolupem solucions de segmentació a nivell de píxel que utilitzen anotacions completes i febles. Pel que fa a la solució totalment supervisada, entrenem una xarxa totalment convolucional utilitzant diferents funcions de pèrdua, amb l'objectiu de resoldre el problema de detecció d'objectius d'àrea petita. Pel que fa a l'enfocament de segmentació semàntica feblement supervisada, proposem una funció de pèrdua nova per contrarestar els efectes de les anotacions febles. Finalment, provam diverses estratègies simples per combinar la detecció de caixes circumdants no orientades amb enfocaments de segmentació semàntica per tal d'obtenir millor rendiment en comparació amb l'aplicació dels dos mètodes de forma independent. Tots els mètodes anteriorment esmentats s'avaluen utilitzant datasets de les dues tasques de visió considerades en aquesta tesi. Els resultats obtinguts ens permeten confirmar un rendiment

competitiu per part dels mètodes desenvolupats.

Dedicated to my wife and my son.

Acknowledgements

My deep gratitude goes first and foremost to Alberto Ortiz Rodríguez and Francisco Bonnín Pascual, my supervisor and my co-supervisor, for their constant encouragement and guidance. They have walked me through all the stages of the writing of this thesis. Without their consistent and illuminating instruction, this thesis could not have reached its present form. I would also like to thank them for their interesting discussion with me, which I have found very informative and useful.

Sincerely appreciate Gabriel Oliver Codina and Alberto Ortiz Rodríguez to give me the opportunity to come to the University of Balearic Islands to work. When I first came to the University of Balearic Islands, I just finished the work experience of software development and programming for embedding systems. I had no idea what the thesis was all about. I would like to thank Alberto Ortiz Rodríguez for devoting a lot of time and effort to introduce the background of my thesis.

Second, I would like to express my heartfelt gratitude to Francisco Bonnín Pascual and Joan Pep Company Corcoles, who help me a lot in my life. As a foreigner, they assist me in solving the life problems and tell me the living habits in Mallorca.

Third, high tribute shall be paid to Emilio Garcia Fidalgo and Alberto Ballesteros, who introduced to me a group of enthusiastic and amiability friends to play football. Special thanks should go to all of the members in the SRV group who have put considerable time and effort in to help me solve problems in my life and my work.

At last, I would like to express my sincere and deepest appreciation to my family, especially to my wife and my son, who supports me to finish the thesis and gives me the motivation to continue researching. I would like to thank my parents for all their love and support throughout my entire life.

Funding

The work reported in this thesis was supported by project IMABIA (Ref. PROCOE/4/2017, Govern Balear, 50% funded by Programa Operatiu FEDER 2014-2020 de les Illes Balears), by the EU-H2020 projects BUGWRIGHT2 (GA 871260) and ROBINS (GA 779776), and by project FUZZYMAR (Ref. PGC2018-095709-B-C21, MCIU/AEI/FEDER, UE).

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AG Attention Gate

ANN Artificial Neural Network

ASPP Atrous Spatial Pyramid Pooling

AUN Attention U-Net

AP Average Precision

BBox (unoriented) Bounding Box

BBox-Seg Use of BBox detection results to improve semantic segmentation

BRNN Bi-directional RNN

CE Cross Entropy

CNN Convolution Neural Network

CRF Conditional Random Field

CART Classification and Regression Tree

DL Dice Loss

DCNN Deep Convolution Neural Network

ENet Efficient Net

FC Fully Connected Layer

FP False Positive

FN False Negative

FL Focal Loss

FCN Fully Conditional Neural Network

FLDA Fisher Linear Discriminant Analysis

GAN Generatice Adversarial Network

HD Hausdorff Distance

FPN Feature Pyramid Network

FFT Fast Fourier Transform

FIS Fuzzy Inference System

GT Ground Truth

GLCM Gray-Level Co-occurrence Matrix

HOG Histogram of Oriented Gradients

HSI Hue Saturation Intensity

HSV Hue Saturation Value

IOU Intersecion over Union

KL Kullback-Leibler

LVQ Learning Vector Quantification

LSTM Long Short-Term Memory

MSE Mean square error

MLP Multi-Layer Perceptron

kNN k-Nearest Neighbours

MCMC Markov chain Monte Carlo

NN Neural Network

NMS Non Maximum Suppression

Prec Precision

RNN Recurrent Neural Network

RBox (Arbitarily) Rotated Bounding Box

RPN Region Proposal Network

RRPN Rotational Region Proposal Network

Rec Recall

ROI Region of Interest

R-CNN Region-CNN

R-FCN Region-based Fully Convolutional Network

RBF Radial Basis Function

SegMap Semantic segmentation results

Seg-BBox Use of SegMap to improve BBox Detection

SSD Single Shot Multibox Detector

SLAM Simultaneously Localization and Mapping

SGD Stochastic Gradient Descent

SOM Self Organizing Mapping

SVM Support Vector Machine

UIB University of the Balearic Islands

UAV Unmanned Aerial Vehicle

WSSS Weakly Supervised Semantic Segmentation

Chapter 1

Introduction

This chapter defines the problems that are considered in this thesis. To this end, Section 1.1 introduces briefly the field of computer vision and refers to the problems addressed, while Section 1.2 focuses on the two application cases that will be used as a benchmark to evaluate the performance of the vision methods that have been developed. Next, Section 1.3 enumerates the goals to attain, and Section 1.4 reviews the contributions that have resulted along the way to achieve the intended goals. Then, Section 1.5 introduces the scope of this research, and Section 1.6 gives an overview about the publications in this thesis. To finish, Section 1.7 outlines the structure of this document.

1.1 Computer Vision

Computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos. Broadly speaking, it seeks to understand and automate tasks that the human visual system can perform: just like humans, as we use our eyes and brains to understand the world around us, artificial vision systems aim at producing the same effect, so that computers can perceive and understand an image or sequence of images and act as appropriate in a given situation. From this generic approach, machine vision has found application as both standalone, autonomous systems *per se* that process images and provide data extracted from the operating environment, e.g. count the number of people in a crowded space, or as another piece of a more complex system, e.g. in robotics, as part of the perception system of a robotic platform, such as in autonomous driving.

From an engineering point of view, computer vision comprises methods for acquiring, processing, analyzing and understanding images of the real world, aiming at producing numerical or symbolic information to be used directly, or as input to another module

in a processing pipeline. Nowadays, steady progress in vision research in all the aforementioned subdomains, coupled with advances in computer technology, are fueling the demand for computer vision systems for a variety of applications.

In this thesis, we deal with the detection and/or recognition of objects of interest in generic scenes. One of the classical solutions to this problem has been image segmentation, whose goal is to distinguish meaningful units in processed images. Once these units have been found, one step further identifies each belonging to a particular class among a set of classes to be recognized, giving rise to the Multi-Class Semantic Segmentation (MCSS) task. From firstly-proposed methods (e.g. region growing [1]) to more robust methods (e.g. level-set [2] and graph-cut [3]), various techniques have been proposed to achieve automatic image segmentation in a wide range of problems. Nevertheless, it has not been until recently that the performance of image segmentation algorithms has attained truly competitive levels, and this has been mostly thanks to the power of machine learning-based methodologies.

On the basis of the concept of Convolutional Neural Networks (CNN) proposed by LeCun and his collaborators (e.g. in the form of the well-known LeNet networks [4]) and followed by the technological breakthrough that allowed training artificial neural structures with a number of parameters amounting to millions [5], deep CNNs have demonstrated remarkable capabilities for problems so complex as image classification, multi-instance multi-object detection or multi-class semantic segmentation. And all this has been accomplished because of the learning the representation capacity of CNNs, embedded in the set of multi-scale feature maps defined in their architecture through non-linear activation functions and a number of convolutional filters that are automatically learnt during the training process by means of iterative back-propagation of prediction errors between current and expected output.

In this thesis, we adopt deep learning-based methodologies to produce novel object detection and recognition methods. The approaches that we follow do not restrict the kind of object to look for in any respect, nor the operating environment where they can be located. The methods produced are therefore intended to be object and environment-agnostic. Nonetheless, we make use of an evaluation benchmark comprising two industry-related applications, the details of which are provided extensively in section 1.2. These two application cases have been selected (1) because of the increasing interest in machine vision-based inspection and control quality applications, and (2) because of the different nature of the two cases and the varied appearance of the objects involved.

1.2 Machine Vision for Visual Inspection and Quality Control

Machine vision systems are emerging as progressively popular solutions to automated on-line quality control and visual inspection applications. Nowadays, with the rise of artificial intelligence, machine vision systems are playing an essential role in modern industry [6–9]. This thesis considers two potential applications of machine learning as part of an evaluation benchmark for the methods here developed. In the first case, a visual inspection system is used for automated corrosion detection on vessel metallic structures. In this task, we need to deal with the automatic detection of one of the most common defects that can affect steel surfaces of large-tonnage vessels, i.e., coating breakdown and/or corrosion in any of its many different forms. In the second case, we deal with the detection of a number of control elements that the sterilization unit of a hospital places in boxes and bags containing surgical tools that surgeons and nurses have to be supplied with prior to starting surgery. These elements provide evidence that the tools have been properly processed after the previous use.

Regarding the first application, it is well known that the overall structure of merchant vessels mostly relies on high-strength steel elements which, in the marine environment, are prone to be affected by corrosion. This defective situation may lead to substantial losses and hidden dangers to crew members and, for this reason, vessels must undergo periodical inspections to evaluate the state of all its metallic structures. By way of example, Fig. 1.1 provides some images of vessel metallic elements affected by corrosion.

Concerning the second application, all surgery tools must undergo cleaning processes before surgery. For this reason, they are introduced inside sterilization boxes fitted with elements used to prove the proper preparation of all the tools included within. This set of elements is shown in Fig. 1.2 and comprises (from left to right and from top to bottom) a label with a barcode used to track the box/bag, a yellow seal to guarantee the box is closed until use, three kinds of paper tape that present a black-, blue- or pink-stripped pattern after sterilization (the white tape changes to black, and the blue tape changes to pink once submitted to the autoclave's temperature) and an internal filter which is placed inside some boxes and that creates a white-dotted pattern when present (the pattern is black-dotted when the filter is missing).

Nowadays both applications are established tasks that require special workers or experts to carry them out. However, environment complexity and some human factors can lead to several problems in performing these tasks. For example, fatigue at work and

Figure 1.1: Examples of corrosion in vessel metallic structures.

stress can lead to ineffective inspections of the ship's structure or poor quality control in the preparation of surgical instrumentation boxes, which in turn can pose additional issues related to information rechecking and incident tracking.

It is clear that the use of machine vision techniques may help in facilitating the two considered tasks, removing the effect of human factors such as fatigue. This, in turn, allows performing faster, what means a reduction in temporal and economic terms.

Among the different machine learning techniques, the so called Deep Learning methods and, in special, those based on Deep Convolutional Neural Network (DCNN) have been used in the recent years to obtain consistent and accurate results in similar applications [10–14].

1.3 Objectives of the Thesis

The main goal of this thesis is the design and development of efficient tools and methodologies for object detection and segmentation of digital images, with a particular emphasis on inspection and quality control applications. To this end, we focus on the use of deep learning, and, more specifically, Deep Convolutional Neural Networks (DCNN), to achieve high accuracy at a reasonable computational cost.

We focus on three different approaches:

- unoriented bounding box detection for object recognition and localization inside the input images,

Figure 1.2: Detection targets in sterilization boxes for surgical tools.

- oriented bounding box detection for enhanced performance on the above, and
- semantic segmentation for pixel-level classification of the same images.

After conducting the appropriate learning procedures, the developed methods are expected to efficiently address the inspection and quality control applications described in section 1.2, namely detect defects on vessel metallic structures, i.e., detect corrosion and coating breakdown, and locate the different control elements present in surgical sterilization boxes (see Fig. 1.2).

To achieve this main goal, we define the following subobjectives:

- Identify and analyze relevant existing solutions for object detection based on DCNNs.
- Develop an accurate unoriented bounding box detector. In order to ensure accurate detection, the detector should be able to identify small targets in complex environments.
- Extend the previous detector to predict oriented bounding boxes, so that the detection results can adhere to the orientation of targets.
- Identify and analyze relevant existing approaches for semantic segmentation based on DCNNs.
- Adopt a semantic segmentation approach to obtain identification results at the pixel-level. Besides, in order to avoid the massive effort required by pixel-level

annotations, this sub-objective also aims to devise and design an approach that can obtain good segmentation performance using user-friendly weak annotations.

- Explore the possibility of adopting a mixed approach between bounding box regression and semantic segmentation to take the best of both approaches.

1.4 Contributions

To fulfil the previous objectives, this dissertation presents the results of a research process which can be summarized in the following contributions:

- An exhaustive survey of earlier works related to visual inspection and quality control applications using object detection and/or semantic segmentation technologies. This review focuses on three aspects:
  - Main DCNN-based object detection algorithms considered as one-stage and two-stage approaches. We also include in this review the oriented detection approaches.
  - Typical DCNN-based semantic segmentation methods, including the current weakly supervised semantic segmentation approaches.
  - Previous approaches on visual inspection and quality control, classified in accordance to the type of features involved and the machine vision technique adopted.
- A flexible detector based on the Single Shot Multi-box Detector (SSD) is proposed to predict the localization and orientation of targets. The detector has the ability to detect small image areas and the orientation of elongated targets. Consequently, this contribution splits into the following two aspects:
  - A Feature Pyramid Single Shot Multi-box Detector (FPSSD) developed to improve the detection performance for small targets.
  - A lightweight convolutional neural network to detect the orientation of the target inside each prediction of FPSSD.
- A novel Weakly Supervised Semantic Segmentation approach using scribble annotations is proposed. This contribution actually comprises:
  - A specific loss function to deal with the reduced supervision achievable by means of scribble annotations. This loss function comprises a first partial cross-entropy term, and the two terms that are described next.

  - A cross entropy-based centroid loss term to predict classes centroids. By adding a specific sub-network, the full network can predict class centroids and perform the segmentation of the input image in one feed-forward procedure.
  - A Mean Square Error loss term to assist in training the network and refining the segmentation results.

- Several strategies for combining bounding box regression and semantic segmentation approaches that aim at improving the detection performance of both approaches when operating separately. This contribution comprises:
  - Three methods using bounding box regression to improve segmentation performance, comprising intersection, union and intersection with threshold approaches.
  - One approach to perform bounding box regression with the help of segmentation results.

1.5 Scope of Research

The work reported in this thesis has been supported by the following projects:

- Project IMABIA (Ref. PROCOE/4/2017, Govern Balear, 50% funded by Programa Operatiu FEDER 2014-2020 de les Illes Balears). The goal of this project is to assist to the inspection, monitoring and identification processes within a hospital environment by means of image processing and artificial intelligence.
- The EU-H2020 projects BUGWRIGHT2 (GA 871260) and ROBINS (GA 779776). These projects aim at filling the technology and regulatory gaps that still represent a barrier to the adoption of Robotics and Autonomous Systems (RAS) in activities related to the inspection of ships.
- Project FUZZYMAR (Ref. PGC2018-095709-B-C21, MCIU/AEI/FEDER, UE). The target of this project is two-fold: (1) advance in the study of fuzzy metrics and indistinguishability operators, and (2) apply them to typical vision and robotics problems, e.g. neural network training.

1.6 Related Publications

Parts of this thesis have been published in international journals and conference proceedings. The following list overviews each individual publication:

- Kai YAO, Alberto ORTIZ, Francisco BONNIN-PASCUAL, A DCNN-based Arbitrarily-Oriented Object Detector for Quality Control and Inspection Applications. (journal paper, under review)

- Kai YAO, Alberto ORTIZ, Francisco BONNIN-PASCUAL, A Weakly-Supervised Semantic Segmentation Approach Based on the Centroid Loss: Application to Quality Control and Inspection, IEEE Access, vol. 9, pp. 69010-69026, 2021

- Kai YAO, Alberto ORTIZ, Francisco BONNIN-PASCUAL, Deep Learning-based Object Detection for a Quality Control Application in the Sterilization Unit of a Hospital, Proceedings of the Annual Workshop of the Health Research Institute of the Balearic Islands, Palma (Spain), 2020

- Kai YAO, Alberto ORTIZ, Francisco BONNIN-PASCUAL, Centroid Loss for Weakly-Supervised Semantic Segmentation in a Quality Control Application, Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna (Austria), 2020

- Alberto ORTIZ, Francisco BONNIN-PASCUAL, Emilio GARCIA-FIDALGO, Joan P. COMPANY, Kai YAO, Visual Inspection of Vessels Cargo Holds: Use of a Micro-Aerial Vehicle as a Smart Assistant, Proceedings of the IMEKO TC-9 Workshop on Metrology for the Sea, Genoa (Italy), 2019

- Kai YAO, Alberto ORTIZ, Francisco BONNIN-PASCUAL, A DCNN-Based Arbitrary-Oriented Object Detector for a Quality Control Application, Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza (Spain), 2019

- Alberto ORTIZ, Francisco BONNIN-PASCUAL, Emilio GARCIA-FIDALGO, Joan P. COMPANY, Kai YAO, Towards the Automation of Visual Inspections of Cargo Holds of Large-Tonnage Vessels, Proceedings of the Jornadas Nacionales de Robotica (Spanish Robotics Conference), Alicante (Spain), 2019

- Alberto ORTIZ, Kai YAO, Francisco BONNIN-PASCUAL, Emilio GARCIA-FIDALGO, Joan P. COMPANY, New Steps towards the Integration of Robotic and Autonomous Systems in the Inspection of Vessel Holds, Proceedings of the Jornadas Nacionales de Robotica (Spanish Robotics Conference), Valladolid (Spain), 2018

1.7 Document Overview

The rest of this dissertation is organized into the following six chapters:

- **Chapter 2** firstly outlines deep learning methodologies, secondly introduces bounding box regression methods, as well as the related evaluation metrics, and thirdly, provides an analysis about the advantages of DCNN for semantic segmentation, introduces some typical semantic segmentation loss functions, and the evaluation metrics for this kind of methods.

- **Chapter 3** firstly reviews the state of the art regarding DCNN-based object detection, including unoriented and oriented object detection approaches. Secondly, we summarize the main techniques of DCNN-based semantic segmentation and review some typical weakly supervised semantic segmentation approaches. In the end, some image processing- and machine learning-based works focusing on visual inspection and quality control are reviewed.

- **Chapter 4** presents an oriented detection solution organized in two stages. In the first stage, a Feature Pyramid SSD is proposed in order to improve the detection performance for small targets. In the second stage, a lightweight convolutional neural network is employed to produce oriented detections for the involved targets on the basis of the outputs of the first stage.

- **Chapter 5** describes two semantic segmentation solutions using full and weak supervision. For the fully supervised approach, we experiment with different loss functions to train the network to improve the segmentation ability for small area targets. A weakly supervised semantic segmentation solution using scribbles annotations is proposed next. Particularly, we focus on the generation of weak annotations, considering scribbles and pseudo masks. We then propose a novel approach based on the idea of clustering to enhance the performance achieved by semantic segmentation alone. At last, we discuss the different feature spaces for our approach and assess its performance using weak annotations of different quality.

- **Chapter 6** explores the combination of the bounding box regression and semantic segmentation approaches. We firstly propose three strategies that make use of bounding box regression to improve semantic segmentation. Next, we develop one method that makes use of semantic segmentation to improve bounding box regression.

- **Chapter 7** concludes the dissertation by summarizing the main contributions and suggesting some future work to extend the research.

Background

In this chapter, we briefly introduce some background concepts for this dissertation. In first place, we provide a fast introduction of basic neural network concepts in Section 2.1. Secondly, the extension to DCNN is introduced in Section 2.2. Then, we briefly describe Bounding Box (BBox) regression, which is an essential technique of DCNN-based object detection algorithms. Both, BBox methods and the underlying evaluation metrics are reviewed in Section 2.3. Next, some background notions regarding semantic segmentation are provided in Section 2.4. Finally, Section 2.5 summarizes this chapter.

2.1 Basic Neural Network Concepts

As a crucial branch of machine learning, deep learning technology has developed significantly, and it brings revolutionary improvements to computer vision tasks. Nowadays, some approaches have obtained significant achievements on some computer vision tasks. The recent enthusiasm for deep learning is not a whim but has experienced a long time of development.

In 1943, McCulloch and Pitts [15] published a paper named "A Logical Calculus of the Ideas Immanent in Nervous Activity", which proposes a neural network and a mathematical model called the MCP model (named by the abbreviation of the two authors names) model. This model imitates the architecture and mechanism of neurons to construct an abstract model. The model is shown in Eq. 2.1, which includes the weights w for the input signal x, the sum operation, and the non-linear activation function δ. The MCP model significantly influenced subsequent scientists and is still widely used in deep learning technology.

$$y_k = \delta(\sum_{j=0}^{m} w_{kj} x_j) \tag{2.1}$$

In 1958, Rosenblatt [16] proposed a neural network consisting of two neurons, namely Perceptron. The authors firstly applied the MCP model a classification task in a machine learning case. The proposal of the Perceptron attracted the interest of a large number of scientists in the research of artificial neural networks, and it is seen as a major landmark in the development of artificial intelligence.

In 1986, Hinton [17] presented the Back-Propagation (BP) algorithm for the Multi-layer Perceptron (MLP) using a sigmoid function for the non-linear mapping that effectively solves the classification task in non-linear classification problems. In their work, they integrate an error back-propagation process based on the feed-forward propagation process from the traditional neural networks. The back-propagation process continuously adjusts the weights of neurons until the output error is reduced to the allowable range or it reaches the pre-set number of training iterations. This work solved the non-linear classification problem and aroused intense academic interest.

Nowadays, neural networks still continue the previous design. A hidden layer of a neural network model is shown in Eq. 2.2, where x represents the model inputs, W denotes the weight matrix, b indicates the bias vector, and δ is an element-wise non-linear function (such as ReLU or TanH).

$$\hat{y} = \delta(Wx + b) \tag{2.2}$$

2.2 Neural Networks for Massive Image Processing

A DCNN, one of the most representative techniques of deep learning for 2D images, can take advantage of the basic properties of natural signals, such as translation invariance, local connectivity, and compositional hierarchies [18]. The DCNN model is made of a recursive application of convolution and pooling, followed by inner product layers (or fully connected layers) at the end of the network. The convolutional layer is a linear transformation that extracts spatial information from the input image. The pooling layer is used to decrease the dimensionality of the inputs by taking the maximum or mean of each block of pixels.

In the convolutional layer, the features of the previous layer are convolved with a convolutional kernel to output new features. The output of the j-th feature map in the l-th layer is computed as:

$$X^{l,j} = \sum_m \sum_n \delta \left(\sum_f \sum_u \sum_v X_{l-1,f}(m-u, n-v) \cdot K_f^{l,j}(u,v) + b^{l,j} \right) \tag{2.3}$$

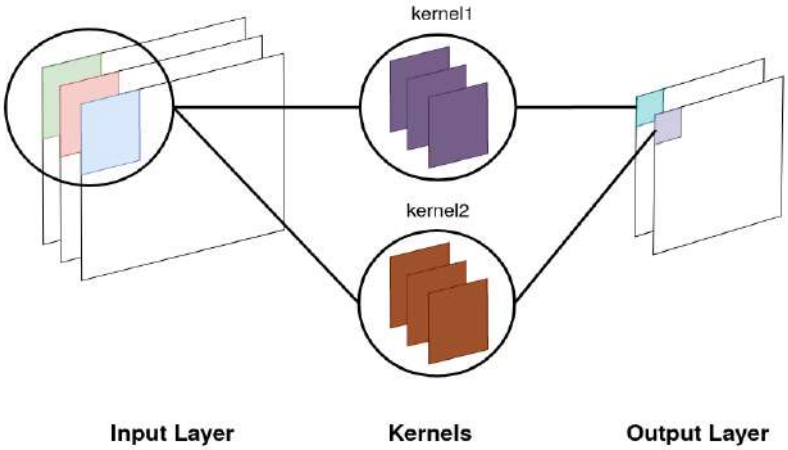


Figure 2.1: An example of convolutional layer in a CNN model. The input features have three channels, while the convolutional layer contains two groups of convolutional kernels. Each kernel is convolved with each image patch (the highlighted parts in the input layer) providing the highlighted parts in the output layer.

where X_{l-1} are the input features (with shape $F \times M \times N$, where F is the number of input channels from the previous layer), K is the convolutional kernel of size $F \times U \times V$, and $b^{l,j}$ is a bias. The convolutional layer can preserve spatial information through a series of filters sliding on the input features, as shown in Fig. 2.1.

After convolution, batch normalization is typically applied to decrease the error rates and prevent gradient explosion [19]. The operation can be seen as,

$$\begin{aligned} \hat{x} &= \frac{x^{l-1} - \mu}{\sigma}, \\ x^l &= \gamma \hat{x} + \beta. \end{aligned} \tag{2.4}$$

where $\{x^{l-1}\}$ denotes the input features of a mini-batch, μ and σ respectively are the mean and standard deviation of $\{x^{l-1}\}$. Then, batch normalization applies a back-propagation algorithm to learn the scale parameter γ and the shift parameter β.

To use a neural network in a regression task, the Mean Squared Error (MSE) is one of the most commonly used loss functions. This can be formalized as:

$$E(\hat{Y}(X), Y) = \frac{1}{2N} \sum_{i=1}^{N} ||y_i - \hat{y}_i||^2 \tag{2.5}$$

where $Y = \{y_i\}$ is the ground truth for the N observed samples $X = \{x_i\}$, and $\hat{Y}(X)$ is

the set of outputs of the model given the input set X. So, by optimizing the target E by means of the back-propagation algorithm, the model can learn the layer weights.

For the classification task, whose target is to predict the probability $\hat{y}$ of one input x being classified with a label y in the set $\{1, ..., C\}$, an element-wise softmax function is typically used to obtain the normalized scores:

$$\hat{y} = \frac{\exp(\hat{y}_c)}{\sum_{c=1}^{C} \exp(\hat{y}_c)} \tag{2.6}$$

where $c \in \{1, 2, \cdots, C\}$ is the set of classes for the inputs. Then, taking the log of $\widehat{y}$ for each sample x_i, the softmax Cross-Entropy loss can be obtained as:

$$E(\hat{Y}(X), Y) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{2.7}$$

with $y_i \in \{1, 2, \cdots, C\}$.

Over-fitting is a typical problem during training in the domain of machine learning. It is noticed when, during training, the model learns the noise of the training set, so that the loss value for the training set $\{X_{\text{train}}, Y_{\text{train}}\}$ decreases but the loss value for the validation set $\{X_{\text{val}}, Y_{\text{val}}\}$ increases. One possibility to overcome this problem is to add a regularization term to the loss function, being L_2 regularization one of the most commonly used. L_2 regularization involves the parameters of each layer, i.e., weights W and biases b, and the regularization coefficients λ_1 and λ_2, resulting in the following target to optimize:

$$L(X, Y) = E(\hat{Y}(X), Y) + \lambda_1 ||W||^2 + \lambda_2 ||b||^2 \tag{2.8}$$

As described before, the DCNN model can solve two main problems in supervised machine learning, which are regression and classification. With the rapid development of GPU acceleration technology, deep learning ushers in an explosive period of development. Recently, DCNNs [4] have become a popular tool for image processing, taking us beyond state-of-the-art performance in many computer vision tasks. Well-known examples of DCNNs and their dates of publication are AlexNet [5] in 2012, VGG-16 [20] in 2014, and ResNet [21] in 2016. These works have proved the potential of DCNN, making DCNN-based models win overwhelmingly in some benchmarks, such as ILSVRC-2014 [22] and CIFAR-10 [23].

For the object detection task, deep learning-based approaches have achieved out-

standing performance. In an object detection task, different detection may have totally different properties, and their problems may vary from each other. In addition, the environment (i.e., light conditions, image clarity, and resolution) make intervene important factors that may affect the detection results.

To overcome these problems, researchers collect a tremendous number of images in various environments for training, and organize benchmarks for object detection that include (but do not limit to) objects at different orientations and scales, and dense scenes considering also overlapped objects. Nowadays, the PASCAL VOC (Visual Object Classes) benchmark [24], which started in 2005, is held annually, including the challenge of image classification and object localization. It contains images from the ImageNet dataset (more than 14 million images) comprising 1000 different classes for classification and 200 classes for detection. Similarly, the Microsoft COCO (Common Objects in COntext) [25] benchmark registers the precise coordinates of the target up to two decimal places. The aim of this dataset is scene understanding in complex daily scenes.

2.3 Bounding Boxes Regression

In general, an object detection approach needs to output the Bounding Box (BBox) coordinates and the category of the target within the BBox. Before the deep learning era, common approaches like Selective Search [26] could provide plenty of region proposals. However, due to the lack of an accurate location, these proposals could not meet realistic requirements. In recent years, some researchers propose using BBox Regression combined with a DCNN to fine-tune the proposals, and obtain precise detection results, such as in [27–30]. This section will introduce the definition and methodology of BBox Regression and then present the main evaluation metrics for object detection.

2.3.1 What is Bounding Boxes Regression?

As shown in Figure 2.2, the red rectangle is a proposal of the Selective Search algorithm. Although the category of the object inside of the red rectangle is the airplane, because the IOU (Intersection over Union, see Section 2.3.3) is lower than 0.5, this prediction cannot be seen as a correct detection in a realistic environment. However, the green rectangle is an accurate location. Therefore, we can fine-tune the red rectangle by making it closer to the green rectangle. In this way, BBox Regression is proposed to fine-tune proposals to obtain accurate detection results.

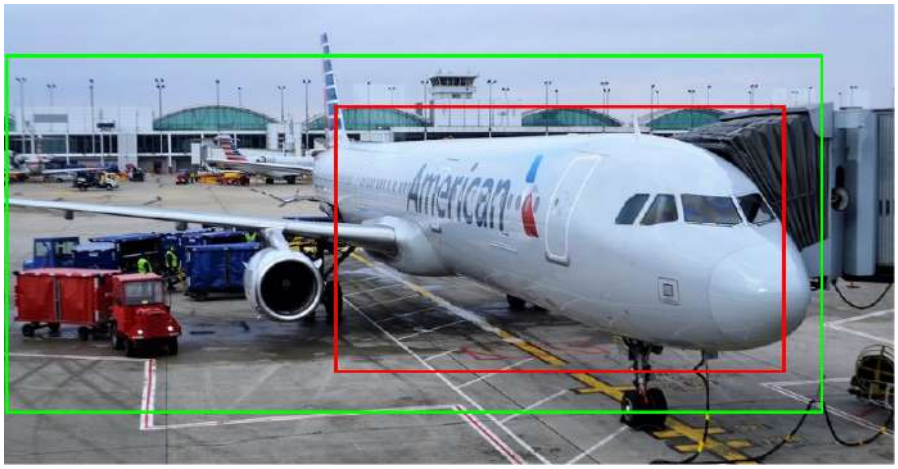


Figure 2.2: The example of a BBox and a proposal of Selective Search algorithm. The red rectangle indicates the proposal from Selective Search, and the green rectangle indicates the accurate detection.

2.3.2 How to do Bounding Boxes Regression?

In general, a tuple (x, y, w, h) is used to represent a BBox, which indicates the coordinate of the BBox center point and its width and height. The purpose of BBox Regression is to find a mapping function f so that maps proposal P into $\hat{P}$, which is close to the ground truth G. The procedure is shown below:

$$\begin{aligned} f(P_x, P_y, P_w, P_h) &= (\hat{P}_x, \hat{P}_y, \hat{P}_w, \hat{P}_h) \\ (\hat{P}_x, \hat{P}_y, \hat{P}_w, \hat{P}_h) &\approx (G_x, G_y, G_w, G_h) \end{aligned} \tag{2.9}$$

The basic idea of finding f is *Scale* and *Shift*. According to the demonstration in R-CNN [31], the authors calculated two coordinates offsets (Δ_x, Δ_y), where $\Delta_x = P_w d_x(P)$ and $\Delta_y = P_h d_y(P)$. Then, they compute a dimensional scale factor (S_w, S_h), where $S_w = \exp(d_w(P))$ and $S_h = \exp(d_h(P))$. Therefore, the mapped proposal $\hat{P}$ is calculated as:

$$\begin{aligned} \hat{P}_x &= P_w d_x(P) + P_x \\ \hat{P}_y &= P_h d_y(P) + P_y \\ \hat{P}_w &= P_w \exp(d_w(P)) \\ \hat{P}_h &= P_h \exp(d_h(P)) \end{aligned} \tag{2.10}$$

As shown in Eq. 2.10, the target of the BBox Regression is to learn four transformations $d_*(P)$. Then, the target transformation can be expressed as $d_*(P) = w_*^T \Phi(P)$,

where $\Phi(P)$ indicates the input feature vector of proposals, and w_*^T are the parameters to be learnt. The loss function can be expressed as:

$$\text{Loss} = \sum_{i}^{N}(g_i - w_*^T\Phi(P))^2 \tag{2.11}$$

where $g_i \in G$.

The optimization target including a L_2 regularization term is:

$$W_* = \arg\min_{w_*} \sum_{i}^{N}(g^i - w_*^T\Phi(P))^2 + \lambda||w_*||^2 \tag{2.12}$$

2.3.3 Evaluation Metrics

Object detection metrics serve as a measure to evaluate how well the detection algorithm performs in an object detection task. Regarding the outputs of a detection algorithm based on DCNN, most models generate a set of detections, where each detection includes the coordinates of a BBox and also a confidence score indicating the probability of the target appearing in the BBox. Different metrics are used to evaluate the performance of these detections given the annotations containing the BBox and the category.

In order to determine how many objects are correctly located, the Intersection over Union (IOU), also referred to as the Jaccard Index, is introduced. IOU is an evaluation metric that quantifies the overlap between the predicted BBox and the ground truth. Let b_i be a predicted BBox provided by the object detector when considering target i, and b_g be the ground truth, then the IOU can be calculated as:

$$IOU(b_i, b_g) = \frac{area(b_i \cap b_g)}{area(b_i \cup b_g)} \tag{2.13}$$

IOU is used to select the True Positive (TP) samples. More precisely, if the IOU between a predicted BBox b_i and a ground truth b_g is higher than a pre-defined threshold ϵ, b_i is considered as a TP. Otherwise, it is considered as a False Positive (FP).

On the other hand, as mentioned before, each predicted BBox provided by a DCNN-based detector is associated with a confidence score. This is used to assess the probability of the target appearing in the BBox. As usual, given a confidence threshold β, a predicted BBox with a confidence score above the threshold is considered as a TP, while a BBox with confidence score below the threshold is considered as a FP.

The two thresholds for the IOU and the confidence score are, thus, tunable parame-

ters which clearly affect in the definition of TP and FP samples, determining the model's performance. Currently, there are several metrics to evaluate the performance of detection algorithms. Among them, Precision, Recall, and Average Precision (AP) are three of the most used. After obtaining the TP and FP, the Precision and Recall can be calculated as follows:

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} = \frac{\text{correct detections}}{\text{all detected boxes}} \\ Recall &= \frac{TP}{TP+FN} = \frac{\text{correct detections}}{\text{all ground truth boxes}} \end{aligned} \tag{2.14}$$

As mentioned above, the values of FP and TP are affected by the IOU threshold setting. In other words, when the threshold is high, it means that the requirement of the detector is high, which makes the precision value high, and recall value low. On the contrary, when the threshold is low, the precision value is low, and recall is high. Both configurations do not provide the most desirable results. The average precision (AP) can measure the overall pros and cons of precision and recall. The AP is based on the precision-recall curve and is the precision averaged across 11 pre-set (Pascal VOC 2007) or all unique (Pascal VOC 2010) recall levels. To obtain the AP, precision is interpolated for a set of different recall levels r by taking the maximum precision p_{interp} whose recall value $\hat{r} > r$.

$$\begin{aligned} AP &= \frac{1}{11} \sum_{r \in \{0,0,1,0,2,\ldots,1.0\}} p_{interp}(r) \quad \text{(Pascal VOC 2007)} \\ AP &= \frac{1}{\text{len(rec)}} \sum_{r \in rec} p_{interp}(r) \quad \text{(Pascal VOC 2010)} \\ p_{interp}(r) &= \max_{\hat{r}>r} p_{interp}(\hat{r}) \end{aligned} \tag{2.15}$$

The calculation of AP only involves one category. However, the task of object detection is usually a multi-category problem. So, mAP is defined as the mean of AP over all K categories:

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K} \tag{2.16}$$

Table 2.1 summarizes the symbols and main metrics used in object detection.

Table 2.1: Summary of symbols typically involved in commonly used metrics for evaluating object detectors.

Metric	Meaning	Definition and description
IOU	Intersection over Union	The overlap between a predicted BBox and the ground truth by means of the Jaccard Index.
β	Confidence threshold	A confidence threshold to decide between TP and FP.
ϵ	IOU threshold	A IOU threshold to decide between TP and FP.
TP(c)	True Positive	A correct detection, where the confidence c and IOU are higher than β and ϵ.
FP(c)	False Positive	An incorrect detection, where the confidence c or IOU are lower than β or ϵ.
Rec	Recall	The fraction of the total amount of relevant instances that were actually retrieved.
Prec	Precision	The fraction of relevant instances among the retrieved instances.
AP	Average Precision	A measure that combines recall and precision for ranked retrieval results given single category.
mIOU	mean Intersection over Union	The mean of IOU over all categories.
mRec	mean Recall	The mean of Rec over all categories.
mPrev	mean Precision	The mean of Prec over all categories.
mAP	Mean Average Precision	The mean of AP over all categories.

2.4 Image Semantic Segmentation

Image Semantic Segmentation pursues assigning a label to every image pixel, what differs from image/BBox classification, which assigns a label to the entire image/BBox. Besides, image semantic segmentation differs from traditional segmentation approaches, such as threshold-based approaches and cluster-based approaches, which split an image into subregions according to its properties (color, texture, energy, etc.). In other words, these traditional approaches do not provide category for every subregion. With the rise of DCNN, image semantic segmentation approaches based on DCNNs can not only predict the category of each pixel, but also significantly improve the segmentation performance. Therefore, Section 2.4.1 will introduce some traditional approaches that are widely used

in the DCNN-based models. Then, we will analyze the advantages of DCNN for image semantic segmentation and introduce some typical loss functions in Section 2.4.2. In the end, the evaluation metrics are introduced in Section 2.4.3.

2.4.1 Traditional Image Semantic Segmentation Approaches

Before the era of deep learning, image processing techniques were widely used for image segmentation. In this section, we summarize some approaches that are also widely used in DCNN-based models.

Using a traditional segmentation approach, such as threshold- or clustering-based approaches, we will probably obtain noisy segmentation results. To address this problem, some authors propose considering the relationship between two pixels by means of a graph-based approach. This is the case of Normalized Cuts (NCuts) [32], dense CRFs [33], and GraphCuts [34]. These approaches suggest that adjacent pixels tend to present analogous properties and labels. Therefore, they construct a graph ($G = (V, E)$) using image pixels as nodes (V) and the similarity between two pixels as the weight of edge (E) between two nodes. On the other hand, the SLIC Superpixels [35] is another approach widely used in the DCNN models, such as [36–39]. All these techniques are briefly described in the following.

Normalized Cuts (NCuts)

The purpose of NCuts is to find the minimum cut for each category in a graph. Assuming a graph $G = (V, E)$, which can be partitioned into two disjoint sets A and B, where $A \cup B = V$ and $A \cap B = \varnothing$. By removing edges connecting the two parts, the degree of dissimilarity between these two pieces can be computed as total weight (w) of the edges which has been removed. In graph theory language, it is called the *cut*, which is defined as $cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$. Therefore, the NCuts can be formulated as below.

$$\begin{aligned} Nassoc(A, B) &= \frac{cut(A, A)}{assoc(A, V)} + \frac{cut(B, B)}{assoc(B, V)} \\ NCuts(A, B) &= \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \\ &= 2 - Nassoc(A, B) \end{aligned} \tag{2.17}$$

where $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$ is the total connection from nodes in A to all nodes in the graph, and $cut(A, A)$ and $cut(B, B)$ are total weights of edges connecting

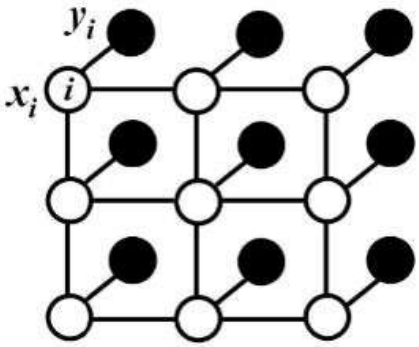


Figure 2.3: Conditional Random Fields (CRFs).

nodes within A and B, respectively.

In order to obtain the optimal partition, Eq. 2.17 is converted to compute the eigenvector of the Laplacian matrix, and the optimal solution is the second smallest eigenvector.

Conditional Random Fields (CRFs)

CRF is a type of structural probabilistic prediction model, that considers neighboring semantic information before prediction, such as the relationship between pixels. Assuming the label of a pixel i in one image is y_i, its observation is x_i. CRF is to construct a graph using every pixel as a node and the similarity between pixels as the weight of the edges, as shown in Fig. 2.3.

Particularly, the fully connected (Dense) CRF [33] is one of the efficient tools for semantic segmentation before the DCNN era. Dense CRF is a probabilistic undirected graphical model, that can be expressed as,

$$P(Y) = \frac{1}{Z} \prod_{C} E_C(Y_C) \tag{2.18}$$

where, Y_c represents the maximal clique, E is the potential function, and Z is the normalization factor.

The potential function $E_C(x)$ consists of a unary potential energy ψ_u and a pairwise potential energy ψ_p. The unary potential energy is computed independently for each pixel by a classifier that produces a distribution over the label assignment. The pairwise potential can be calculated using texture, location, and color descriptors. The pairwise

potential has the form:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w^m k^m(f_i, f_j) \tag{2.19}$$

where each k^m is a Gaussian kernel, the vectors (f_i, f_j) are feature vectors for pixels i and j in an arbitrary feature space, w^m is the linear combination weights, and μ is a label compatibility function.

In the end, the sum of unary and pairwise potential energy is the loss function, as shown in Eq. 2.20. By minimizing the loss function, the segmentation results can be obtained.

$$E_C(x) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j) \tag{2.20}$$

Superpixels

The target of superpixel-based approaches is to combine a group of pixels with similar properties into a large superpixel. For example, considering an input image with more than 160,000 pixels, this can result with about a thousand superpixels after processing it with a superpixels algorithm. For instance, considering a 640×480 image, the resulting graph comprises 307200 nodes in total. However, after using a superpixel-based algorithm (1000 superpixels, for example), the graph only has 1000 nodes, which is more efficient than before. For this reason, superpixel-based approaches are usually employed as a preprocessing step in vision tasks.

SLIC Superpixels algorithm is the abbreviation of the Simple Linear Iterative Cluster Superpixels. The general procedure comprises the following steps: (i) the input image is converted to the CIE-Lab color space, and a 5-d vector is created for each pixel using the (L, a, b) color values and the coordinates (x, y) of a pixel; (ii) analogous to the cluster approach, the SLIC algorithm generates K centroids and then searches the space around each centroid for pixels that are close to the centroids. The distance between pixel p_i and p_j is shown in Eq. 2.21.

$$\begin{aligned} d_c &= \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \\ d_s &= \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \\ D &= \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2} \end{aligned} \tag{2.21}$$

Algorithm 1: The SLIC Superpixels algorithm.

Input:
The input image I and required K superpixels.
Output:
The list L of superpixels containing the label L_i for every pixel.
Initialization:
Convert the image into the CIE-Lab color space.
Sample pixels at regular grid step S and initialize cluster centoids $C_k = [l_k, a_k, b_k, x_k, y_k]^T$.
Move the initialized centroid to the position of the smallest gradient in the 3×3 area.
Initialize the label L_i for every pixel i to -1.
Initialize the distance d_i for every pixel i to ∞.
while $E \leq$ *threshold* **do**
  for *each cluster centroid* C_k **do**
    for *each pixel located in the* $2S \times 2S$ *area around* C_k **do**
      Compute the distance d'_i (Eq. 2.21) between the pixel i and centroid C_k.
      if $d'_i < d_i$ **then**
        $d'_i = d_i$, $L_i = k$.
      end
    end
  end
  Update the new cluster centroids C'_k.
  Compute the residual error E between C'_k and C_k.
end

where, N is the total number of pixels in the image, $N_s = \sqrt{\frac{N}{K}}$, N_c is the hyper-parameter, and the tuple $(x,\ y)$ provides the coordinates of pixel p. Meanwhile, the algorithm classifies the pixels of the image using the same label as the closest centroid; and (iii) the algorithm calculates the average value for all pixels in the K superpixels, and obtains K centroids. In the end, it repeats the last two steps until the algorithm converges. The SLIC Superpixels procedure is shown in Algorithm 1.

2.4.2 DCNN-based Semantic Segmentation

Currently, DCNN-based approaches have achieved overwhelming superior performance for image semantic segmentation in comparison with traditional approaches. To explain the reasons, in this section we first analyze the advantages of DCNN for image semantic segmentation. Then, having into account that defining an optimized target is a crucial

factor in DCNN-based approaches, we list some typical loss functions.

2.4.2.1 The Advantages of Using DCNNs in Vision Tasks

A neural network is a complex compound function in essence, which has a strong and comprehensive fitting ability. As long as the activation function is properly selected and the number of neurons is large enough, even considering a 3-layer neural network (i.e., comprising a single hidden layer), the network is proved to be able to fit any continuous mapping function from a given input vector to a given output vector [40–42]. This is the so called Universal Approximation Theorem. In [41], the authors prove that the Universal Approximation Theorem is correct under the condition of using the sigmoid activation function, and in [40], the authors prove that the Universal Approximation Theorem does not depend on the specific activation function but is guaranteed by the structure of the neural network.

A CNN can be seen as a fully connected and weights shared neural network, so the Universal Approximation Theorem is also applicable to it. In [43], the authors theoretically explain the success in computer vision of CNNs, and they apply a mathematical framework to analyze the CNN properties. The convolutional operation can be divided into two steps, the first step is the linear transformation, and the second step is the activation function (nonlinear) transformation. The former step can be regarded as a linear mapping that projects the input data into a lower-dimensional space; the latter step is a compressed nonlinear transformation of data. In this way, a CNN has properties and fitting ability similar to those of a generic neural network.

For the image semantic segmentation, the hierarchical architecture of a DCNN can provide the contextual semantic information for the segmentation target in the entire image, which can be understood as features that reflects the relationship between the target and its environment. Meanwhile, the shallow layers of a DCNN can provide low-level features, such as position, texture, and color, etc. Therefore, due to having outstanding fitting ability and, at the same time, providing rich and diverse features, DCNNs can effectively solve image semantic segmentation problems.

2.4.2.2 Semantic Segmentation Loss Functions

The selection of the loss function is an important step when designing a DCNN for image semantic segmentation. In recent years, some researchers have proposed several loss functions in specific fields for better performance. In [44], the author summarizes 15 loss functions for image semantic segmentation, which have been proved to achieve good

performance in various fields. These loss functions can be divided into four categories, namely Distribution-related Loss Functions, Metric-related Loss Functions, Boundary Loss Functions, and Compound Loss Functions. In the following, we briefly introduce some typical loss functions in each category.

Distribution-related Loss Functions

Nowadays, the Cross-Entropy (CE) loss function is one of the most commonly used loss functions. This is formulated as:

$$L_{CE} = -\sum_{i}^{C} y_i \log(p_i) \tag{2.22}$$

where, C indicates the category number, y_i indicates the ground truth and p_i indicates the prediction. However, some researchers are interested in obtaining fine-grained information about the image, so they have developed some specific loss functions for semantic segmentation.

Focal Loss [45] function is an improvement over the standard CE loss. It is achieved by changing the shape of curve of CE so that the weights of hard samples will be increased during training. In this way, it can decrease the influence of an imbalance situation in the dataset. It is formulated as:

$$FL(p_i) = -(1 - p_i)^{\gamma} \log(p_i) \tag{2.23}$$

where γ is the scale factor.

Metric-related Loss Functions

The Dice coefficient is a widely used metric in the computer vision community, which is applied to calculate the similarity between two images. In 2016, it was also adapted as a loss function known as the Dice Loss [46].

$$DL(y, p) = 1 - \frac{2yp + 1}{y + p + 1} \tag{2.24}$$

Here, 1 is added both in the numerator and the denominator to ensure that the function is not undefined in edge case scenarios, such as $y = p = 0$. The Dice Loss is popular in medical image segmentation tasks.

The Tversky Loss [47] is based on the Dice Loss. It adds weights to FP (false-

positives) and FN (false-negatives) with the β coefficient's help. Similarly to the Dice Loss, the Sensitive Specific Loss [48] is inspired by the Sensitivity and Specificity metrics and is designed to be used in those cases that need more attention on the True Positive samples.

The Hausdorff Distance (HD) is a metric used by segmentation approaches to track the performance of a model. It is defined as:

$$d(P, Y) = \max_{p \in P} \min_{y \in Y} ||p - y||^2 \tag{2.25}$$

In the HD Loss [49], the authors proposed three variants, which are designed on the basis of how the Hausdorff distance is used as part of the loss function: (i) taking the maximum of all HD errors, (ii) taking the minimum of all errors obtained by placing a circular structure of radius r, and (iii) taking the maximum of a convolutional kernel placed on top of missing segmented pixels.

Boundary Loss Function

The Boundary Loss [50] is applied to highly unbalanced segmentation tasks. The form of this loss is the distance measure on the spatial contour, instead of throughout the area. In this way, it solves the problem caused by the region loss of highly unbalanced segmentation tasks:

$$Dist(\partial G, \partial S) = \int_{\partial G} ||y_{\partial S}(p) - p||^2 dp \tag{2.26}$$

It is used to evaluate the change between ∂G and ∂S, where p is a point on boundary ∂G and $y_{\partial S}(p)$ denotes the corresponding point on boundary ∂S.

Compound Loss Function

The Compound Loss (CL) [51] is defined as a weighted sum of the Dice Loss and a modified CE. It attempts to leverage the flexibility of the Dice Loss regarding class-unbalanced tasks, while using CE for curve smoothing. It is defined as:

$$L_{CL}(y, p) = \alpha L_{CE}(y, p) - (1 - \alpha) DL(y, p) \tag{2.27}$$

2.4.3 Evaluation Metrics

Regular performance evaluation metrics for image segmentation [52] include pixel accuracy (P_{acc}), mean accuracy (M_{acc}), mean intersection over union (mIOU), frequency weighted IOU (FW_{IOU}), mean recall (mRec), mean precision (mPrec), and F_1 (Dice)

score. Let n_{ij} be the number of pixels of class i predicted to belong to class j, where there are n_{cl} different classes, and let $t_i = \sum_j n_{ij}$ be the total number of pixels of class i, and $n_i = \sum_j n_{ij}$ be the total number of predicted pixels of class i. Seven metrics can be computed as follows:

- The pixel accuracy (P_{acc}) is a ratio between the amount of properly classified pixels and the total number of pixels.

$$P_{acc} = \frac{\sum_i n_{ii}}{\sum_i t_i} \tag{2.28}$$

- The mean pixel accuracy (M_{acc}) is a slightly improved P_{acc}, where the percentage of correct pixels is computed for each category and then averaged over the number of categories.

$$M_{acc} = \frac{1}{n_{cl}} \frac{\sum_i n_{ii}}{t_i} \tag{2.29}$$

- The mean intersection over union (mIOU) is the most commonly used metric to evaluate the performance of segmentation, and it computes a ratio between the intersection and the union of the segmentation result and the ground truth.

$$\text{mIOU} = \frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \tag{2.30}$$

- The frequency weighted intersection upon union (FW_{IOU}) is improved over the raw mIOU, which weights each category's importance depending on their appearance frequency.

$$FW_{\text{IOU}} = \frac{1}{\sum_k t_k} \sum_i \frac{t_i n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \tag{2.31}$$

- The mean Recall and Precision are also commonly used to assess the performance of segmentation approaches. True Positive (TP), False Positive (FP) and False Negative (FN) samples are obtained from the pixel-wise segmentation result and the ground truth. Using a macro-average approach [53], the mean Recall (mRec) and mean Precision (mPrec) are obtained. The mRec effectively describes the

completeness of the true positive samples related to the ground truth,

$$\text{mRec} = \frac{1}{n_{cl}} \sum_i \frac{TP_i}{TP_i + FN_i} = \frac{1}{n_{cl}} \sum_i \frac{TP_i}{T_i} \tag{2.32}$$

while the mPrec indicates the proportion of predicted positive samples among the predicted samples.

$$\text{mPrec} = \frac{1}{n_{cl}} \sum_i \frac{TP_i}{TP_i + FP_i} = \frac{1}{n_{cl}} \sum_i \frac{TP_i}{P_i} \tag{2.33}$$

- Finally, the F_1 score (F-score or F-measure) is the harmonic mean of precision and recall:

$$F_1 = \frac{2 \cdot \text{mPrec} \cdot \text{mRec}}{\text{mPrec} + \text{mRec}} \tag{2.34}$$

2.5 Conclusion

In this chapter, we have first provided a fast overview of the deep learning history and then briefly introduced the deep convolutional neural network. Second, we have introduced the background of object detection and image semantic segmentation approaches. Regarding object detection, we have introduced the definition and mechanism of BBox Regression, which is the basic technique of most current approaches, and we have summarized the evaluation metrics, which are commonly used to assess the performance of this kind of approaches. Regarding image segmentation, firstly, we have introduced some traditional approaches that are widely used in existing deep learning models, secondly, we have analyzed the advantages of using DCNNs for image segmentation, and finally, we have reviewed some typical loss functions for semantic segmentation.

Related Work

In this chapter, we review previous works related to object detection and image segmentation based on deep learning and aiming at generic purposes. However, since we always have in mind the two application cases that we consider in this thesis, we also review previous works related to inspection and quality control, mostly using classical image processing and shallow machine learning techniques.

We firstly review some typical DCNN-based object detection algorithms in Section 3.1. Secondly, we review main techniques of DCNN-based semantic segmentation in Section 3.2. In the end, we review some works based on traditional methods for visual inspection and quality control tasks in Section 3.3.

3.1 Deep Learning for Object Detection

DCNNs have lots of outstanding advantages, among others a hierarchical architecture to learn image representations with multiple levels of abstracted information and the capacity to learn complicated non-linear functions and feature representations directly and automatically from the training set with minimal domain knowledge. Inspired by the advantages of DCNNs, many researchers have focused on object detection using images as main input to this kind of neural network.

In this section, firstly, we review some typical object detection frameworks in Section 3.1.1. Then, we revise other approaches for specifically oriented objects detection in Section 3.1.2.

3.1.1 Detection Frameworks

Recently, techniques for object feature representation and classifiers used for recognition have progressed rapidly, which can be demonstrated by the significant changes from

manual features to automatically learned features. Initially, DCNN-based approaches applied optimized sliding window strategies and generated plenty of proposals by searching over multiple scales and aspect ratios in the image. However, with the increase of the number of proposals and the size of input images, the requirements regarding computational resources and the needs of operating efficiency also increased. Therefore, the design of efficient and effective detection frameworks has played a key role. Commonly adopted strategies include cascaded architectures, sharing feature computation, and reduction of proposals generation.

Current approaches can be roughly divided into two categories:

1. Two-stage detection frameworks comprising a proposal generation network and a Regional CNN (R-CNN) for BBox regression.

2. One-stage detection frameworks consisting in an end-to-end network, which combines the proposals with BBox regression.

3.1.1.1 Region-Based (Two-Stage) Frameworks

In two-stage frameworks, regions of interest (ROI) containing the detected target are independently generated in the first stage. In the second stage, CNNs features are extracted according to these ROIs, and a Regional CNN (R-CNN) is employed to predict the category and regress the coordinates of the BBox.

R-CNN

Girshick et al. [54] present a straightforward and scalable algorithm named R-CNN, which improves the mAP over 30% regarding the best score of previous approaches in the PASCAL VOC 2012 benchmark, achieving 53.7% mAP. In this study, the authors integrate the AlexNet [5] with a selective search algorithm [26]. In the first stage, class agnostic region proposals are generated by using the selective search algorithm, and in the second stage, the authors use the extracted region proposals from the first stage as input to fine-tune a CNN model pre-trained using a large-scale dataset, such as ImageNet. All region proposals with IOU value higher than 0.5 are defined as positive samples, and the rest are negative samples. In order to obtain the category, a Support Vector Machine (SVM) classifier is trained using fixed-length features extracted from CNNs, to classify the category of target in the proposal. On the other hand, BBox regression is implemented to obtain the location of objects.

Although it improves the detection performance significantly, R-CNN has notable drawbacks:

1. In order to train the SVM classifier and BBox regression, the training process is expensive in both disk space and time. For one image, the authors obtain almost 2000 region proposals and use them to train the SVM classifiers and BBox regression.

2. Training is a multi-stage process, which is difficult to optimize.

3. Inference is also slow due to the process of extracting proposals from the test image.

SPP-Net

The R-CNN only allows a series of input images with a fixed resolution (224×224). He et al. [55] present the SPP-Net that allows input images of different sizes. Before this, all of the pre-trained neural networks had a strict demand for image size, e.g., 224×224 (ImageNet), 32×32 (LeNet), and so on. For this reason, a series of operations such as cropping or warping were required, which could cause loss and distortion of the image information to a certain extent. The reason why these networks require fixed-scale input is the fully connected (FC) layer. However, the convolutional layer can accept the arbitrary dimension of the input. Therefore, the authors propose a spatial pyramid pooling layer situated after the last convolutional layer to obtain fixed-length features. In this way, the input images can be of any size. Another improvement is that they compute the feature maps from the entire image, and then extract feature maps for the arbitrary size proposals. The SPP-Net is 102 times faster than R-CNN during inference using ZFNet [56] as backbone and achieves better detection performance in the PASCAL VOC 2007 benchmark.

Fast R-CNN

Despite the improvements, SPP-Net is still time-consuming since 2000 candidate regions have to pass through R-CNN individually. Fast R-CNN [27] is developed to solve the disadvantages of R-CNN, providing more accurate detection results in less time. Improvements introduced in Fast R-CNN can be summarized as:

1. R-CNN and SPP-Net use a multi-stage training strategy. Instead, Fast R-CNN makes use of a ROI pooling layer to extract a fixed-length feature for each region proposal. Then the extracted feature maps are used in the second stage. The ROI

pooling layer is added after the last convolutional layer in the VGG-16 network, so that the input image only needs to pass the whole network once to obtain the feature maps and proposals at the same time.

2. Fast R-CNN utilizes a multi-task loss function, which jointly trains the classification task and the BBox regression task.

3. Fast R-CNN employs an end-to-end training strategy, which can update the parameters of the whole network.

Compared to SPP-Net, Fast R-CNN performs three times faster in training and ten times faster in testing, which achieves higher accuracy in the PASCAL VOC 2012 benchmark.

Faster R-CNN

Despite the significant improvement in speed and accuracy, Fast R-CNN still obtains region proposals using the selective search algorithm, what limits the detection efficiency. Ren et al. [57] develop a Faster R-CNN to solve this problem. To be precise, they propose a Region Proposal Network (RPN) to efficiently and accurately generate proposals, which is added after the last convolutional layer in the VGG-16 network. The classification task is performed as in R-CNN.

Specifically, the RPN initializes 9 anchor boxes of different scales and aspect ratios at each position of the feature maps provided by the last convolutional layer. The procedure of RPN is as follows: (1) the method slides over every point in the feature maps, extracting 9 anchors with different scales and aspect ratios, namely, three scales $(128^2, 256^2, 512^2)$ and three aspect ratios (1:1, 1:2, 2:1); (2) for every proposal, if the overlap rate is above 0.7, the proposal is classified as a positive sample, and if the overlap ratio is below 0.3, the proposal is classified as a negative sample. The remaining proposals with overlap between 0.3 and 0.7 are discarded. RPN shares features of the last convolutional layer with Fast R-CNN, enabling highly efficient region proposal computation.

By using the VGG-16 network as backbone, Faster R-CNN processes 5 frames per second (FPS) on a single GPU, while achieves the state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO benchmarks.

Feature Pyramid Network (FPN)

Typical DCNN models are based on hierarchical architectures which can provide multi-scale information regarding different features. These feature pyramids are commonly used in detection systems. Before the rise of deep learning, image pyramids were already used in many traditional approaches, where images are resized to different sizes to obtain multi-dimensional features. Although image pyramids can provide various hierarchical features, it is obvious that the calculation requirement is huge. Taking advantages of the hierarchical architecture of DCNNs, Lin et al. [58] propose the Feature Pyramid Network (FPN) which extracts feature maps from different convolutional layers, and then obtain predictions for the different scale features. In this way, the authors can obtain the feature pyramid in one forward propagation procedure. In order to obtain diverse semantic information, they develop a top-down path to combine the high-resolution and low-semantic feature maps from shallow layers with the low-resolution and high-semantic feature maps from deep layers. Using FPN in the Faster R-CNN architecture, their approach achieves state-of-the-art performance on the COCO detection benchmark. In addition, the runtime of their approach on a single GPU is reasonable.

Region-based Fully Convolutional Network (R-FCN)

Currently, most of deep neural networks are designed to classify the category of the image, such as AlexNet, VGG-16, and ResNet. So, one of the most important properties of the classification network is the shift-invariance, which means that when the object moves in the image, the classification result (confidence value) does not change much from before. However, object detection approaches need the network to be sensitive to the position of objects, so the shift-invariance may have a negative influence on the performance of object detection. To address this problem, Dai et al. [29] propose a position-sensitive ROI pooling layer. More precisely, the authors split every ROI into a k $\times$ k grid. Secondly, they extract features for each grid, and a softmax function is used to compute a score for every grid cell. As a result, they obtain a positive-sensitive score map from the softmax function, which is used to select high-quality proposals for the second stage. In the second stage, BBox regression is applied to obtain the final prediction, which is the same as the second stage in Faster R-CNN. Their approach can adopt a fully convolutional network as backbone, and it obtains an 83.6% mAP on the PASCAL VOC 2007 test set with 101-layer ResNet.

Mask R-CNN

He et al. [59] proposed Mask R-CNN, which aims at obtaining pixel-wise instance segmentation results and object detection results. Mask R-CNN also adopts a two-stage pipeline based on the Faster R-CNN approach: using the RPN to obtain proposals in the first stage, and predicting the category, position, and segmentation for the target within the BBox in the second stage. One of their main contributions is that they develop a ROIAlign layer to replace the ROI pooling layer. The ROIAlign layer is proposed to avoid the misalignments caused by the round operation of the ROI pooling layer. Basic calculations in most of the deep learning platforms are performed by using float numbers. However, when the proposal's coordinates need to map back to the input dimension, float numbers have to be converted into integer numbers, what introduces errors in this operation. As a solution, the ROIAlign layer applies interpolation calculation instead of the round operation. Another main contribution is that they add a new branch to obtain instance segmentation results. The authors apply a joint training strategy to train the three tasks, including object classification, object detection using BBox, and instance segmentation. By using the FPN architecture, Mask R-CNN achieves top results for the COCO object instance segmentation and BBox object detection benchmark.

Cascade R-CNN

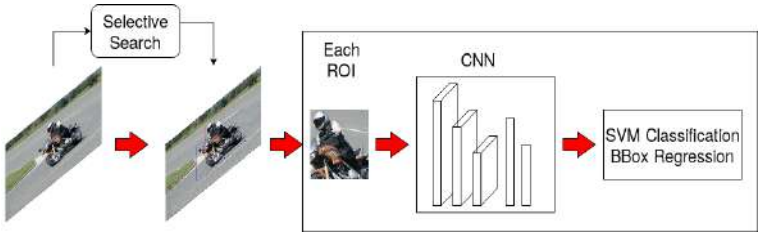
In Faster R-CNN, the distribution of proposals from RPN in the training and inference stages is different. In the training stage, there are lots of proposals from RPN that have high IOU value, while there are only a few proposals with high IOU in the inference stage. This problem leads to a decrease in the detection performance of the inference stage. Addressing this problem, Ouyang et al. [60] propose a Chained Cascade Network based on the Faster R-CNN architecture. The Cascade R-CNN expands the two stages of Faster R-CNN into a multi-stage R-CNN. The authors consider that each stage needs to select its corresponding quality proposals according to a special IOU threshold. For instance, in the second layer, they set the threshold of IOU to 0.5 to select proposals; in the third layer, the threshold is set to 0.6, and so on. Their cascade architecture can effectively solve the problem regarding the different distribution of the proposals in the training and inference stages, while they obtain higher quality proposals than Faster R-CNN. Their approach provides the best performance in the MS COCO Detection Challenge.

Libra R-CNN

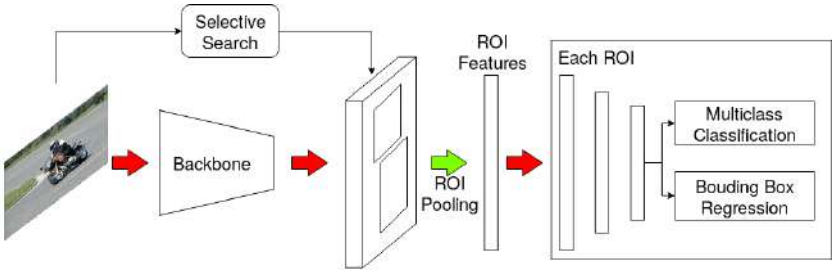
In object detection tasks, the number of proposals of the background is usually higher than the number of foreground samples, causing an imbalance between the positive and negative samples during training and inference. Recently, some researchers focus on solving the proposals' imbalance problem. Pang et al. [61] propose the Libra R-CNN focusing on solving the imbalance problem during training. They analyze the imbalance problem in three levels: sample level, feature level, and objective level. As a solution, they develop three novel components: IOU-balanced sampling, balanced feature pyramid, and balance L1 loss, respectively. More precisely, addressing the IOU-imbalanced problem, the authors find that most of the hard samples' IOU are lower than 0.05, then they uniformly sample hard samples based on the IOU value. For the feature imbalance, a four-step operation, including rescaling, integrating, refining, and strengthening, is applied to enhance the features presentation from the feature pyramid. In the end, for the BBox regression, the authors find that outliers (i.e. the regression loss value is above 1) contributed more than 70% of the gradient, while a large number of inliers (i.e. the regression loss value is below 1) only contribute 30%. Thus, the authors develop a balanced L1 loss to solve the objective level imbalance problem. Libra R-CNN significantly improves the detection performance in 2.5 points regarding FPN on the COCO Detection Challenge.

NAS-FPN

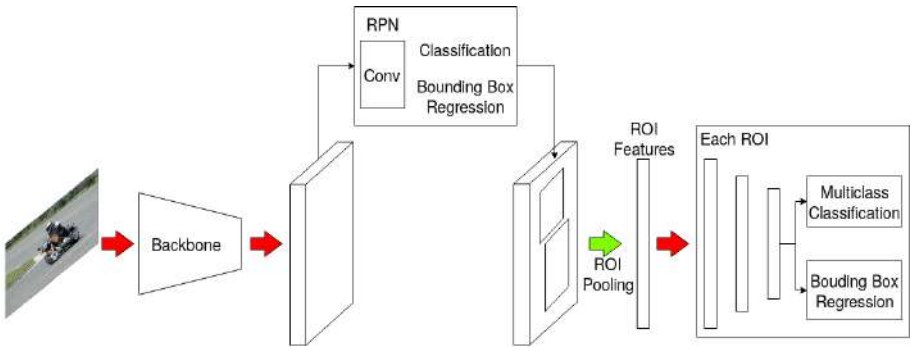
Ghaisi et al. [62] propose a novel feature pyramid architecture, which obtains better performance than FPN. The motivation of this work is to automatically perform a hierarchical connection of feature maps instead of the top-down path connection used in FPN. The advantage of feature fusion is that it can retain the details of low-level features on high-level features, especially for edge, texture, and shape features, which are ambiguous in the top layer features but are very helpful for positioning accuracy. Some previous works, such as PANet [63], introduced skip connection in feature pyramids. However, the architecture of this model was designed before training. To solve this problem, the authors apply a path search strategy with reinforcement learning to obtain better feature fusion architecture in the FPN. In the NAS-FPN, they train a controller using reinforcement learning to choose the best model architecture in the given search space, and the controller uses the accuracy of the sub-model in the search space as the reward signal to update the parameters. The experimental results on the MS COCO dataset show that NAS-FPN has good flexibility and high performance.



R-CNN [54]



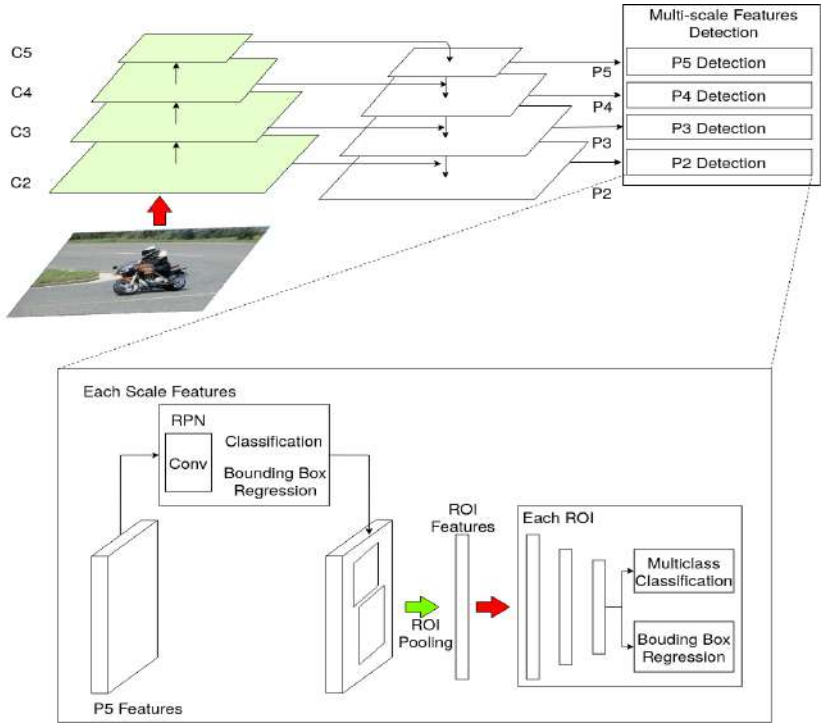
Fast R-CNN [27]



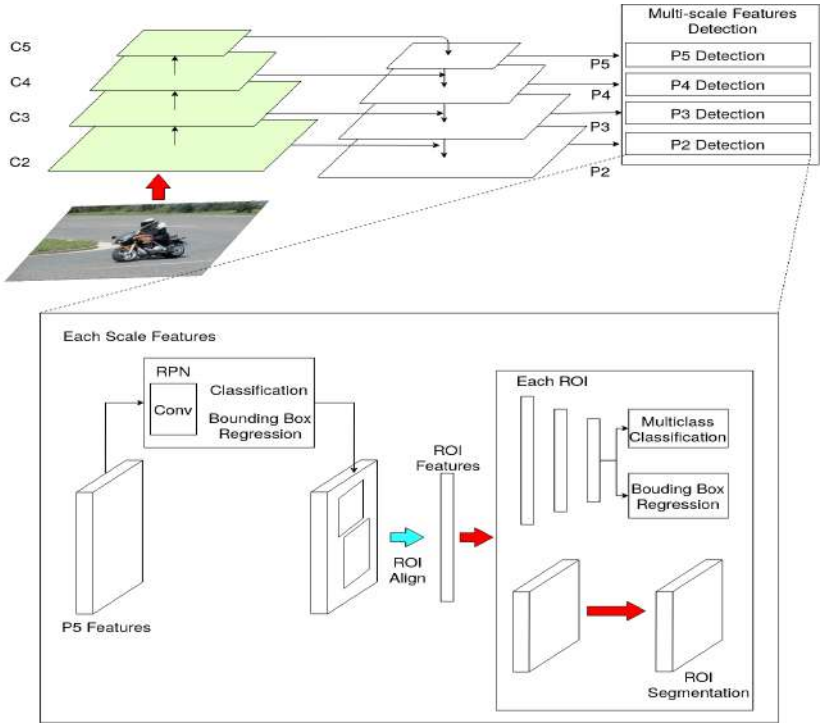
Faster R-CNN [57]

Figure 3.1: High-level diagrams of the two-stage object detection frameworks. (1/2)

In Figure 3.1 and Figure 3.2, we describe visually several two-stage object detection frameworks including some typical detectors, namely R-CNN [54], Fast R-CNN [27], Faster R-CNN [57], FPN [58] and Mask R-CNN [59].



FPN [58]



Mask R-CNN [59]

Figure 3.2: High-level diagrams of the two-stage object detection frameworks. (2/2)

3.1.1.2 One-Stage Frameworks

One-stage networks aim to directly predict the class category and the BBox coordinates using a single feed-forward DCNN, so that they do not include a separate procedure for region proposals generation. Unlike two-stage approaches, one-stage frameworks do not have a specific R-CNN stage for each ROI, so these networks are more efficient than the two-stage-based approaches in general.

OverFeat

OverFeat is proposed by Sermanet et al. [64] and represents the basis for the DCNN-based one-stage object detectors. In this work, the authors develop a fully convolutional neural network to conduct the object classification, localization, and detection. They select AlexNet as backbone and use three separate sub-nets for the three different tasks. In this way, all tasks are solved using a single framework and a shared backbone. The main contributions of this approach are summarized below:

- Multi-Scale Classification. OverFeat incorporates an offset pooling layer to output multi-scale features in order to reduce the computation of multi-scale classification. On the other hand, the authors propose a fully convolutional architecture to allow multi-dimensional input images.

- Combination Prediction. This method obtains *top-k* features from the classification network based on multi-scale and sliding window, and the regression network is used to regress the BBox coordinates on each scale. In the end, the final detection results are obtained by merging multi-scale detections.

Compared with R-CNN, OverFeat has a significant speed advantage but is less accurate. However, the idea of using the multi-scale and sliding window strategies on shared feature maps has inspired later works.

YOLO

Redmon et al. [28] propose the YOLO network for object detection. They discard the proposal generation procedure. Instead, they suggest that object detection can be considered as a regression problem that separately predicts the BBox and its associated category probability. More precisely, YOLO divides the image into an $S \times S$ grid, where each grid is responsible to predict the confidence score and the BBox coordinates. In this way, YOLO discards the whole region proposal generation step. The advantage of

YOLO is the velocity, which can reach 45 images per second through a 24-layer convolutional neural network. Besides, the authors also apply their approach to another simple convolutional neural network, increasing its velocity up to 155 images per second.

Later, Redmon et al. [65] propose YOLO9000 with several improvements: (a) Batch Normalization is used in the network, which brings 2% improvement on mean Average Precision (mAP); (b) an anchor box mechanism like the anchor boxes in Faster R-CNN is introduced, which increases the recall from 81% (provided by standard YOLO) to 89% on the Pascal VOC 2007 test set; (c) the authors use the K-means clustering method to select the anchor boxes instead of human-selected; and (d) the authors utilize a multi-scale training strategy to improve the robustness of the network. YOLO9000 has the ability to detect 9418 category objects and shows good performance.

Single-Shot Multibox Detector (SSD)

Liu et al. [30] propose a new detection approach based on MultiBox [66] named Single Shot MultiBox Detector (SSD), which has a speed similar to YOLO and obtains an accuracy similar to Faster R-CNN. The main contribution of SSD is that the authors combine the bottom- and top-layer features and apply the prior box strategy to assist in BBox regression. By using prior boxes during training, their network converges faster and turns out to be easy to optimize. On the other hand, their approach has the ability to detect multi-scale objects using multi-scale features: large-scale features (bottom-layer features) are used to detect small objects, while the small-scale features (top-layer features) are used to detect big objects. The authors utilize a VGG-16 network as backbone, but changing the last two fully connected layers by two convolutional layers. After the base network, they add four convolutional layers to complement the base network. Like most object detection approaches, SSD predicts a fixed number of BBoxes and scores, followed by a Non-Maximum Supression (NMS) step to produce the final detection. For a 300×300 input image, SSD obtains 74.3% mAP on the VOC2007 test at 59 FPS versus 7 FPS / 73.2% mAP achieved by Faster R-CNN and 45 FPS / 63.4% mAP achieved by YOLO.

CornerNet

Law and Deng [67] proposed an anchor free approach for object detection, named CornerNet. The authors found several disadvantages of anchor-based detection approaches: 1) since the anchor has to be generated at each point on the feature map, it produces an imbalance condition between positive samples and negative samples; 2) anchor boxes in-

troduce several hyperparameters, such as the scale and aspect ratio of the anchor boxes, which are relatively complicated to determine for different datasets. As its name indicates, CornerNet uses the top-left corner and bottom-right corner to predict BBoxes. In CornerNet, the backbone network consists of two stacked Hourglass networks [68] with a simple corner pooling approach. Their general idea is that the two Hourglass networks are responsible for detecting the top-left and bottom-right corners, respectively, using an embedding vector for each corner. Each of them is used to determine whether the pair comprising a top-left corner and a bottom-right corner belongs to the same object. CornerNet achieves 42.1% mAP on MS COCO, outperforming all previous one-stage detectors.

CenterNet

CenterNet [69] was inspired by CornerNet. The authors found that the prediction of CornerNet has lots of False Positive samples, and this problem is caused by the grouping of a top-left corner and a bottom-right corner for one BBox. Indeed, it is difficult to find the correct pair of two corners, because the two corners are always outside the object and the embedding vectors of the two corners can not perceive the internal information of the object. As a solution, CenterNet predicts not only two corners but also the center point. So, if the region determined by the two corner points contains the center point, this prediction is kept, otherwise it is discarded. This is equivalent to performing post-processing on the prediction of CornerNet, which obtains more accurate predictions. On the MS COCO dataset, CenterNet achieves 47.0% mAP, outperforming all existing one-stage detectors.

In Figure 3.3, we show the high-level diagrams of some typical one-stage frameworks: YOLO [28], SSD [30], and CenterNet [69].

3.1.1.3 Summary

In Table 3.1, we compare the performance of various object detection approaches on two benchmark datasets, namely PASCAL VOC 2012 [70] and MS COCO [25]. The evaluated approaches include the one-stage and two-stage frameworks we discussed before.

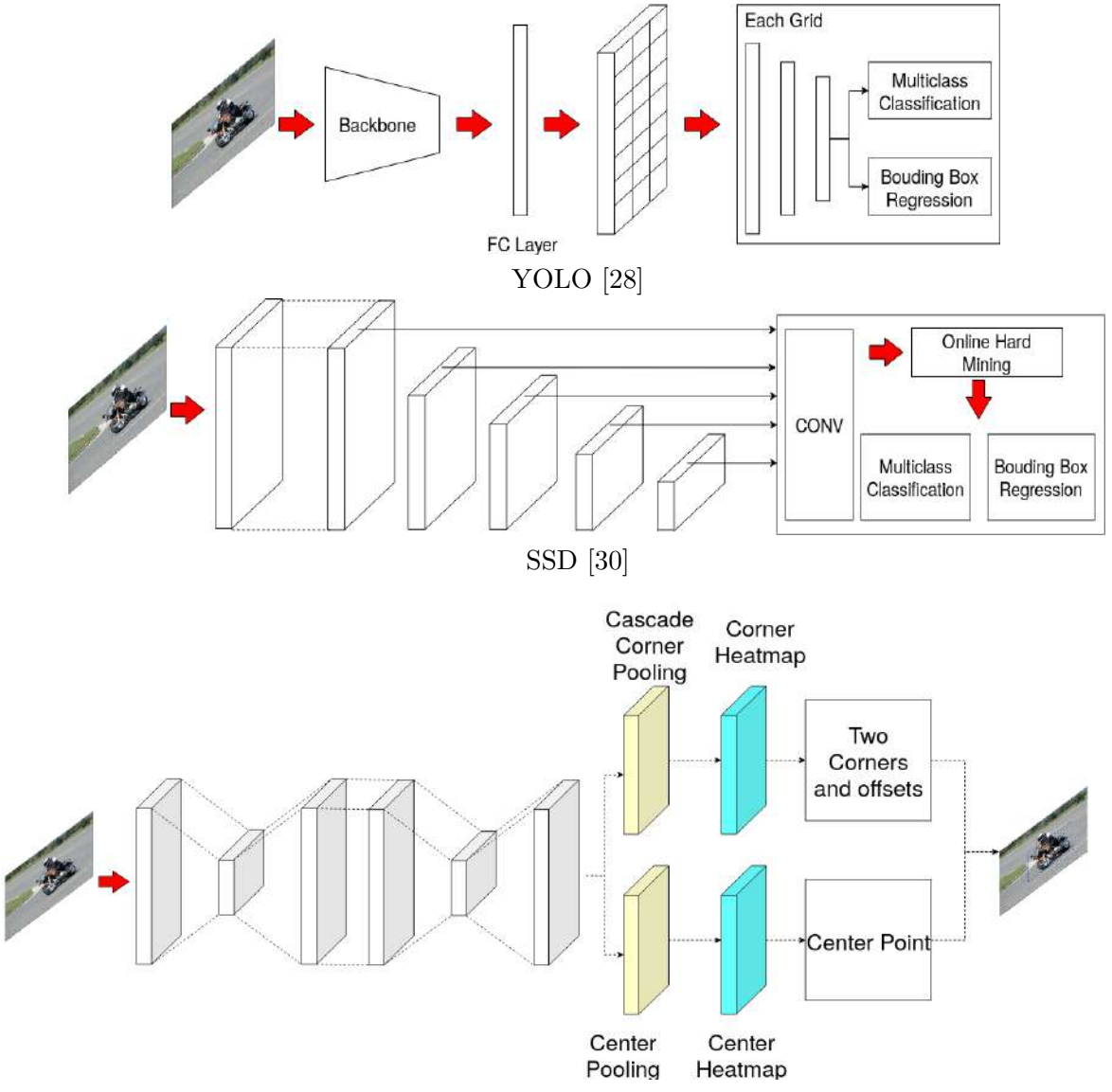


Figure 3.3: High-level diagrams of one-stage object detection frameworks.

3.1.2 Oriented Object Detection

Object detection algorithms using straight BBoxes have achieved excellent performance, but they meet difficulties for detecting oriented and elongated objects, such as text detection in natural scenes and elongated targets in satellite images. Approaches inspired by deep learning technology enjoy the advantage of automatic feature learning, avoiding the design and test of a large amount of potential hand-crafted features. Most of the current approaches are based on general object detection architectures, such as Faster R-CNN and SSD. On the other hand, a segmentation backbone is used to generate the independent categorical score masks, and then the oriented detection results are obtained from these score masks. In the following sections, we review some of these approaches, considering approaches based on region proposals in Section 3.1.2.1 and methods based on regional attention in Section 3.1.2.2.

3.1.2.1 Approaches based on Object Detection

Most of current approaches [71–80] are inspired by general object detection technologies [30, 57, 58]. These approaches adopt generic detection approaches but change the proposal strategy and BBox regression stage to estimate the orientation of objects.

As shown in Fig. 3.4 [A], [81] adopts an architecture based on SSD to locate the text in natural scenes. The authors pre-define some prior boxes by analyzing the text shape in the dataset, and then use a sliding window strategy to select a series of oriented and straight prior boxes by comparing the mIOU value. At the end of their network, they propose a smooth L_n loss for regressing the orientation of text, which has better performance than L_2 loss and smooth L_1 loss in terms of robustness and stability.

Similarly, TextBoxes [80] and TextBoxes++ [79] adopt SSD to fit the various orientations and dimensions of text. Their work is based on the SSD architecture and makes use of horizontal default boxes. In this way, their approaches require fewer default boxes in comparison with oriented prior boxes, so that TextBoxes and TextBoxes++ can converge fast and are easy to optimize. On the other hand, benefiting from the high efficiency of SSD, their approach can quickly and accurately obtain the straight and oriented detection results in one feed-forward procedure. On the ICDAR 2015 challenge, they achieve 0.817 F1-score at 11.6 FPS. As an improvement, TextBoxes++ connects a text recognition network to the detection results, and outputs quadrilateral and Rotated Bounding Boxes (RBox) in parallel. In this way, they can effectively identify the contents of the text, achieving state-of-the-art performance for word spotting challenges.

Dickenson and Gueguen [78] propose a method for extracting and symbolizing build-

Table 3.1: Comparative results of different object detection algorithms.

Framework	Proposal	Multi-scale	VOC 2012 (mAP)	COCO (AP)	FPS
R-CNN [54]	Selective Search	×	58.2%	-	0.03
SPP-Net (ZFNet) [55]	Selective Search	✓	59.2%	-	0.38
Fast R-CNN [27]	Selective Search	✓	63.1%	-	0.5
Faster R-CNN [57]	RPN	✓	73.2%	-	5
FPN (ResNet-101) [58]	FPN	✓	-	39.0%	4-6
R-FCN (ResNet-101) [29]	RPN	✓	83.6%	31.5%	1
Mask R-CNN (ResNet-101) [59]	FPN	✓	63.1%	37.1%	5
CC-Net (VGG16) [60]	Chained Cascade RPN	✓	80.4%	-	0.44
Libra R-CNN (ResNet-101) [61]	Libra RPN	✓	-	37.2%	-
NAS-FPN (AmoebaNet) [62]	NAS-FPN	✓	-	48.3%	4-6
YOLO (DarkNet) [28]	-	×	63.4%	-	45
YOLO9000 (DarkNet) [65]	K-means prior Anchor	✓	78.6%	-	40
MultiBox (Inception) [66]	-	-	29%	-	2
SSD (VGG16) [30]	Default Box	✓	74.3%	34.9%	59
CornerNet (Hourglass) [67]	-	✓	-	41.1%	7
CenterNet [69]	-	✓	-	47.0%	3.7

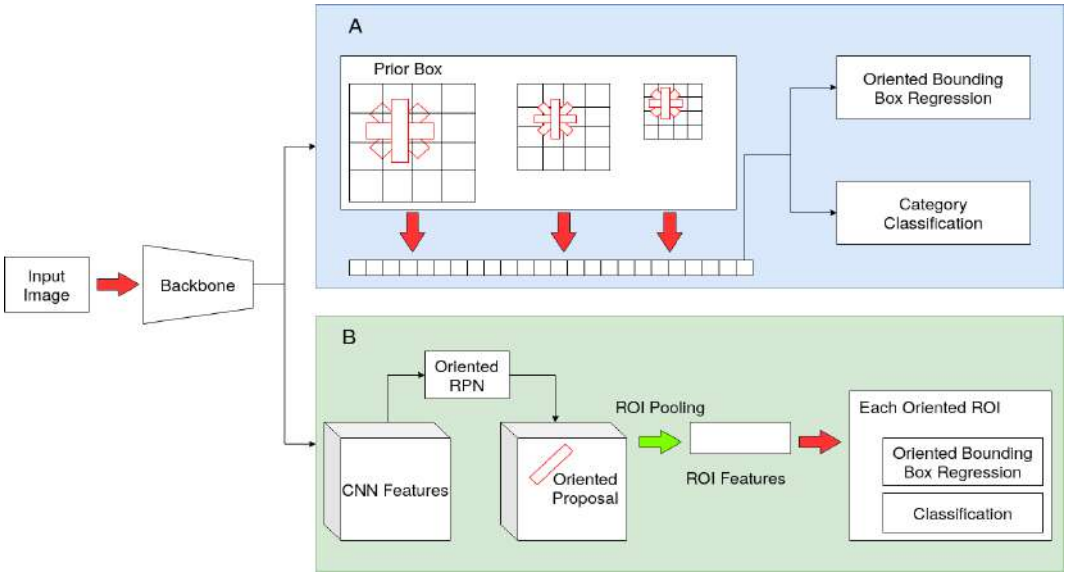


Figure 3.4: High-level diagrams of oriented detection approaches based on BBox detection algorithms: (A) using the SSD and (B) using the R-CNN architecture.

ing footprints (BFP) by using a convolutional neural network based on SSD. The authors regress the coordinates and orientation of BFP through one fully convolutional network, where the regression terms are based on the center point, width, height, and orientation angle of BFP. Experiments are conducted on the four cities included in the DeepGlobe Challenge dataset and their approach achieves the best performance.

Other methods [71–77] adopt two-staged object detection frameworks based on the Faster R-CNN architecture, as shown in Fig. 3.4 [B]. This is the case of R2CNN [73]. In its first stage, the authors make use of RPN to generate straight BBoxes, which are the minimal enclosed rectangles of oriented texts. Secondly, a multi-task loss function is employed to simultaneously output the straight BBoxes, the text/non-text confidence value, and the inclined angle to represent the orientation. Similarly, R2CNN++ [71] also employs the Faster R-CNN architecture to detect the small objects in aerial images. However, R2CNN++ incorporates an inception fusion network (IF-Net), which applies the inception module in GoogLeNet [82] to solve the detection problem for small objects. Following the IF-Net, they design a multi-dimensional attention network (MDA-Net) to generate an attention map, consisting of spatial and channel attention scores, to remove the noise in the feature maps. Then, a Rotational Region Proposal Network (RRPN) is developed to provide oriented priors. Finally, a smooth L_1 loss is used to regress the coordinates and the inclined angle of each BBox. Their model achieves 71.16% and

75.35% mAP for straight BBox detection and RBox detection in DOTA [83] dataset, respectively.

In [72], the authors present a Rotational Region Proposal Network (RRPN) based on Faster R-CNN to obtain the orientation of objects. They propose a Rotational Region-of-Interest (RROI) pooling layer to project arbitrary-oriented proposals to the feature maps. They experiment with their approach on three real-world text detection datasets, i.e., MRSA-TD500 [84], ICDAR2013 [85], and ICDAR2015 [86], obtaining more accurate results than previous approaches.

Unlike previous approaches, [74] designs a new corner-based region proposal network to generate the quadrilateral region proposals by detecting and linking the corners of text bounding-boxes. The new method does not apply anchor boxes; instead, they apply a corner proposals strategy to save massive computation due to the procedure of anchor boxes. On the other hand, a Dual-Pooling layer is embedded in the region-wise subnetwork for data augmentation. The experimental results show that the Dual-Pooling layer can improve the utilization of positive samples in the training set. Their approach achieves an F1-score of 0.876 on ICDAR 2013, and 0.845 on ICDAR 2015.

3.1.2.2 Approaches Based on Regional Attention

Instead of using region proposals, some works [87–91] focus on using the attention maps from DCNN to obtain oriented detections. As shown in Fig. 3.5, regional attention maps can provide a rough position prediction, while a post-processing module is used to obtain accurate RBox predictions.

He et al. [87] propose a DCNN regression-based approach for multi-oriented scene text detection. There are several novel contributions in this work: first, they develop a direct RBoxes regression strategy to obtain the inclined angle of the object, instead of the indirect regression terms like Faster R-CNN and SSD; second, the whole pipeline of their approach only has two parts, one is an FCN-based neural network that is used to obtain the regional attention maps, and the other is to apply the Recalled Non-Maximum Suppression as a post-processor to get the final predictions. In this way, their approach can obtain oriented rectangle and irregular quadrilateral predictions, achieving 81% F1-score on the ICDAR2015 benchmark.

Similarly, He et al. [88] apply a hierarchical inception module on a fully convolutional network and they propose a single-shot text detector. Their network includes three main parts: a convolutional component, a text-specific component, and a BBox prediction component. The convolutional component is inherited from the SSD detector. The

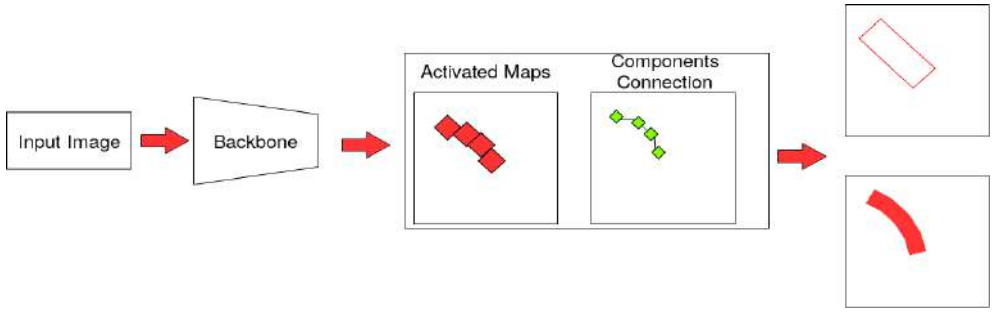


Figure 3.5: High-level diagrams of oriented detection approaches based on Regional Attention: using an FCN [52] similar architecture, a word-level or text-level attention map is generated, then a post-processor is applied to obtain the final predictions.

text-specific component includes a text attention module and a hierarchical inception module, which are used to, respectively, predict the probability heatmap and obtain better convolutional features. As a result, the text attention module can significantly reduce the negative effects of noise. Finally, the BBox prediction component applies RBoxes regression to predict five parameters (the coordinates of the center point, width, height, and oriented angle). Besides, it can also obtain oriented detection at word-level. They conduct experiments on the ICDAR2015 dataset and achieve an F1-score of 77%, outperforming the state-of-the-art results.

Zhou et al. [89] propose an effective approach for text detection in nature scenes, comprising two stages: a Fully Convolutional Network (FCN) and an NMS merging stage. The FCN directly produces text region predictions at the pixel-wise level, abandoning redundant and time-consuming proposals procedures. Besides, thanks to the accurate regional attention map, their approach can obtain text detection in RBoxes at word-level. In the second stage, the authors propose a Locality-Aware NMS to use a threshold strategy to obtain the predicted geometric shape. Quantitative experiments are conducted on ICDAR 2015, COCO-Text, and MRSA-TD500 datasets. The experimental results demonstrate that their method outperforms the state-of-the-art approaches in terms of both accuracy and efficiency.

Liao et al. [90] propose a rotation-sensitive regression detector (RRD) for oriented scene text detection. As usual, text detection in natural images involves two tasks, which are text presence detection (classification task) and RBox regression. As mentioned in [29], the classification task needs position invariance, while the RBox regression must be sensitive to the location of objects. Therefore, the authors consider using an individual features-based strategy for classification and detection tasks to prevent the performance degradation. In this way, they propose to perform classification and regression on differ-

ent branches. From the oriented detection branch, the authors extract rotation-sensitive features by rotating the convolutional filters. Their approach achieves state-of-the-art performance on four oriented scene text benchmarks, including ICDAR 2015, MRSA-TD500, RCTW-17 and COCO-Text.

3.2 Deep Learning for Semantic Segmentation

Image segmentation is another basic problem in Computer Vision, whose target is to assign a label for each pixel of a given image and group them into several visually meaningful or interest regions, aiming at scene understanding. Recently, the success of deep learning techniques in various high-level computer vision tasks, specifically for the problem of image classification and object detection, has motivated researchers to explore the capabilities of such networks for pixel-level labeling problems like semantic segmentation.

Currently, image semantic segmentation has various application scenarios, such as detecting road condition that plays a crucial role in a self-driving car, medical image lesion segmentation that can significantly improve the work efficiency of doctors, and segmenting targets in remote sensing images that can extract different terrain, locate roads and forests, etc. On the other hand, image semantic segmentation is used as a pre-processing step to convert the original image into more abstract and computer-friendly forms, which not only can remain the crucial feature in the image but also effectively reduce useless information and improve work efficiency.

With the rising of DCNN, the performance of segmentation has been significantly enhanced. The key advantage of these deep learning techniques is that DCNNs have the ability to learn the appropriate feature representation for the task in an end-to-end model rather than using manual features, which requires domain expertise to spend much time on fine-tuning to make them work on a particular scenario.

On the other hand, DCNN-based image segmentation approaches usually require a large number of annotations for the training dataset, where every pixel in the image must be marked. In the practical work, pixel-wise labelling of images is a time-consuming work for human beings. Therefore, the pixel-wise annotation of datasets limits the performance of current segmentation approaches. As a solution, some researchers focus on using weak annotations for semantic segmentation. Now, there are three main types of weak annotations, which are image tags, BBoxes, and other annotation forms, such as points or scribbles. The purpose of weakly-supervised semantic segmentation is to apply partially marked pixels to train the semantic segmentation network.

In the following sections, we will introduce the semantic segmentation techniques from two aspects: (1) introduce the main technologies of image semantic segmentation, and (2) review some typical weakly-supervised semantic segmentation approaches.

3.2.1 Main Techniques in Semantic Segmentation

A DCNN features a small receptive field in the shallow layers of the network through a series of small convolution kernels. As the number of network layers increases, the receptive field in the deep layers enlarges. In this way, DCNNs have the ability to obtain local features in the shallow layers and global information in the deep layers. Current DCNN-based semantic segmentation approaches fully apply features from shallow and deep layers to accomplish the segmentation task, achieving remarkable performance.

In this section, we introduce the main techniques on DCNN-based semantic segmentation, including decoder variants, CRFs, dilated convolution, feature pyramids, attention mechanisms, adversarial learning, and RNN.

3.2.1.1 Decoder Variants

Nowadays, most of the segmentation approaches apply an auto-decoder network based on a classification network. Long et al. [52] firstly introduced DCNNs to image segmentation achieving outstanding performance. In this work, the authors construct a fully convolutional neural network (FCN) with the replacement of the fully connected layers by convolutional layers in VGG-16. Then, the authors refer to the architecture of the auto-encoder after the modified VGG-16 network. Particularly, the authors apply the VGG-16 network as the encoder and use a deconvolution layer [92] for upsampling, incorporating some convolutional layers as a decoder. During training, the authors apply the back-propagation algorithm to automatically learn the weights of each layer. The architecture is shown in Fig 3.6. The FCN is trained end-to-end, allows arbitrary input size, and produces correspondingly-sized output with efficient inference and learning. FCN achieves state-of-the-art performance in PASCAL VOC 2012 challenge.

Apart from FCN, some researchers are also inspired by the architecture of the auto-encoder to design a symmetrical network for image semantic segmentation. SegNet [93] is a typical auto-decoder architecture. The authors design a transformation network that can adopt the classification network for semantic segmentation. In this work, the encoder is a network model that also uses the VGG-16 network, mainly for analyzing object information. The decoder is intended to transfer the information from the encoder part into the final segmentation results, where each pixel is represented by the index

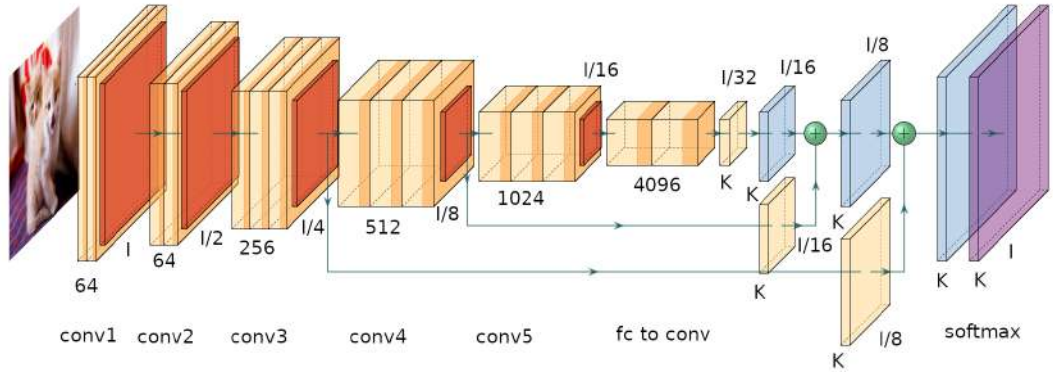


Figure 3.6: High level diagram of the FCN architecture. Several convolutional layers are applied to replace the fully-connected layers from the VGG-16 network, as shown in the *fc to conv* part. Next, feature maps from different layers are fused and a deconvolution layer is used for upsampling. In this way, the output results of the same size as the input.

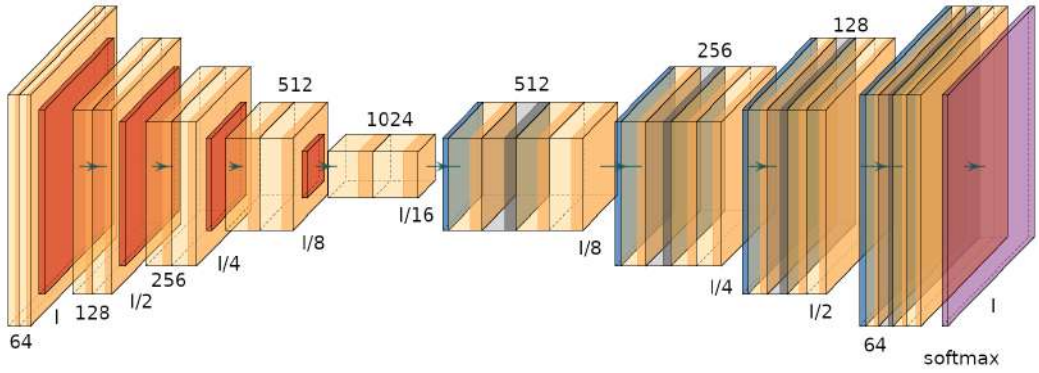


Figure 3.7: High level diagrams of the SegNet architecture. It presents a typical symmetrical auto-encoding and auto-decoding structure.

corresponding to its categorical information. The decoder of SegNet is composed of a series of up-sampling and convolutional layers, and then a softmax classifier is finally connected to predict each pixel's category. Unlike FCN, the up-sampling layer in SegNet is the linear interpolation operation. The SegNet architecture is shown in Fig. 3.7.

Ronneberge et al. [94] propose a similar architecture named U-Net for biomedical image segmentation. The architecture consists of a connection path to combine the features of the encoder and the decoder. The symmetrical architecture of U-Net, which is shown in Fig. 3.8, can obtain precise segmentation results for small targets. Compared to SegNet, U-Net considers fusing the feature maps of the encoder and decoder together by adding feature maps in the corresponding positions in the decoder part. In this way,

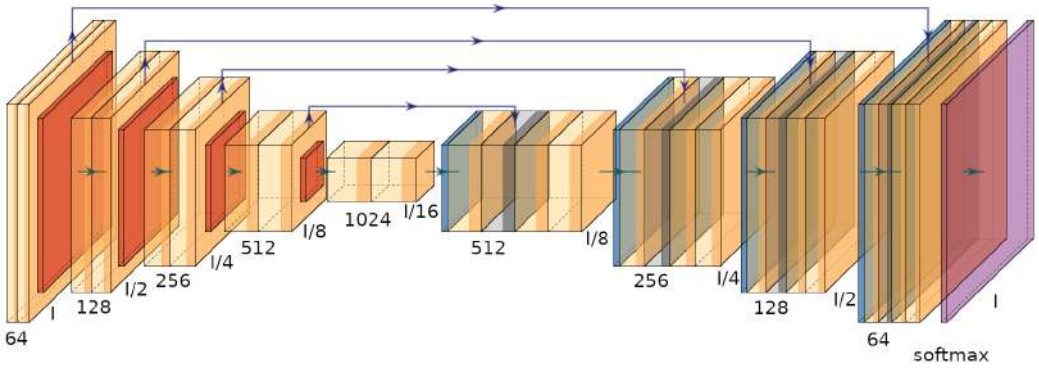


Figure 3.8: High level diagrams of the U-Net architecture.

their approach has the ability to combine the detailed information from the shallow layers with the semantic information from the deep layers. U-Net can be trained end-to-end with very few images and obtains better performance than the previous best method.

3.2.1.2 Conditional Random Fields

Although segmentation approaches based on auto-encoders have achieved excellent performance, the auto-decoder approaches obtain features through a small convolutional kernel (usually 3×3 or 5×5) sliding on the input image to obtain local features. Therefore, these approaches cannot integrate information from the global context of the image, which is useful to obtain the accurate boundary of targets and improve the detection ability of small targets. Currently, some researchers focus on combining such global information in DCNN-based models, by means of adding fully connected Conditional Random Fields (Dense CRFs) as a post-processor for the segmentation results. Dense CRFs [33] build a fully connected graph using each pixel as a node and the similarity between two pixels as the weight of the corresponding edge in the graph. Dense CRFs compute an energy function as the optimized target. This energy function is shown in Eq. 3.1, where the unary term $\psi_u(x_i)$ usually is the output of the softmax function, and the pixels pair-wise terms $\psi_p(x_i,\ x_j)$ are computed by a Gaussian kernel over the RGBXY feature space. The Dense CRFs model can effectively improve the segmentation performance without much extra calculation. This combination is particularly important for capturing global interaction and fine-tuning the local details, and it can combine the high-level features from the DCNN layers with the low-level features from

the input image.

$$E(x) = \sum_i \psi_u(x_i^0) + \sum_E \psi_p(x_i^1,\ x_j^1) \tag{3.1}$$

In Eq. 3.1, x^0 indicates the output of the softmax function, and x^1 indicates the RGBXY feature space corresponding to the input image.

DeepLab series of works [95, 96] make use of a fully connected (dense) CRF model as post-processor to optimize segmentation results. Dense CRFs consider using the whole image to build a fully connected graph, so the system can consider the global interactions in the low-level feature space. At the same time, they make use of a unary term using the softmax output with semantic information from the network. In this way, they combine the semantic information from the DCNN with the interaction information from low-level features. As a result, the Dense CRFs can effectively improve the segmentation performance of decoder-based approaches for object boundaries.

Another typical work applying a Dense CRF to refine the segmentation is the CRFas-RNN [97]. Compared with the auto-encoder, their work can obtain a more robust model and more precise results. Unlike DeepLab, the authors consider the inference procedure of Dense CRFs as an RNN, and the Dense CRFs are integrated into the segmentation network by applying an end-to-end training strategy, instead of as a post-processor of the segmentation results. The main contribution of this work is that the iterative solving and reasoning process of Dense CRFs is regarded as the RNN operation, which is embedded in FCN and jointly trained together. In this way, the authors combine the color and space prior knowledge into the DCNN segmentation model, obtaining more refined results. Their approach achieves state-of-the-art performance on the challenging Pascal VOC 2012 segmentation benchmark.

Using Dense CRFs within a semantic segmentation model can effectively improve the performance of the segmentation network.

3.2.1.3 Dilated Convolution

Most segmentation approaches apply a classification network to obtain multi-scale features. However, DCNN models usually use a pooling layer to reduce the image size while increasing the receptive field. In the field of image segmentation, since prediction is performed at pixel-level, it is necessary to enlarge the small resolution feature maps to the original dimension. Therefore, DCNN-based approaches have a process of reducing the size first and then reverting to the original size. Currently, no matter whether max-pooling or average-pooling is used, lots of detailed information is lost in the down-

sampling processing. Some researchers designed the dilated convolution to solve this problem by increasing the receptive field, allowing each convolution output to contain a larger range of information. Experimental results prove that the dilated convolution can be applied well in problems where images require global information or speech text requires longer sequential information dependencies.

Figure 3.9 provides a graphical explanation of dilated convolution. The dilation rate l controls the increasing factor of the receptive field. As shown in Fig. 3.9, stacking l-dilated convolution makes the size of the receptive field grow exponentially, while the number of parameters for the filters keeps at a linear growth. It means that dilated convolution allows efficient dense feature extraction on any arbitrary resolution.

In [98], the authors apply the dilated convolution for semantic segmentation tasks. The main idea of this work is to use dilated convolution to aggregate multi-scale contextual information without pooling layers. The authors neither apply max-pooling nor average-pooling. Instead, they design a new convolutional architecture that systematically uses dilated convolutions for multi-scale context aggregation. In detail, the authors modify the VGG-16 network for semantic segmentation by discarding the last two pooling layers, and each of the pooling layers is replaced by a dilated convolutional layer with a dilation factor of two. The obtained results prove that the dilated convolutional layer can effectively improve segmentation accuracy.

ENet [99] is another example of the application of dilated convolutions for semantic segmentation. The authors consider that the forward calculation of some previous approaches, such as approaches based on VGG-16, takes a long time, processing only four or five FPS on the terminal device, which can not meet the requirements of mobile devices. In this work, the authors introduce several tricks for semantic segmentation:

- They make use of dilated convolutions to decrease the resolution of the feature map, solving the problem due to down-sampling. Compared to FCN (32 times down-sampling), ENet only performs 8 times down-sampling.

- They consider that the classification network layers should not directly contribute to the segmentation task. Instead, the backbone network should act as a good feature extractor and only process the input for the decoder of the network.

- The architecture of the encoder/decoder is not mirror-symmetrical, unlike most of the segmentation networks. In ENet, the decoder only occupies a small part, while the encoder is larger and mainly performs image information processing and filtering.

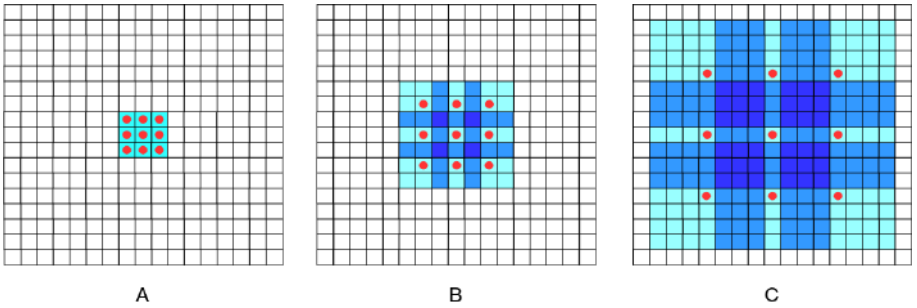


Figure 3.9: Diagram illustrating dilated convolution. In this example, only the 9 red points of the 3×3 kernel are involved in the convolution operation, while the rest of the receptive field is ignored. A: 1-dilated convolution with a 3×3 kernel and its receptive field is 3×3; B: 2-dilated convolution and its receptive field of 7×7; C: 4-dilated convolution and its receptive field of 15×15.

- In most network architectures, ReLU and Batch Normalization layers are used after convolutional layers. However, the authors find that using ReLU decreases model accuracy. Instead, they apply Batch Normalization and PReLU layers after convolutional layers.

Experimental results show that ENet can achieve 46.8 FPS, providing good performance on the Cityscapes, CamVid, and SUN datasets.

3.2.1.4 Feature Pyramid Approach

The image pyramid technique is a traditional approach in image processing, which applies successively multi-scale sampling to obtain multi-scale features. There are two common kinds of image pyramids: the Gaussian pyramid, which is used to decrease the resolution of the input image; and the Laplacian pyramid, which is used to reconstruct the input image.

In the field of semantic segmentation, spatial pyramid pooling is an example of using the image pyramid approach. For a start, the work described in [96] proposes the Atrous Spatial Pyramid Pooling (ASPP) approach to robustly segment objects at multiple scales. The ASPP module concatenates different feature maps obtained from several dilated convolutional layers. More precisely, it includes one 1×1 convolution and three 3×3 convolutions with different dilation rates. Batch normalization is used after each of the parallel convolutional layers to accelerate the converge procedure. The ASPP module uses different dilation rates of convolutional layers in parallel and obtains image context at multi-scale, as shown in Fig 3.10. In this way, the authors can obtain

multi-scale feature maps in one feed-forward procedure. The ASPP module is similar to the context module used in [99], but it is applied directly to the feature maps in the middle layers of the network instead of to the final output. Their work obtains a new state-of-the-art performance at the PASCAL VOC 2012 benchmarks, reaching 79.7% mIOU in the test set.

Lin et al. [100] proposed a convolutional network with a sliding pyramid pooling module to obtain multi-scale feature maps. Regarding scene recognition, contextual relations are ubiquitous and provide important clues for segmentation. Spatial context relationships can provide compatible relationships between objects. For instance, a car might appear on a highway, and it is impossible that a car is in the sky. Addressing this idea, the authors explore two types of spatial context to improve segmentation performance: patch-patch and patch-background. For the patch-patch context, a CRF-based approach is proposed to compute an energy function using CNN-based patch pair-wise energy to obtain the semantic information of two adjacent patches. Unlike previous CRFs-based works, whose purpose is to depict object boundaries, the patch-patch context is used here to improve the coarse-level segmentation performance. For the patch-background context, the authors make use of an ASPP module. As shown in Fig. 3.10, the ASPP can obtain multi-scale feature maps in one feed-forward procedure, discarding the inefficient multi-scale input. Their approach obtains state-of-the-art performance on NYUDv2, PASCAL VOC 2012, PASCAL-Context and SIFT-flow datasets.

The Laplacian pyramid is another commonly used pyramid in semantic segmentation. In [101], a multi-scale reconstruction architecture is developed, which applies the skip connections from high-resolution feature maps and multiplicative gating to successively refine segment boundaries and then reconstruct the image from low-resolution maps. In this work, the authors consider that the pooling layer can destroy the spatial and internal information of convolutional features. Therefore, they develop a reconstruction architecture to solve this problem. In their reconstruction results, it is clear that the results from bottom layers contain lots of semantic mistakes, but have clear boundaries and positions. On the other hand, the results from the top layers have ambiguous object boundaries, but contain correct semantic predictions. Inspired by ResNet, the authors make use of a convolutional network to learn residual segmentation, that is, they make use of the high-resolution feature maps from the shallow layers to learn the accurate boundary, and low-resolution feature maps from deep layers to learn the semantic information. Then, they combine the accurate boundary prediction from shallow layers with the semantic segmentation results from deep layers to obtain the final segmentation results. Their approach reaches state-of-the-art performance on the PASCAL VOC and

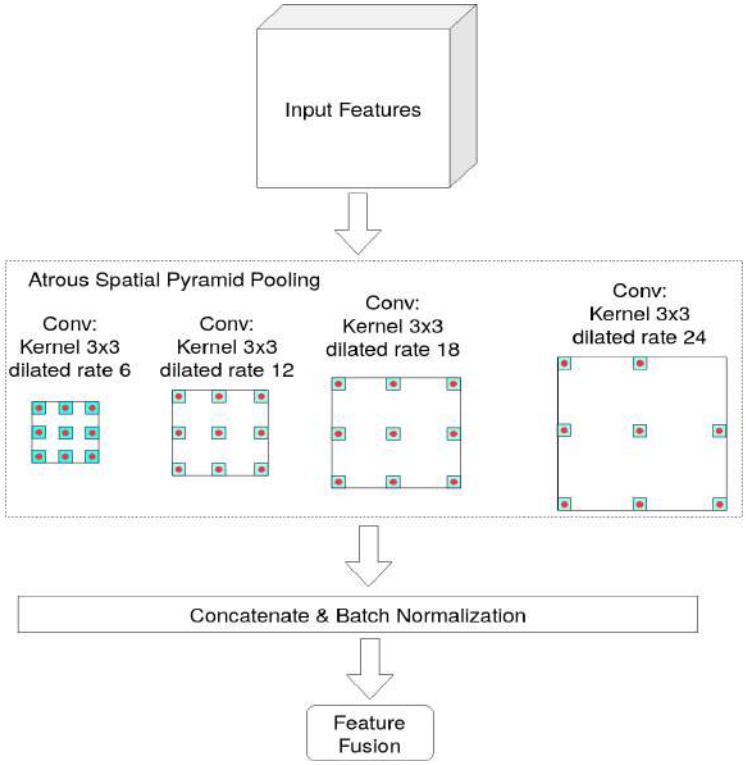


Figure 3.10: A diagram of the Atrous Spatial Pyramid Pooling Module.

CityScapes benchmarks.

3.2.1.5 Attention Mechanism

Attention mechanism constitutes a widely used technique in Natural Language Processing (NLP), such as text and speech processing, morphological analysis, lexical semantics, etc., and it is also widely used in semantic segmentation, aiming at improving the expressiveness of feature maps. The role of attention modules can be considered as using information transferred from several subsequent layers/feature maps to select the most valuable information in the input vectors.

Currently, there are some works [102–105] using the attention modules based on U-Net. Lian et al. [102] propose the ATTention U-Net in order to help the network to learn the discriminative features for the iris detection. Their approach has two steps: (1) the first step is to regress a BBox of the potential iris region and to generate an attention mask; (2) the second step is to use the attention mask as a weighted function to merge with discriminative feature maps in the model, making the segmentation model

to pay more attention to the target region. The authors evaluate their approach with UBIRIS.v2 and CASIA.IrisV4 databases, achieving mean error rates of 0.76% and 0.38%, respectively.

Oktay et al. [103] integrate attention gates (AGs) into a skip connected U-Net for pancreas segmentation in medical images. AGs can eliminate the necessity of using explicit external tissue/organ localization modules of cascaded convolutional neural networks, and it can be easy to integrate into other CNN architectures. Experimental results show that AGs consistently improve the prediction performance of U-Net.

Similarly, Li et al. [104] develop a fully automatic approach based on deep learning for breast mass segmentation. Their approach contains an encoder based on a densely connected convolutional network (DenseNets) and a decoder using attention gates (AGs) to integrate bottom- and top-level feature maps. Their approach is evaluated on the public and authoritative Digital Database for Screening Mammography (DDSM) database, achieving the current state-of-the-art performance.

Ni et al. [105] propose a Refined Attention Segmentation Network (RASNet) to simultaneously segment surgical instruments and identify their category. Similarly to previous works, the attention module is adopted to help the network to focus on semantic areas. In this case, the targets are small, what introduces the imbalance problem in their application. To address this problem, the authors apply the weighted sum of the cross-entropy loss and the Jaccard index loss functions. Their approach achieves state-of-the-art performance in the MICCAI EndoVis Challenge 2017 (94.65% mean Dice score and 90.33% mIOU).

Some researchers combine the attention mechanism with different architectures, such as [106, 107]. Chen et al. [106] integrate the attention model in an FCN. In their work, the authors develop two attention modules to learn a soft weight for multi-scale features at each pixel position. More precisely, they propose a skip-net to fuse the multi-scale features and a share-net to obtain the shared weights using the attention mechanism. As a result, the proposed attention module not only enhances the expressiveness of feature maps, but also allows to diagnostically visualize the features at different positions and scales. Through extensive experiments, the experimental results demonstrate the effectiveness of the attention module on three challenging datasets, including PASCAL-Person-Part, PASCAL-VOC 2012, and MS-COCO 2014.

Similarly, Li et al. [107] propose a Pyramid Attention Network (PAN) to exploit the impact of global contextual information in semantic segmentation, which combines the attention module and spatial pyramid to extract precise dense features. The authors introduce a Feature Pyramid Attention Network module, which imposes a spatial pyra-

mid attention structure on the high-level features and is combined with a global pooling strategy to learn the best feature representations. In addition, the global context feature information is obtained by the global up-sampling operation in each up-sampling layer, which serves as a guide for low-level features. In this way, the network has the ability to locate detailed categorical information. The proposed approach achieves the best performance on the PASCAL VOC 2012 dataset. Without the pre-training process using the MS COCO dataset, their model obtains 84.0% mIOU on the PASCAL VOC 2012 benchmark.

3.2.1.6 Adversarial Leaning

Goodfellow et al. [108] proposed a new framework to train the generative model via an adversarial learning strategy. Their Generative Adversarial Network (GAN) takes samples following a fixed distribution and transforms them using a DCNN to approximate the distribution of training samples. The principle of GAN is simple. A typical GAN model comprises two networks: a Generator ($G(\theta_g)$) and a Discriminator ($D(\theta_d)$). $G(\theta_d)$ is a network that is used to generate the expected output. It receives random noise $z \sim p_g(z)$ and generates $G(z, \theta_g)$. $D(\theta_d)$ is a discriminant network to determine whether an input image is true or false, which means that the input image is the ground truth or the generated result, respectively. $D(\theta_d)$ is trained using the generated results $G(z, \theta_g)$ and the ground truth Y to maximize the loss of $D(\theta_d)$. At the same time, they train the generator $G(\theta_g)$ to minimize $\log(1 - D(G(z, \theta_g), \theta_d)$. Therefore, the combined function is:

$$\min_G \max_D V(D, \theta_d, G, \theta_g) = E_{z \sim p_d(z)}[\log D(z, \theta_d)] + E_{z \sim p_g(z)}[\log(1 - G(z, \theta_g))] \quad (3.2)$$

Recently, some researchers consider applying GAN to image semantic segmentation. In [109], the authors view the procedure of segmentation as a generation process. In particular, the segmentation network outputs a label map using the raw image as input, so the segmentation network is used as the generator in their work. On the other hand, the discriminator's task is to determine whether the input image is generated by the segmentation network or taken from the ground truth. The discriminator in this work is significantly different from the discriminator in the original GAN, having a dual input: the input raw image and the segmentation ground truth. When the raw image and the segmentation ground truth are the input, the ground truth for the discriminator is 1; and when the input is the pair comprised by the raw image and the generated segmentation result, the discriminator ground truth is 0. Using this framework, the authors hope

the segmentation network can generate a segmentation result that is difficult for the discriminator to distinguish. They conduct experiments on the PASCAL VOC 2012 dataset, and the experimental results show that the adversarial training strategy can improve the segmentation performance.

Zhu et al. [110] use an adversarial training strategy to improve the robustness of the model and avoid over-fitting when using a small-scale dataset. In their work, they apply FCN for mammographic mass image segmentation and use Dense CRFs to refine the segmentation results. Furthermore, they introduce two measures to improve the performance of the segmentation:

- Before feeding the input image into the segmentation network, they generate a probability map, where a high value in the probability map represents that the possibility of belonging to the target is high. They combine the probability map with cross-entropy loss to train the segmentation network.

- Inspired by Goodfellow's work [108], the adversarial sample can make the original model classification to fail. The authors believe that if a model is robust, even if it is an adversarial sample, the segmentation result should be correct. In this way, they apply the adversarial training to improve the robustness of the segmentation network.

Their approach reaches the state-of-the-art performance on the INBreast and DDSM-BCRP datasets.

Similar to Zhu's work [110], some researchers [111, 112] consider using adversarial learning to improve the segmentation performance. In [111], the authors apply an adversarial training approach to improve on brain Magnetic Source Imaging (MRI) segmentation performance. They develop a discriminator based on an auto-encoder architecture to distinguish a binary mask between manual ground truth and predicted segmentation. For the generator, they use a dilated convolutional layer in FCN to increase the receptive field. The experimental results show that using the adversarial training strategy can improve the segmentation performance on two different datasets. Similarly, Rezaei et al. [112] also apply GAN for brain tumor segmentation. In this work, the authors exploit the conditional Generative Adversarial Network (cGAN) and train a semantic segmentation DCNN with an adversarial training strategy, where the discriminator is used to classify the inputs coming from the ground truth or from the output of the segmentation network. The proposed model achieves superior performance on the BraTS 2017 dataset.

Neff et al. [113] apply the adversarial training strategy for medical image segmentation. In this field, the task of pixel-level annotation for the supervised task is time-consuming and expensive, and it is also difficult for staff without professional experience. In their work, they develop a new variant of GAN, which is used to synthesize medical images and also to generate segmentation masks for the supervised task. The discriminator is the same as in [109], which has a dual input architecture to distinguish whether the input pair is from the real image batch or from the synthetic image batch. This approach is evaluated on lung segmentation involving thorax X-ray images. The experimental results show that GANs can be used to synthesize training data for this task.

Zhang et al. [114] propose a novel approach for biomedical image segmentation using unannotated and annotated images. The network consists of two networks: (1) a segmentation network (SN) is used to conduct semantic segmentation; (2) an evaluation network (EN) is used to assess the segmentation quality. Firstly, the SN is trained using annotated images for segmentation, and the EN is encouraged to distinguish between segmentation results of unannotated images and annotated ones. Secondly, the SN is used to produce segmentation results for unannotated images such that the EN can not distinguish these from the annotated images. The experimental results show that the proposed model is effective in utilizing unannotated images to obtain considerably better segmentation results.

3.2.1.7 Recurrent Neural Network-based Approaches

Recurrent Neural Networks (RNN) are designed for handling sequences of data. Among them, the long short-term memory (LSTM) is a type of RNN that is developed to solve the vanishing gradient problem in the typical RNN architecture [115]. Nowadays, some researchers consider applying RNNs for image semantic segmentation. In comparison with segmentation networks based on DCNNs, RNNs have the capability to consider the sequential continuity between continuous images. On the other hand, RNN-based modules have better performance when they deal with bigger targets that have more inter-slice information than when performing small target segmentation.

In [116], the authors apply Clockwork RNN (CW-RNN) for muscle perimysium segmentation. The CW-RNN module is proposed in [117], which aims at improving the potential long-term dependency with less parameters. The CW-RNN module requires 100 times less running time in comparison with CNN-based models. By applying the CW-RNN model in U-Net, the experimental results on the muscle perimysium dataset

shows a 5% improvement in mean accuracy.

In [118], a ReNet [119] model is expanded for semantic segmentation, named Reseg. This model has several stages. Firstly, the input image is fed into a VGG-16 to obtain feature maps. Secondly, the obtained features are sent to a ReNet model, which consists of four RNN layers, extracting features in four directions: top-down, bottom-up, left-right, and right-left, respectively. Then, a simple up-sampling model is used to recover the dimension to the size of the input. At last, a softmax function is used to output the probability of each pixel.

Based on Reseg, Li et al. [120] propose a novel Long Short-Term Memorized Context Fusion (LSTM-CF) model that captures and fuses contextual information from multiple channels of photometric and depth data. The LSTM-CF module consists of four parts: a module for vertical depth context extraction, a module for vertical photometric context extraction, a memorized fusion module for incorporating vertical photometric and depth contexts as true 2D global contexts, and a final layer for pixel-wise scene labeling given concatenated convolutional features and global contexts. Their approach obtains state-of-the-art performance on the large-scale SUNRGBD dataset and the NYUDv2 dataset.

Bai et al. [121] apply the LSTM concept to segment sequences of medical images. In this work, they combine a fully convolutional network with an LSTM model for image semantic segmentation, which integrates both spatial and sequential information. In their task, the annotations in the medical images are sparse. To solve this problem, the authors perform non-rigid label propagation on annotations and introduce an exponentially weighted loss function to train the network. In their network, the FCN is proposed to tackle pixel-wise classification, and they design a convolutional LSTM (C-LSTM) to model the sequential dependency in image sequences. In the end, they use a convolutional layer to obtain the final label maps sequence. This approach is evaluated on the aortic MR dataset, obtaining an average Dice score of 0.960 for the ascending aorta and 0.953 for the descending aorta.

A number of works introduced in this section are enumerated in Table 3.2. Besides, we also introduce the segmentation techniques used and their segmentation performance.

3.2.2 Weakly-Supervised Semantic Segmentation

Nowadays, Weakly-Supervised Semantic Segmentation (WSSS) has become a hot topic. Unlike fully supervised semantic segmentation approaches, which make use of accurate pixel-wise annotations, weakly-supervised approaches expect to use rough annotations, such as BBoxes, image tags, and other forms of annotations, to ultimately provide an

Table 3.2: Comparative results of different semantic segmentation algorithms.

Network	Segmentation Technique	Evaluation Dataset & Performance
SegNet [93]	Auto Decoder	PASCAL VOC, 59.1% mIOU
FCN [52]	Skip Connection Decoder	PASCAL VOC, 62.2% mIOU
U-Net [94]	Auto Decoder, Feature Fusion	ISBI Cell: 77.5% F score
DeepLab [96]	Dilated Convolution, Dense CRFs	PASCAL VOC, 79.7% mIOU
Fisher Yu [98]	Dilated Convolution	PASCAL VOC, 67.6% mIOU
PSPNet [122]	Skip Connection Decoder, Spatial Pyramid Pooling	PASCAL VOC, 85.4% mIOU
DeepLabv3 [82]	Atrous Spatial Pyramid Pooling	PASCAL VOC, 86.9% mIOU
FeatMap-Net [100]	Image Pyramid, Dense CRFs	PASCAL VOC, 78.0% mIOU
G. Ghiasi [101]	Laplacian Pyramid Reconstruction, Dense CRFs	PASCAL VOC, 77.5% mIOU
ATT-UNet [102]	Auto Decoder, Attention Mechanism	UBIRISv2, 76.0% F score
Attention U-Net [103]	Auto Decoder, Attention Mechanism	CT Pancreas, 82.5% F score
PAN [107]	Feature Pyramid Attention	PASCAL VOC, 78.4% mIOU
Shuyi Li [104]	Auto Decoder, Attention Mechanism	DDSM, 78.4% F score
P. Luc [109]	Adversarial Training	PASCAL VOC, 54.3% mIOU
P. Moeskops [111]	Auto Decoder, Adversarial Training	Brain MSI, 90.1% F score
ReSeg [118]	Bi-directional LSTM	CamVid, 58.8%
LSTM-CF [120]	Bi-directional LSTM, RGB-D Segmentation	NYUDv2, 49.4% Accuracy
M Rezaei [112]	Conditional Adversarial Network	BraTS 2017, 68.0% F score
Spatial CW-RNN [116]	Auto Decoder, CW-RNN	Skeletal Muscle Microscopy Images, 91.8% Mean Accuracy
W Bai [121]	Auto Decoder, RNN	MR Images, 96.0% F score
RACE-net [123]	RNN	DRISHTI-GS1, 97.0% F score
H Li [124]	Auto Decoder, BiLSTM	NIH Pancreas-CT, 83.9% F score

accurate pixel-level prediction. The reason behind the success and popularity of WSSS is simple: researchers need to spend 1.5 minutes to annotate a single image at the pixel level for the CityScapes dataset, while labeling a BBox for one image only takes 7 seconds, and image-level annotations only need one second [125]. Therefore, using the WSSS approach can effectively improve the work efficiency.

The difficulty of WSSS lies in the fact that it falls between the fully supervised approach and the unsupervised approach, and that the segmentation procedure is quite different depending on the kind of weak annotation used. BBoxes are typical representations for object position, which locate one object using a rectangle. Image tags supervision means that only image classification labels are used to train the segmentation network. Other forms of supervision include scribbles, point annotations, marking each type of semantic label as annotations, etc. Although there is a particular gap between the performance of models trained by weak annotations and the performance of models trained by full supervision, some works have proved that they can meet the requirements in certain scenarios.

3.2.2.1 Weakly-Supervised Semantic Segmentation using Image Tags

Although image tags annotation is the weakest annotation for segmentation, the annotation can be obtained very efficiently. Many researchers have currently worked for narrowing the gap between image tags supervised approaches and fully supervised approaches. Most current works apply Classification Activation Maps (CAMs) [126] as the segmentation proposals, which are obtained from the classification network through a global average pooling layer. In [126], the authors find that the Global Average Pooling (GAP) can be used to extract the object location from feature maps. The classification network, except the softmax layer, is used to extract features. A global pooling layer is then used after the last convolutional layer to compute the average for each feature. Then, a weighted sum is performed to obtain CAMs. In this way, the rough localization of objects is obtained in CAMs, and they can be used as the segmentation proposals to train the segmentation network. After the publication of this approach, lots of works focus on using CAMs to obtain accurate segmentation results.

Wei et al. [127] propose a new adversarial erasing strategy based on CAMs for semantic segmentation. Classification networks are only responsive to small and sparse discriminative regions for targets, which is too sparse to train a semantic segmentation network. In order to solve this problem, they continuously erase the target regions from feature maps and train the classification network in an adversarial manner. Through

erasing the current high responsive areas in CAMs, their approach can sequentially discover new target areas to get the whole outline of the target. Their approach obtains 50% and 55.7% mIOU scores on PASCAL VOC 2012 validation and test sets respectively, only using image tags as annotations.

Inspired by [127], Zhang et al. [128] also propose a weakly-supervised object localization approach using adversarial learning, named Adversarial Complementary Learning (ACoL). Differently from the iterative training strategy used in [127], the authors build two parallel-classifiers: the first classifier is used to generate CAMs, and the second classifier firstly erases the highest response area in the first classifier CAMs and then trains itself using image tags supervision. In the end, they fuse CAMs from the two classifiers to generate the final localization. The authors evaluate this approach using the ILSVRC dataset obtaining a localization error rate of 45.14%, which is the new state-of-the-art performance.

In [129], the authors find that most of the previous approaches make use of the highest response areas in CAMs. However, only using the highest response, the corresponding CAMs are usually biased towards a partial area of the target and sometimes even incorrectly locate the background area. To solve this problem, they propose a Combinational Class Activation Map (CCAM), which applies a linear combination using low and high responses in CAMs. To evaluate the response in CAMs, they apply a softmax function to obtain the probability of the category for each pixel. In this way, their CCAM can be used to locate targets and suppress the background area. As a result, their model obtains superior performance compared to previous approaches on representative target localization benchmarks, including ILSVRC 2016 and CUB-200-2011.

Kolesnikov and Lampert [130] develop a three-stage approach for weakly-supervised segmentation using image tags annotation, named Seed, Expand, and Constrain (SEC). In this work, the authors define a seed for each target as the starting point of segmentation. Then, they expand these starting points into the whole target based on similarity. Finally, they use boundary information to determine the boundary of expansion through Dense CRFs. The compound loss function is developed to jointly train the three tasks, which are to determine the starting point in CAMs, the degree of expansion, and the final refinement of the target boundary. They show in experimental results that a deep convolutional neural network trained by their compound loss function leads to substantially better segmentations than previous state-of-the-art methods on the challenging PASCAL VOC 2012 dataset.

Huang et al. [131] propose an improvement on the basis of SEC, named Deep Seeded Region Growing (DSRG). Since the seed area in SEC is set before training the segmen-

tation network, the incorrect localization in CAMs will be inherited in this stage. As a solution, DSRG applies a dynamic supervision strategy to train the segmentation network. More precisely, the seed areas expansion from CAMs is used to expand the seed area in each training iteration, and the expanded seed area is used to replace the current seed area in the next iteration. Then, the authors develop a combined loss function, including a seeding loss, which is used to train the network to obtain better seed areas in CAMs, and a boundary loss, which is used to fine-tune the boundary of segmentation results to match the ground truth boundary. Besides, a DSRG, which integrates the SRG approach into a DCNN, is developed to expand the seed areas in a back-propagation manner. Their approach obtains 63.2% mIOU on the PASCAL VOC 2012 test set and 26.0% mIOU on the COCO dataset.

Kwak et al. [37] propose a Superpixel Pooling Network (SPN) for weakly-supervised semantic segmentation using image tags supervision. Their approach can be divided into two steps: in the first step, they apply a classification network connecting with several up-sampling layers as the backbone; in the second step, the authors develop a superpixel pooling layer to generate the segmentation proposals. The superpixel pooling layer makes use of the superpixel segmentation results as input, which is used to reflect the low-level image features. In this way, their approach is trained in turn by using the SPN's results as the annotation of the segmentation network. After several iterations, their approach obtains good performance on the PASCAL VOC 2012 segmentation benchmarks.

Zhou et al. [132] propose an instance segmentation approach using image tags supervision. The motivation of this work is that the maximum response value in CAMs has strong visual semantic cues for detection targets. In their work, the authors first integrate the information of the categorical peak responses in CAMs and then use a back-propagation algorithm to expand it to an area with a large amount of object instance information. The expanded results are called Peak Response Maps (PRMs), which provides detailed representations of instance objects and can be used as segmentation proposals. After obtaining the PRMs, they combine the information from class-aware cues, instance-aware cues, and object priors from the segmentation proposals together to predict instance masks. Their approach obtains the state-of-the-art performance on popular benchmarks of WSSS, including PASCAL VOC 2012 and MS COCO.

Zhang et al. [133] propose a Self-Produced Guidance (SPG) approach for semantic segmentation under image tags supervision based on CAMs. The SPG is a stage-wise approach to learn the accurate location of targets by using high confidence regions within CAMs. After obtaining CAMs, these are divided into three groups by categories, namely target, background, and undefined regions. Then, the three groups are merged into the

classification network to obtain the final self-produced supervision map. The whole network is trained in an end-to-end manner, and the authors adopt the cross-entropy loss function for the classification and self-produced guidance learning. The proposed SPG achieves the new state-of-the-art performance on the ILSVRC validation set.

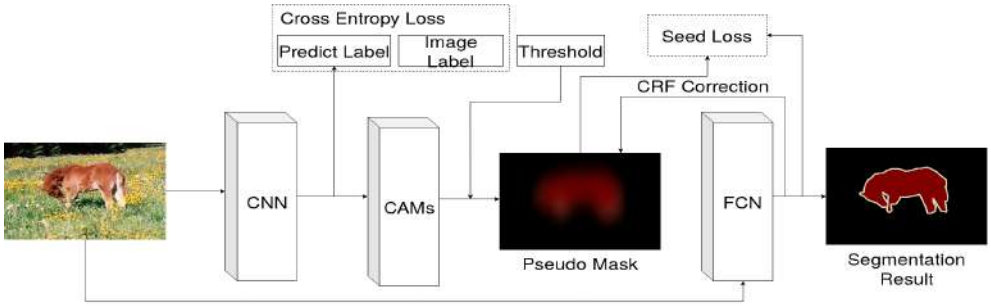
In Fig. 3.11 and Fig. 3.12, several high level diagrams of some WSSS works using image tags supervision are displayed.

3.2.2.2 Weakly-Supervised Semantic Segmentation using Bounding Boxes

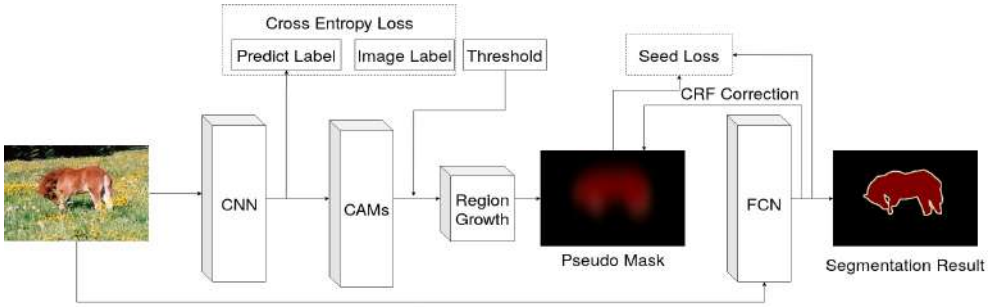
Bounding Boxes are another very popular weak annotation for semantic segmentation. Comparing with image tags, BBoxes can provide a rough localization of the target, which do not need CAMs from a classification network. Although BBoxes entirely surround the objects, they also contain background pixels that contaminate the ground truth of the training set. However, nowadays, many researchers propose methods based on BBoxes as supervision to train a DCNN to obtain pixel-wise predictions.

Khorava et al. [134] propose gradually removing the background pixels from BBox annotations for semantic segmentation, and they explore a recursive training strategy for segmentation network under weakly supervision. More precisely, the segmentation results from a segmentation network are used as the ground truth to train the network in the next iteration. In this procedure, they apply GrabCut [135] and BBox annotations to refine and denoise the segmentation results. Between each iteration, the authors improve the ground truth with three post-processing strategies: (1) any pixel in the segmentation results outside the box annotations is reset to the background label; (2) if we suppose a segmented area which is small compared to its corresponding BBox (i.e., IOU<50%), then box area is used as the ground truth instead of the segmentation result; (3) dense CRFs are applied after each iteration to better respect the target's boundary. Overall, their weak supervision approach can obtain 95% of the quality of the fully supervised approach.

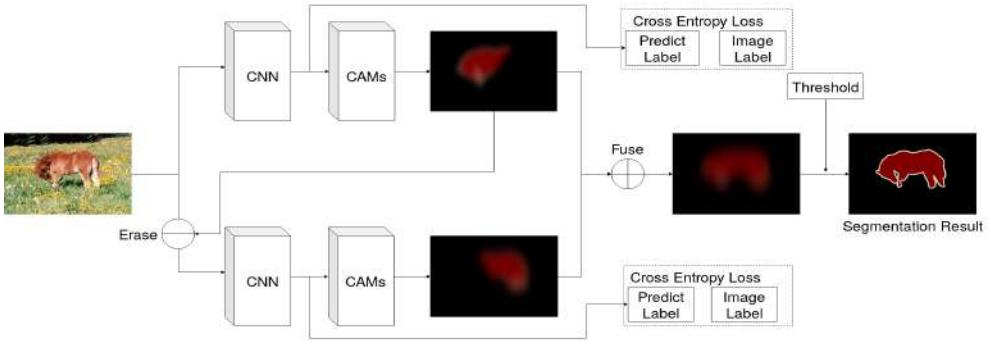
Zhao et al. [136] propose an instance segmentation combined with an object detection algorithm under BBoxes supervision. Their object detection approach consists of an instance segmentation sub-net and a classification sub-net, where the two sub-nets share several convolutional layers of the backbone. Finally, a GraphCut-based mask refinement procedure is developed to obtain the segmentation results. Regarding the segmentation sub-net, it consists of a 1×1 convolutional layer and a position-sensitive pooling layer [29], providing position-sensitive score maps, which are the instance segmentation results. As for the classification network, it is applied to predict the category of the target.



(a) SEC [130]

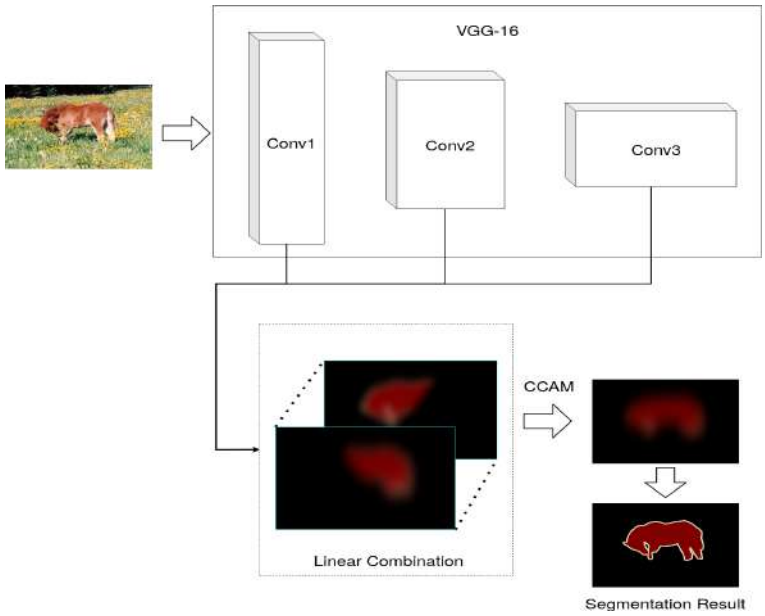


(b) DSRG [131]

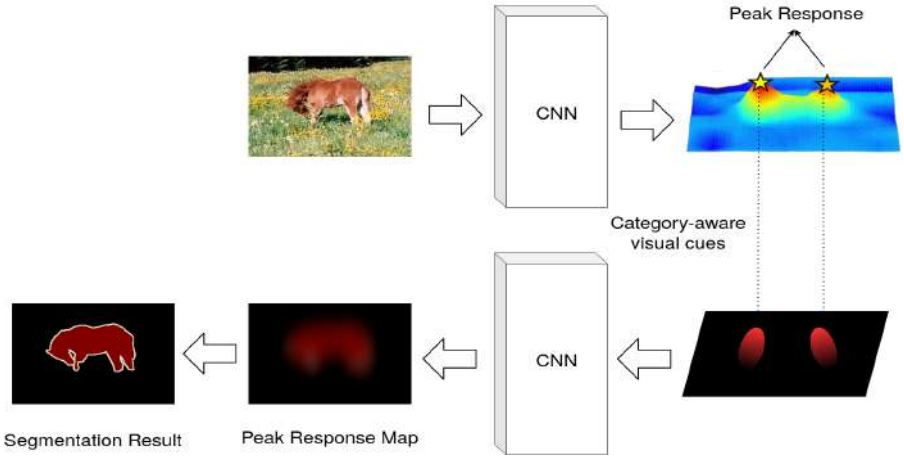


(c) Adversarial Complementary Learning [128]

Figure 3.11: High level diagrams of some weakly-supervised semantic segmentation approaches using image tags. (1/2)



(a) CCAM [129]



(b) PRM [132]

Figure 3.12: High level diagrams of some weakly-supervised semantic segmentation approaches using image tags supervision. (2/2)

In the end, the BBox annotations and instance segmentation results are the input of the GraphCut refinement module, which also provides the supervision for the network training in the next iteration. The authors verify their approach using PASCAL VOC 2007 and 2012 benchmarks. The experimental results prove the effectiveness of their approach.

Hu et al. [137] also undertake object detection by means of WSSS using BBox annotations. This work deals with the Visual Genome dataset, which comprises 3000 visual concepts, what makes hard to obtain pixel-wise annotations. To solve this problem, the authors develop a weight transfer function to predict the segmentation results according to the pre-trained weights from the object detection branch. The weight transfer function is a type of transfer learning, where knowledge gained from the detection task is used to help the segmentation task. Their approach is based on Mask R-CNN [59]. The difference is that the authors apply the weight transfer function between the object detection branch and the instance segmentation branch, which can transfer the category-specific information from the model's BBox detectors to its instance segmentation predictors. Note that the object detection branch includes two types of detection weights: the ROI classification weights w^c_{cls} and the BBox regression weights w^c_{det}. The authors experiment with using either two types or a single type of detection weights. Their approach explores a new research direction for the WSSS problem.

Song et al. [138] introduce a Box-driven Class-wise Masking model (BCM) to remove the irrelevant regions of each class, and develop a filling rate guided adaptive loss (FR-loss) to help the model to ignore the incorrectly labeled pixels. The architecture of their network is based on FCN. For the BCM model, the output features of FCN are evenly sliced into N branches, where N is the number of categories. For each branch, the authors apply a binary attention model to produce a weights map, and they introduce a Mean Square Error (MSE) function to compute the error between the attention map and its corresponding mask (ground truth). The authors consider that the percentage of the same category targets within the corresponding BBoxes should be similar, so they compute an FR-loss function by using the area of the segmentation result and its corresponding BBox area. In this way, their approach adjusts the model training with global statistical information, which can help to reduce the negative effects due to the incorrectly labeled pixels. Their approach obtains good performance on the PASCAL VOC 2012 benchmark.

3.2.2.3 Weakly-Supervised Semantic Segmentation Using Scribble and Point Annotations

As mentioned before, image tags are too ambiguous to detect targets at the pixel level, and the BBox annotations include incorrectly labeled pixels, which introduces noise and instability during training. Therefore, some researchers consider applying other forms of weak annotations, such as points and scribbles, which only mark a few pixels of the targets. The idea behind using annotations is to try to avoid the ambiguous and boundary areas of targets. Therefore, the marked pixels in the point or scribble annotations are correctly labeled, and they can be used as segmentation cues to guide the segmentation network.

Bearman et al. [139] develop a semantic segmentation system under points supervision. Compared to image tags, the point is a consistent and predictable way to indicate the position of the target. In their work, an FCN is used as backbone, and they train the FCN model using a partial cross-entropy loss function. The authors find that weakly-supervised learning usually is prone to the local minimum and that the predicted results usually focus only on a small area of the target or even predict the pixels belonging to the background. To solve this problem, they develop a new term for the loss function based on an objectness prior. The objectness prior is a probability indicating whether the pixel belongs to any category. The objectness prior loss function cooperates with the partial cross-entropy loss function. As a result, their approach yields an improvement of 12.9% mIOU over the image-level annotation on the PASCAL VOC 2012 dataset.

Unlike [139], Papadopoulos et al. [140] consider using extreme points as weak annotations for object detection and semantic segmentation. The extreme points indicate the left-, right-, top-, and bottom-most points of the target area. In their work, they change the traditional way of annotating BBoxes, consisting in using the combination of the top-left-corner, and the bottom-right-corner to four extreme points. To do that, the authors develop an annotator to obtain the four extreme points. In detail, the annotator reads a set of instructions and then goes through an interactive training stage that comprises a qualification test at the end, in order to decide the quality of the extreme clicks, i.e. mouse clicks used to indicate the box corners. For the segmentation annotation, the authors develop a binary pair-wise energy function to distinguish the foreground and background. For object detection, an edge detector is applied to detect the outline of the target, combined with the extreme clicks to obtain the ROI that contains the target. Finally, the ROI and the extreme clicks are used to feed to a GraphCut for foreground segmentation. They achieve good results on the PASCAL VOC 2007 and 2012 datasets.

Scribbles are another widely used annotation for the WSSS problem. Xu et al. [141] propose an unified approach that incorporates various forms of weak supervision, including image tags, BBoxes, and scribbles. Their approach applies a one-vs-all linear SVM classifier using CNN features to obtain a pixel-wise result, considering image level tags (ILT) supervision. Then, they feed the output of the classifier into an inference function to predict categories for the test-image on the SIFT-flow dataset. Their approach outperforms the state-of-the-art approach by 12% on per-class accuracy.

Lin et al. [125] propose an image segmentation approach using scribble annotations based on a graphical model that jointly propagates information from the scribbles to unmarked pixels. Since scribbles only mark a few pixels, which can not provide enough information to train the segmentation network, the authors build a graphic model based on superpixels to solve this problem. The loss function used in their work comprising two parts. The first part is based on scribbles. Formally, if a superpixel overlaps with a scribble, then it has zero cost, if a superpixel does not overlap with any scribble, it can be assigned to any absent label in this image with equal probability. The second part of the unary term respects the output of a fully convolutional network. The pairwise terms are used to model the similarity between two superpixels and are used to propagate information to the unmarked pixels. Their approach shows good segmentation results on the PASCAL-Context dataset.

Tang et al. [142] propose a compound loss function combining the partial cross-entropy loss and Normalized Cuts (NCuts) [32] regularization, which introduces consistency of low-level features such as texture, color, and position characteristics. The partial cross-entropy loss function considers seeds as pixels whose labels are known, and the NCuts regularization term softly evaluates the consistency of all pixels. In the NCuts regularization term, the authors use a standard Gaussian kernel W_i over the RGBXY feature space to build the Laplacian matrix, which attends the feed-forward and backward path computation. They obtain high-quality segmentation results, which are close to the performance of a fully supervised approach. In their later work [143], the authors directly integrate NCuts and Dense CRFs regularization terms into the loss function, avoiding extra post-processing inference steps. In this work, the authors propose several regularized losses for the WSSS problem based on the Potts/CRF model, normalized cuts, and kernel cuts. By directly integrating regularization terms in the loss function, their approach does not need to generate the segmentation proposals before training. Comprehensive experiments with their regularized weakly-supervised loss function are conducted, and their approach can obtain the state-of-the-art performance, achieving a performance similar to the fully supervised approach.

Table 3.3: Summary of Weakly-Supervised Semantic Segmentation Algorithms.

Ref.	Weak Supervision	Main Techniques
[126]	Image tags	CAMs
[127]	Image tags	CAMs, adversarial training, erasing
[130]	Image tags	CAMs, SEC, modified loss
[131]	Image tags	CAMs, region growing, SEC
[128]	Image tags	CAMs, adversarial training, erasing
[37]	Image tags	CAMs, superpixel pooling
[132]	Image tags	CAMs, peak response map
[133]	Image tags	CAMs, self-produced guidance
[144]	Image tags	CAMs, peak response map, instance activation map
[145]	Image tags	CAMs, attention dropout
[129]	Image tags	Combinational class activation maps
[146]	BBoxes	EM approach, dense CRFs
[134]	BBoxes	Multi-scale combinatorial grouping, GraphCut refinement
[136]	BBoxes	Object detection, GraphCut refinement
[137]	BBoxes	Mask R-CNN, transform learning
[138]	BBoxes	Dense CRFs, filling rate loss
[139]	Image tags, center points	Objectness prior, modified loss
[147]	Extreme points	Deep extreme cut
[141]	Image tags, scribbles	Superpixels segmentation
[125]	Scribbles	Superpixels segmentation, GraphCut refinement
[142]	Scribbles	Modified loss, NCuts regularization
[143]	Scribbles	Modified loss, NCuts regularization, CRF regularization

Table 3.3 lists some typical algorithms of weakly-supervised semantic segmentation.

3.3 Classical and Modern Solutions for Image-based Inspection and Quality Control

In this section, we overview the existing techniques for visual inspection and quality control systems. We firstly review some approaches based on image processing techniques in Section 3.3.1, and then, in Section 3.3.2, we review some approaches based on machine learning and deep learning techniques.

3.3.1 Image Processing-based Approaches

Colour Histograms

Histogram comparison statistics are widely used in defect detection and quality control systems. Commonly used statistical measures include range, mean standard derivation, variance, and median. Although these approaches may seem a bit simple, some works have proved that they turn out to be low-cost and efficient solutions in several applications, such as [148–150].

In [148], the authors evaluate different pre-processing image enhancement filters for corrosion detection based on the red channel histogram. Their approach includes two steps. Firstly, the authors apply corrosion extraction based on the red channel histogram to display the area of corrosion. Secondly, they select a set of filters to deal with the input images, including mean filter, median filter, Gaussian filter, wavelet de-noising, Weiner filter, Bayer filter, and anisotropic diffusion. The experimental results show that the Bayer filter provides the best results for the considered task.

Roberts [150] proposes a color histogram-based corrosion detector to deal with images taken from a Micro-Aerial Vehicle (MAV). Their approach transforms images into the HSV color space and makes use of a threshold to select the corroded areas. Aijazi et al. [149] develop a novel automatic detection approach for detecting defects on the vessel surface. They apply several scanners to scan different positions around the ship, while RGB images associated with each scanner are obtained. Then, the RGB images are converted into the HSV space to separate the illumination invariant color component from the intensity. At last, corrosion of different sizes and shapes is automatically detected by a histogram threshold-based approach. The authors propose two different threshold-based histogram approaches depending on the level of corrosion/defects on the large ship hull. The first approach makes use of a global threshold-based histogram, while the second approach is based on adaptive thresholds.

Co-occurrence Matrices

Spatial gray-level co-occurrence matrices (GLCM) are one of the most well-known and widely used approaches for defect detection and quality control systems. These second-order statistics are accumulated into a set of 2D matrices, each of which measures the spatial dependency of two gray-levels for a given displacement vector. Using GLCM, texture features, such as energy, entropy, contrast, homogeneity, and correlation can be easily calculated. Some example approaches using GLCMs for defect detection are

[151–154].

Bento et al. [151] design a new approach to inspect corrosion in oil and gas tanks and pipelines. This method considers using the GLCM attributes to describe texture changes in metallic surface images and applying Self Organizing Mapping (SOM) to classify the corroded area. The accuracy of their approach can attain 93% on the validation data set from the real world.

Bonnin-Pascual and Ortiz [152] propose an approach for corrosion detection on the surface of vessels. The method is built around a two-stage scheme. In the first stage, roughness related to the energy of the symmetric GLCM is computed using intensity values, and then a threshold of energy is used to select suspicious patches. In the second stage, a classifier based on a codeword dictionary is used to finally identify corroded patches. In this approach, a codeword consists of stacked histograms for the red, green and blue color channels. The same authors propose a second method also using the energy computed for the GLCM in [155]. This approach differs from the previous one on the color stage, which this time it is based on a classifier consisting in a two-dimensional histogram of corrosion colors in the hue-saturation space. Both methods prove to be fast and effective for detecting corrosion in vessel metallic surfaces.

Iivarinen [153] presents two unsupervised segmentation approaches using histogram-based texture analysis for surface defect detection, which are the GLCM-based method and the local binary pattern-based method. Both approaches detect defects by computing texture features from a small image window. The unsupervised segmentation approach is trained with fault-free surface samples. So, if the distribution of the test sample is different from the distribution of training samples, their approach classifies it as a defect.

Odemir et al. [154] compare the performance of different approaches for automated visual inspection of textile images. In their work, the authors implement and test some model-based and feature-based approaches on real fabric images, such as Markov Random Fields, the Karhunen-Loeve Transfer, 2D Lattice Filters, Laws Filters, co-occurrence approaches, and FFT-based approaches. For their task, the Markov Random Fields model gives the best results.

Ünsalan and Erçil [156] consider to make use of the texture analysis approach to inspect rust on steel surfaces. In their experiments, three texture analysis techniques (GLCM, Markov Random Fields (MRF), and Histogram of Image (HIS)) are tested based on three different color spaces, including the YIQ, RGB, and HSI color spaces. Afterwards, the Nearest Neighbor classifier is used for the classification of the steel surface. The experimental results indicate that their approach can be used in the automated

inspection and classification process of metallic surfaces.

Choi and Kim [157] also consider that corrosion detection can be seen as an analysis procedure using features extracted from digital images, such as color, texture, and shape, instead of the traditional detection approaches based on electrochemical procedures. In order to distinguish corrosion in the surface, the authors make use of interpretations from a HIS model. As for the texture features, the GLCM is used. In their experiments, five types of corrosion are examined, obtaining good detection results.

In [158], the authors present an approach using color and texture descriptors to perform corroded and non-corroded surface area discrimination. This approach makes use of GLCMs to obtain texture descriptors and HSI color histograms for the color features. In order to classify corroded areas on the carbon steel surfaces, a Fisher classifier is applied and evaluated.

Morphological Operators

In the computer vision area, the basic idea of a morphological image operation is to measure and extract the corresponding shapes in the image with a particular form of structural element. The application of morphological image processing can simplify image data, maintain their essential shape characteristics, and remove irrelevant structures. The main usages of morphology operations in image processing include noise removal, boundary extraction, area filling, connected component extraction, convex shell computation, refinement, coarsening, etc.

Tanaka and Uematsu [159] develop an approach using morphological operators to detect cracks in the road surface. Their approach considers that cracks can be seen as a succession of dark points by using a linear filter to process the image. Their approach detects cracks through several procedures, such as black pixel extraction, saddle points detection, linear feature extraction, and connectivity analysis. Their approach obtains good performance in their task. The same task is approached by Zheng et al. [160], who employ a machine-learning algorithm to automatically learn morphological processing parameters, such as the structuring elements and the segmentation threshold. Their approach is implemented and tested on a number of road surfaces, and the experimental results show an accurate detection for cracks on the road surface.

Yoshioka and Omatu [161] propose an approach to detect cracks on the surface of steel products. An automatic detection system, including a digital camera and an online processor, is used to obtain photos from the factory and detect cracks in real-time. In their approach, they propose a new method using rotational morphology in order to

distinguish between structural lines on the steel surface and cracks. The rotational morphology is an expansion of mathematical morphology with rotated structuring elements. In this way, their approach fits the direction defined in the original image, while other directions correspond to cracks.

In order to overcome the difficulty of previous methods for detecting cracks of different thicknesses, Jahanshahi et al. [162] propose a contact-less remote-sensing crack detection and quantification methodology based on 3D scene reconstruction, image processing, and pattern recognition techniques. At the beginning, the depth information of the scene is obtained by scaling the reconstructed 3D model. Then, a morphological crack segmentation operator is introduced to extract crack-like patterns. Moreover, a contact-less crack thickness quantification procedure is introduced, which is based on the depth perception of the scene. In the end, the pixels in each centerline are aligned with the corresponding orientation to compute the hypotenuse length, which is counted in the horizontal and vertical directions. Their approach is executed on autonomous or semi-autonomous robotic systems.

Similarly, Zhang et al. [163] develop a crack detector for subway tunnel safety monitoring. With a platform fitted with high-speed CMOS cameras, photos of the tunnel surface are obtained and stored. Then, the authors apply morphological image processing techniques and threshold operations to segment the potential crack areas. In the feature extraction process, a histogram-based descriptor is used to describe the spatial shape difference between cracks and other irrelevant objects. The approach can remove 90% of misidentified objects.

Apart from previous approaches, Zhao et al. [164] develop an anisotropic clustering approach to differentiate small cracks from the noisy concrete background. A globally convex segmentation model is utilized to provide appropriate candidate points and parameters for the cluster processing. In this way, the clusters corresponding to cracks can be easily sifted out according to their elongated shape. The detected cracks are similar to manually traced ground truth cracks.

Filter-based Approaches

Filter-based approaches [165–173] share a common characteristic: they apply a set of filters on the spatial domain, frequency domain, and joint spatial-frequency domain to an input image, and perform a statistical calculation.

The Laplacian filter is successfully applied in [166]. In the beginning, the authors firstly employ a weighted averaging filter to smoothen input images. The Laplacian filter

is applied in the x-axis followed by a two-level threshold, and in the y-axis followed by a single threshold. The threshold level is updated for images with no defects and it is not updated for images including defects. The resulting method proves to be time efficient and robust, being able to successfully process images from various sorts of scenes and amounts of noise. Liu et al. [168] employ similar filter-based techniques to detect defects on steel surfaces by using a discriminant function. In order to decrease the influence of the steel strip texture, essential statistical approaches, such as difference, gradient, mean, and variance, are used. Their approach improves the feasibility and accuracy of previous approaches.

In [169], the authors also apply a filter-based approach to detect six types of defects in steel sheets. In their work, the authors develop a new pattern recognition system based on using both the Euclidean distance and Fractal Geometric Properties (FGP). Firstly, the Wiener filter is applied to reduce the noise in the input image. Then, a coarse object location is obtained by using the Sobel edge detector. Finally, they present a new segmentation algorithm based on object recognition. In their segmentation algorithm, the self-learning procedure based on fuzzy logic and membership function theory is used to classify the category of the target using the Euclidean distance and FGP as features. Their approach obtains accurate diagnostic results.

In [167], an automatic surface inspection system based on a background difference approach is proposed. This approach is based on hypothesizing that the features of background or defect-free areas have a homogeneous distribution. On the other hand, a significant change between the input and its background distribution is considered as a defective area. Their system captures surface images of steel strips through 20 CCD cameras. By means of the background difference approach, defects are distinguished from the background of images in the gray-scale input image. Their approach can detect the main defects of hot rolled strips more effectively than previous approaches. In a different approach, Cong et al. [170] divides the input images in small (20×20) sub-images. By characterizing the gray-scale pixel distribution of sub-images, five kinds of defects on cold-rolled strips are detected. More precisely, the authors first use a sequential extraction technique to extract the background image as a standard image. Secondly, the authors inspect the small sub-images for remarkable defect features. Similarly, Djukic and Spuzic [171] propose a statistical discriminator to detect small area defects on a hot rolled steel surface. Firstly, the discriminator is employed to learn the distribution of the areas of images without defects. Then, this distribution is used to select patches with significant differences as candidates for defects.

Li et al. [173] propose a Local Annular Contrast (LAC) algorithm for ROI identifi-

cation of three types of rod defects. The LAC is based on the fact that pixels belonging to defects have a higher contrast concerning the surrounding background. Firstly, the average filter is employed to smooth the noise from the steel bar surface images. Then, the LAC-based algorithm is applied to inspect defects on the grayscale images. The LAC radius in their approach is required to be greater than the size of defects, which can ensure their approach did not miss the area of the defect. Their approach only needs 13 ms to inspect one steel bar surface image, and its detection accuracy exceeds 95%.

Bonnin-Pascual and Ortiz [155] propose a machine learning technique to classify corroded surfaces. Their approach utilizes the Adaptive Boosting paradigm (AdaBoost) by using Classification and Regression Trees (CART) as a weak classifier. They obtain statistical features by convolving Law's texture energy filters on patches centered at corroded and non-corroded areas. Their approach obtains an efficient performance for vessel corrosion detection.

Wavelet-based Approaches

The wavelet transform is another widely used technique in computer vision. Wavelets are a set of self-similar mathematical functions that are used to approximate more complex functions via the super positioning principle. Using a wavelet transform in spatial and frequency domains, the transient elements in one image can be represented by a smaller amount of information, what allows refining image features. Some works [174–181] propose using wavelet-based approaches for inspection in industrial manufacturing environments, for road detection, etc.

Subirats et al. [174] present an automated approach for crack detection on pavement surface images. Their approach consists of two steps. Firstly, the authors use a separable 2D continuous wavelet transform for multi-dimensional inputs. Secondly, they search the maximal value of the wavelet coefficient. Finally, a post-processor gives a binary image indicating the presence of cracks on the pavement surface images. Similarly, Jeon et al. [181] use a wavelet transform for detecting corner cracks on the surface of steel billets. The wavelet model allows reducing the effect of different scale defects. Meanwhile, texture and morphological features are used to identify the corner cracks among the defective candidates. The experimental evaluation proves that their approach is effective for this specific application.

Wu et al. [175] develop an algorithm based on the undecimated wavelet transform and mathematical morphology, focusing on the problem of false alarms by scales and watermarks on hot rolled steel plates. The candidate position of defects is detected as

follows: first, the authors obtain the position of defects by the modular maximum of the inter-scale correlation of the wavelet coefficient; then, defective areas are classified by using prior knowledge of the surface characteristics. The recognition rate of their approach is up to 90.23%. Similarly, Jeon et al. [177] employ a discrete wavelet transform to detect defects on the billet's surface under various lighting conditions. Their approach focuses on solving the incorrect detection caused by oxidized substances covering the steel billet surface, which is similar to the corner cracks. Also, in order to differentiate corner cracks from pseudo defects, the authors investigate the morphological features of corner cracks for classification.

In [179], Ghorai et al. apply five different kinds of wavelet: Haar, Daubechies 2 (DB2), Daubechies 4 (DB4), biorthogonal splines, and multi-wavelets. More precisely, the five kinds of wavelets are used to extract features by decomposing small windows (32×32) of surface images into different resolution levels. Using a SVM classifier, their approach is able to identify different kinds of defects. The authors apply their approach to detect 24 different defect categories and compare its performance with texture-based approaches. The experimental results show that wavelet-based approaches are better suited for their defect detection application on steel surfaces.

Table 3.4 details the main technologies of the different approaches reviewed in this section.

3.3.2 Machine Learning-based Approaches

Support Vectors Machines

A Support Vectors Machine (SVM) [187] is a supervised Machine Learning algorithm that exhibits a constant classification accuracy. It is a very effective technique for general supervised pattern recognition, and a binary classifier employed to predict the category (-1 or 1) of the input. SVMs have also been extended to solve multi-category classification problems using one-versus-all and one-versus-one strategies [179, 182, 185, 186, 188–190].

In [188], the authors focus on detecting primary surface defects on cold rolling strips while the product line is running normally. Their system consists of a high-speed linear CCD image controller and an FPGA to improve the low-level image processing speed. The whole classification algorithm includes two steps. In the first step, the defect information is extracted using an image segmentation algorithm based on a fuzzy logic technique and the information entropy theory. In the second step, they extract image features of the defects by the homomorphic filtering algorithm and make use of a fuzzy

Table 3.4: List of defect detection and quality control methods based on image processing techniques.

Feature Approaches	References	Detection Targets
Histogram Properties	[148–150, 152, 158, 182]	Corrosion
	[156]	Metal surface defects
Co-Occurrence matrices	[151, 152, 157, 158]	Corrosion
	[156, 183]	Metal surface defects
	[153]	Surface defects
	[154]	Textile product defects
Morphological Operators	[159–163]	Cracks
	[157]	Corrosion
	[177]	Corner crack
	[175]	Defects of hot rolled steel plates
	[184]	Surface defects of cold rolling mill steel
Filter Approaches	[178, 185, 186]	Cracks
	[165, 167, 171]	Cracks, pits, tear on hot steel track
	[169, 170]	Defects of cold steel strip
	[168, 172]	Defects of steel strips
	[166]	Cracks, spot, dark line of steel bar
	[173]	Surface defects of steel bar
Wavelet Approaches	[177, 181]	Corner cracks
	[174]	Defects of pavement
	[175]	Defects of hot rolled steel plates
	[176]	Defects of hot wire rod
	[179]	Defects of hot strip
	[178]	Cracks of steel welded seam
	[180]	Corrosion

SVM (FSVM) classifier to obtain the detection results.

Tsutsumi et al. [182] develop a system to evaluate the degree of corrosion of galvanized steel in power transmission towers. Their algorithm is designed to use a SVM with the Radial Basis Function (RBF) kernel. More precisely, the authors obtain images of steel structures using a digital camera, clip out the rust steel area and resize it to 320×320 pixels. Then, the authors construct a 192-dimensional hue-saturation-value (HSV) space histogram as the input features of the classifier. In the end, SVMs classifier with an RBF kernel is in charge of the defect classification. Their approach achieves 85.6% accuracy.

Yamana et al. [189] propose an approach for crossarm reuse judgment based on SVMs. The authors compare the classification performance with three conventional methods, which are k-Nearest Neighbors (kNN), Multi-Layer Perceptrons (MLP), and Radial Basis Functions (RBF) network. In order to perform a three-category classification task, they employ three two-class classifications, “reuse” vs. “retire”, “retire” vs. “reuse after

plating", and "reuse after plating" vs. "reuse". In this way, if two of the three judgments are the same, the results are considered as correct predictions. On the contrary, if the three predictions are different from each other, results are considered as incorrect. To avoid a dimensional explosion, they compress the original input size from 640×480 to 4×3 and 20×15. As a result, their approach achieves a degree of accuracy as high as 99.0% for the two-class classification case and 97.2% for the three-class classification case.

In order to deal with non-linear problems, SVMs can employ kernel functions to transform samples to a different feature space, where they are linearly separable. Among the existing kernels, the aforementioned RBF kernel is a popular choice. Choi et al. [178] extract 46 geometric features and eight gray-level features from segmented ROIs. The authors compare the detection performance of using different kernels on a SVM, which are the linear kernel, the polynomial kernel, the RBF kernel and the sigmoid kernel. As a result, a SVM with an RBF kernel obtains the best performance for detecting seam cracks in steel plates. Similarly, [185] also employs SVM with RBF kernels for defect detection. Their approach is robust and capable of providing good discrimination when the input data contains some noise, and it can detect steel surface defects in real-time, where its speed is less than 6 ms per one-megabyte image. Their system is tested on image data from hot rolling manufacturing productions.

In [186], the authors propose a knowledge-based SVM (PK-MSVM), which combines a feature extraction technique with knowledge processing. Inspired by the experiences of steel plant engineers, the authors select three important characteristics of the defects: the length to width ratio, and the longitudinal and transverse locations. The authors implement a set of experiments with and without a prior process knowledge as comparative experiments. Also, some multi-categorical comparative experiments are conducted using SVM variations, such as one-against-one, one-against-all, and the multi-categorical Hastie algorithm for SVMs.. The experimental results show that the PK-MSVM obtains the best performance.

Unsupervised and Semi-supervised Classifiers

In real industrial environments, ground truth of defects is difficult to obtain from experimented engineers or experts. Therefore, some researchers consider applying unsupervised or semi-supervised classifiers. These classifiers learn common categorical features from the input patterns, and are able to identify those features by a self-guided strategy without or with less help of ground truth. In order to detect defects, Self-Organizing

Maps (SOM) [183, 191, 192] and Learning Vector Quantization (LVQ) [193, 194] are commonly applied.

Self-Organizing Maps is an unsupervised learning neural network for feature recognition. It simulates that neurons in different regions of human brain have different characteristics and functions. So, the different groups of neurons in SOM have different response characteristics, and this learning process is done automatically. SOM uses a neighborhood function to generate low-dimensional spaces for the training set, transforming complex nonlinear statistical relationships into simple linear relationships. Caleb and Steuer [191] conduct experiments using a SOM network for two tasks: to explore the subgroups within each category and re-analyze the defects misclassified by others supervised learning approaches. In detail, the authors apply the SOM algorithm for the misclassified defects areas to obtain a clear separation of defects or non-defects regions. On the other hand, their SOM approach can obtain subgroups of defects regions, which is used to separate the different types of defects. As a result, their approach improves the accuracy of six types of defects and has the ability to detect an ambiguous manifestation of defects.

Martins et al. [192] present an automatic inspection system to monitor the rolled steel quality. Their approach is based on (1) image processing techniques, such as the Hough Transform to classify three types of defects with well-defined geometric shapes, and (2) two well-known feature extraction approaches: Principal Component Analysis (PCA) and SOM. Their system is successfully validated on real-world datasets and achieves an overall accuracy of 87%.

Iivarinen et al. [183] propose a segmentation scheme to detect surface defects. Particularly, the authors develop an unsupervised neural network based on a SOM to estimate the distribution of faulty-free samples by using RGB values as features. So, if the distribution of a sample differs from the estimated distribution, the sample contains defects. Their approach can be used in general scenes, and it can also be applied to detect faults in different types of surfaces.

Learning Vector Quantification (LVQ) belongs to prototype-based clustering algorithms, which try to find a set of prototype vectors, where each prototype vector represents a cluster. Unlike the K-means clustering algorithm, who needs to set the unknown K before training, LVQ uses the label of samples in the training set as supervision information and relies on this information to get the prototype vectors. The advantage of LVQ is that the prototype vector is easy to interpret for experts in the respective application domain, and it can be applied to multi-categorical classification problems.

By way of example, an inspection and quality control system using the measurement

of the angular distribution over a 25-degree cone angle of the scattering is conducted, calibrated, and evaluated for coated sheet and steel samples in [194]. In their system, features are chosen from a specially designed photo-detector array, where the value of each feature is the sum of the pixel intensities in a section of the image plane. Then, the scaled features are fed into a classifier based on Kohonen's Learning Vector Quantification (LVQ2) for classification. Their system is used to evaluate on the CrO2 coated steel samples, classifying for fault or non-fault samples.

In [193], surface defects of hot rolled strips are identified by applying the LVQ algorithm. Features, extracted from spectrum images by using a Fast Fourier Transform (FFT) technique, include 224 channels as the input vector of LVQ. Then, all features are regrouped to obtain a set of optimized character parameters. Afterwards, the optimized feature includes 54 channels: 50 channels in the crisscross region, one Sum of Valid Pixels (SVP) feature, and three Repletion Ratio of Center Region (RRCR) features. Then, the optimized features are fed in an LVQ model to classify 11 types of defects on hot rolled strips. As a result, this approach presents a recognition rate between 84.56% and 92.92% for the hot strips dataset.

Classical (shallow) Artifical Neural Networks (ANN)

In the recent years, ANN-based approaches have been extensively used for quality control tasks in many industrial scenarios, such as vessel inspection [195–197], factory inspection and quality control [10, 12, 198–200], civil infrastructure inspection [201–204], railway inspection [13, 205] and steel surface inspection [184, 206, 207]. An important advantage of ANNs over SVMs is that they are easy to expand to multi-categorical scenes, unlike SVM which need to employ one-against-one and one-against-all strategies.

Ortiz et al. [195] propose a system to detect corrosion and coating breakdown (CBC) from vessel images. In this work, an artificial neural network approach is developed, which is used on a Micro-Aerial Vehicle platform. In their work, a network with three fully connected layers is deployed to detect corrosion. The authors discuss several configurations, using a different number of hidden neurons, to determine the optimal setup for the classifier. The authors combine the information from RGB and HSV color spaces, and the texture from center-surround differences as features to train the network. The authors further propose a different detector in [196] based on two well-known object detection techniques, Faster R-CNN [27] and SSD [30], for detecting corrosion in vessel structures. The two approaches are fine-tuned with the VGG-16 [20] backbone. Experimental results show that Faster R-CNN obtains better performance in general, providing

higher precision values. In order to obtain more accurate detection, in [197], the FCN [52] is employed to detect corrosion at the pixel level for vessel hull structures. Since the corroded areas only take a small area in some images, what introduces an imbalance situation for training, the authors apply different loss functions to improve the segmentation performance. More precisely, they employ the Focal Loss [45], the Dice Loss [208] and the softmax Cross-Entropy Loss to train the segmentation network. In the end, the Dice Loss brings the best performance for their task.

In [198], Park et al. develop a visual inspection method for detecting dirties, scratches, burrs, and wears on the surface of different materials. They extract image patches of defect areas to be used as training samples, and construct several networks of different layers in order to obtain the best performance. The authors find that a single CNN can provide good performance to detect various types of defects on textured and non-textured surfaces.

Petricca et al. [199] propose a comparison between image processing techniques and a Deep Learning approach for automatic metal corrosion (rust) detection. In this work, the authors focus on image classification, so that an image showing a corroded area is considered as rust, while an image without any corroded area is considered as non-rust. The image processing approach comprises a reddish pixel selection approach using the HSV color space. On the other hand, the authors fine-tune the AlexNet model by using a dataset from a real scenario. The experimental results show that the CNN model obtains better accuracy (78% versus 69%).

Similarly, Cha et al. [12] propose an inspection system based on object detection technology. They apply the Faster R-CNN to detect structural damage on the surface of civil infrastructures. The results of Faster R-CNN achieve 90.6%, 83.4%, 82.1%, 98.1%, and 84.7% precision for five types of damage.

Ren et al. [200] propose a solution based on the Decaf [209] network for Automated Surface Inspection (ASI) in the factory. In their work, they deal with two tasks: a classification task and a segmentation task. For the classification task, the authors apply transfer learning technology to fine-tune the Decaf network. To be precise, they train the last layer of the Decaf network, while the weights of other layers remain unchanged. For the segmentation task, they split the images into several patches, and obtain the activation maps from the classification network for every patch. Then, they stitch all of the activation maps together, as the heatmap of the whole image. In the end, a threshold approach is applied to obtain segmentation results. For the classification task, their approach obtains an accuracy between 0.66% and 25.5% higher than previous approaches, while for the segmentation task, their approach achieves 0 error escape rate.

In order to detect defective products at the factory, Wang et al. [10] propose a machine vision-based inspection system. To be precise, they present a new deep learning-based vision inspection approach to identify and classify defective products. Firstly, they apply a Gaussian filter to the input images to remove random noise. Secondly, a ROI containing the product inside is generated using the Hough transform. In the end, a CNN comprising several inverted residual blocks is used to classify the defective products within the input ROI. Their approach can quickly detect defective products with high accuracy.

Other researchers use convolutional neural networks to detect cracks in public facilities, such as building and bridges. Oullette et al. [201] employ a standard Genetic Algorithm (GA) that trains a 2-layer CNN to detect 15 different predefined subtypes of cracks. The GA-based approach obtains 92.3% accuracy for crack detection in the test set. Zhang et al. [202] propose a solution to detect cracks in pavement based on a convolutional network. The architecture of the convolutional network comprises four convolutional layers, four max-pooling layers, and two fully-connected layers. The convolutional network is trained using the Stochastic Gradient Descent (SGD) method with batch samples of 48×48 pixels. They compare the results of the convolutional network, SVM, and Boosting algorithms, and conclude that the discriminative convolutional network-based model outperforms the hand-crafted features in describing complex patch contexts.

Cha et al. [204] propose a new neural network to detect cracks in infrastructures instead of using a pre-trained neural network. They collect 40K images with a resolution of 256×256 pixels to train their network and 55 images with a resolution of 5888×3584 pixels for testing by using a sliding window technique. They get 98% accuracy as a result, which is outstanding performance compared with Canny and Sobel edge detection methods.

Yang et al. [203] propose an Unmanned Aerial Vehicle (UAV) system to detect concrete spalling and cracks in civil infrastructures. For the inspection task, the authors train a CNN on a Concrete Structure Spalling and Crack (CSSC) dataset. They randomly crop sub-images from the spalling and crack regions to create the training set, and then train a VGG-16 based network to classify the defects. As a result, they achieve over 70% accuracy on the test set and more than 93% accuracy on the CSSC dataset. Cha et al. [204] also present a CNN-based approach for crack inspection on concrete surfaces. Unlike traditional approaches, their approach does not need to compute hand-crafted features. Instead, they use a CNN to learn features automatically. The authors construct a specific CNN for the inspection task, and their approach achieves 98% accuracy.

The railway system is also a typical industrial scenario that needs to be inspected effectively. Santur et al. [13] develop a DCNN-based inspection system to detect oil and dust residues on the railway. In this study, they develop a platform to inspect the status of the railway's surface using images from a 3D laser camera. By using a DCNN, their approach can detect the railway surface quickly and contact-free, and it is effective to detect some defects such as fractures, scratching, and fissures with high accuracy. Similarly, Eisenbach et al. [205] apply a DCNN to detect cracks on the train tracks. Their aim is to automatically detect cracks on millions of high-resolution railway surface images. Since a DCNN needs massive memory to deal with high-resolution images, the authors split the whole image into many patches and use a classification network to detect cracks in every patch. Compared to the approaches based on image processing, their approach achieves state-of-the-art performance on the GAPs dataset [13].

Concrete tunnels have become increasingly important in the current life, and the number of tunnels has significantly increased in the recent years. Nowadays, some researchers focus on using DCNN-based approaches instead of engineers to carry out some work, such as maintenance, inspection, and evaluation, so as to ensure the normal operation of concrete tunnels. Makantasis et al. [210] apply several low-level features extracted from the input image, and construct an MLP as a detector to assess the condition of the tunnel. More precisely, firstly, the authors combine the edges, frequency, entropy, texture, and Histogram of Oriented Gradients (HOG) features as inputs for the MLP. Secondly, a Gaussian image pyramid is exploited to provide scale invariance features. In the end, a three layers CNN is used to detect deformation at the pixel level. Such an approach achieves fast and accurate predictions. Similarly, Protopapadakis et al. [211] develop a new CNN-based approach for tunnel inspection. To overcome the difficulties deriving from this environment, such as low operational times, limited hardware resources, and variable lighting conditions, they apply a domain-specific heuristic post-processing technology. Instead of using complicated hand-crafted features, the authors use a CNN to extract high-level features from a classification network, and then a threshold-based heuristic approach is used to classify crack areas.

The inspection of the factory facilities is another crucial industrial scenario. Masci et al. [206] employ a CNN to classify seven kinds of defects on steel plates. Compared to a SVM classifier, which is trained with commonly used feature descriptors, their CNN-based approach obtains two times better performance. Yazdchi et al. [184] propose two methods of classification, which are an ANN and a Fuzzy Inference System (FIS), to detect five types of defects in cold-rolled strips. For the FIS, the authors apply the Sugeno's method to describe data, which uses a few rules and is therefore efficient.

Nevertheless, the accuracy using FIS features is 82%, much lower than the 97% accuracy of the CNN using seven statistical features. In [207], a modified loss function is proposed to classify the surface defects on steel strips. Their modified loss function aims at solving the slow convergence of the network caused by the saturation areas.

Regarding a different industrial application, Shihavuddin et al. [212] propose a CNN-based solution for the timely detection of surface damages on wind turbine blades. In their approach, they use a drone to obtain high-resolution images of wind turbines and develop an object detection-based automated damage suggestion system. Their vision-based detection approach is based on Faster R-CNN. They evaluate different architectures and regularization to obtain the best detection performance. The experimental results demonstrate that their approach can achieve human-level precision in terms of suggested damage location and classification on wind turbine blades.

Finally, Table 3.5 summarizes the machine learning-based methods for defect detection and quality control, which are classified by the machine learning algorithms used in their approach.

Table 3.5: List of defect detection and quality control methods based on machine learning techniques.

Learning Approaches	**References**	**Detection Targets**
SVM	[178, 185, 186]	Cracks of steel welded seam
	[162, 185]	Cracks
	[188]	Defects of cold rolling strip
	[189]	Reused crossarm
	[179]	Defects of hot steel strip
	[190]	Defects of cold steel strip
Clustering	[151, 157, 213]	Corrosion
	[164]	Cracks
	[214]	Cold steel strip
	[156]	Metal surface defects
SOM	[191]	Defects of hot steel strip
	[192]	Defects on rolled steel
	[151]	Corrosion
LVQ	[194]	Defects of cold steel strip
	[193]	Defects of hot steel strip
Fuzzy Logic Classifier	[188, 215]	Defects of cold rolled strip
	[216]	Defects of flat steel
ANN	[195–197]	Corrosion
	[10, 12, 198–200]	Factory facilities inspection
	[201–204]	Defects of civil infrastructure
	[13, 205]	Defects of railway
	[184, 198, 199, 206]	Defects of steel surface

Multi-Scale and Orientation-Aware Bounding-Boxes Regression

In this chapter, we propose a new bounding boxes-based regression solution aiming at detecting objects that may require being aware of their orientation for the detection to be useful. Besides, in order to be able to recognize small objects, the solution adopts a multi-scale approach.

The detector that we propose in this chapter is based on the Single Shot Multi-box Detector (SSD) since it attains prominent detection performance in on-line operation. After fine-tuning SSD and discussing on its performance, we propose the Feature Pyramid SSD (FPSSD) for enhanced detection at multiple scales. We as well design a lightweight neural network for oriented bounding boxes regression. To shorten the terminology, as already done in previous chapters, we will refer to unoriented and oriented bounding boxes as, respectively, BBox(es) and RBox(es). We validate our approach on both the visual inspection and the quality control tasks.

The structure of this chapter is as follows. In Section 4.1, we overview the methodology behind the SSD, while Section 4.2 adopts a transfer learning-based approach to tune SSD for the application cases considered in this dissertation, and discusses on the results achieved. Section 4.3 enumerates a number of loss functions employed in bounding boxes-based object detection. Section 4.4 proposes the new orientation-aware object detector, while Section 4.5 reports on the experimental results obtained. Finally, Section 4.6 concludes our work.

4.1 Overview of the Single Shot Multi-box Detector (SSD)

SSD is a one-stage object detection approach. It is organized around a standard VGG-16 network as the backbone network, with its last fully-connected layer is replaced by

a convolutional layer. Compared with most detection algorithms based on R-CNN [31], such as [27] and [57], SSD does not require any extra procedure to generate proposals. Alternatively, a mechanism of default boxes is used, presenting numerous default boxes for BBox regression. On the other hand, unlike R-CNN based methods, which obtain feature maps from a single convolutional layer, in SSD, feature maps are extracted from layers conv4_3, fc7, conv8_2, conv9_2, and conv10_2. Subsequently, feature maps are connected and sent to a multi-task loss function to regress the coordinates of BBoxes, and to infer the confidence value for the respective category.

The idea of the use of default boxes is to provide prior knowledge for BBox regression, which can improve the stability of the training. Similar to the box anchor in Faster R-CNN [57], the parameters of the default boxes are set by different pre-selected scales and aspect ratios. Then a sliding window strategy is applied for each pixel on feature maps to generate default boxes. Next, two matched pairs are found, i.e., the matched pair of ground truth and predicted BBoxes, as well as the matched pair of ground truth and default box. The matched default boxes are considered prior boxes, and are adopted to regress targets in the loss function.

At the end of the network, a multi-task loss function is adopted, consisting of a Smooth_{L1} Loss for BBox regression (L_{reg}), and a softmax Cross-Entropy Loss for classification (L_{cls}). In more detail, for BBox prediction, SSD adopts a set of indirect regression targets, such as (c_x, c_y, w, h), which denote, respectively, the coordinates of the center point of the BBox and its width and height. Besides, a softmax activation function is employed for multi-category classification.

Let us define x as the output of the network, containing the regression terms of the predicted BBox p and let us define g as the ground truth. In addition, let us define c as the category of the object. The loss function is defined below:

$$L(x, c, p, g) = \frac{1}{N}(L_{cls}(x, c) + \alpha L_{loc}(x, p, g)) \tag{4.1}$$

where N denotes the number of matched prior boxes, and a is a parameter to balance the impact between the classification loss term and the BBox regression loss term on the final loss. In the original method, α is set to 1 by cross-validation.

The BBox regression term consists of two optimized box-delta parameters $(\hat{t}_i, t_j)$, where $\hat{t}_i$ is determined from the predicted BBox p_i and the prior box d_k, and t_j is calculated from ground truth g_j and the prior box d_k. Moreover, the box-deltas are calculated with respect to the center point (c_x, c_y) and the scale (w, h) of the BBox, as

expressed below:

$$\begin{aligned} \hat{t}_i^{cx} &= \frac{p_i^{cx} - d_k^{cx}}{d_k^w}, \quad \hat{t}_i^{cy} = \frac{p_i^{cy} - d_k^{cy}}{d_k^h} \\ \hat{t}_i^{w} &= \log \frac{p_i^{w}}{d_k^w}, \qquad\quad \hat{t}_i^{h} = \log \frac{p_i^{h}}{d_k^h} \\ t_j^{cx} &= \frac{g_j^{cx} - d_k^{cx}}{d_k^w}, \quad t_j^{cy} = \frac{g_j^{cy} - d_k^{cy}}{d_k^h} \\ t_j^{w} &= \log \frac{g_j^{w}}{d_k^w}, \qquad\quad t_j^{h} = \log \frac{g_j^{h}}{d_k^h} \end{aligned} \tag{4.2}$$

Thus, the BBox regression loss function is

$$L_{reg}(x, p, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{c_x, c_y, w, h\}} x_{ij}^p \text{Smooth}_{L1}(\hat{t}_i^m - t_j^m) \tag{4.3}$$

where $x_{ij}^p = \{0, 1\}$ is an indicator function: when the k-th default box matches with the j-th ground truth box (whose category is p), x_{ij}^p is 1, otherwise x_{ij}^p is 0. *Pos* denotes the set of predicted boxes that are matched to a default box with the Jaccard index higher than 0.5, while *Neg* denotes the set of predicted boxes with the Jaccard index lower than 0.5. Besides, SSD uses hard negative mining to select easily misclassified negative examples to construct the *Neg* set, to be three times the size of the *Pos* set. The expression of Smooth_{L1} can be found in a posterior section.

For the classification task, the loss function is a softmax Cross-Entropy Loss over the multi-category confidence value c as defined below,

$$L_{cls}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^n) \tag{4.4}$$

where

$$\hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \tag{4.5}$$

In Eq. (4.4), $\hat{c}_i^n$ stand for the confidence values corresponding to negative predictions.

4.2 Transfer Learning-based Detection Results for SSD

In this section, we report on the results obtained from SSD within the context of transfer learning for the two application scenarios considered. In more detail, we retrain the detector, freezing all network layers except for the last layer, and we repeat this procedure

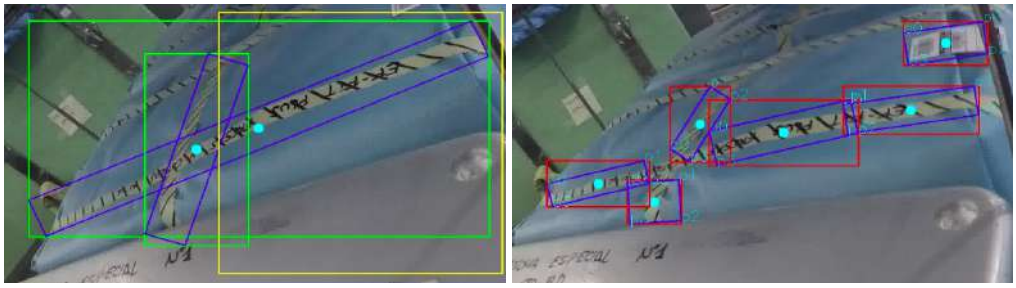


Figure 4.1: Examples of ground truth annotations for datasets A and B of the quality control task.

for the two datasets as independent tasks, i.e., we do not intend from the network to be able to perform detections for the two problems simultaneously.

For the quality control task, as can be seen in Fig. 4.1, we have defined two different datasets (extracted from the final dataset) as for the boxes associated to the targets of every image. Dataset A defines one box for every target of the training image (Fig. 4.1 (left)). Although this seems natural for relatively square and small targets, such as the label and the seal, it is not as straightforward for the paper tape, because of its elongated shape, and hence we define dataset B (Fig. 4.1 (right)), which splits the object in several parts to favor a better training and latter detection of this kind of objects.

In SSD, a proper selection of default boxes becomes crucial for achieving a high detection success, as already noted in [30]. A clustering approach. e.g., using K-means, can be adopted here to set the parameters of the default boxes. In Table 4.1, we report some results for datasets A and B, and up to three different sets of default boxes, namely for 4, 5 and 6 different aspect ratios. The detection performance is measured in terms of the standard metrics defined in Table 2.1, such as mean Recall (mRec), mean Precision (mPrec), mean Intersection Over Union (mIOU), and the mean Average Precision (mAP).

From Table 4.1, we can see that results for dataset B are better than for dataset A, what proves, from the BBox regression point of view, the larger difficulty of dealing with large, elongated objects with regard to square-like, smaller objects. As can be observed, even for dataset B, predicted bounding boxes contain a large area of background, while the detected target, e.g., the paper tape, only occupies a small fraction of the BBox. This suggests the idea of developing an arbitrarily-oriented detector as a proper solution, for this particular case, but also for general detection of objects characterized by a certain orientation, irrespectively of their detailed shape.

In Fig. 4.3, we show two detection examples of the visual inspection dataset. As

Table 4.1: Detection performance of the SSD algorithm for datasets A and B.

Dataset	# def. boxes	mRec	mPrec	mAP	mIOU
A	4	0.5806	0.9019	0.5688	0.7550
	5	0.8195	0.9074	0.8036	0.7735
	6	0.7677	0.8696	0.7213	0.7462
B	4	0.9214	0.9847	0.9225	0.8681
	5	0.9019	0.9631	0.8522	0.8098
	6	0.9309	0.9841	0.9320	0.8719

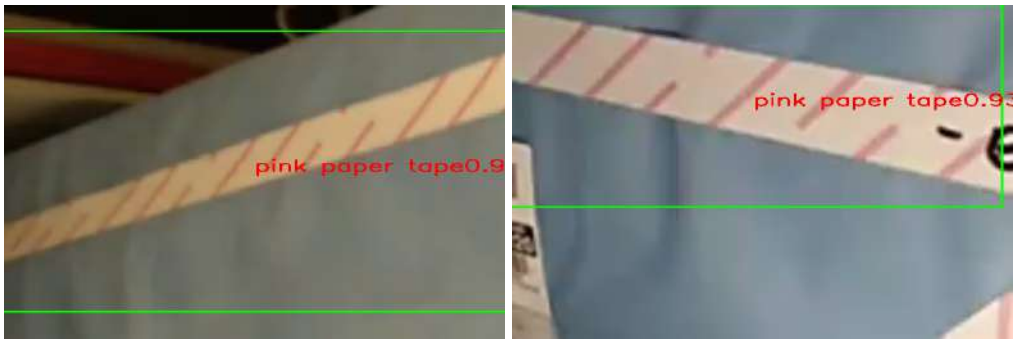


Figure 4.2: Detection results of SSD on Dataset B.

can be observed, this application case requires dealing with maybe highly irregular and possibly elongated shapes, what again supports the idea of developing the arbitrarily-oriented detector. Besides, it may be necessary to be able to simultaneously detect large and small areas affected by corrosion.

4.3 Loss Functions for Oriented Objects Detection

Generally speaking, the BBox detection algorithms based on SSD tend to exhibit high performance. However, a straight rectangle cannot locate effectively objects of elongated shape. For instance, Fig. 4.4 (a) illustrates that the red rectangle cannot accurately locate the paper tape. Notice in particular that the BBox contains a large area of background, while the area of paper tape takes only a small portion of the rectangle, which probably brings instability to the training phase. However, the RBox in Fig. 4.4 (b) locates more accurately the paper tape.

Many existing works focusing on oriented detection comply with a BBox detection

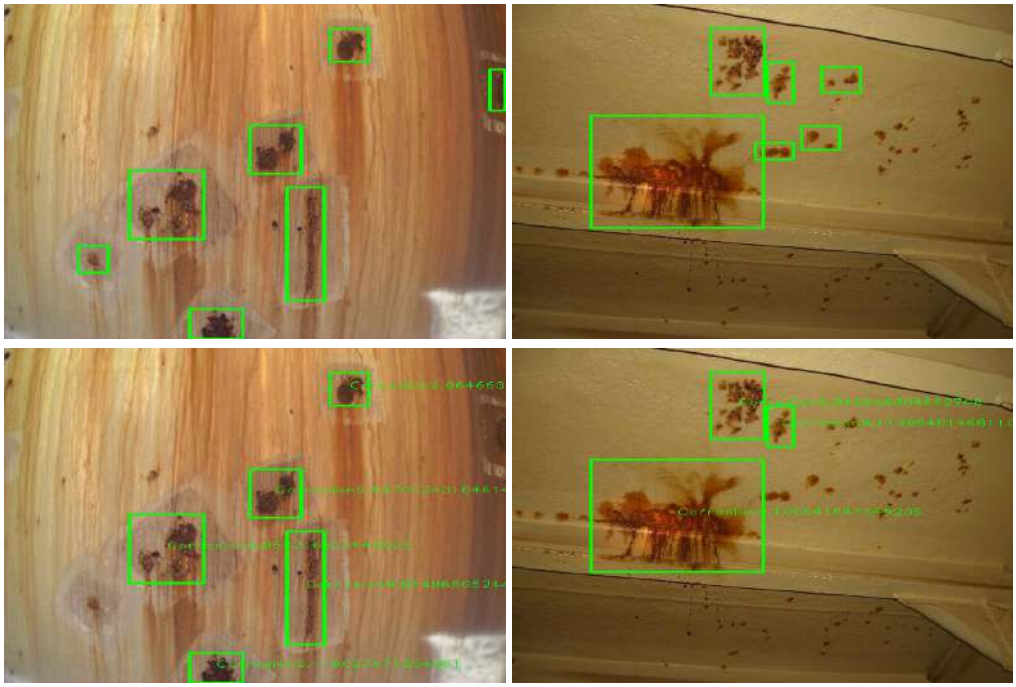


Figure 4.3: Detection results of SSD for the visual inspection dataset. The upper row shows two examples of ground truth , while the lower row shows the detections predicted by SSD.

architecture. Specifically, those studies achieve RBox detection by adding a regression loss function for the RBox parameters. Like most object detection methods, the oriented detection approach needs to classify the category of the BBox, as well as to regress the localization of the target. For the classification task, the Cross-Entropy (L_{ce}) loss function is extensively used with the softmax activation function, as expressed in Eq. (4.4). For the regression tasks, especially for RBox parameters regression, different solutions have been proposed. Next, we discuss the parameterization of an RBox and some relevant loss functions.

The Smooth_{L1} loss function is one of the most common loss functions for the bounding box regression task, such as in [71, 72, 75, 76, 217–219], as defined below:

$$L_{reg} = \sum_{i \in S} \text{Smooth}_{L1}(p_i, p^*) \tag{4.6}$$

where:

$$\text{Smooth}_{L1}(p_i, p^*) = \begin{cases} 0.5 \cdot (p_i - p^*)^2 & \text{if } |p_i - p^*| < 1 \\ |p_i - p^*| - 0.5 & \text{otherwise} \end{cases} \tag{4.7}$$

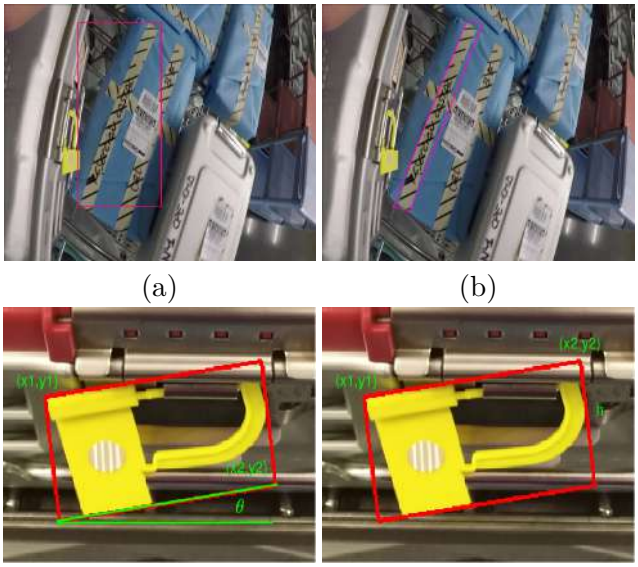


Figure 4.4: Examples of BBox and RBox detections for the quality control dataset are shown in respectively (a) and (b). Different definitions of RBoxes are illustrated in (c) and (d).

where, p and p^* denote, respectively, the prediction and the ground truth. The derivative of Smooth_{L1} is expressed as:

$$\frac{d}{dx}(\text{Smooth}_{L1}(x)) = \begin{cases} -1 & \text{if} \quad x \leq -1 \\ x & \text{if} \quad x \leq 1 \\ 1 & \text{if} \quad x > 1 \end{cases} \tag{4.8}$$

where x represents the difference between p and p^*.

In [81], Liu et al. develop a continuous regression loss function based on Smooth_{L1}, named Smooth_{Ln}, as expressed in Eq. (4.9). Their loss function is intent to achieve a trade-off between robustness and stability:

$$\text{Smooth}_{Ln} = (|x| + 1)\ln(|x| + 1) - |x| \tag{4.9}$$

The Mean Squared Error (MSE) is another typical regression loss function for BBox and RBox regression, as used in [220, 221], and as defined below:

$$L_{\text{MSE}} = ||p - p^*||^2 \tag{4.10}$$

Regarding the parameterization of RBoxes, most existing studies usually make use of five parameters: the coordinates of two vertexes and the height h or orientation θ of the RBox. By way of example, in Fig. 4.4 (c), the regression targets of the RBox contain two vertexes on the diagonal and the orientation of the RBox. On the other hand, Fig. 4.4 (d) shows another parameterization, involving the height of the RBox.

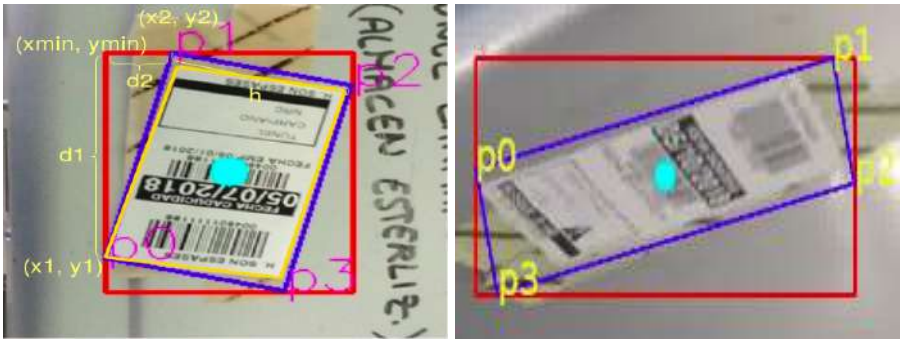
Following [76], the BBox regression problem is categorized into two types, which are direct regression and indirect regression. The indirect regression method derives from R-CNN [31] and consists in computing a set of offsets using the ground truth and prior boxes, as expressed in Eq. (4.2), such as [72, 73, 76, 78, 217, 219]. As its name describes, the direct regression method directly makes use off the error between the prediction and the ground truth [87, 89, 221].

For our tasks, since the orientation angle is periodic, which can confuse the network, we adopt an approach inspired by the one presented in Fig. 4.4 (d). Unlike the text detection task, where the orientation of the text is assigned to a unified orientation to facilitate reading. Besides, the orientation of the targets in our tasks is commonly random and diverse. Thus, prior orientations for the RBox regression task provides no assistance. Moreover, using prior boxes for RBox regression brings massive calculation to transfer the indirect regression terms to the real orientation parameters, thereby reducing the system efficiency. Thus, we select a direct method for RBox regression, which is detailed in Section 4.4.1.

4.4 An Orientation-Aware Multi-box Object Detector

The detector proposed in this section comprises two stages. The first stage employs a method based on SSD to regress the BBox containing the objects of interest. The set of default boxes employed by SSD is determined after a clustering analysis on the training set. Moreover, a feature pyramid architecture is adopted to fuse different feature maps from the backbone network. This approach primarily searches relevant BBoxes and improves the object localization performance. At the second stage, a specifically designed network is adopted to regress the parameters of the RBox maximally contained in everyone of the BBoxes.

In the following, we describe first the parameterization employed for BBoxes and RBoxes in Section 4.4.1. Second, the SSD-based method is presented in Section 4.4.2. Then, the K-means clustering-based strategy for selecting default boxes is introduced in Section 4.4.3. Lastly, a full description of the RBox regression method can be found in Section 4.4.4.



(a) (b)

Figure 4.5: Parameterization of BBoxes and RBoxes in (a) and order of vertexes for the RBoxes in (a) and (b).

4.4.1 Parameterization of Unoriented and Oriented Bounding Boxes

Figure 4.5 illustrates how BBoxes and RBoxes are parameterized in our methodology, as well as how the ground truth is generated. Regarding the latter, first, the yellow lines describe a four-side polygon minimally enclosing the object, from which the minimal RBox is generated, indicated by means of a four-side purple polygon. Then, a minimal BBox is obtained from the rotated BBox, shown as a red rectangle. The anchor point coordinates are denoted by the blue point in Fig. 4.5 (a,b), which are parameterized by (c_x, c_y) and the box size by (w_b, h_b), as in the original SSD.

As shown in Fig. 4.5 (a), an RBox is described by the intersects (d_1, d_2) between the sides of the straight rectangle and the upper side of the rotated rectangle. Optionally, a parameter h is added to select one of the two possible rectangles that may arise from the tuple (d_1, d_2). To determine (d_1, d_2) individually, a clockwise order is defined onto the four corners of the RBox, as indicated in Fig. 4.5 (a) and (b). Thus, the network of the second stage is intended to regress the vector of values (d_1, d_2, h).

The loss function finally employed to regress BBox parameters (same as in Eq. (4.3)) is as follows:

$$L_{reg}(x, p, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{c_x, c_y, w, h\}} x_{ij}^{p} \text{Smooth}_{L1}(\hat{t}_i^m - t_j^m) \tag{4.11}$$

To finish this section, we include next the MSE-like loss function involved in the

regression of the RBox parameters:

$$L(d,g) = \frac{1}{2N}\sum_{i\in N}(||d_1^i - g_1^i||^2 + ||d_2^i - g_2^i||^2 + ||d_h^i - g_h^i||^2) \tag{4.12}$$

where d denotes the predicted (d_1, d_2) and height h, g represents the ground truth and N is the size of the mini-batch.

4.4.2 Feature Pyramid Single Shot Multi-box Detector (FPSSD)

The Single Shot Multi-box Detector considers the use of feature maps from different layers to regress BBoxes. More precisely, SSD applies feature maps of shallow layers to detect small targets. Conversely, it uses feature maps of deep layers to detect big targets. However, the feature maps of shallow layers contain numerous detailed features (e.g., edges, shapes, textures, etc.), whereas they lack the semantic information. In our approach, we adopt the strategy of fusing deep layers with the shallow layers to produce enhanced features. In this way, both detailed and semantic information is encoded in the feature maps, in a sort of pyramid of features to enable the detection of diverse-scale targets.

The idea of the Feature Pyramid originates from the Image Pyramid method. The Image Pyramid is a method to analyze the image at multiple resolutions, generated by multi-scale sampling of the original image via Gaussian kernels. The Image Pyramid imitates the multi-scale features of the image. As assisted by the hierarchical architecture of a CNN, a feature pyramid can be constructed in one feed-forward procedure, so the computational cost of multi-scale sampling can be greatly diminished. Therefore, the Feature Pyramid can efficiently address the multi-scale problem with a relatively low cost.

Several existing approaches aim at using the Feature Pyramid in SSD, such as [58, 222, 223]. Figure 4.6 illustrates four methods for fusing feature maps. Method (a) in Fig. 4.6 depicts the most common strategy, which merges feature maps layer by layer by element-wise addition (or concatenation) and performs detections at each scale. Though the Feature Pyramid has been proved to be able to efficiently improve the detection performance for small targets in [58], it requires massive computation, which is what this work wants to avoid, among other goals. Another method adopts a lightweight fusion approach named FSSD [222] as shown in Fig. 4.6 (b). In this method, the top-down and the down-top paths are independent of each other. First, it fuses the feature maps from the top to the bottom layers in the top-down path by enlarging

the resolution of feature maps from top layers to bottom layers. Then, in the top-down path, the resolution of feature maps is increased to four groups by interpolation. Lastly, the different resolution feature maps are combined by the *concatenation* layer and sent to the loss function. Though this method is capable of saving computational cost as compared with method (a), the feature maps applied in the detector are singular and lack some semantic information. In Fig. 4.6 (c), the depicted method employs a strategy identical to FPN [58] to fuse the feature maps. Moreover, to reduce the computation, also a concatenation layer is implemented to combine the different feature maps. Subsequently, the combined feature maps are fed into the detector. Lastly, Fig. 4.6 (d) illustrates the strategy of the original SSD. It shows that SSD does not have the feature fusion module, and it lacks the capability to capture both low-level details and high-level semantic information.

To finish, Fig. 4.7 describes diagrammatically the architecture of FPSSD. In the proposed method, the feature maps are extracted from conv4_3, fc7, conv6_2, conv7_2, conv8_2, and conv9_2 of the original SSD network. Besides, we use deconvolution layers to increase the dimension of the corresponding feature map. Since feature maps from top layers have more output channels than feature maps from bottom layers, a 1×1 convolutional layer termed *lateral connection* in [58] is used to unify the output channels of all feature maps. Lastly, feature maps integrated with top and bottom layers are sent to the detector to predict the category and localization of targets.

4.4.3 Default Boxes Selection

SSD predefines nine default boxes per feature map location by imposing different size combinations (w_k, h_k). Since the shape of the ground truth can vary significantly, and this approach makes use of a group of selected default boxes termed as prior boxes that are adopted to calculate the regression targets, a proper selection of prior boxes turns out to be critical to achieve prominent detection performance. As already suggested in [65, 66], such a proper selection contributes to the stability of the underlying optimization process, converges faster, and effectively optimizes the IOU between predicted and correct boxes. Taking inspiration from the aforementioned, our detection approach also employs prior boxes, although they are chosen automatically in accordance with the available data, by means of a clustering process.

To be more specific, the well-known K-means algorithm is run over the BBoxes that belong to the ground truth, and the box width and height act as the clustering features. Instead of the Euclidean distance, typically used by K-means implementations, we define

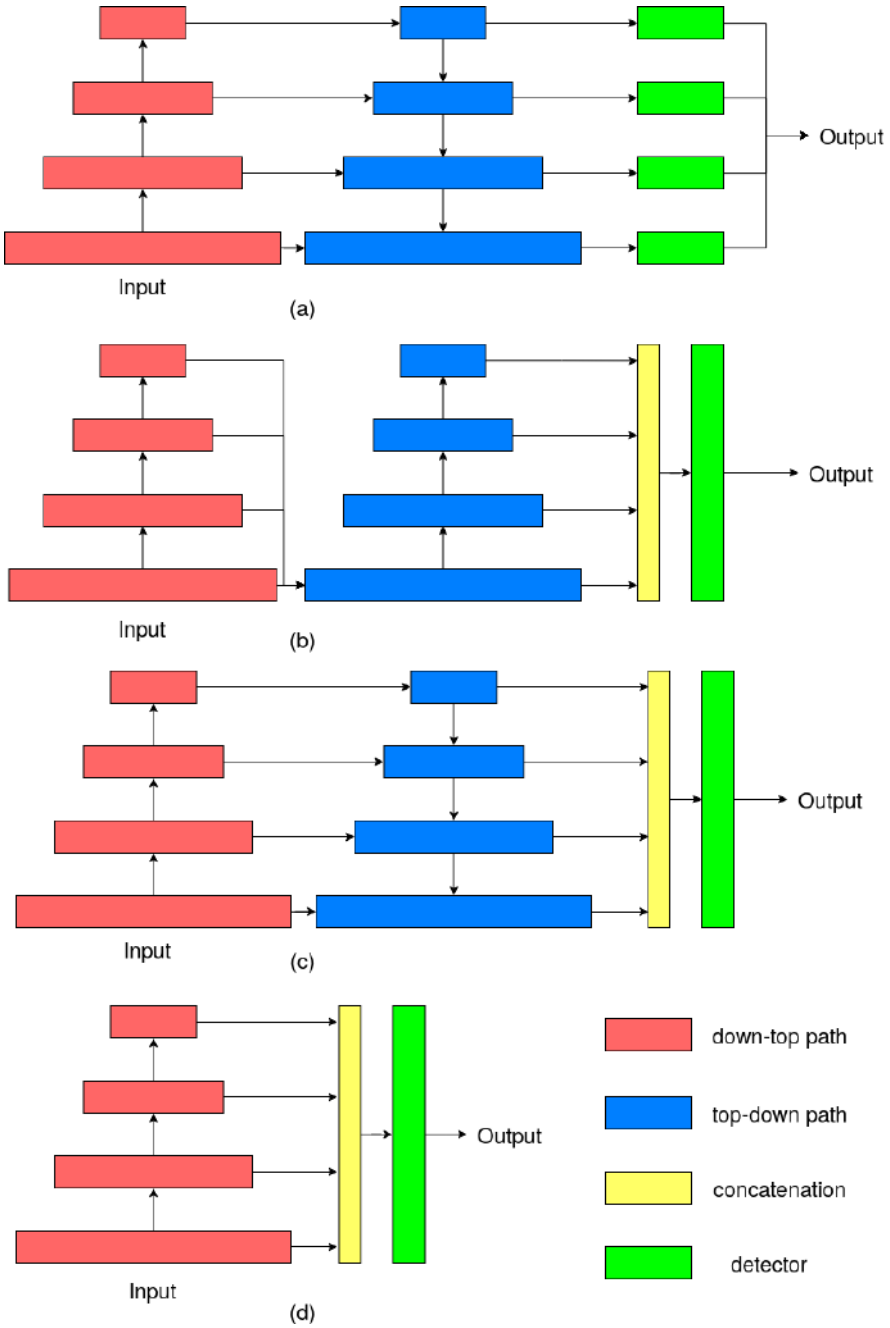


Figure 4.6: Different strategies for fusing feature maps: (a) feature maps are fused from top to bottom layer by layer, as adopted by FPN; (b) a lightweight architecture merges feature maps from top to bottom; (c) our approach, as implemented in FPSSD; (d) the original SSD strategy uses feature maps from different layers separately.

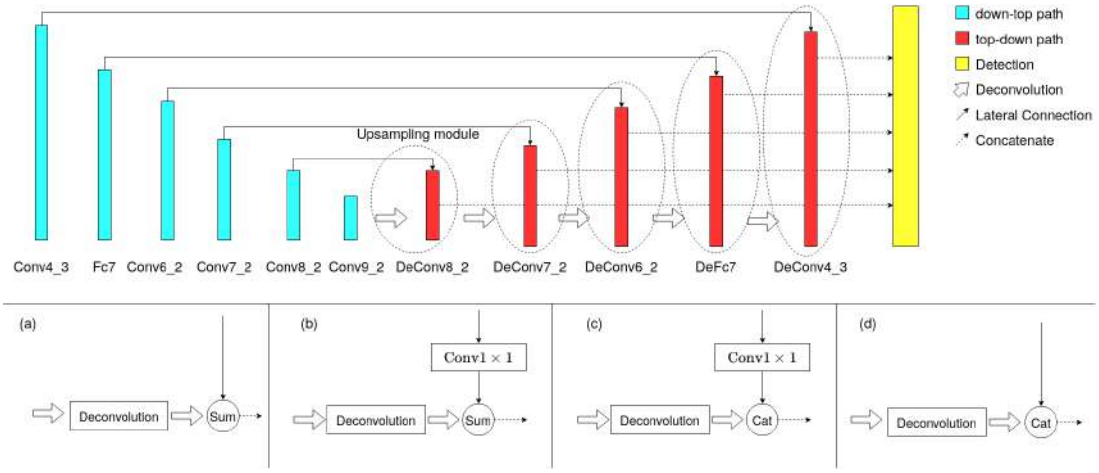


Figure 4.7: Architecture of the Feature Pyramid Single-Shot Multibox Detector (FPSSD). In the second row, four building blocks illustrating the upsampling models are shown: (a) Feature Pyramid Non-Lateral Connection and Elements-wise Sum; (b) Feature Pyramid Lateral Connection and Elements-wise Sum; (c) Feature Pyramid Lateral Connection and Concatenation; (d) Feature Pyramid Non-Lateral Connection and Concatenation.

the IOU as a distance metric since the former tends to miss large BBoxes. Accordingly, the distance between a sample box b_i and the cluster centroid c_j is defined as:

$$\begin{aligned} d(b_i, c_j) &= 1 - \text{IOU}(b_i, c_j) \\ &= 1 - \frac{o(b_i, c_j)}{a(b_i) + a(c_j) - o(b_i, c_j)} \end{aligned} \tag{4.13}$$

where $o(\cdot, \cdot)$ denotes area overlap and $a(\cdot)$ denotes area.

Table 4.2 lists the mIOU values for hand-picked default boxes and for automatically selected boxes generated through clustering for the quality control and visual inspection datasets. Specific to the hand-picked cases, the default boxes are predefined similarly to the Pascal VOC dataset. As can be observed from Table 4.2, four clusters automatically chosen outperform ten pre-defined default boxes, demonstrating that high-quality and better-parameterized default boxes can be obtained by means of our methodology. As could be expected, the more clusters, the better performance is obtained (the trend can be observed to continue for more than six clusters), although the number of clusters should not be high to keep the running time reasonable.

Table 4.2: Mean mIOU (mIOU) of default hand-picked boxes vs. automatically selected using clustering.

Dataset	Approach	# def. boxes	mIOU (%)
quality control	Hand-Picked	4	36.75
	Hand-Picked	5	42.51
	Hand-Picked	6	49.53
	Hand-Picked	10	55.44
	Clustering	4	58.05
	Clustering	5	61.70
	Clustering	6	63.56
visual inspection	Hand-Picked	4	35.93
	Hand-Picked	5	37.96
	Hand-Picked	6	42.75
	Hand-Picked	10	61.82
	Clustering	4	61.58
	Clustering	5	63.37
	Clustering	6	65.31

4.4.4 RBox Regression

To implement the RBox detection, the BBox regression network acts as the input to a specifically designed lightweight network in charge of inferring the RBox parameters. In this section, we describe and discuss the architecture of that network.

The lightweight convolutional network is developed taking as a basis LeNet [224] and introducing a number of modifications: (1) the input size is 63×63 after incorporating an additional convolution layer at the input of the original network, in order to avoid reducing the image to LeNet's 28×28 pixels, which can mean losing too much information; (2) after each convolutional layer, we incorporate a normalization layer to favor a faster convergence during training by means of batch normalization [19], which is known to decrease the effect of covariate shift; (3) since the RBox parameters (d_1, d_2, h) range from 0 to 1, a sigmoid layer has been placed between the last fully connected layer and the loss layer; (4) lastly, the final layer implements the Euclidean loss expressed in Eq. (4.12), also mentioned in Section 4.4.1. The architecture of the RBox regression network can be found in Fig. 4.8.

4.5 Experimental Results and Discussion

This section reports on the experimental results obtained for FPSSD and the RBox regression network, and assesses their detection performance for the two tasks considered in this dissertation. The experimental setup is first presented in Section 4.5.1. Secondly,

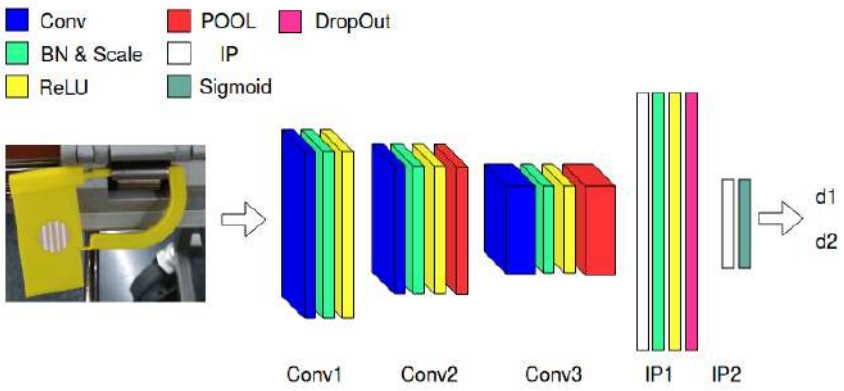


Figure 4.8: Architecture of the RBox regression network. In the drawing, *IP* stands for a fully connected layer, *BN & Scale* represents a batch normalization and scale layer, *ReLU* denotes a ReLU layer, *POOL* indicates a Max-Pooling layer, *Sigmoid* and *Dropout* refer to, respectively, sigmoid and dropout layers.

in Section 4.5.2, we present the results corresponding to BBox regression. Finally, the results for RBox regression can be found in Section 4.5.3.

4.5.1 Experimental Setup

Experimental Environment

Identically to the original SSD, VGG-16 is used as the backbone network. On the other side, the confidence threshold for detection is set to 0.7, as it is commonly done in object detection applications. As already said, we adopt a clustering-based definition of default boxes, comprising a total of 6 prior boxes. This configuration has been employed in all the experiments of this chapter.

To optimize the network weights, we have employed the SGD optimizer with weight decay and momentum, both respectively set to 0.001 and 0.9. We have adopted a multiple steps learning strategy, where the learning rate is set to 10^{-5} for the first 8000 iterations, the next 6000 iterations we use a learning rate of 10^{-6}, and the last 6000 iterations change the learning rate to 10^{-7}.

The quality control dataset consists of 484 images comprising a total of 7 categories, while the visual inspection dataset contains 214 images and comprises two categories. The images have been resized to 512×512 pixels.

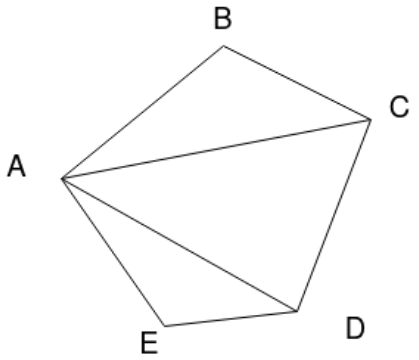


Figure 4.9: Calculation for the area of a convex polygon.

The full detector has been trained on a PC platform fitted with an Intel i9-9900K processor with 64 Gb RAM and an Nvidia RTX 2080 Ti GPU.

Assessment Metrics

To evaluate the performance of our approach, the mean Intersection Over Union (mIOU or Jaccard index) is calculated on the basis of the predicted BBoxes and the ground truth. Moreover, we also report on the standard recall (R), precision (P) and F_1 score for both BBox and RBox regression.

To assess the performance of oriented detection, we also provide the mean RBox IOU (mRIOU) as a supplementary performance metric. Unlike the mIOU of a BBox, the shape of the intersection of two RBoxes can be irregular, so the area of intersection must be found by means of a specific algorithm.

An example for the calculation of the convex polygon area is shown in Fig. 4.9, the area of polygon S_p equals the area sum of three triangles, i.e. $S_p = S_{\Delta\mathrm{ABC}} + S_{\Delta\mathrm{ACD}} + S_{\Delta\mathrm{ADE}}$, where the area of each triangle is the result of the cross product of two vectors: for instance, $S_{\Delta\mathrm{ACD}} = \frac{1}{2}\vec{\mathrm{AC}} \times \vec{\mathrm{AD}}$. For any convex polygon P, the vertices of the polygon are arranged counterclockwise as $\{v_1, v_2, \cdots, v_n\}$, and the vertex coordinates are $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$. On the basis of the aforementioned, the area S_p of the polygon is given by:

$$S_p = \frac{1}{2}\sum_{i=2}^{n}(x_1 y_i - x_i y_1) \tag{4.14}$$

Besides, to compare the regression results directly, we also employ the mean absolute error (MAE), i.e. the difference between the prediction and the ground truth on the test

set, as calculated by Eq. (4.15):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_p - x_g| \tag{4.15}$$

where x_p denotes prediction and x_g represents the corresponding ground truth. Therefore, lower MAE means better regression results.

Comparative Experiments

To assess the BBox detection performance, we take SSD512 as the baseline, using 6 default boxes and 512×512 input images. On the other side, we also include in the comparison TextBoxes++ [79], a solution for text detection in images, which is also based on SSD. It consists of two stages: in the first stage, TextBoxes++ outputs straight and oriented detection results, while, for the second stage, the authors design a neural network to recognize the context of the text. Thus, we select the first stage of TextBoxes++ to compare with our RBox detection approach.

4.5.2 BBox Detection Results

In this part, we evaluate the BBox regressor for the visual inspection and quality control tasks. First, an ablation study is conducted focusing on the lateral connections and the different methods for feature maps fusion in the Feature Pyramid architecture. Secondly, we assess experimentally the BBox regressor and compare it with the aforementioned baseline.

4.5.2.1 Ablation Study

Table 4.3 lists the performance of FPSSD on our two tasks. As can be seen, configurations (a) and (d) employ lateral connections, while configurations (b) and (c) do not. On the other side, configurations (a) and (b) adopt *element-wise sum* to fuse feature maps, while configurations (c) and (d) apply the *concatenation* layer to combine feature maps. The four configurations are considered to select the best way to fuse feature maps. The standard mAP, mean Recall, mean Precision, and F_1 score are calculated to compare the resulting performance.

Table 4.3: Ablation study: effect of lateral connections and different feature maps fusion approaches in the FPSSD architecture. FPNL: Feature Pyramid Non-Lateral, FPL: Feature Pyramid Lateral, SUM: element-wise sum, CAT: concatenation layer. All values are percentages.

Task	Configuration		mAP	mRec	mPrec	F_1
Quality Control		SSD512	80.55	81.11	96.63	88.19
	Fig. 4.7(a)	FPSSD 512 + FPNL + SUM	84.71	85.43	93.24	89.16
	Fig. 4.7(b)	FPSSD 512 + FPL + SUM	86.44	87.15	95.50	91.13
	Fig. 4.7(c)	FPSSD 512 + FPL + CAT	86.32	86.82	95.79	91.08
	Fig. 4.7(d)	FPSSD 512 + FPNL + CAT	85.83	86.46	91.24	88.79
Visual Inspection		SSD512	82.18	83.11	94.34	88.37
	Fig. 4.7(a)	FPSSD 512 + FPNL + SUM	81.31	82.41	95.13	88.31
	Fig. 4.7(b)	FPSSD 512 + FPL + SUM	90.91	91.13	100.0	95.36
	Fig. 4.7(c)	FPSSD 512 + FPL + CAT	81.33	82.64	94.33	88.10
	Fig. 4.7(d)	FPSSD 512 + FPNL + CAT	81.72	82.62	95.63	88.65

Effect of the lateral connection and the feature map fusion strategy

The lateral connection is a convolutional layer with a 1×1 kernel, which is adapted to unify the output channels from top to bottom layers. In the VGG-16 network, the output channels of the top layers are more than the output channels of the bottom layers. Accordingly, after up-sampling, the output channels have to be unified. On the other hand, in the feature pyramid architecture, the scale of feature maps decreases gradually through a series of pooling layers, and the deconvolution operation is adopted to expand the scale of feature maps. In the mentioned process, the locations of targets in feature maps will generate offset. Using the lateral connection, the locations of targets can be passed from the finer level of the top layer via the lateral connections to the bottom layer.

Specifically, regarding the quality control task, the FPSSD obtains better performance than the baseline almost in all cases. We see that the values of mAP and recall of cases (a) to (d) are higher than the original SSD, as shown in Table 4.3. Comparing the performance between cases (a) and (b) on the quality control dataset, case (b) leads to higher mAP, recall, and precision than case (a). Identically, the metrics for case (c) are higher than those of case (d). Thus, the lateral connections are conducive to improving the detection performance for the quality control task.

As for the visual inspection task, the performance observed for FPSSD is rather different to the performance of the quality control task. Cases (a), (c) and (d) give rise to lower mAP and recall compared with the baseline. It is considered that there are two main factors leading to these results: the detected target of the visual inspection task

is corrosion, which is not an object in practice, characterized by an irregular shape and various colors in the same image. Another factor is that some small regions of defects are excessively small to be detected. However, for this dataset, case (b), using lateral connections, gives rise to the highest metric values, significantly larger than the baseline.

Finally, in our two tasks, we are mainly interested in detecting all the targets despite this means increasing the number of false positives in the prediction. Therefore, we seek to attain high values of recall, allowing a reduction in precision. In Table 4.3, the maximum recall is obtained for case (b), reaching 87.15% and 91.13% for the two datasets, employing the lateral connections in the FPSSD architecture.

Concatenation or Element-wise Sum?

The *concatenation* and *element-wise sum* are two common methods for combining two input vectors. In most deep learning frameworks, the *concatenation* layer refers to a utility layer connecting its multiple inputs to one single output according to the channel set by the user, while the *element-wise sum* adds the feature values at the same position of the two input feature maps.

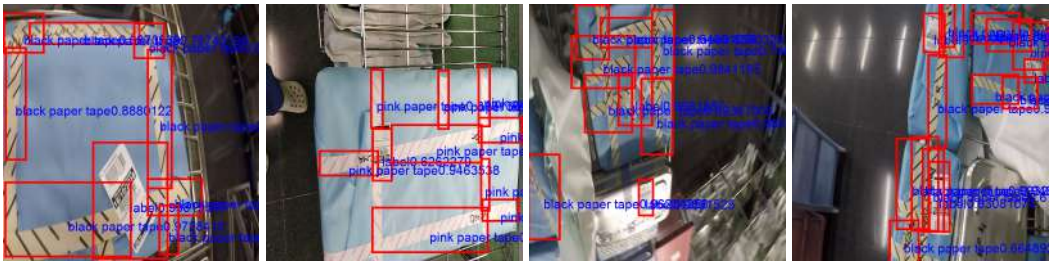
For the quality control task, the FPSSD using the *element-wise sum* to fuse feature maps, i.e., case (b), gets 86.44% mAP and 87.15% recall, both higher than the configuration using the *concatenation* layer, i.e., case (c), 86.32% mAP and 86.82% recall. On the other hand, the precision of case (b) is 95.50%. Although it is 0.29% lower than the precision achieved by the baseline, it is already a positive result. For the visual inspection task, case (b) adopting *element-wise sum* to fuse feature maps reaches the highest values for all metrics.

Summing up, FPSSD using *element-wise sum* and the lateral connections has resulted to be the best configuration, according to the experiments performed involving the quality control and the visual inspection tasks.

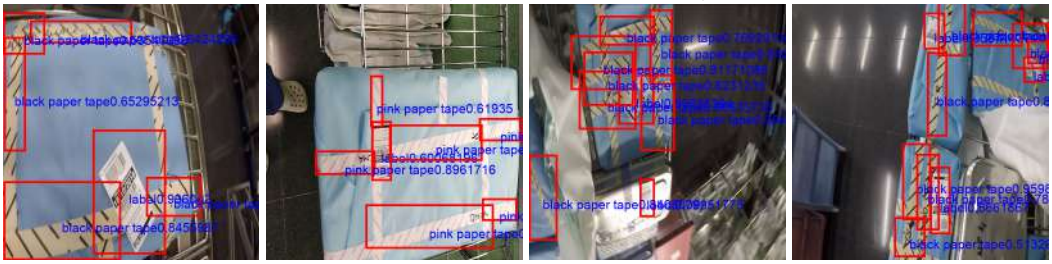
4.5.2.2 Global Performance Evaluation

Figure 4.10 shows some examples of detection results of FPSSD. Results for the original SSD are also shown for comparison. As can be observed, the proposed method achieves better performance than SSD512.

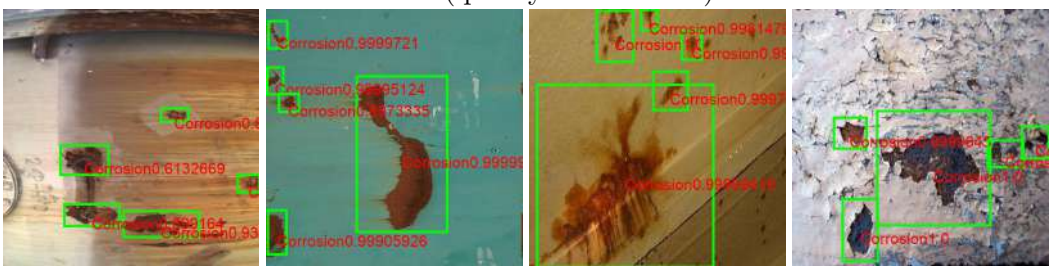
Regarding the quality control task, the most difficult aspect is to detect properly the three kinds of paper tape and discriminate its category. As can be observed in Fig. 4.10, the detections of the first row are more accurate than those of the second row. Even in a very intensive scene, FPSSD can detect almost all parts of paper tape, while the



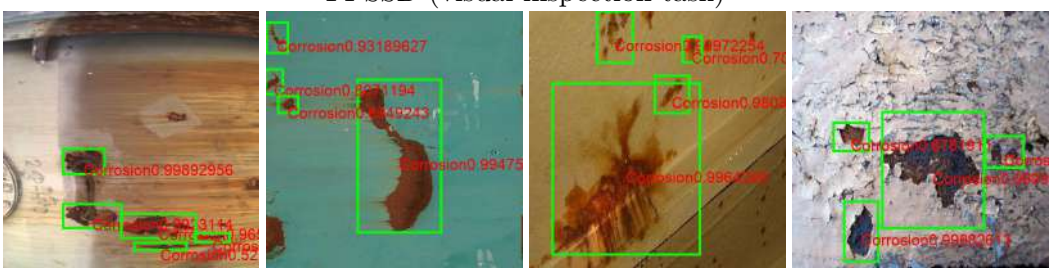
FPSSD (quality control task)



SSD512 (quality control task)



FPSSD (visual inspection task)



SSD512 (visual inspection task)

Figure 4.10: Detection results of FPSSD: (1st row) FPSSD and the quality control dataset, (2nd row) SSD512 and the quality control dataset, (3rd row) FPSSD and the visual inspection dataset, (4th row) SSD512 and the visual inspection dataset.

Table 4.4: Performance results of FPSSD.

Task	Class	mRec	mPrec	F_1	mAP	mIOU
Visual Inspection (FPSSD)	Corrosion	0.9113	1.0	0.9536	0.9091	0.9375
Visual Inspection (SSD512)	Corrosion	0.8311	0.9434	0.8837	0.8218	0.8486
Quality Control (FPSSD)	Label	0.9177	0.9779	0.9468	0.9097	0.8707
	Seal	0.8566	0.9697	0.9096	0.8461	0.8382
	Black tape	0.7191	0.8793	0.7912	0.7055	0.7695
	Blue tape	0.9139	0.9421	0.9278	0.9093	0.8468
	Pink tape	0.8219	0.9614	0.8862	0.8206	0.8236
	Internal filter	1.0	1.0	1.0	1.0	0.9166
	Average	0.8715	0.9550	0.9113	0.8644	0.8443
Quality Control (SSD512)	Label	0.8821	0.9691	0.9236	0.8783	0.8484
	Seal	0.8301	0.9769	0.8975	0.8289	0.8256
	Black tape	0.5468	0.9342	0.6898	0.5387	0.7673
	Blue tape	0.8839	0.9328	0.9077	0.8773	0.8346
	Pink tape	0.7387	0.9668	0.8375	0.7261	0.8259
	Internal filter	0.9841	1.0	0.9920	0.9841	0.9111
	Average	0.8111	0.9663	0.8819	0.8055	0.8404

baseline misses several pieces of it. For the visual inspection dataset (third and fourth rows in Fig. 4.10), the proposed approach also achieves better performance, being able to detect small regions of corrosion which the baseline leaves undetected.

Table 4.4 reports on the quantitative results. For the quality control task, it is clear that FPSSD obtains higher performance than SSD512. In Table 4.4, FPSSD achieves 0.7191, 0.9139, and 0.8219 recall levels for the three types of paper tape, which are 0.1723, 0.03, and 0.0832 higher than the recall values of the original SSD512. As for the average value of mIOU, for FPSSD it results to be 0.8443, which is similar to the average mIOU (0.8404) of SSD512. Besides, the average recall of FPSSD is also higher, 6.04%, than the average recall attained by SSD512. Thus, FPSSD can detect more targets than SSD512 and produce similar quality BBoxes as SSD512. For the other targets of the quality control dataset, namely the Label, Seal and Internal filter categories, FPSSD attains higher performance than SSD512. Looking at the average performance, FPSSD behaves also better than the baseline for all metrics except for the average precision, although the average of F_1 is again higher for FPSSD.

For the visual inspection task, the recall and precision of FPSSD are 8.02% and 5.66% higher than for SSD512. On the other side, observing the mIOU values, FPSSD reaches 93.75% while SSD512 only attains 84.86%, demonstrating that FPSSD can produce higher quality BBoxes than SSD512.

Summing up, FPSSD leads to a quite competitive performance on the two datasets. In respect to the visual inspection task, it has shown able to detect both generic corrosion and small-scale corrosion. As for the quality control dataset, it has been capable of identifying the intended targets in dense scenes.

4.5.3 RBox Regression Results

As already mentioned, though FPSSD can obtain good performance on BBox detection on both tasks, however, for some elongated targets, the results of FPSSD are inaccurate, and some BBoxes contain several parts of other objects, as can be observed specially from the detection results of the quality control dataset, and also from the visual inspection dataset (see Fig. 4.10). As already explained along the previous sections, the aforementioned two issues are expected to be solved by means of RBox regression.

4.5.3.1 Discussion on the RBox Targets Regression Approach

According to Fig. 4.5 (a), if (d_1, d_2, h) is known, a unique RBox can be obtained inside one BBox. If only (d_1, d_2) are available, the orientation of the RBox can be determined uniquely, but there are two possible values for h. They correspond to the distances from the intersection points leading to d_1 and d_2 and the opposite, respective intersection points in the orthogonal direction.

Figure 4.11 shows some examples of RBoxes detections for the two tasks, and for two and three regression targets. Results for a fine-tuned version of AlexNet are also provided in the same figure as a baseline. In these figures, the red points indicate the offsets corresponding to d_1 and d_2, while the green line represents the third regression term h. Furthermore, the black line is used to connect two red points to show the orientation of the target. As can be observed, the black line in the first row (using two regression targets) adheres better to the orientation of the target than the detections of the second row (using three regression targets).

On the other side, Table 4.5 shows the MAE for each regression target to assess the regression performance of the two approaches considered. As can be observed, the MAE values for d_1 and d_2 for the two-target case are lower than the corresponding MAE values for the three-targets case. Moreover, the average MAE of the two-target case is

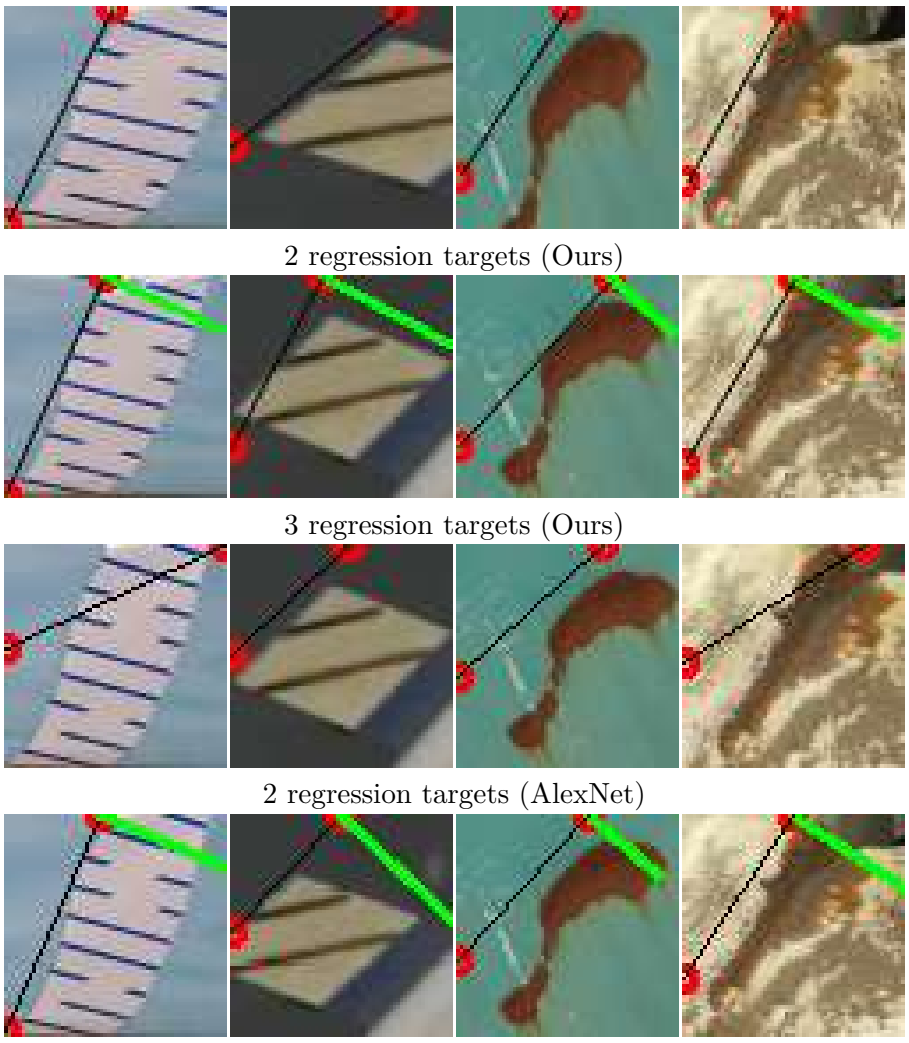


Figure 4.11: RBox regression results: (1st & 2nd rows) regression for two and three targets and our RBox regression network; (3rd & 4th rows) regression for two and three targets and a fine-tuned version of AlexNet. The red dots indicate the intercepts corresponding to d_1 and d_2.

Table 4.5: MAE values for the different regression targets considered for the RBox regression approach.

Task	Approach	d_1	d_2	h	average
Quality Control	Ours (2 Terms)	0.1059	0.1017	-	0.1038
	Ours (3 Terms)	0.2289	0.1862	0.0557	0.1569
	AlexNet (2 Terms)	0.2038	0.1915	-	0.1976
	AlexNet (3 Terms)	0.2430	0.1997	0.0932	0.1786
Visual Inspection	Ours (2 Terms)	0.1556	0.1612	-	0.1584
	Ours (3 Terms)	0.3151	0.3105	0.0889	0.2381
	AlexNet (2 Terms)	0.1722	0.1915	-	0.1818
	AlexNet (3 Terms)	0.2722	0.3744	0.2501	0.2989

Table 4.6: mRIOU values for the RBox regression network. (Ours-2T and Ours-3T stand for the two- and three-target regression methods.)

Task →	Quality Control						Visual Insp.
Category → Method ↓	Label	Seal	Blue Tape	Pink Tape	Black Tape	Intl. Filter	Corrosion
TextBoxes++	0.4851	0.5332	0.3197	0.2683	0.2957	0.5712	0.4615
Ours-2T	**0.7102**	**0.7123**	**0.6247**	**0.5604**	**0.4993**	**0.7669**	**0.5932**
Ours-3T	0.5160	0.4790	0.3136	0.2962	0.3188	0.5917	0.5419

also lower than the average MAE for the three-target case. Summing up, the approach inferring two targets is the one achieving better performance, and consequently it is the one to be selected.

On the other hand, we fine-tune the fully-connected layers of AlexNet to regress the RBox parameters. For the two tasks, it can be seen clearly that the average MAE of AlexNet is higher than for our network, as shown in Table 4.5.

4.5.3.2 Assessment of the RBoxes detector

At last, we connect the FPSSD and the RBox regression network to get oriented detection in the inference stage. The input of the RBox regression network is the prediction of FPSSD, which, because of being a prediction, could be slightly displaced with regard to the true object location, increasing hence the challenge to estimate correctly the object orientation.

Figure 4.12 shows final detection results from our solution. The results are organized

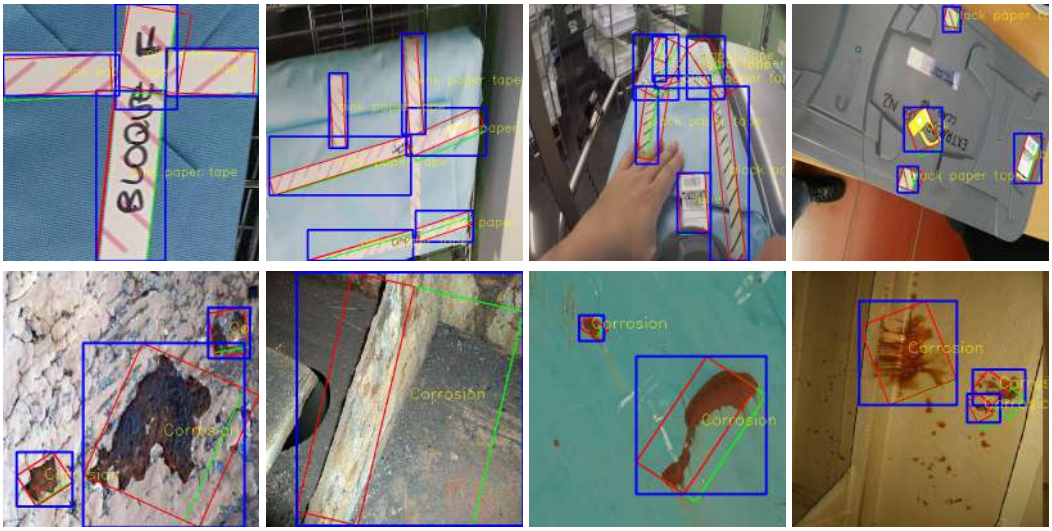
in three groups: the first group results from the two-target regression approach, the second group is from the three-target regression method, and the last group contains results from TextBoxes++ [79]. In the first group, we show two oriented rectangles, in red and green, corresponding to the two possible solutions for every (d_1, d_2) pair.

As shown in Fig. 4.12, the RBox regression network for two-target regression gives rise to more accurate detections than TextBoxes++. The three-target regression approach, although it gives rise to a single solution, is not as accurate than the two-target case. As discussed in the previous section, the robustness of the three-target approach is weaker. As for TextBoxes++, although the network has converged while being fine-tuned for the two datasets, the performance has not resulted to be above the other approaches, and this seems to be caused by the reduced scale of the targets appearing in the two datasets.

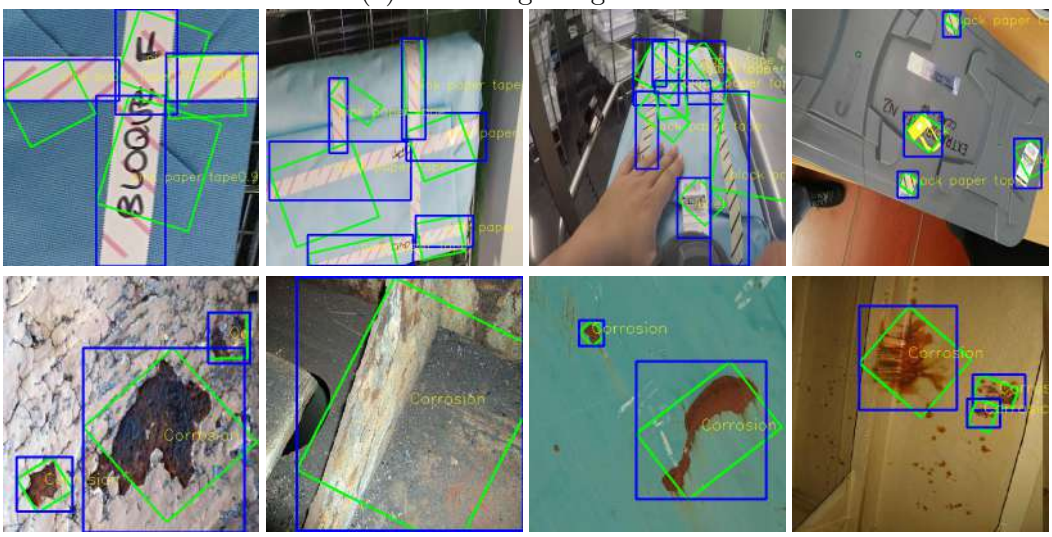
Table 4.6 compares the three approaches by means of mRIOU values. For a fair comparison, we select the biggest RBox for the case of using two-target regression. The mRIOU for the case of two regression targets is higher than the value for the case of three regression targets for all of the categories on the quality control dataset. For the elongated objects (the three types of paper tape), the resulting mRIOU values are 0.6247, 0.5604, and 0.4993, which are significantly higher than for the others. As for the visual inspection task, the mRIOU for the case of two regression targets is also higher than for the other methods.

4.6 Conclusions

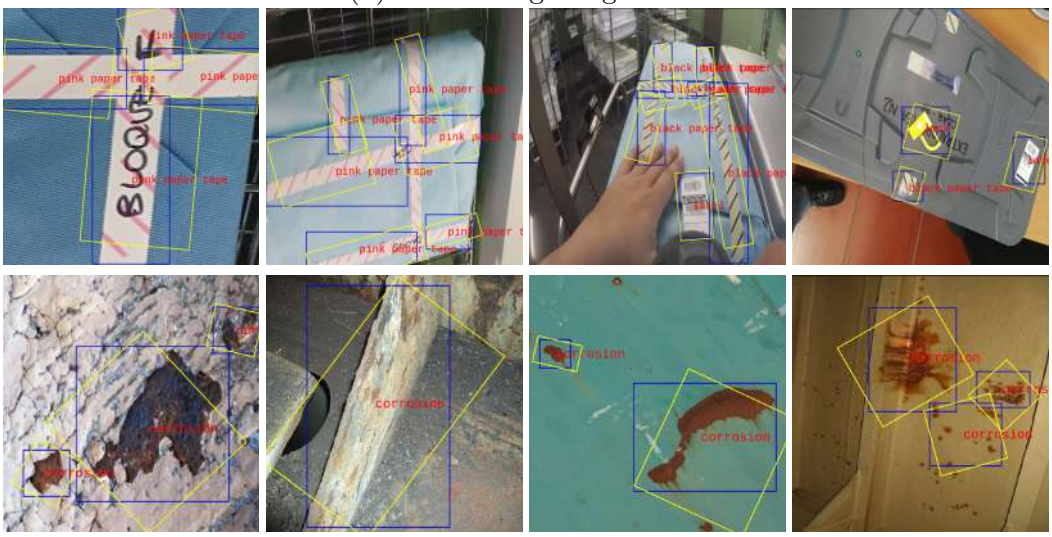
A two-stage arbitrarily-oriented object detection method for regressing the parameters of RBoxes has been described, and assessed on both the visual inspection and the quality control tasks. The first stage of our solution comprises a feature pyramid architecture that has been designed and embedded in an SSD network to fuse the available feature maps, giving rise to the FPSSD network. Besides, the default boxes for BBox regression have been chosen on the basis of a clustering process over the available datasets. The experimental results have shown that our solution outperforms SSD. In the second stage, a simple but effective neural network has been designed to regress the parameters of RBoxes. The design process has considered two parameterizations of RBoxes, to select the most appropriate, which has been the one comprising two regression targets. The experimental results of the whole solution show improved performance over other detection approaches.



(a). Two-target regression



(b). Three-target regression



(c). TextBoxes++

Figure 4.12: Examples of final detections from the two tasks.

Chapter 5

Image Semantic Segmentation

Despite the general prominent performance exhibited by the BBox detector, when lots of objects appear in a small area, the BBox-based detection approach tends to degrade in performance, and also, most importantly, detection results become less informative and even messy. Due to this reason, we have considered the adoption of an alternative semantic segmentation approach to detect targets at the pixel level, increasing the localization accuracy.

The loss function is a crucial part of DCNN-based segmentation approaches. As described in Section 2.4.2, several loss functions have been developed in order to solve some specific problems. Hence, given the fact that not all of them can be useful for any possible segmentation context and task, the question arises as to what loss function is the most suitable for our specific tasks. In this respect, we investigate three popular loss functions, namely the Cross-Entropy loss, the Focal loss, and the Dice loss, to train an FCN-8s network in order to observe the effect of these loss functions on the segmentation performance and select the most appropriate one.

In this chapter, we also address the cost of generating the pixel-level ground truth necessary for semantic segmentation against the quality of the segmentation output. As it is well known, the preparation of pixel-wise annotations is a time-consuming process, which require a large effort if high-quality results are expected. Recently, the use of weak annotations instead has become a hot topic. In this respect, we propose a new weakly-supervised image segmentation methodology aiming at high-quality segmentation with a reasonable labelling effort. We evaluate the resulting performance on the two tasks that we consider in this dissertation.

The chapter is organized into three sections. Section 5.1 reports on the segmentation performance of FCN-8s when trained by means of several and different loss functions. Section 5.2 introduces the weakly-supervised semantic segmentation solution and pro-

vides the detailed configuration and the experimental results that have been obtained. Section 5.3 sums up our work and discusses on possible improvements.

5.1 Fully Supervised Semantic Segmentation

In this section, we attempt to use a fully convolutional neural network architecture (FCN) [52], which is trained end-to-end, to obtain pixel-level detection results for both the visual inspection and the quality control tasks. The FCN learns and predicts dense outputs from a whole-image-at-a-time by dense back-propagation/feed-forward computation.

During first experiments, we found that the detector lacked the ability to detect small objects in both tasks. The reason for this behavior is twofold: on the one hand, if the training set is imbalanced, the model is prone to fail for the category with less data; on the other hand, the network architecture itself imposes limitations to the size of the smallest target detectable.

As part of our work, we have considered up to three different loss functions for the FCN-8s [52] architecture. Due to their characteristics, these variants are considered to allow the detector to locate both large and small objects, the latter typically corresponding to the challenging detection case.

5.1.1 Network Architecture

In this work, the FCN [52] does not contain any fully connected layer, thus reducing the loss of spatial information due to the compulsory scale operations on the input image. Instead, the authors use convolutional layers to replace fully convolutional layers based on VGG-16 [20], and the full network allows arbitrary input dimensions, and also accepts arbitrary output dimension.

Typical DCNNs, such as LeNet, AlexNet, and VGG, recursively apply a series of basic components, including convolution, pooling, and ReLU layers, to the input features. However, the output dimension of these networks is reduced by subsampling to keep filters small and computational requirements reasonable. In order to obtain the dense predictions, the feature maps need to be recovered to the dimension of the input, reducing the number of channels at the same time. Two approaches of up-sampling are used in DCNN models, which are bilinear interpolation and deconvolution.

Typical bilinear interpolation computes each output from the nearest four inputs by a linear map that depends only on the relative positions of the input and output cells. Deconvolution is the inverse operation of convolution, which reverses the forward

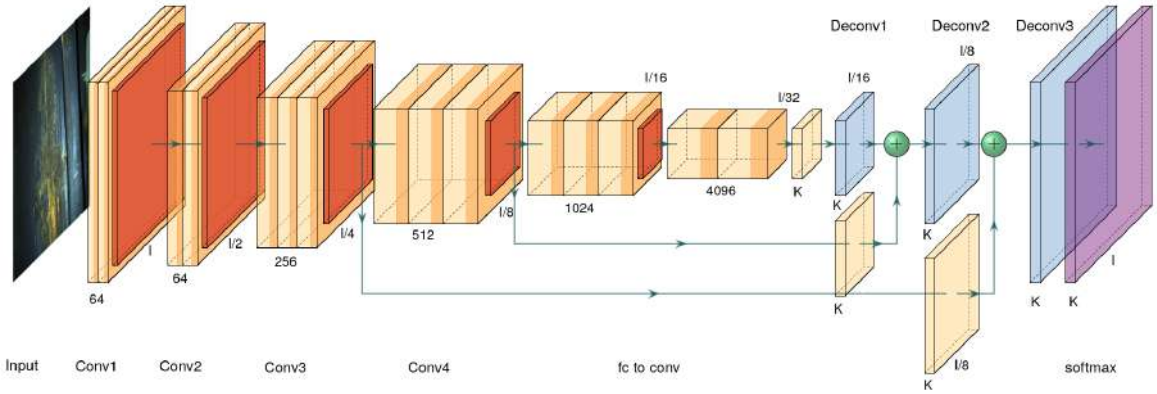


Figure 5.1: The architecture of FCN-8s, which combines semantic deep features with detailed shallow features to obtain accurate dense predictions by means of lateral connections.

and backward passes of convolution. On the other side, a stack of deconvolution layers and activation functions can learn weights for up-sampling, contrary to bilinear interpolation which is a fixed operation. Besides, a deconvolution layer can be implemented within a DCNN model for end-to-end learning by back-propagation from a pixel-wise loss function. By way of example, the architecture of FCN-8s [52] is shown in Fig. 5.1.

For a start, we consider the visual inspection task and hence a traditional binary classification problem, solved by means of FCN-8s to discriminate areas with corrosion from areas with no or minor corrosion. To detect the defects, an RGB image of size $w \times h \times 3$ is used as the input, and FCN-8s outputs the corresponding binary prediction of size $w \times h$. We proceed similarly for the quality control task, just increasing the number of classes and re-training.

5.1.2 Relevant Segmentation-oriented Loss Functions

As described in Section 2.4.2, researchers have developed different loss functions to focus the training on specific details of the input images. In this section, we involve the Focal loss [45], the Dice loss [46], and the Cross-Entropy loss for both tasks.

5.1.2.1 Cross-Entropy Loss

Cross-Entropy is an important concept in information theory, as it is used to measure the dissimilarity between two probability distributions. The Binary Cross-Entropy loss is widely used for binary classification tasks, and as well for semantic segmentation and

hence pixel-level classification, where it also succeeds. The definition of Binary Cross-Entropy is as follows:

$$L_{BCE}(y_i, \widehat{y}_i) = -(y_i \cdot \log(\widehat{y}_i) + (1 - y_i) \cdot \log(1 - \widehat{y}_i)) \tag{5.1}$$

where, $\hat{y}_i$ is the predicted value (after e.g. a sigmoid or a softmax activation function) and y_i is the ground truth, both for pixel i.

For the multi-category case, the cross entropy loss adopts the following form:

$$L_{CE}(y, \hat{y}) = -\sum_i y_i \cdot \log(\hat{y}_i) \tag{5.2}$$

where $\cdot$ is the dot product, y_i is the ground truth in one-hot encoding and $\hat{y}_i$ is a vector of predicted probabilities, one per class.

5.1.2.2 Focal Loss

The Focal loss [45] is a variation of the Cross-Entropy loss, as shown in Eq. (5.3). Differently from the standard Cross-Entropy loss, the Focal loss contains a modulating factor γ and balance parameter α. The modulating factor γ adjusts the smoothly rate at which easy examples are down-weighted and extends the range in which an example receives a low loss value when γ is increased. The balanced parameter α is used in conjunction with γ, as can be seen in the following:

$$L_{FL}(\widehat{y}_i) = -\alpha(1 - \widehat{y}_i)^{\gamma} \log(\widehat{y}_i) \tag{5.3}$$

α can be set by inverse class frequency or treated as a hyper-parameter to be set by cross-validation. When γ is 0 and $\alpha = y_i$, the focal loss becomes into the standard Cross-Entropy loss.

Figure. 5.2 shows the focal loss for several settings of γ. Commonly, the threshold for the predicted probability is set to 0.5, so these samples are marked as well-classified samples in Fig. 5.2. With γ increasing, the loss values of well-classified samples decrease, which means that the impact of well-classified samples reduces and the training focuses more on the difficult-to-classify samples, i.e., the minority class(es). The Focal loss was proposed to deal with highly imbalanced problems [45].

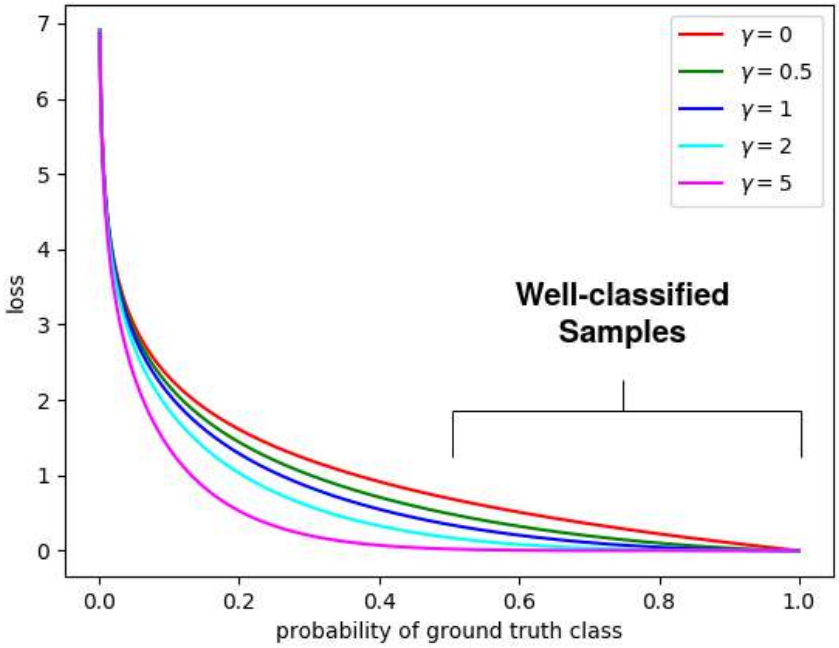


Figure 5.2: Behaviour of the Focal loss for different γ values, and α set to 1. As shown, the x axis represents predicted probability, and the y axis represents the corresponding loss value.

5.1.2.3 Dice Loss

The Dice loss derives from the Sørensen-Dice coefficient, which can be used to evaluate the similarity between two binary vectors, e.g., $y = (y_1, \cdots, y_N)$ and $\hat{y} = (\hat{y}_1, \cdots, \hat{y}_N)$. In [208], the authors change the Dice coefficient into a loss function for 3D medical image segmentation. The Dice loss function is shown in Eq. (5.4):

$$L_{DL} = 1 - \frac{2\sum_i y_i \cdot \hat{y}_i + \epsilon}{\sum_i |y_i|^2 + \sum_i |\hat{y}_i|^2 + \epsilon} \tag{5.4}$$

where, as before, $\cdot$ is the dot product, y_i is the ground truth in one-hot encoding format and $\hat{y}_i$ is a vector of predicted probabilities, one per class. ϵ prevents the denominator from becoming 0.

Unlike the CE loss, which needs to set a balance parameter between the foreground and the background, the Dice loss only focuses on the foreground of the image, and the background has less contribution to the loss value. The Dice loss is also oriented to

imbalanced problems.

5.1.3 Experiments and Discussion

5.1.3.1 Experimental Setup

The FCN tested here applies VGG-16 as an encoder, and is trained by means of the Focal loss (FL), the Dice loss (DL), and the softmax Cross-Entropy loss (SO). As for the decoder network, its weights have been initialized using the Kaiming method [225]. The experiments reported in this section have been preceded by series of tuning iterations to get the best possible results. For all experiments, the Stochastic Gradient Descent with Momentum optimizer (SGDM) has been selected. The maximum number of training iterations has been set to 200 epochs, with a mini-batch size of 8, and a momentum of 0.9. As on previous occasions, a machine with a GPU NVIDIA GeForce RTX 2080 Ti has been used in all the experiments. For the Focal loss, the modulating factor γ has been set to 2 and α_i to 0.25.

5.1.3.2 Gradient Analysis

For a start, we first analyze the effect of the different loss functions on the back-propagation training step. In Fig. 5.3, we plot gradient maps from the last layer, i.e., the activation functions using the same input image. As can be observed in the first row of Fig. 5.3, the gradient values after the first iteration of the three models are different. Firstly, looking at the DL gradient map, it is clear that the gradient values of the background are 0, which do not update weights during back-propagation, while the gradient values of the foreground are not 0. As for the SO loss, both foreground and background contribute to the back-propagation, and the absolute values of both are similar. Regarding the FL loss, the absolute values of the gradient for background and foreground are different. As explained before, we can change the modulating factor γ to fine-tune both values.

After training, we can see in the second row of Fig. 5.3 that the gradient's absolute values for the object boundaries are higher than for other image locations, so all three models attempt to learn to obtain accurate detections. Differently to the FL and SO losses, the gradient values of points inside the targets are not zero for the DL loss. Thus, we can conclude that the DL loss focuses more on the foreground area than the other loss functions. On the other hand, the gradient values at the target interior points are around 0.2 for DL.

Table 5.1: Detection results for FCN-8s and different loss functions.

Dataset	Loss	PA	MA	mIOU	fwIOU	P	R	F_1
Visual Inspection	DL	0.96	0.92	0.87	0.93	0.92	0.88	0.90
	FL	0.95	0.87	0.83	0.90	0.95	0.83	0.89
	SO	0.96	0.91	0.86	0.92	0.91	0.75	0.82
Quality Control	DL	0.85	0.84	0.86	0.91	0.82	0.78	0.80
	FL	0.83	0.81	0.83	0.87	0.80	0.79	0.79
	SO	0.86	0.82	0.82	0.88	0.78	0.79	0.78

By observing the gradient maps, we know that the Cross-Entropy loss treats foreground and background fairly, and when the area of the foreground is relatively small, the network tends to learn more the background, thereby ignoring the foreground. The purpose of the Focal loss is to readjust the proportion of foreground's and background's contribution to the back-propagation. As a result, the Focal loss creates an imbalanced environment that is more inclined to the foreground. Finally, the Dice loss focuses more on exploring to predict better the foreground, and the background has a more limited effect during training. Therefore, when the detected target is very small, the Dice Loss is expected to have a larger performance. From the third row in Fig. 5.3, we can effectively see that the DL model achieves the best performance for small targets.

5.1.3.3 Experimental Results

Table 5.1 compares the results obtained from the Focal loss, the Dice loss, and the Softmax Cross-Entropy loss function. The performance is measured in terms of the standard metrics for semantic segmentation defined in [52], such as pixels accuracy (PA), mean accuracy (MA), mean region intersection over union (mIOU), and frequency weighted IOU (fwIOU), as well as through the traditional precision (P), recall (R) and F_1 metrics.

As can be observed, metric values in Table 5.1 are quite similar for the different loss functions when segmentation metrics are considered. Looking at the recall values, the DL-based model obtains the highest score for the visual inspection task, while the SO-based model obtains the lowest recall. On the other side, the DL-based model also obtains the highest MA, mIOU, and fwIOU. Therefore, all in all, the DL-based model seems to be the best option for the visual inspection task.

As for the quality control task, any of the three models give rise to an evident performance gap. As discussed before, the purpose of the Focal loss and the Dice loss is

to solve the imbalance problem. We recall that, for semantic segmentation, the imbalance problem manifests essentially as undetected small targets. In our quality control dataset, there is not an extreme imbalance situation. However, we can see that the DL model obtains the highest mIOU, fwIOU, and Precision, and for the other metrics the value attained is almost the same as the highest one. Therefore, the DL loss results to be also effective for the multi-category segmentation problem, and we thus select the DL-based model for the quality control task.

Figure 5.4 shows examples of segmentation results for both tasks. For the visual inspection task, the DL-based model obtains better segmentation performance for a small area of corrosion, as expected. By adjusting the modulating parameters γ in the Focal Loss, we can see that the performance, for small corrosion areas, is also better than for the SO-based model. As for the FL-based model, it still misses some small corrosion areas compared to the DL-based model. As for the quality control task, in the comparison of the segmentation results of the three models, results from the DL-based model turns out to be better than from the others, while the results of the FL- and SO-based models are similar.

5.1.4 Conclusions

The loss function is an important element of the optimization stage in deep learning-based segmentation approaches. It is crucial to determine the effects of the different loss alternatives for the tasks under consideration. In this work, we have investigated how to solve the segmentation problems for small targets by using various loss functions to train the DCNN model, namely the Dice loss, the Focal loss, and the softmax Cross-Entropy loss. In the experiments we have employed the same network and kept the same training configuration.

We have compared the three models by means of standard performance metrics, after training with the aforementioned three loss functions. The Dice loss has resulted the best on average for the two tasks.

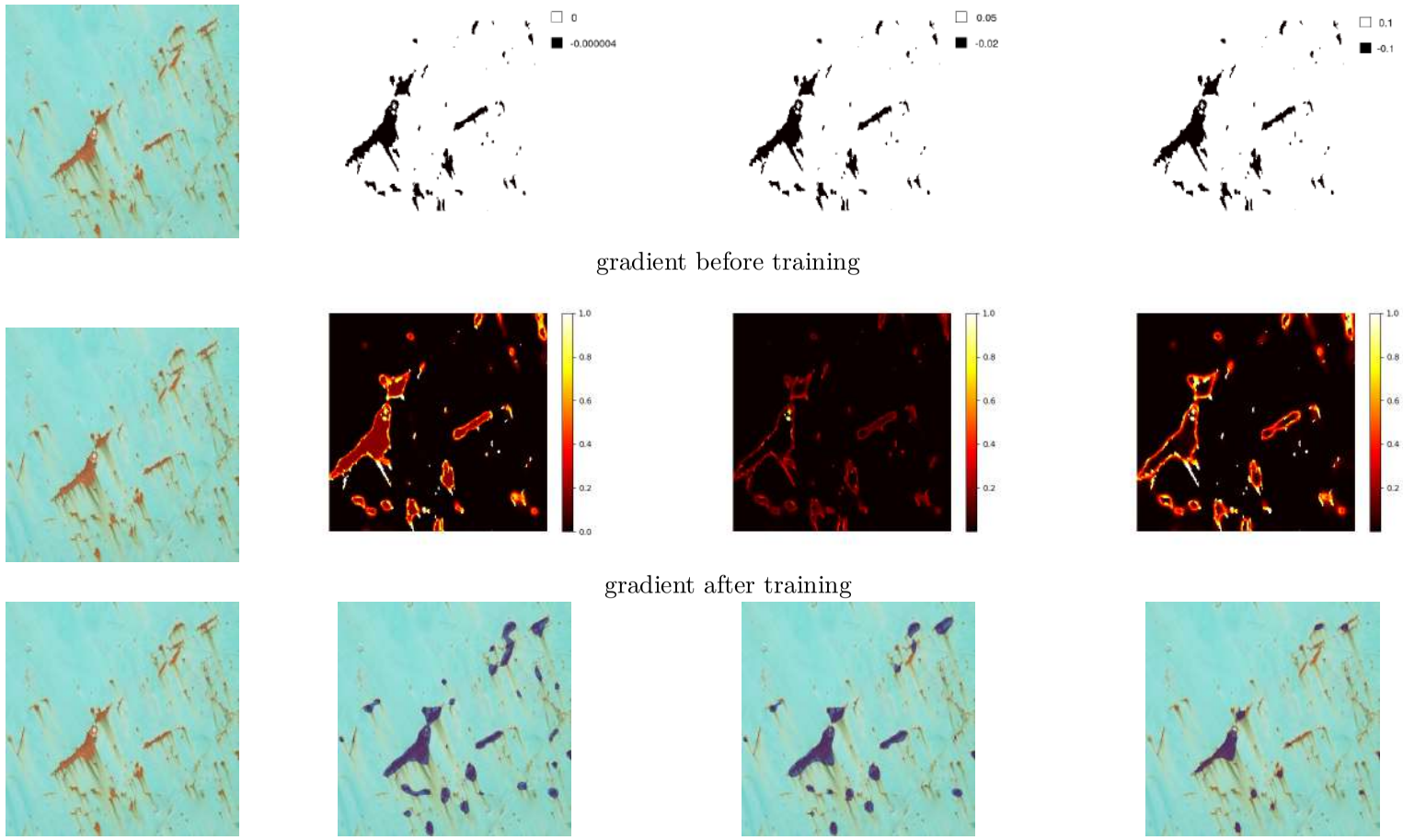


Figure 5.3: Gradient maps from the Dice loss, the Focal Loss, and the softmax Cross-Entropy loss: (a) input images, (b) to (d) gradient maps from respectively, DL, FL, and SO. The first row shows the gradient maps absolute value after the first training iteration. The second row shows the gradient maps absolute value after the last iteration. To compare the gradient maps conveniently, we normalized the gradient maps into the range from 0 to 1. The third row shows the final detection results.

5.2 Weakly-Supervised Semantic Segmentation

Despite obtaining good segmentation performance through fully supervised approaches, we find that it is very costly to obtain the pixel-level ground truth for real applications. In general terms, DCNN-based semantic segmentation needs a large number of pixel-level annotations to train the neural network. Although powerful interactive tools have been developed for annotating targets at the pixel level, e.g. [226], which makes it sufficient for the user to draw a minimal polygon surrounding the target, it still takes a few minutes on average to label the target area for one single picture. Besides, it is difficult for inexperienced staff to ensure accuracy in marking corrosion at the pixel level in the visual inspection dataset. Because of that, in this section, we consider Weakly-Supervised Semantic Segmentation (WSSS) solutions.

Up to now, WSSS methods have been widely discussed. There are various forms of user interaction suitable for segmentation annotations, such as image tags, scribbles, and bounding boxes. Among all of them, the scribbles option allows for easy, friendly and flexible annotations, i.e., it is just necessary to drag the cursor over the targets to detect, using different colors for the different categories, supplying more information than image tags and not involving pixels from other categories, unlike bounding boxes.

Consequently with all the aforementioned, in this section, we develop a new WSSS solution and show its validity for the two detection problems considered in this dissertation.

5.2.1 Methodology and Network Architecture

Figure 5.5(a) illustrates fully-supervised semantic segmentation approaches based on DCNN, which applies a pixel-wise training strategy. In this way, the network is trained to predict analogous results to the full labelling ground truth, thus achieving good segmentation performance levels in general. However, this model ignores the reality that pixels of the same category tend to be similar to their adjacent pixels. In the WSSS problem, due to lack of accurate and sufficient pixel-wise labels, the similarity of pixels can be taken advantage of to obtain the categories of unlabelled pixels. In this respect, there are some works reliant on the similarity of pixels to train the WSSS network, for example, a dense CRF was used in [227], the GraphCut approach was adopted in [136], and a superpixels segmentation algorithm was used in [125].

Inspired by ScribbleSup [125], in this section, we also consider combining superpixels and scribble annotations to obtain the category information of unlabelled pixels and

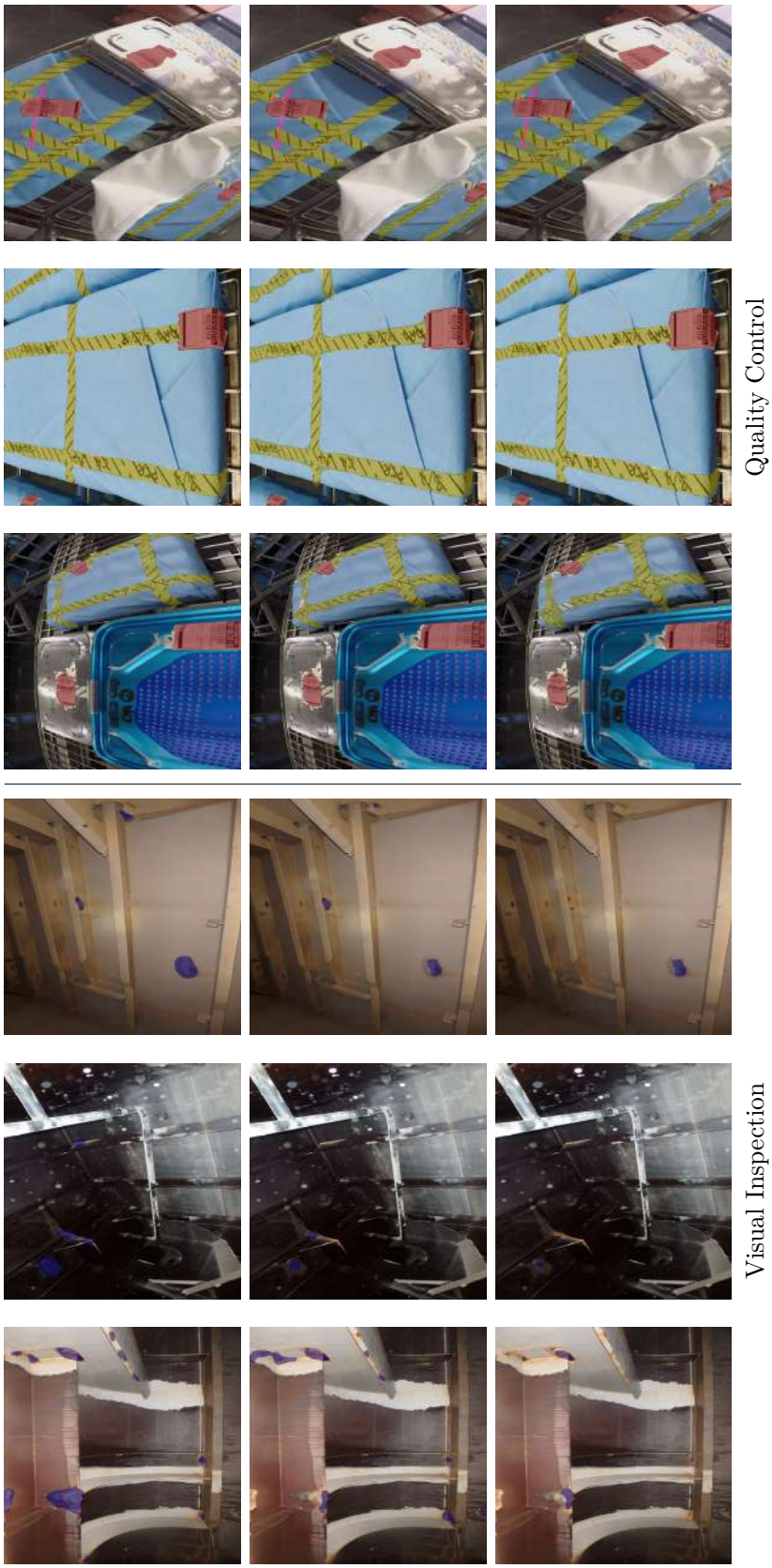


Figure 5.4: Examples of segmentation results for the visual inspection task (left) and for the quality control task (right): (1st row) segmentation results for the DL-based model, (2nd row) segmentation results for the FL-based model, and (3rd row) segmentation results for the SO-based model.

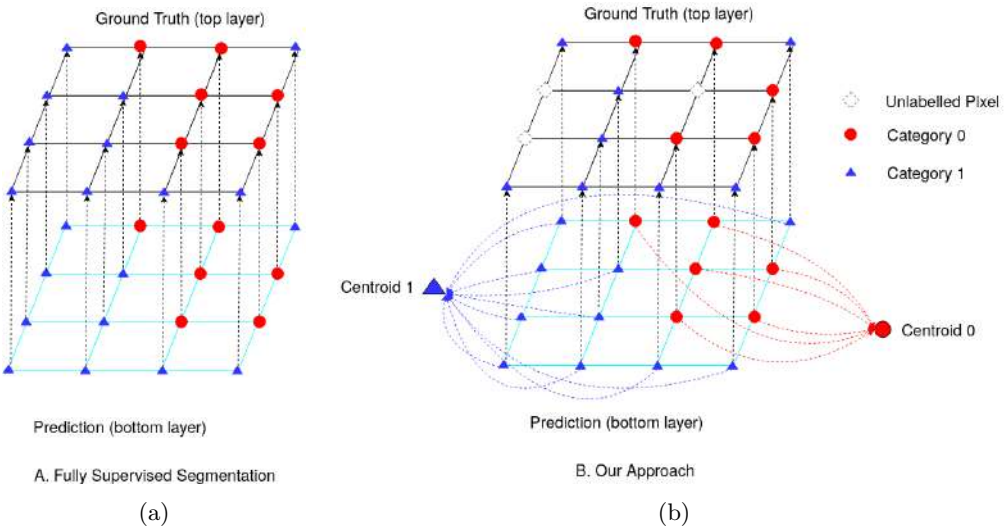
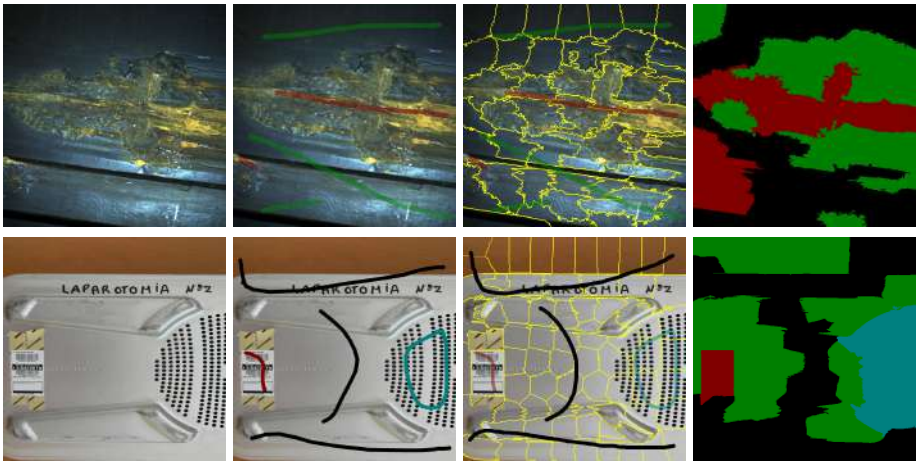


Figure 5.5: Illustration of (a) full supervision and (b) our weakly-supervised approach for semantic segmentation: (a) all pixels are labelled to make the prediction [bottom layer of the drawing] resemble the ground truth [top layer of the drawing] as much as possible after pixel-wise training; (b) to solve the WSSS problem, the category information from the incomplete ground truth, i.e., the weak annotations, is propagated towards the rest of pixels making use of pixel similarity and minimizing distances to class centroids derived from the weak annotations.

generate pseudo-masks as segmentation proposals, which can make the network converge fast and achieve competitive performance. Fig. 5.6 shows the scribble annotations and pseudo masks derived from superpixels. For example, Fig. 5.6 (b) and (c) respectively show the scribble annotations and the superpixels-based segmentation obtained for two images of the two application cases considered. The corresponding pseudo-masks, which contain more annotated pixels than the scribbles, are shown in Fig. 5.6 (d). As can be observed, not all pixels of the pseudo-masks are correctly labelled, and so a solution robust to these inaccuracies has been developed, as suggested in Fig. 5.5 (b). To this end, we incorporate the Centroid Loss and a normalized MSE term into the full loss function. This is discussed in Section 5.2.2.2.

In this section, we describe the methodology of our approach. To begin with, we refer to how weak annotations are handled and how the pseudo-masks are obtained in Section 5.2.1.1. Then, the architecture of the network is described in Section 5.2.1.2 and the different loss terms are discussed in Sections 5.2.2.1 (partial Cross-Entropy loss, L_{pCE}), 5.2.2.2 (Centroid Loss, L_{cen}) and 5.2.2.3 (normalized MSE-term, L_{mse}, and the full loss function L). At last, the calculation process of our approach is introduced in Section 5.2.2.4.



(a) (b) (c) (d)

Figure 5.6: Weak annotation and propagation example: (a) original images; (b) scribbles superimposed over the original image; (c) scribbles superimposed over the superpixels segmentation result; (d) resulting pseudo-masks. Regarding the scribble annotations: (1st row) red and green scribbles respectively denote corrosion and background; (2nd row) black, red and blue scribbles respectively denote background, tracking label and the internal filter texture. As for the pseudo-masks: (1st row) red, black and green pixels respectively denote corrosion, background and unlabelled pixels; (2nd row) red, blue, black and green pixels respectively denote the tracking label, the internal filter texture, the background and the unlabelled pixels.

5.2.1.1 Weak annotations and pseudo-masks generation

Figure 5.6 (b) shows two examples of the scribble annotations, one for the visual inspection case (top) and the other for the quality control case (bottom). Since scribbles represent only a few pixels, the segmentation performance that the network attains using exclusively scribbles is far from satisfactory for any task that is considered. To improve the network performance, we combine the scribbles with an over segmentation of the image to generate pseudo-masks as segmentation proposals for training. First, an over segmentation of the input image is generated using the superpixels algorithm Adaptive-SLIC (SLICO) [35]. Figure 5.6 (c,top) shows an over segmentation in 50 superpixels, while 80 are selected for Fig. 5.6 (c,bottom). Next, those pixels belonging to a superpixel that intersects with a scribble are labelled with the same class as the scribble, as shown in Fig. 5.6 (d). In Fig. 5.6 (d,top), the black pixels represent the background, the red pixels indicate corrosion, and the green pixels denote unlabelled pixels. In Fig. 5.6 (d,bottom), black and green pixels denote the same as for the top mask, while the red pixels represent the tracking label and the blue pixels refer to the internal filter class.

5.2.1.2 Network Architecture

In this work, we adopt U-Net [94] as the base network architecture. As it is well known, U-Net evolves from the fully convolutional neural network concept and consists of a contracting path followed by an expansive path. Benefitting from its symmetrical architecture, U-Net has shown to be effective to detect small area lesions in biomedical images, though it has been shown to exhibit good performance in general for natural images even for small training sets.

Moreover, similarly to the Attention U-Net (AUN) [103], in our segmentation network, we have also embedded attention modules for improving the segmentation ability with small targets. The attention module has been widely used in Natural Language Processing (NLP) [228–231]. In these works, the attention module is applied to focus on the digit number that needs to be recognized in complex scenes. In the field of computer vision, a flexible trainable attention module was introduced in [103, 232–234] to improve the segmentation performance. In this work, AGs are integrated into the decoding part of U-Net to improve its ability to segment small targets.

For completeness, we include in Fig. 5.7 a schematic about the operation of the AG that we make use of in this work, which, in our case, implements Eq. (5.5) as described below:

$$\begin{aligned} (x_{i,c}^{l})' &= \alpha_i^l \, x_{i,c}^l \\ \alpha_i^l &= \sigma_2(W_\phi^T(\sigma_1(W_x^T x_i^l + W_g^T g_i + b_g)) + b_\phi) \end{aligned} \tag{5.5}$$

where the feature map $x_i^l \in \mathbb{R}^{F_l}$ is obtained at the output of layer l for pixel i, c denotes a channel in $x_{i,c}^l$, F_l is the number of feature maps at that layer, the gating vector g_i is used for each pixel i to determine focus regions and is such that $g_i \in \mathbb{R}^{F_l}$ (after up-sampling the input from the lower layer), $W_g \in \mathbb{R}^{F_l \times 1}$, $W_x \in \mathbb{R}^{F_l \times 1}$, and $W_\phi \in \mathbb{R}^{1 \times 1}$ are linear mappings, while $b_g \in \mathbb{R}$ and $b_\phi \in \mathbb{R}$ denote bias terms, σ_1 and σ_2 respectively represent the ReLU and the sigmoid activation functions, $\alpha_i^l \in [0, 1]$ are the resulting attention coefficients, and $\Phi_{\text{att}} = \{W_g, W_x, b_g; W_\phi, b_\phi\}$ is the set of parameters of the AG.

The attention coefficient $\alpha \in [0, 1]$ identifies salient image regions and prunes feature responses to preserve only the specific task relevant activation maps. Unlike the Squeeze-and-Excitation (SE) block as referred to in [232], which obtains attention weights in channels for filter selection, the AGs involved in our approach are used to calculate attention weights at the spatial level.

As shown in Fig. 5.8, one attention gate is fed by two input tensors, one from the

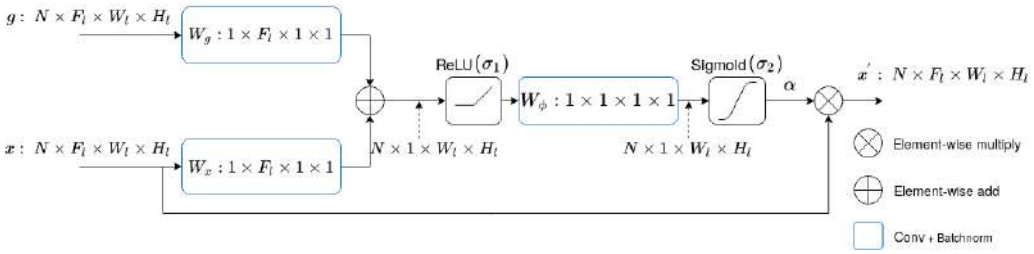


Figure 5.7: Schematic diagram of an Attention Gate (AG). N is the size of the mini-batch.

encoder and the other form the decoder, respectively g and x in Fig. 5.7. By integrating AG, spatial regions are selected by analyzing the semantic activation from x and the contextual information provided by the gating signal g which is collected from a coarser scale. The contextual information carried by the gating vector g is hence used to highlight salient features that are passed through the skip connections. In our case, g enters the AG after an up-sampling operation that makes g and x have compatible shapes (see Fig. 5.7).

Additionally, a sub-network in the AUN is integrated to calculate the centroid loss, as can be observed in Fig. 5.8. This sub-network is intended to predict class centroids on the basis of the scribbles that are available in the image, with the aim of improving the training of the main network from the possibly noisy pseudo-masks, and hence achieve a higher level of segmentation performance. As a result, the network shown in Fig. 5.8 processes two sorts of ground truth during training, scribble annotations Y_{scr} to train the sub-net for proper centroid predictions, and pseudo masks Y_{seg} for image segmentation. Apart from that, it yields two outputs, a set of centroids P_{cen} and the image segmentation results P_{seg} (while during inference only the segmentation output P_{seg} is relevant). Predicted cluster centroids are used to calculate the Centroid Loss term L_{cen} (described in Section 5.2.2.2) of the full loss function L, which comprises two more terms (as described in Section 5.2.2.3). The whole network is trained through a joint training strategy following an end-to-end learning model. During training, the optimization of L_{cen} induces updates in the main network weights via back-propagation that are intended to reach enhanced training and therefore produce better segmentations.

The sub-net is embedded in the intermediate part of the network. As shown in Fig. 5.8, the sub-net consists of three blocks, each of which consists of a fully connected layer, a batch normalization layer, and a ReLU activation function. The shape of P_{cen} is $C \times M$, where, N represents the batch size, C indicates the number of categories, and M denotes the dimension of centroid features. To compute the centroids, we consider using

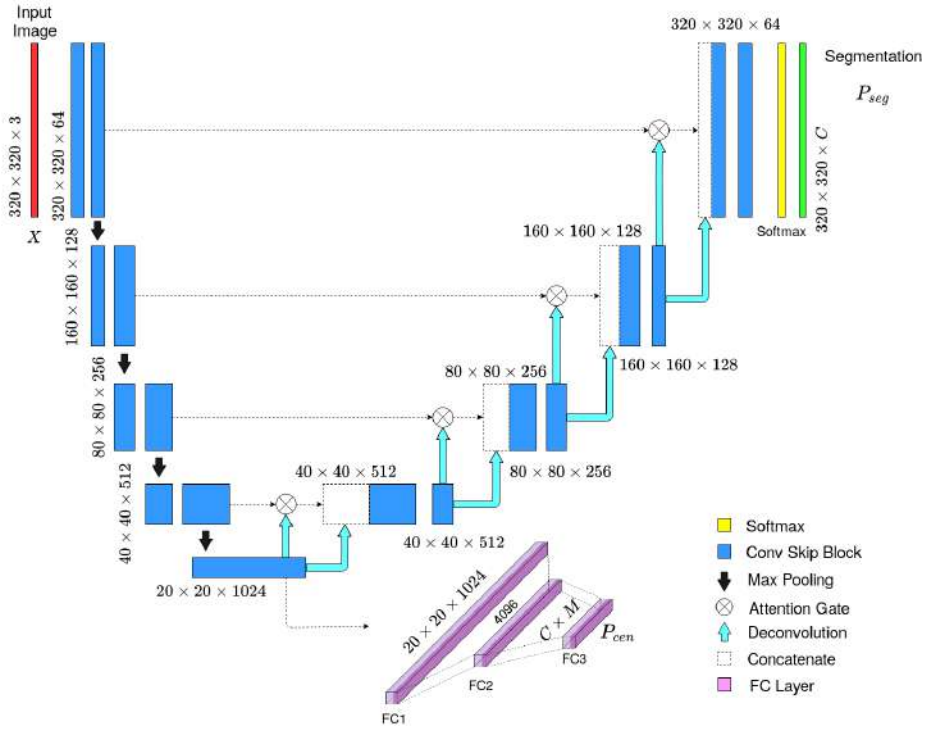


Figure 5.8: Block diagram of the Centroids AUN model. The size decreases gradually by a factor of 2 at each scale in the encoding part and increases by the same factor in the decoding part. In the latter, AGs are used to help the network focus on the areas of high-response in the feature maps. The *Conv Skip* block is the *skip connection* of ResNet [21]. The sub-network of the lower part of the diagram is intended to predict class centroids. In the drawing, C denotes the number of classes and M is the dimension of the class centroids.

the softmax outputs as features, hence comprises C components, though we foresee to combine them with K additional features from the classes which are incorporated externally to the operation of the network, and hence $M = C + K$. On the other hand, the shape of P_{seg} is $C \times H \times W$, where (W, H) is the size of the input image.

5.2.2 Loss Function

5.2.2.1 Partial Cross-Entropy Loss

Given a C-class problem and a training set Ω, comprising a subset Ω_L of labelled pixels and a subset Ω_U of unlabelled pixels, the Partial Cross-Entropy Loss L_{pCE}, widely used for WSSS, computes the Cross-Entropy only for labelled pixels $p \in \Omega_L$, ignoring $p \in \Omega_U$:

$$L_{\text{pCE}} = \sum_{c=1}^{C} \sum_{p \in \Omega_L^{(1)}} -y_{g(p),c} \ \log \ y_{s(p),c} \tag{5.6}$$

where $y_{g(p),c} \in \{0, 1\}$ and $y_{s(p),c} \in [0, 1]$ represent respectively the ground truth and the segmentation output. In our case, and for L_{pCE}, $\Omega_L^{(1)}$ is defined as the pixels labelled in the pseudo-masks (hence, pixels from superpixels not intersecting with any scribble belong to Ω_U and are not used by Eq. (5.6)). Hence, $y_{g(p),c}$ refers to the pseudo-masks, i.e., Y_{seg}, while $y_{s(p),c}$ is the prediction, i.e., P_{seg}, as supplied by the softmax final network layer.

5.2.2.2 Centroid Loss

As can be easily foreseen, when the network is trained using the pseudo-masks, the segmentation performance depends on how accurate the pseudo-masks are and hence on the quality of superpixels, i.e., how they adhere to object boundaries and avoid mixing classes. The Centroid Loss function is introduced in this section for the purpose of compensating a dependence of this kind and improving the quality of the segmentation output.

In more detail, we define the Centroid Loss term L_{cen} as another partial Cross-Entropy loss:

$$L_{\text{cen}} = \sum_{c=1}^{C} \sum_{p \in \Omega_L^{(2)}} -y^*_{g(p),c} \ \log \ y^*_{s(p),c} \tag{5.7}$$

defining in this case:

- $\Omega_L^{(2)}$ as the set of pixels coinciding with the scribbles,
- $y^*_{g(p),c}$ as the corresponding labelling, and

$$y^*_{s(p),c} = \frac{\exp(-d_{p,c})}{\sum_{c'=1}^{C} \exp(-d_{p,c'})} \tag{5.8}$$

$$d_{p,c} = \frac{||f_p - \mu_c||^2}{\sum_{c'=1}^{C} ||f_p - \mu_{c'}||^2} \tag{5.9}$$

where: (1) f_p is the feature vector associated to pixel p and (2) μ_c denotes the centroid predicted for class c, i.e., $\mu_c \in P_{\text{cen}}$. f_p is built from the section of the softmax layer of the main network corresponding to pixel p, though f_p can be extended with the incorporation of additional external features, as already mentioned. This link between L_{pCE} and L_{cen} through the softmax layer makes both terms decrease through the joint optimization, in the sense that for a reduction in L_{cen} to take place, and hence in the full loss L, also L_{pCE} has to decrease by better predicting the class of the pixels involved. The additional features that can be incorporated in f_p try to introduce further information from the classes, e.g. low-level features, such as colour or texture.

In practice, this loss term *pushes* pixel class predictions towards, ideally, a subset of the corners of the C-dimensional hypercube, in accordance with the scribbles, i.e., the available ground truth. Some similarity can be certainly established with the K-means algorithm. Briefly speaking, K-means iteratively calculates a set of centroids for the considered number of clusters/classes, and associates the samples to the closest cluster in feature space, thus minimizing the intra-class variance until convergence. Some DCNN-based clustering approaches reformulate K-means as a neural network optimizing the intra-class variance loss by means of a back-propagation-style scheme [235, 236]. Different from the latter, in this work, Eq. (5.7) calculates a set of prototype vectors, that are centroids of each category, using the true classes defined by the labelling of the scribbles $y^*_{g(p),c}$, and minimizes the distances from the scribbles samples to predicted centroids.

5.2.2.3 Full Loss Function

Since L_{pCE} applies only to pixels labelled in the pseudo-mask and L_{cen} is also restricted to a subset of image pixels, namely the pixels coinciding with the scribbles, we add a

third loss term in the form of a normalized MSE loss L_{mse} to behave as a regularization term that involves all pixels for which a class label must be predicted $\Omega_L^{(3)}$, i.e., the full image. This term calculates the normalized distances between the segmentation result for every pixel and its corresponding centroid:

$$L_{\text{mse}} = \frac{\sum\limits_{p \in \Omega_L^{(3)}} d_{p,c(p)}}{|\Omega_L^{(3)}|} \tag{5.10}$$

where $|\mathcal{A}|$ stands for the cardinality of set $\mathcal{A}$, and $d_{p,c(p)}$ is as defined by Eq. (5.9), with $c(p)$ as the class prediction for pixel p (and $\mu_{c(p)}$ the corresponding predicted centroid), taken from the softmax layer.

Finally, the complete loss function is given by:

$$L = L_{\text{pCE}} + \lambda_{\text{cen}} L_{\text{cen}} + \lambda_{\text{mse}} L_{\text{mse}} \tag{5.11}$$

where λ_{cen} and λ_{mse} are trade-off constants.

5.2.2.4 Loss Function Calculation Process

The calculation process of our approach is described in Fig. 5.9.

The input of our loss function includes four components, which are the scribble annotations, the predicted centroids, the additional input features, and the current prediction from the segmentation network. First, we calculate the Cross-Entropy loss associated to the pseudo-mask L_{pCE}.

Second, we determine the features f_p of each pixel $p \in \Omega_L^{(2)}$ using the corresponding output of the softmax layer, optionally combined with additional information, e.g. normalized RGB color features. Meanwhile, the predicted centroids $\mu_c, c \in 1:C$ from the sub-net are obtained. Next, the normalized Euclidean distance $d_{p,c}$ is calculated using the features f_p and the centroids μ_c. Then, in order to implement the Centroid loss, the output of the softmax activation function is employed and a one-hot format ground truth $y^*_{g(p),c}$ is assigned according to the category of the scribble. Finally, L_{cen} is calculated.

Third, the L_{mse} is computed, comprising an MSE loss to minimize the distance between the pixel features f_p, where $p \in \Omega_L^{(3)}$, and its corresponding centroid μ_c. The purpose of L_{mse} is to refine the segmentation results by optimizing the distance between the segmentation results and the corresponding centroids.

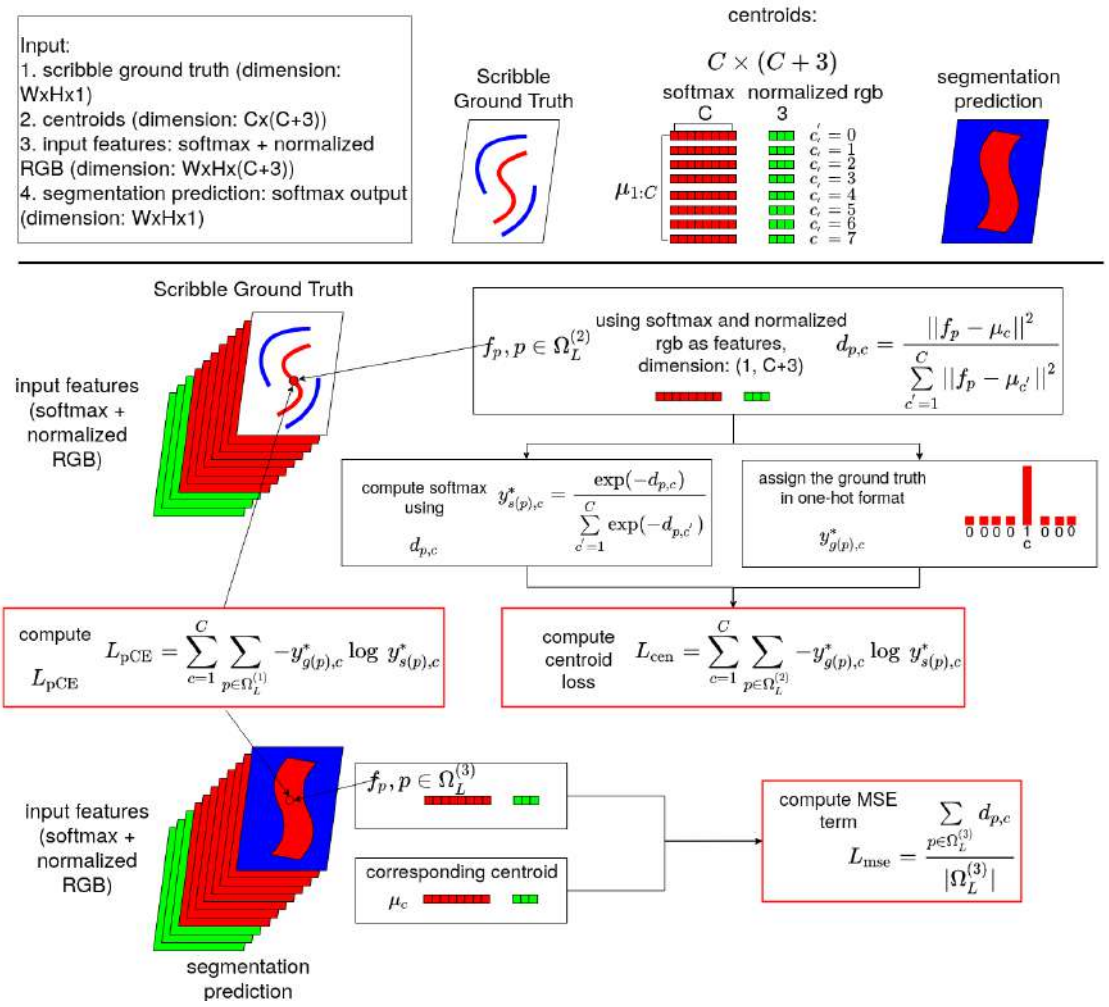


Figure 5.9: Forward calculation of the full loss function.

5.2.3 Experiments and Discussion

In this section, we report on the results obtained for the two application cases that constitute our benchmark. For a start, Section 5.2.3.1 describes the experimental setup. Next, in Section 5.2.3.2, we discuss about the feature space where the Centroid Loss is defined and its relationship with the weak annotations, while Section 5.2.3.3 evaluates the effect on the segmentation performance of several combinations of the terms of the loss function L, and Section 5.2.3.4 analyzes the impact of weak annotations and their propagation. Subsequently, our approach is compared against two previously proposed methods in Section 5.2.3.5. To finish, we address the final tuning and show segmentation

results, for qualitative evaluation purposes, for some images of both application cases in Section 5.2.3.6.

5.2.3.1 Experimental Setup

Dataset

The test set obtained from the quality control and visual inspection application cases is applied to evaluate the performance of our approach. The quality control dataset contains 484 images in total. As on the previous occasions, two thirds of the dataset are designated for training and the rest are intended for testing. For the visual inspection dataset, comprising of 241 images, the same strategy is adopted to split the dataset. Both datasets have been in turn augmented with rotations and scaled versions of the original images, together with random croppings, to increase the diversity of the training set. At last, as already explained, the ground truth for both datasets comprise scribbles and pseudo-masks (generated in accordance to the process described in Section 5.2.1.1).

By way of illustration, Fig. 5.10 shows, for the two application cases, some examples of weak annotations with different settings as for the width of the scribbles and the number of superpixels used for generating the pseudo-masks.

Evaluation metrics

In order to evaluate the performance of our approach, we consider using the mIOU, the mean Recall, the mean Precision and the F_1 score as the evaluation metrics.

In all experiments, we make use of fully supervised masks/ground truth for both datasets in order to be able to report accurate calculations on the segmentation performance. This ground truth has been manually generated only for this purpose, it has been used for training only when referring to the performance of the fully- and weakly-supervised approach, for comparison purposes between the full- and the weakly-supervised solutions.

To finish, in a number of experiments we also report on the quality of the pseudo-masks, so that the segmentation performance reported can be correctly appreciated. To this end, we calculate a weak mIOU (wmIOU) using the definition of mIOU between the psedo-mask and the fully-supervised mask involved, whose purpose is to value the quality of the pseudo-masks.

Implementation details and main settings

As on previous occasions, all experiments have been conducted using the Pytorch framework running in a PC fitted with an NVIDIA GeForce RTX 2080 Ti GPU, a 2.9GHz 12-core CPU with 32 GB RAM, and Ubuntu 64-bit. The batch size is 8 for all experiments and the size of the input image is 320×320 pixels, since this has turned out to be the best configuration for the aforementioned GPU.

As already mentioned, the AUN for semantic segmentation and the sub-net for centroid prediction are jointly trained following an end-to-end learning model. The network weights are initialized by means of the Kaiming method [225], and they are updated using a 10^{-4} learning rate for 200 epochs.

Best results have been obtained for the balance parameters λ_{cen} and λ_{mse} set to 1.

Overall view of the experiments

The experiments that are going to be discussed along the next sections consider different configurations for the different elements that are involved in the semantic segmentation approach. These configurations, which are enumerated in Table 5.2, involve:

- different widths of the scribble annotations used as ground truth, namely 2, 5, 10 and 20 pixels,
- different amounts of superpixels for generating the pseudo-masks, such as 30, 50 and 80,
- two ways of defining the feature space for the class centroids: from exclusively the softmax layer of AUN and combining those features with other features from the classes.

Notice that the first row of Table 5.2 refers to experiments where the loss function used for training is just the partial Cross-Entropy, as described in Eq. (5.6), and therefore can be taken as a lower baseline method. The upper baseline would correspond to the configuration using full masks and the cross-entropy loss L_{CE} for training, i.e., full supervised semantic segmentation, which can also be found in Table 5.2 as the last row.

Apart from the aforementioned variations, we also analyze the effect of several combinations of the loss function terms, as described in Eq. (5.11), defining three groups of experiments: Group 1 (G1), which indicates that the network is trained by means of only L_{pCE}, and also is used as the lower baseline; Group 2 (G2), which denotes that the network is trained by means of the combination comprising L_{pCE} and L_{cen}; and Group

3 (G3), for which the network is trained using the full loss function as described in Eq. (5.11).

Finally, we compare our segmentation approach with two other alternative approaches also aimed at solving the WSSS problem through a modified loss function. These loss functions are the Constrained-size Loss (L_{size}) [237] and the Seed, Expand, and Constrain (SEC) Loss (L_{sec}) [130]:

$$L_{\text{size}} = L_{\text{pCE}} + \lambda_{\text{size}} L_{\mathcal{C}(V_S)} \tag{5.12}$$

$$L_{\text{sec}} = L_{\text{seed}} + L_{\text{expand}} + L_{\text{constrain}} \tag{5.13}$$

As for λ_{size} in the $L_{\mathcal{C}(V_S)}$, it is set to 10^{-3}. Regarding L_{sec}, it comprises three terms, which are the seed loss L_{seed}, the expand loss L_{expand}, and the constrain loss $L_{\text{constrain}}$. In our case, we feed L_{seed} from the scribble annotations, while we adopt the same configuration as in the original work for L_{expand} and $L_{\text{constrain}}$.

5.2.3.2 About the Centroid loss feature space and the weak annotations

Generally speaking, color and texture features are widely used in image semantic segmentation. Different from color information, the features extracted from the top layers of CNN carry only semantic information. Due to down sampling operations such as pooling, these features on longer convey such detailed information as texture, color, shape, and spatial relationships. To address this problem, the experiments reported in this section consider the incorporation of color data from the classes into the calculation and minimization of the Centroid and the MSE loss functions, L_{cen} and L_{mse}. More specifically, we adopt a simple strategy by making use of normalized RGB features [1]:

$$\text{nRGB}_p = \frac{1}{R_p + G_p + B_p} (R_p, G_p, B_p) \tag{5.14}$$

As described in Section 5.2.1.2, the shape of P_{cen} is $C \times M$, where $M = C + K$, and K is the number of additional features from the classes that we incorporate into the network optimization problem. When only the features extracted from the last layer of the network are utilized, M equals the number of categories C, where C is 2 (corrosion and no-corrosion) in the visual inspection task, and C is 7 in the quality control task. After the integration with normalized RGB color space, M equals to $C + 3$. Of course, more sophisticated hand-crafted features can be incorporated into the process, though the idea of this experiment has been to make use of simple features.

[1] If $R_p = G_p = B_p = 0$, then $\text{nRGB}_p = (0, 0, 0)$.

Table 5.2: Labels for the experiments involving our WSSS approach, on the width of the scribbles and the number of superpixels employed for generating the pseudo-mask. Lower and upper baselines are also explicited. In this table, L_{CE} stands for the Cross-Entropy loss, L_{pCE} represents the partial Cross-Entropy loss, L_{cen} denotes the centroids loss, and L_{mse} indicates the MSE loss. Finally, SMX stands for softmax.

Configuration	**Label**	**Scribbles width**	**Num. superpixels**	**Centroid features**	**Supervision**	**Loss function**
lower baseline	E-SCR2	2	-	-	only scribbles	L_{pCE}
	E-SCR5	5	-	-		
	E-SCR10	10	-	-		
	E-SCR20	20	-	-		
	E-SCR20-SUP30	20	30	-	pseudo-masks	L_{pCE}
	E-SCR20-SUP50	20	50	-		
	E-SCR20-SUP80	20	80	-		
	E-SCR2-N	2	-	SMX	only scribbles	$L_{\text{pCE}} + L_{\text{cen}}$ $[+L_{\text{mse}}]$
	E-SCR2-NRGB	2	-	SMX & RGB		
	E-SCR5-N	5	-	SMX		
	E-SCR5-NRGB	5	-	SMX & RGB		
	E-SCR10-N	10	-	SMX		
	E-SCR10-NRGB	10	-	SMX & RGB		
	E-SCR20-N	20	-	SMX		
	E-SCR20-NRGB	20	-	SMX & RGB		
	E-SCR20-SUP30-N	20	30	SMX	pseudo-masks	$L_{\text{pCE}} + L_{\text{cen}}$ $[+L_{\text{mse}}]$
	E-SCR20-SUP30-NRGB	20	30	SMX & RGB		
	E-SCR20-SUP50-N	20	50	SMX		
	E-SCR20-SUP50-NRGB	20	50	SMX & RGB		
	E-SCR20-SUP80-N	20	80	SMX		
	E-SCR20-SUP80-NRGB	20	80	SMX & RGB		
upper baseline	E-FULL	-	-	-	full mask	L_{CE}

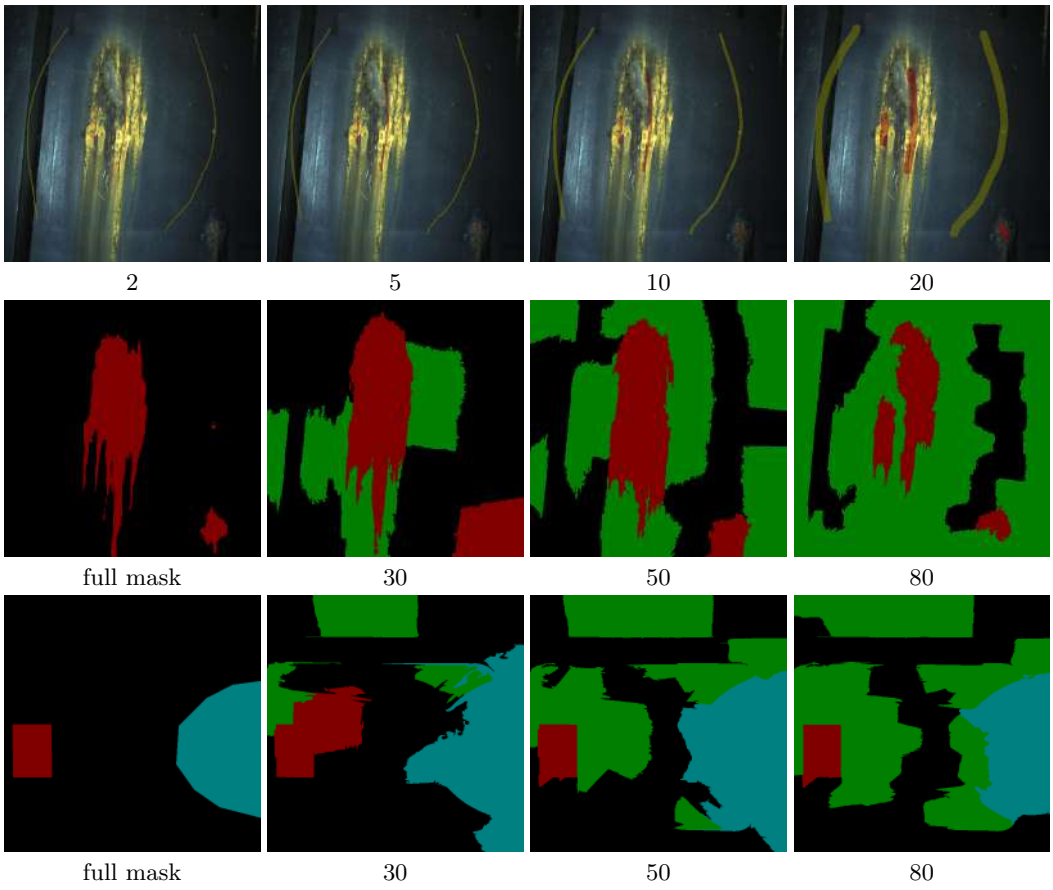


Figure 5.10: Examples of weak annotations and their propagation for the two application cases: (1st row) examples of scribble annotations of different widths, namely, from left to right, 2, 5, 10 and 20 pixels, for the visual inspection case; (2nd and 3rd rows) the leftmost image shows the fully supervised ground truth, while the remaining images are examples of pseudo-masks generated from 20-pixel scribbles and for different amounts of superpixels, namely 30, 50, and 80, for the two images of Fig. 5.6 and, hence, for the visual inspection and the quality control application cases. (The colour code is the same as for Fig. 5.6.)

Tables 5.3 and 5.4 evaluate the performance of our approach for different combinations of loss terms, for the two feature spaces of centroids outlined before, and for different configurations of weak annotations, such as the width for scribbles and the number of superpixels for pseudo-masks. Meanwhile, we also consider two possibilities of producing the final labelling: from the output of the segmentation network and from the clustering deriving from the predicted class centroids, i.e., label each pixel with the class label of the closest centroid; from now on, to simplify the discussion despite the language abuse, we will refer to the latter kind of output as that resulting from *clustering*. Finally, Table 5.3 only shows results for the visual inspection task because the scribbles alone have been shown not enough for obtaining proper segmentations in the quality control case.

From Table 5.3, it can be seen clearly from the table that the segmentation and clustering mIOU of experiments E-SCR*-NRGB is lower than the mIOU in experiments E-SCR*-N, and that there is a big gap in performance, which suggests that the RGB features actually do not contribute on improving the segmentation performance when scribble annotations alone are used as supervision information for the visual inspection dataset.

Besides, Table 5.4 lists the experimental results in terms of mIOU with pseudo-masks as segmentation ground truth. For both datasets, contrary to the results shown in Table 5.3, the performance of experiments E-SCR20-SUP*-NRGB is similar to that of experiments E-SCR20-SUP*-N. Additionally, the mIOU of some experiments where the integrated features (i.e., softmax and colour) are used is even higher than if only softmax features are used (e.g. E-SCR20-SUP80-N/NRGB, sixth row of Table 5.4).

From a general point of view, both Tables 5.3 and 5.4 show that our approach requires more labelled pixels to achieve higher segmentation performance when the integrated features are employed. In contrast, the use of softmax features only requires the scribble annotations to produce good performance for the visual inspection task. Moreover, our approach using softmax features obtains higher mIOU than using the integrated features in most of the experiments. Thus, softmax features are adopted to perform future experiments.

5.2.3.3 Effect of the loss function terms

In this section, the effect of L_{cen} and L_{mse} on the segmentation results is analyzed by means of experiments in groups G1, G2, and G3, as already discussed in Section 5.2.3.1.

Table 5.3 shows that the mIOU of experiments in G2 is significantly higher than that

Table 5.3: Segmentation performance for different centroid feature spaces and different widths of the scribble annotations. **N* denotes that only the SMX (*softmax*) features are used to compute L_{cen} and L_{mse}, while **NR* denotes that the feature space for centroids prediction comprises both SMX and RGB features. *Seg* denotes that the segmentation output comes directly from the segmentation network, while *Clu* denotes that the segmentation output is obtained from clustering.

Task	Experiments	wmIOU	L_{pCE}	L_{cen}	L_{mse}	mIOU (Seg)	mIOU (Seg,*N)	mIOU (Seg,*NR)	mIOU (Clu,*N)	mIOU (Clu,*NR)
Visual Inspection	E-SCR2	0.2721	✓			0.3733	-	-	-	-
	E-SCR5	0.2902	✓			0.4621	-	-	-	-
	E-SCR10	0.3074	✓			0.4711	-	-	-	-
	E-SCR20	0.3233	✓			0.5286	-	-	-	-
	E-SCR2-*	0.2721	✓	✓		-	0.6851	0.4729	0.6758	0.3889
	E-SCR5-*	0.2902	✓	✓		-	0.6798	0.4989	0.6706	0.6020
	E-SCR10-*	0.3074	✓	✓		-	0.6992	0.5130	0.6710	0.6267
	E-SCR20-*	0.3233	✓	✓		-	0.6852	0.5562	0.6741	0.6164
	E-SCR2-*	0.2721	✓	✓	✓	-	0.6995	0.4724	0.6828	0.3274
	E-SCR5-*	0.2902	✓	✓	✓	-	0.7134	0.4772	0.7001	0.2982
	E-SCR10-*	0.3074	✓	✓	✓	-	0.7047	0.4796	0.6817	0.3130
	E-SCR20-*	0.3233	✓	✓	✓	-	0.6904	0.5075	0.6894	0.6187

Table 5.4: Segmentation performance for different centroid feature spaces and for different amounts of superpixels to generate the pseudo-masks. **N* denotes that only the SMX (*softmax*) features are used to compute L_{cen} and L_{mse}, while **NR* denotes that the feature space comprises both SMX and RGB features. *Seg* denotes that the segmentation output comes directly from the segmentation network, while *Clu* denotes that the segmentation output is obtained from clustering.

Task	Experiments	wmIOU	L_{pCE}	L_{cen}	L_{mse}	mIOU (Seg)	mIOU (Seg,*N)	mIOU (Seg,*NR)	mIOU (Clu,*N)	mIOU (Clu,*NR)
Visual Inspection	E-SCR20-SUP30	0.6272	✓			0.6613	-	-	-	-
	E-SCR20-SUP50	0.6431	✓			0.7133	-	-	-	-
	E-SCR20-SUP80	0.6311	✓			0.7017	-	-	-	-
	E-SCR20-SUP30-*	0.6272	✓	✓		-	0.6848	0.6847	0.7081	0.6859
	E-SCR20-SUP50-*	0.6431	✓	✓		-	0.7447	0.7368	0.7372	0.7136
	E-SCR20-SUP80-*	0.6311	✓	✓		-	0.7242	0.7355	0.7127	0.6761
	E-SCR20-SUP30-*	0.6272	✓	✓	✓	-	0.6919	0.7071	0.6987	0.7076
	E-SCR20-SUP50-*	0.6431	✓	✓	✓	-	0.7542	0.7133	0.7491	0.7294
	E-SCR20-SUP80-*	0.6311	✓	✓	✓	-	0.7294	0.7246	0.7268	0.7118
Quality Control	E-SCR20-SUP30	0.4710	✓			0.5419	-	-	-	-
	E-SCR20-SUP50	0.5133	✓			0.6483	-	-	-	-
	E-SCR20-SUP80	0.5888	✓			0.7015	-	-	-	-
	E-SCR20-SUP30-*	0.4710	✓	✓		-	0.6882	0.6889	0.6142	0.6062
	E-SCR20-SUP50-*	0.5133	✓	✓		-	0.7236	0.7203	0.6644	0.6480
	E-SCR20-SUP80-*	0.5888	✓	✓		-	0.7594	0.7337	0.6768	0.6451
	E-SCR20-SUP30-*	0.4710	✓	✓	✓	-	0.7030	0.6237	0.5910	0.6077
	E-SCR20-SUP50-*	0.5133	✓	✓	✓	-	0.7291	0.7046	0.6605	0.6372
	E-SCR20-SUP80-*	0.5888	✓	✓	✓	-	0.7679	0.7409	0.6687	0.6780

of experiments in G1, where the gap of the maximum mIOU in G1 and G2 is 0.2066. As for the segmentation performance for in G3 experiments, it is systematically above that of G2 experiments for the same width of the scribble annotations and if centroids are built only from the softmax features. When the colour features are incorporated, segmentation performance decreases from G2 to G3.

Furthermore, Table 5.4 shows that the segmentation performance improves by adding L_{cen}, e.g., G2, in comparison with experiments in G1, while the performance for the G3 experiments is superior to that of G2 experiments, and this is observed for both tasks. Therefore, by incorporation L_{cen} and L_{mse} terms into the loss function, the segmentation mIOU is gradually improved for our two tasks.

Regarding the clustering performance, the mIOU of experiments in G3 is also higher than that of experiments in G2. In addition, it can be found out in Table 5.3 and Table 5.4 that the mIOU of clustering for certain G2 experiments (E-SCR20-SUP30-N for both datasets and E-SCR20-SUP80-N for the quality control dataset) is slightly higher than for G3 experiments, while the segmentation mIOU for G2 is lower than for G3. It is suggested that L_{mse}, in some experiments, makes the segmentation quality from clustering deteriorate.

Overall, the incorporation of L_{cen} and L_{mse} improves segmentation performance for both tasks, and labelling from segmentation turns out to be superior to that deriving from class centroids.

5.2.3.4 Impact of weak annotations and their propagation

In this section, we evaluate our approach under different weak annotations and their propagation, and discuss on their impact on segmentation performance for both tasks. To this end, we plot in Fig. 5.11 the mIOU (complementarily to Tables 5.3 and 5.4), and also the recall and the precision values resulting after the supervision of different sorts of weak annotations for the two tasks. As can be observed in these plots, the curves corresponding to the G3 experiments are above than those for G1 and G2 groups for all the performance metrics considered.

Regarding the visual inspection task, Fig. 5.11(a) shows that the mIOU values for the G2 and G3 experiments are above those for G1 (the lower baseline), which follows a similar shape as the wmIOU values, while the curves from G2 and G3 groups keep a relative stable level for the different sorts of weak annotations. As for the quality control task, the curves of mIOU values for all groups are similar among all groups and similar to the wmIOU values, as shown in Fig. 5.11(d). Obviously, the curves in Fig. 5.11

(a) and (d) show that the scribbles are enough for describing the classes in the case of the visual inspection task, which is a binary classification problem, while this is not true for the quality control case, a multi-class problem, and this makes necessary resort to the pseudo-masks (G2 and G3 groups) to achieve a higher performance. Globally, our approach obtains higher mIOU values than the that of G1, and it corroborates the improvement of the Centroid loss on the segmentation performance, despite its ultimate contribution to the segmentation performance is also affected by the quality of the weak annotations involved, i.e., the pseudo-masks deriving from scribbles and superpixels for the cases of the G2 and G3 groups.

Additionally, observing the precision curves corresponding to Fig. 5.11 (c) and (f), the precision curves of weak annotations show a sharp decline when the weak annotations shift from scribbles to pseudo masks. As seen in the pseudo-masks in the second and third rows of Fig. 5.10, when the number of superpixels is low, e.g., 30, the pseudo-masks contain lots of incorrectly labelled pixels, while the number of labelled pixels is significantly higher than that of the scribble annotations. Thus, the precision curves of weak annotations show a downward trend while the recall curves exhibit an upward trend in Fig. 5.11 (b) and (e). On the other side, we can see that, in general, precision and recall values are higher for the G3 group than for the G2 group, and both curves are above those for the G1 group, and this behavior replicates for the two tasks. Finally, the output from clustering does not clearly lead to a different performance, better or worse, over the alternative outcome from the segmentation network, showing that clustering is less appropriate for the quality control task from the point of view of the recall metric.

From a global perspective, the experiments we have performed reveal that (a) the segmentation quality benefits from the use of pseudo-masks, (b) our approach can obtain a better segmentation performance than the lower baseline, i.e., using of exclusively scribbles, (c) despite the incorrectly labelled pixels contained in the pseudo-masks, our approach can provide an accurate and stable segmentation result for the two tasks.

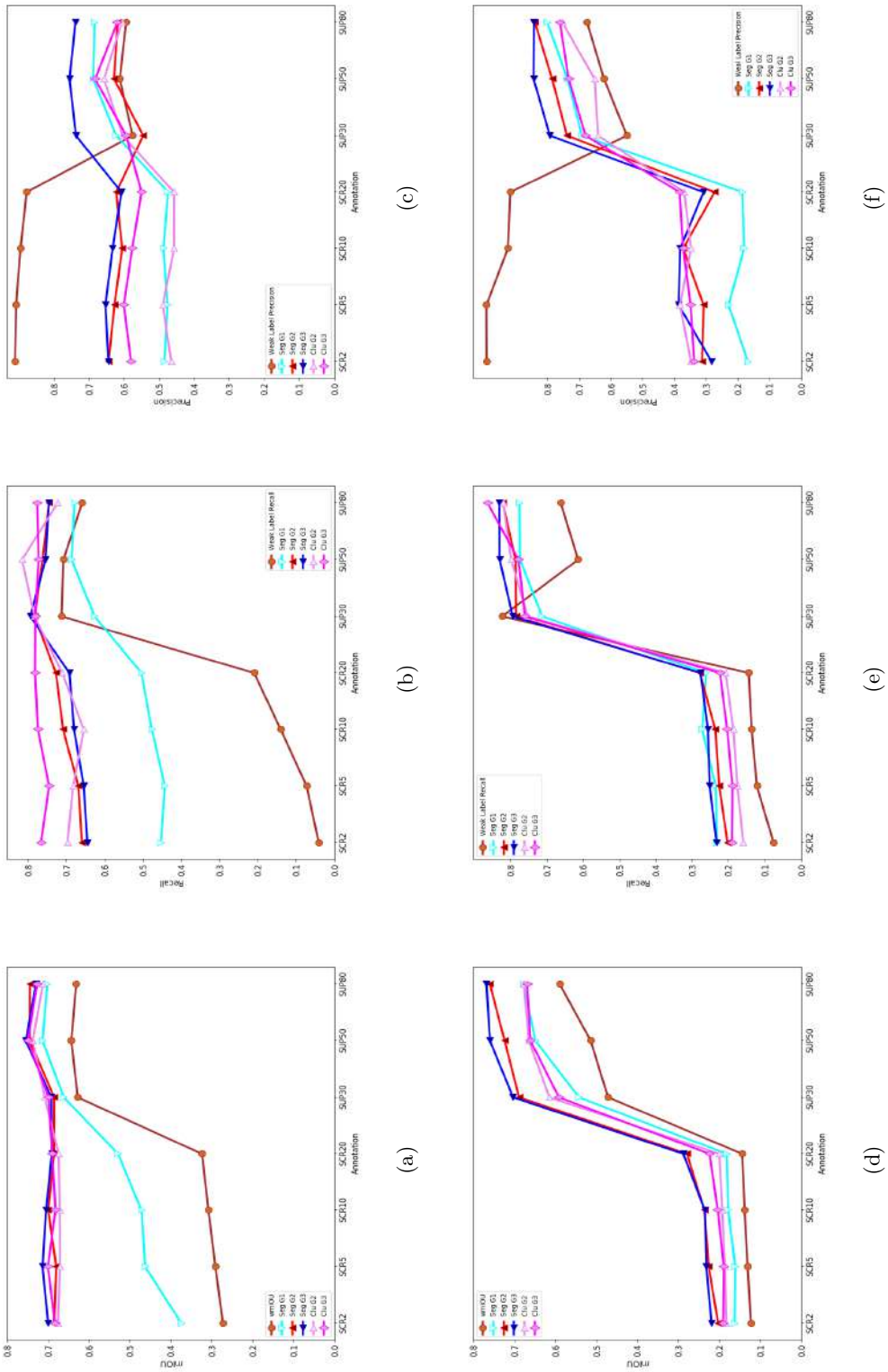


Figure 5.11: Performance metrics for our approach under different sorts of weak annotations. The first row plots are for the visual inspection task, while those of the second row are for the quality control task. In both rows, from left to right, the three figures plot respectively the mIOU, the mean Recall, and the mean Precision. SUP30, SUP50 and SUP80 labels correspond to the use of 20 pixel-wide scribbles.

5.2.3.5 Comparison with other loss functions

In Table 5.5, we compare the segmentation performance of our approach for the two tasks with that resulting from the use of the Constrained-size Loss L_{size} [237] and the SEC Loss L_{sec} [130] for different variations of weak annotations.

As for the visual inspection task, the network trained with L_{sec} is clearly inferior to the one resulting for our loss function, and the same can be said for L_{size}, although, in this case, the performance gap is shorter, even negligible when the width of the scribbles is of 20 pixels. When the pseudo masks are applied, the performance of the three approaches shows no significant difference. However, our approach obtains the highest mIOU (0.7542). As for the quality control task, the mIOU of our approach is higher than that of the other two approaches.

Summing up, we can conclude that the loss function proposed in Eq. (5.11) outperforms both the Constrained-size Loss L_{size} and the SEC Loss L_{sec} on the visual inspection and the quality control tasks.

5.2.3.6 Final tuning and results

As discussed in Sections 5.2.3.3 and 5.2.3.4, the network trained by means of our approach, which in particular comprises the full loss function described in Eq. (5.11), obtains the best segmentation performance against other approaches and for the two tasks considered in this work. In order to improve the segmentation performance, the dense CRF is added as a post-processing stage of the outcome of the network. The final segmentation and clustering performance metrics of G3 experiments are shown in Table 5.6, including mean recall, mean precision, mIOU and wmIOU. The performance of the upper baseline method (E-FULL) is also reported.

Regarding the visual inspection task, the experiment E-SCR20-SUP50-N leads to the best segmentation mIOU (0.7542), as shown in Table 5.6. After the incorporation of the dense CRF, the mIOU reaches 0.7859, while the performance gap with the E-FULL is 0.0474. Referring to the visual inspection task, the case E-SCR20-SUP30-N obtains the highest recall value (0.7937), although its precision (0.7081) and F_1 score (0.7485) are not the highest. In other words, the segmentation result of E-SCR20-SUP30-N contains more incorrect predictions than case E-SCR20-SUP50-N. Consequently, the case E-SCR20-SUP50-N leads to the best performance among the other cases, with a slightly increase after the CRF post-processing stage. The performance of clustering is not far in quality to these values, but it attains the a lower mIOU (0.7491) and F_1 score (0.7250).

Table 5.5: Comparison of different loss functions for both the visual inspection and the quality control tasks. mIOU values are provided. Best performance is highlighted in bold.

Task	Weak Annotation	L_{size} [237]	L_{sec} [130]	Ours
Visual Inspection	E-SCR2-N	0.6098	0.4366	**0.6995**
	E-SCR5-N	0.6537	0.4372	**0.7134**
	E-SCR10-N	0.6754	0.5486	**0.7047**
	E-SCR20-N	**0.6909**	0.5624	0.6904
	E-SCR20-SUP30-N	**0.7068**	0.6397	0.6919
	E-SCR20-SUP50-N	0.6769	0.7428	**0.7542**
	E-SCR20-SUP80-N	0.7107	0.6546	**0.7294**
Quality Control	E-SCR20-SUP30-N	0.4724	0.5808	**0.7030**
	E-SCR20-SUP50-N	0.4985	0.6262	**0.7291**
	E-SCR20-SUP80-N	0.5051	0.6918	**0.7679**

As for the quality control task, the mIOU (0.7679) of E-SCR20-SUP80-N is the highest among all cases, with the second best F_1 value (0.8350). In this task, the case E-SCR20-SUP50-N obtains the highest precision and F_1 score, which are 0.8439 and 0.8368 respectively, though at a very short distance to the E-SCR20-SUP80-N case. After dense CRF, the corresponding mIOU is 0.7143. The most appropriate configuration, which seems to be 20-pixel scribbles and 80 superpixels for psedudo-mask generation, attains the highest mIOU (0.7707). The gap in this case with regard to full supervision is 0.0897. Similarly to the visual inspection task, the clustering performance in this task is inferior to the segmentation performance, which attains the lower mIOU (0.7491) and F_1 score (0.7250) values.

From a general point of view, the results obtained indicate that 20-pixel scribbles, together with a rather higher number of superpixels, making the pseudo-masks better adhere to the object boundary, are the best options for both tasks. Compared with the lower baseline (G1 group), our approach involving the L_{cen} and L_{mse} attains higher segmentation performance, with a slight decrease regarding full supervision. On the other hand, the segmentation performance is better than the results deriving from clustering for our two tasks.

Figure 5.12 shows examples of segmentation results for the visual inspection task. As can be observed, the segmentations resulting from our approach are very similar to those from the upper baseline (E-FULL). Moreover, the results from clustering contain more incorrect predictions, i.e., false positives. As for the quality control task, the same conclusion can be drawn from Fig. 5.13.

After all these experiments, it is clear that the Centroid loss makes it possible to train properly the segmentation network using a small number of labelled pixels. Although the performance of the FSSS approach is inferior to that of a fully supervised approach, it is believed that this is a reasonable gap for both tasks, given the challenges arising from the use of weak annotations.

5.2.4 Conclusions

In this section, a weakly-supervised semantic segmentation approach incorporating a variation of AUN as a segmentation network has been proposed, and its performance has been evaluated on the two application tasks that are considered in this dissertation. The loss function employed comprises three terms, which are a partial Cross-Entropy term, the so-called Centroid Loss and a regularization term based on the mean square error. The whole loss function is jointly optimized within an end-to-end learning model. As has been reported in the experimental results section, our approach can achieve a competitive performance with a significantly lower cost of image labelling. Despite the use of weak annotations of varying quality, the proposed WSSS approach leads to good segmentation performance and reduces the negative impact of inaccurare labels.

On the other hand, the performance gap between the weakly-supervised approach and the corresponding fully-supervised approach has shown to be rather reduced. Regarding the mIOU, it is around 0.08-0.1 below in absolute terms for the WSSS approach. As for the precision and recall, the difference between both approaches is almost zero for the quality control task, and around 0.10-0.15 below for the visual inspection task. As for the latter, it has been shown feasible to use scribbles alone and achieve reasonable segmentation results. However, only scribble annotations have proved to be not enough for a multi-category problem such as the quality control task. As a solution, pseudo-masks deriving from the combination of scribbles and superpixels have been employed as approximate ground truth in both tasks, with the aforementioned performance.

5.3 Overall Conclusions on Semantic Segmentation

In this chapter, we have tackled the problem of image semantic segmentation with full and weak supervision in order to obtain finer and more accurate detections with regard to bounding box-based approaches. After approaching the fully-supervised semantic segmentation, we have considered how to reduce the cost of image labelling which is traditionally associated to full supervision, and have developed a weakly-supervised so-

Table 5.6: Segmentation results for the full loss function (G3). *Seg* denotes that the segmentation output comes directly from the segmentation network, while *Clu* denotes that the segmentation output is obtained from clustering. *CRF refers to the performance (mIOU) after dense CRF post-processing. Best performance for WSSS is highlighted in bold for each task.

Dataset	Experiments	wmIOU	mIOU (seg)	mRec (seg)	mPrec (seg)	F_1 (seg)	mIOU (clu)	mRec (clu)	mPrec (clu)	F_1 (clu)	*CRF (seg)
Inspection	E-SCR2-N	0.2721	0.6995	0.6447	0.6452	0.6449	0.6828	0.7663	0.5803	0.6605	0.7068
	E-SCR5-N	0.2902	0.7134	0.6539	0.6542	0.6540	0.7001	0.7447	0.6015	0.6655	0.7212
	E-SCR10-N	0.3074	0.7047	0.6797	0.6332	0.6556	0.6817	0.7741	0.5772	0.6613	0.7241
	E-SCR20-N	0.3233	0.6904	0.6917	0.6081	0.6472	0.6894	**0.7816**	0.5507	0.6461	0.7172
	E-SCR20-SUP30-N	0.6272	0.6919	**0.7937**	0.7081	0.6987	0.7485	0.7806	0.5946	0.6750	0.7489
	E-SCR20-SUP50-N	0.6431	**0.7542**	0.7543	**0.7567**	**0.7491**	**0.7555**	0.7725	**0.6830**	**0.7250**	**0.7859**
	E-SCR20-SUP80-N	0.6311	0.7294	0.7452	0.7397	0.7268	0.7424	0.7758	0.6200	0.6892	0.7693
	E-FULL	1.0	0.8333	0.8537	0.9119	0.8818	-	-	-	-	0.8218
Quality Control	E-SCR20-SUP30-N	0.4710	0.7030	0.7937	0.7924	0.7930	0.5910	0.7600	0.6798	0.7177	0.7142
	E-SCR20-SUP50-N	0.5133	0.7291	0.8298	**0.8439**	**0.8368**	0.6605	0.7777	0.7332	0.7548	0.7143
	E-SCR20-SUP80-N	0.5888	**0.7679**	**0.8303**	0.8398	0.8350	**0.6687**	**0.8630**	**0.7606**	**0.8086**	**0.7707**
	E-FULL	1.0	0.8604	0.8058	0.8432	0.8241	-	-	-	-	0.8459

lution.

Regarding fully-supervised semantic segmentation, within the framework of a Fully Convolutional Network, this work has focused on exploring the effects of different loss functions, namely the Dice loss, the Focal loss and the softmax cross entropy loss, analyzing these effects separately for the two application tasks considered. The experimental results have shown that the Dice loss leads to the best segmentation performance for both tasks.

As for weakly-supervised segmentation, we have proposed an approach using scribbles as ground truth, making use of the idea of Attention U-Net (AUN) for the segmentation network and optimizing a specially designed loss functions comprising a partial Cross-Entropy loss, a centroid loss, and a normalized MSE-based term. The approach proposed has been reported to be able to achieve satisfactory performance with a significantly lower effort in labelling work.

As expected, the performance of full supervision is above that achieved by weak supervision. Nevertheless, the experimental results have shown that the performance gap between the weakly-supervised approach and the fully-supervised approach has shown to be rather reduced for the two tasks considered.

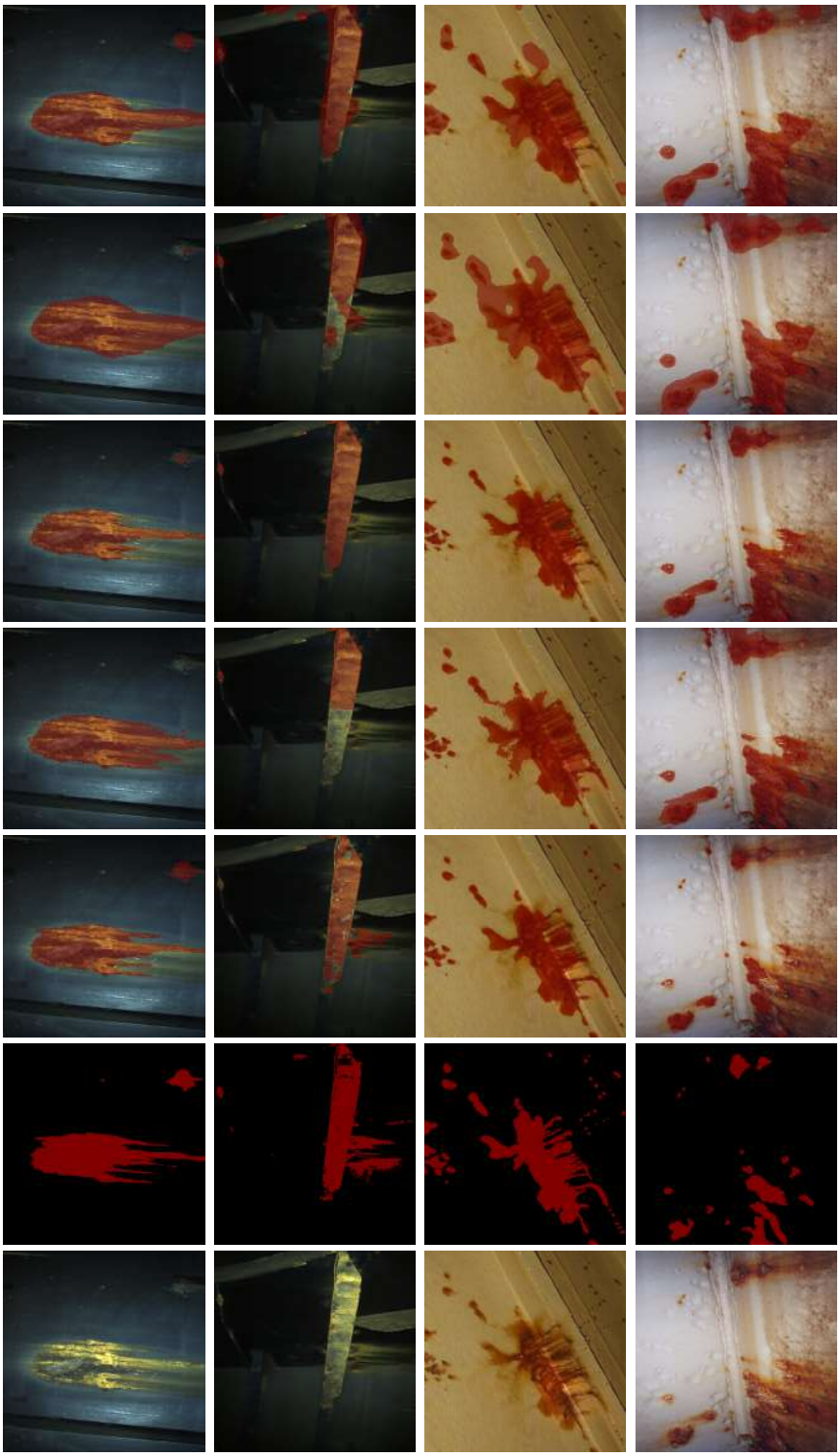


Figure 5.12: Examples of segmentation results for the visual inspection task: (1st column) original images, (2nd column) full mask, (3rd column) results of the fully supervised approach, (4th & 5th columns) segmentation output after dense CRF from E-SCR20-N and E-SCR20-SUP50-N, (6th & 7th columns) segmentation output of clustering results for the same configurations.

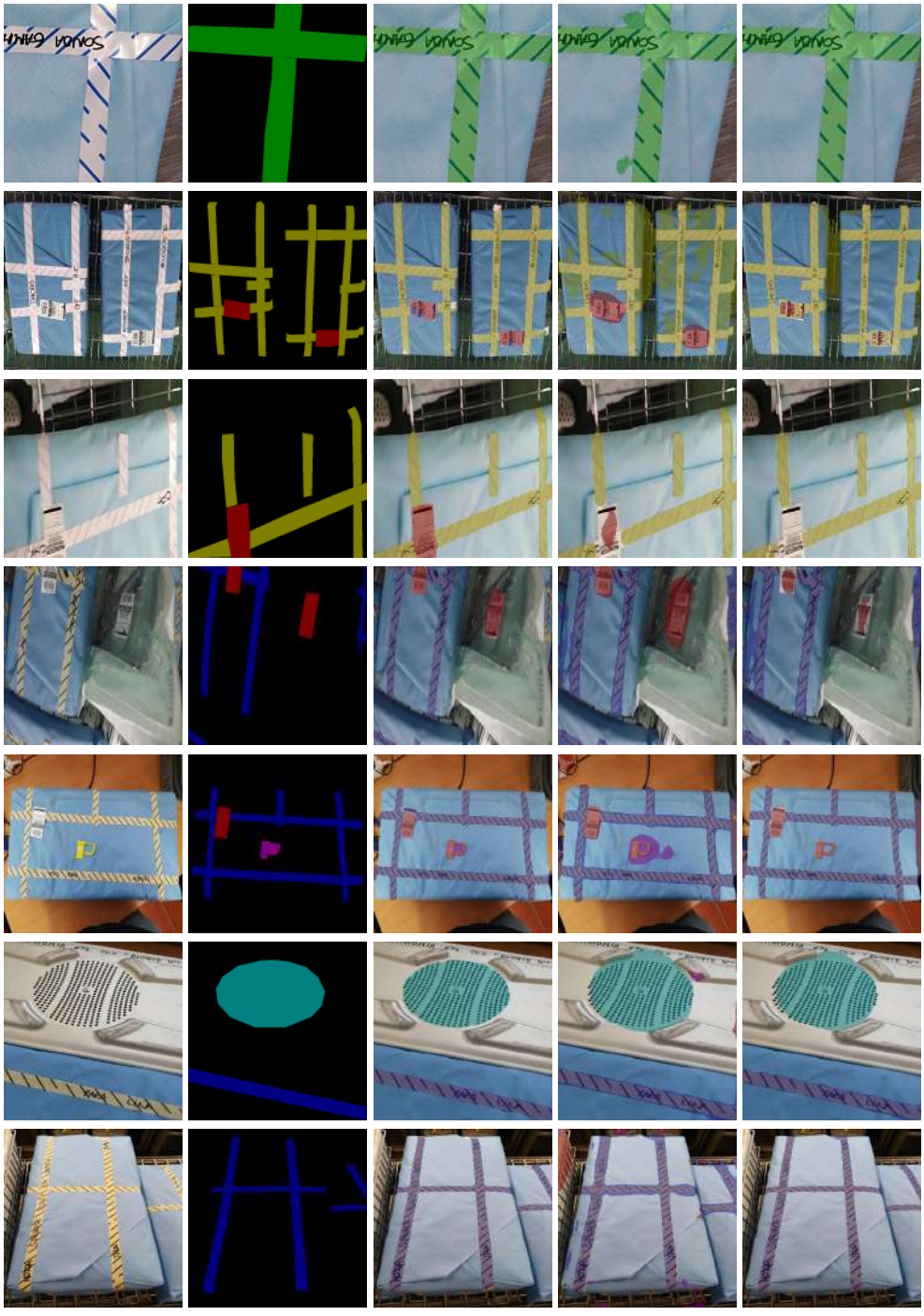


Figure 5.13: Examples of segmentation results for the quality control task: (1st column) original images, (2nd column) full mask, (3rd column) results of the fully supervised approach, (4th and 5th columns) are segmentation outputs after dense CRF from the E-SCR20-SUP80-N and clustering outputs from the E-SCR20-SUP80-N, respectively.

On the Combination of Semantic Segmentation and Bounding Boxes Detection

After all the experiments performed and reported along the previous chapters, we have found that the segmentation and BBox detection approaches can be complementary in terms of detection performance. On the one hand, BBox-based detectors are trained to detect full objects and consequently capture better the essence of the objects to detect because of the global view that is acquired as part of the training process. This approach is, however, typically unable to deal with small targets. On the other hand, pixel-level classification provides more accurate and finer detections, to deal with small targets, but losing the aforementioned global view. These are the reasons why, in this chapter, we attempt and test some combination strategies.

Firstly, we illustrate the rationable behind the combination of the two approaches in Section 6.1. Secondly, we elaborate on some simple combination strategies in Section 6.2. More precisely, we consider two basic strategies: (a) use the BBox detection results to improve the segmentation performance, namely the *BBox-Seg* strategy, and (b) use the image segmentation results to improve BBox detection, namely the *Seg-BBox* strategy. In Section 6.3, we evaluate the different combination strategies and discuss on the results obtained. At last, an overview of this chapter is given in Section 6.4.

6.1 Some Illustrative Cases

To visualize the complementarity of the detection results of the two approaches, we select the FPSSD model developed in Chapter 4 as the BBox detector and the Attention U-Net (AUN) model described in Chapter 5 as the segmentation approach for both tasks. In Fig. 6.1, we can observe that, on some occasions, the BBoxes predicted do not contain the whole target but a part of it, and this happens for both tasks; on other occasions, the

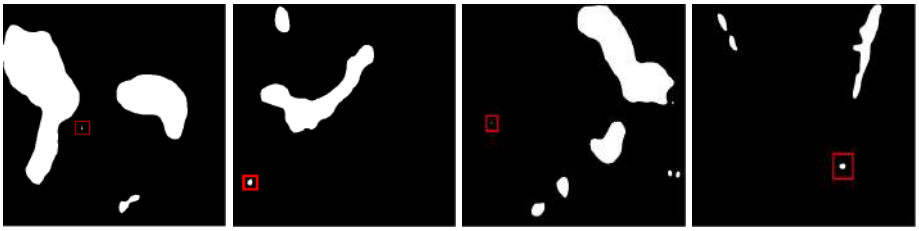
Algorithm 2: BBox-Seg – Intersection of bounding box detection and segmentation results.

Input:
Segmentation results P_{seg}
BBox detection results P_{det}
C is the number of categories
Initialization:
Define IM as an empty mask to store the resulting segmentation mask
begin
  One-hot encode P_{seg} into P_{seg_oh}
  Transform P_{det} to mask and one-hot encoding format P_{det_oh}
  for *each* $c = 1,2, \cdots, C$ **do**
    $\text{IM}[c] = P_{seg\_oh}[c] \cap P_{det\_oh}[c]$ ▷ *bitwise AND operation*
  end
  One-hot decode IM.
end
Output:
Pixel-level intersection IM between P_{seg} and P_{det}.

BBox represents a large fraction of the image in order to ensure the full object lies inside it and hence contain a lot of background. On the other side, the segmentation output can comprise some small, spurious areas that do not correspond to any of the targets under consideration because of the lack of global view for those targets (see e.g. the red rectangles in the segmentation results in Fig. 6.1), being false positives. As already said, in this chapter, we address the development of some simple strategies that combine BBox detection with semantic segmentation to mitigate the aforementioned situations.

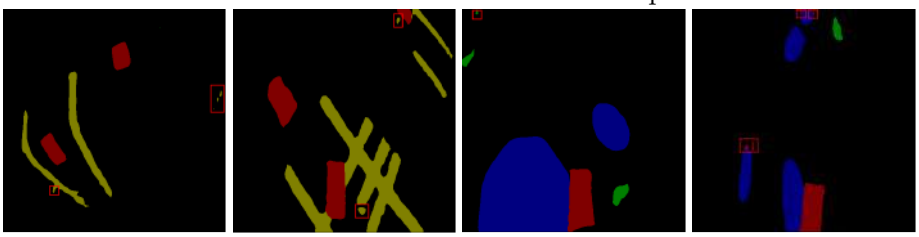
6.2 Combination Strategies

This section details the different combination strategies that have been considered and assessed. First of all, we develop the *BBox-Seg* combination strategies in Section 6.2.1. In general terms, we perform three different operations to combine the two approaches: intersection of results, union of results, and intersection of results combined with a threshold over the resulting regions' area. The *Seg-BBox* strategy is addressed in Section 6.2.2.

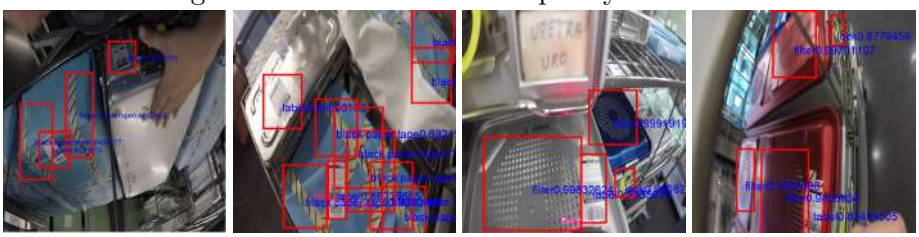


Segmentation results for the visual inspection task

BBox detection results for the visual inspection task



Segmentation results for the quality control task



BBox detection results for the quality control task

Figure 6.1: Some examples of segmentation and BBox detection results for the two tasks. The input images for the first and third rows can be found in the second and fourth rows.

6.2.1 BBox-Seg Strategies

This section describes several *BBox-Seg* combination strategies which, given a set of predicted bounding boxes, try to enhance the segmentation output produced by a semantic segmentation approach. Consequently, the algorithms which are developed in this section output a segmentation mask, resulting from the corresponding combination strategy.

As a first combination strategy, we consider using the intersection of two detection results, as shown in Algorithm 2. Firstly, the BBox detection results are transformed into mask format by filling the whole bounding box with the category index. Secondly, both the segmentation results P_{seg} and the BBox detection mask P_{det} are encoded into one-hot format as P_{det_oh} and P_{seg_oh}, what ensures that each channel in the one-hot format only contains 0 and 1 values. In this way, the intersection can be obtained by means of a bitwise AND operation between P_{det_oh} and P_{seg_oh} for every channel. A final one-hot decoding step gives rise to the resulting intersection mask.

Secondly, we also consider the union of two detection results, as shown in Algorithm 3. The whole calculation is relatively similar to the intersection approach, although in this case one must take care of performing the union operation only for intersecting detections.

Finally, because the intersection strategy can miss certain detection results, whereas the union strategy can contain plenty of background, we suggest a third strategy that sets a threshold to discard certain intersection areas between the segmentation and the BBox detection results. More precisely, we define a threshold γ that means a minimum percentage of intersection area with respect to the whole image. Therefore, when the intersection area divided by the image size is higher than γ, the segmentation area is preserved; otherwise, the segmentation area is discarded. The whole procedure is formally stated in pseudo-code as Algorithm 4.

6.2.2 Seg-BBox Strategy

As already commented, a BBox detection approach can miss some small area targets due to the limitations of the underlying CNN which, otherwise, could be effectively captured by a semantic segmentation approach. The *Seg-BBox* combination strategy that is described in this section intends to work out this shortcoming by incorporating additional detections from the segmentation output. The result of *Seg-BBox* is hence an enhanced list of bounding boxes, unlike the *BBox-Seg* strategies which output an improved segmentation mask.

Algorithm 3: BBox-Seg – Union of bounding box detection and segmentation results.

Input:
Segmentation results P_{seg}
BBox detection results P_{det}
C is the number of categories
Initialization:
Define UM as an empty mask to store the resulting segmentation mask
begin
  One-hot encode P_{seg} into P_{seg_oh}
  Transform P_{det} into a mask in one-hot encoding format P_{det_oh}
  for *each bounding box* $p_{det\_oh} \in P_{det\_oh}$ **do**
    $c =$ category of p_{det_oh}
    $p_{det,c} = p_{det\_oh}[c]$
    for *each connected component* $p_{seg,c} \in P_{seg\_oh}[c]$ **do**
      if $p_{seg,c} \cap p_{det,c} \neq \emptyset$ **then**
        $\text{UM}[c] = \text{UM}[c] \cup (p_{seg,c} \cup p_{det,c})$ ▷ *bitwise OR operations*
      end
    end
  end
  One-hot decode UM ▷ *use priority to decode pixels with multiple class detections*
end
Output:
Pixel-level union UM between P_{seg} and P_{det}

The combination process is as follows: (1) each bounding box resulting from the BBox detection approach is appended to the list of resulting bounding boxes; (2) image regions of the segmentation output that do not overlap sufficiently with any bounding box of the same class are incorporated into the list of resulting bounding boxes; (3) to implement the previous step, we find the minimum-area rectangle enclosing the image region and add it as a new bounding box. The aforementioned in (3) is performed only if the rectangle is above a minimum size γ_1 and if the overlapping with the bounding box is below γ_2 (otherwise, we would add a double detection). The whole procedure is formally stated in pseudo-code as Algorithm 5.

6.3 Experiments and Discussion

In this section, we discuss on the experimental results that we have obtained for the different combination strategies proposed in this chapter. Firstly, we describe the setting

of hyper-parameters and the experimental setup in Section 6.3.1. Secondly, we report on the experimental results of the BBox-Seg strategies in Section 6.3.2. Finally, Section 6.3.3 informs of the experimental results of the Seg-BBox combination strategy.

6.3.1 Experimental Setup

The experimental setup comprises the same datasets that have been employed in previous chapters, i.e. the different algorithms have been evaluated on both the visual inspection and the quality control tasks. As could be expected, the FPSSD model described in Chapter 4 an the AUN model of Chapter 5 have been involved in the tests as, respectively, the BBox detector and the semantic segmentation approach. For these experiments, the AUN has been trained by means of the fully supervised ground truth.

For performance evaluation, we employ, similarly to previous chapters, the mIOU, as well as the mean recall (mRec), the mean precision (mPrec), and the corresponding F_1 score.

6.3.2 Evaluation of the BBox-Seg Strategies

6.3.2.1 BBox-Seg: intersection strategy

As shown in Table 6.1, for the quality control task, the precision and F_1 score of the intersection strategy are, respectively, 3.7% and 0.72% higher than the precision and the mIOU attained by purely the segmentation network (AUN). However, the recall value declines by 1.81%. As for the visual inspection task, the intersection strategy achieves higher precision, whereas the recall and the mIOU are lower than for AUN. Summing up, although the precision achieved by the intersection is higher (as could be expected), the other metric values do not confirm the superiority of this strategy with respect to the original approach.

On the other side, Fig. 6.2 confirms the increase/decrease in precision/recall, as the intersection strategy tends to remove some regions of the segmentation output, giving rise to less false positives but maybe missing some positives.

6.3.2.2 BBox-Seg: union strategy

As shown in Table 6.1, the recall value of the union strategy gets the highest for both tasks, unlike the performance observed for the intersection strategy. However, as expected, the BBox detection results incorporate a large area of background into the

Algorithm 4: BBox-Seg – Intersection of bounding box detection and segmentation results, with threshold

Input:
Segmentation results P_{seg}
BBox detection results P_{det}
C is the number of categories
γ is the threshold on the size of the intersection area
(w, h) are the width and height of the image
Initialization:
Define IMγ as an empty mask to store the resulting segmentation mask
begin
  One-hot encode P_{seg} into P_{seg_oh}
  Transform P_{det} into a mask in one-hot encoding format P_{det_oh}
  for *each bounding box* $p_{det\_oh} \in P_{det\_oh}$ **do**
    c = category of p_{det_oh}
    $p_{det,c} = p_{det\_oh}[c]$
    for *each connected component* $p_{seg,c} \in P_{seg\_oh}[c]$ **do**
      if $\frac{area(p_{seg,c} \cap p_{det,c})}{w \times h} \geq \gamma$ ▷ *area(r) is the area of the image region r*
      then
        IM$\gamma[c]$ = IM$\gamma[c] \cup (p_{seg,c} \cap p_{det,c})$ ▷ *bitwise AND/OR operations*
      end
    end
  end
  One-hot decode IMγ
end
Output:
Pixel-level intersection IMγ between P_{seg} and P_{det}

results, leading to a lot of pixel-wise false positives. In this way, the precision and mIOU show a significant reduction in comparison to the intersection results.

The aforementioned is corroborated in Fig. 6.2, as the union operation gives rise to the detection of almost all the targets of the image, though at the expense of increasing the number of false-positive pixels.

6.3.2.3 BBox-Seg: intersection-with-threshold strategy

We have evaluated this third BBox-Seg strategy for decreasing value of γ, from 0.01 to 0.0005. As can be observed in Table 6.1, the precision and recall metrics increase/decrease generally with an increase in γ for both tasks, as expected. Regarding the mIOU, the

change is not that monotonous with variations in γ: the highest mIOU (86.9%) is obtained for γ=0.002 for the quality control task, which is above that achieved by any of the other strategies, while the highest mIOU (84.78%) corresponds to γ=0.001 for the visual inspection task, though the best performance is attained by the AUN on this occasion.

6.3.2.4 Discussion on the BBox-Seg strategies performance

All in all, we can see that the BBox-Seg strategies do not bring a significant improvement to the segmentation performance. On the one hand, irrespective of whether simple intersection or intersection with threshold is used, the low performance of the BBox detector with small targets can make discard some correct detections resulting from the segmentation-based detector. On the other hand, since BBoxes tend to contain a part of the background, the union strategy tends to bring plenty of false-positive pixels.

6.3.3 Evaluation of the Seg-BBox Strategy

As can be seen in Table 6.2, when the IOU threshold γ_2 increases, the mIOU of the final detection results decreases, while the recall rises, from 90.51% to 95.51% for the quality control task, and from 92.73% to 100% for the visual inspection task. Specifically, regarding the quality control task, when γ_2 is 0, the Seg-BBox strategy effectively improves the recall in comparison with FPSSD. It also yields a higher mIOU (84.93%) and F_1 score (92.33%) than FPSSD. As for the visual inspection task, when γ_2 is 0, the recall of the Seg-BBox strategy is 1.6% higher than that of FPSSD, as well as the mIOU, which is slightly above. Summing up, Seg-BBox requires the two thresholding operations to outperform FPSSD, with significant improvement on recall in both tasks for all values of γ_2, while improvements on precision and mIOU take place for values of γ_2, mostly for $\gamma_2 = 0$.

Figure 6.5 shows examples of detection results of FPSSD and detection results of Seg-BBox only using the γ_1 threshold. Figure 6.6 and Figure 6.7 show examples of detection results for the Seg-BBox strategy for different values of γ_2. It is clear that the best detection performance is obtained for both tasks when γ_2 is 0.

6.4 Conclusions

In this chapter, we have addressed the combination of the semantic segmentation and the BBox detection approaches to improve the detection performance of both methods.

In the first part of the chapter, we have developed and tested several strategies aiming at combining BBox detection with semantic segmentation in order to improve the latter's performance, namely the *BBox-Seg* combination strategies. In detail, we consider applying three kinds of combination strategies, namely intersection, union, and intersection with threshold. The experiments performed have not shown a real advantage from this kind of combination.

In the second part of the chapter, we have proposed an algorithm for improving BBox detection performance with the help of the semantic segmentation output, namely the *Seg-BBox* combination strategy. To this end, we transform the segmentation output into a set of bounding boxes which are added to the list of initial detections provided that the BBox (1) is not too small and (2) does not overlap significantly with a previous detection. This combination outperforms effectively the BBox detection algorithm, particularly when no overlap is permitted with the original BBoxes ($\gamma_2 = 0$).

Algorithm 5: Seg-BBox combination strategy.

Input:
Segmentation results P_{seg}
BBox detection results P_{det}
C is the number of categories
γ_1 is the threshold on the size of a bounding box from P_{seg}
γ_2 is the threshold on the IOU between bounding boxes
Initialization:
Define C_{bbox} as an empty list to store the bounding boxes resulting from the combination
begin
  One-hot encode P_{seg} into P_{seg_oh}
  Transform P_{det} into a mask in one-hot encoding format P_{det_oh}
  for *each bounding box* $p_{det\_oh} \in P_{det\_oh}$ **do**
    append p_{det_oh} to C_{bbox}
    $c =$ category of p_{det_oh}
    $p_{det,c} = p_{det\_oh}[c]$
    for *each connected component* $p_{seg,c} \in P_{seg\_oh}[c]$ **do**
      $p_{segbox,c} =$ minimum-area rectangle of $p_{seg,c}$
      if $area(p_{segbox,c}) \leq \gamma_1$ ▷ *area(r) is the area of the image region r*
      then
        continue
      end
      if $IOU(p_{segbox,c}, p_{det,c}) < \gamma_2$ ▷ *IOU(a, b) denotes the Intersection Over Union for BBoxes a and b*
      then
        append $p_{segbox,c}$ in one-hot encoding format to C_{bbox}
      end
    end
  end
  One-hot decode C_{bbox}
end
Output:
List of bounding boxes C_{bbox} resulting from the combination of P_{seg} and P_{det}.

Table 6.1: Evaluation of the BBox-Seg combination strategies, namely *intersection*, *union*, and *intersection with threshold*. Threshold γ is set as indicated. (Best result shown in bold face.)

Dataset	Networks	Metrics	Seg. Network (AUN)	Intersection	Union	Intersection with Threshold (γ)					
						0.01	0.008	0.005	0.002	0.001	0.0005
quality control	ANU + FPSSD	mIOU	0.8604	0.8532	0.5551	0.8496	0.8358	0.8505	**0.8690**	0.8407	0.8412
		Rec	0.8058	0.7877	**0.8759**	0.7721	0.7894	0.7979	0.8087	0.8110	0.7916
		Prec	0.8432	**0.8802**	0.6477	0.8539	0.8630	0.8522	0.8687	0.8473	0.8406
		F_1	0.8241	0.8313	0.7447	0.8109	0.8245	0.8241	**0.8376**	0.8287	0.8153
visual inspection	ANU + FPSSD	mIOU	0.8333	0.8175	0.5376	0.8149	0.8067	0.8275	0.8140	**0.8478**	0.8298
		Rec	0.8537	0.8251	**0.9362**	0.8030	0.8266	0.8284	0.8313	0.8328	0.8225
		Prec	0.9119	**0.9273**	0.6682	0.8653	0.8610	0.8589	0.8387	0.8323	0.8385
		F_1	**0.8837**	0.8732	0.7798	0.8329	0.8434	0.8485	0.8349	0.8325	0.8304

Table 6.2: Evaluation of the Seg-BBox combination strategy. Threshold γ_1 is set to 0.005 $\times$ area(image) to discard small bounding boxes. Threshold γ_2 is set as indicated. Seg-BBox (AUN) does not employ γ_1 nor γ_2, Seg-BBox (γ_1) only applies γ_1, Seg-BBox ($\gamma_{1,2}$) applies both. (Best result shown in bold face.)

Dataset	Networks	Metrics	FPSSD	Seg BBox (ANU)	Seg-BBox (γ_1)	Seg-BBox ($\gamma_{1,2}$)			
						$\gamma_2 = 0.0$	$\gamma_2 = 0.3$	$\gamma_2 = 0.5$	$\gamma_2 = 0.7$
quality control	ANU + FPSSD	mIOU	0.8443	0.5417	0.6176	**0.8493**	0.8196	0.8292	0.7953
		Rec	0.8715	0.6108	0.6246	0.9051	0.8766	0.9312	**0.9551**
		Prec	**0.9550**	0.6554	0.6858	0.9424	0.8813	0.8067	0.7859
		F_1	0.9113	0.6323	0.6537	**0.9233**	0.8789	0.8644	0.8622
visual inspection	ANU + FPSSD	mIOU	0.9375	0.5961	0.6365	**0.9446**	0.8627	0.8220	0.7827
		Rec	0.9113	0.6108	0.6482	0.9273	0.9452	0.9807	**1.0**
		Prec	**1.0**	0.6554	0.6858	0.9865	0.9022	0.8023	0.7860
		F_1	0.9536	0.6323	0.6665	**0.9559**	0.9231	0.8825	0.8801

AUN segmentation results for the visual inspection task

BBox-Seg intersection results for the visual inspection task

BBox-Seg union results for the visual inspection task

AUN segmentation results for the quality control task

BBox-Seg intersection results for the quality control task

BBox-Seg union results for the quality control task

Figure 6.2: Qualitative comparison between AUN and the BBox-Seg intersection/union strategies.

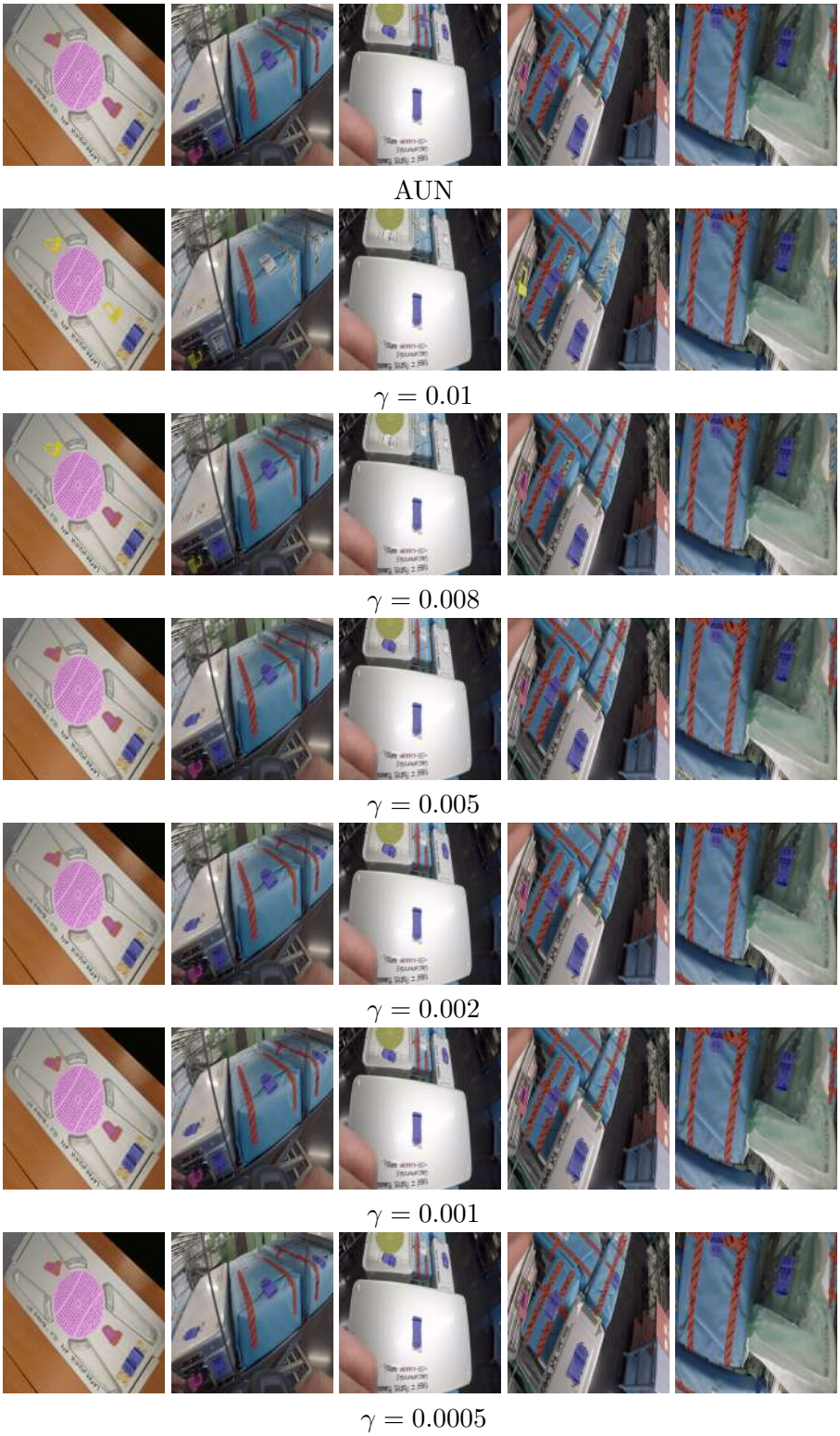


Figure 6.3: Qualitative comparison between AUN and the BBox-Seg intersection-with-threshold strategy for the quality control task and different values of γ.

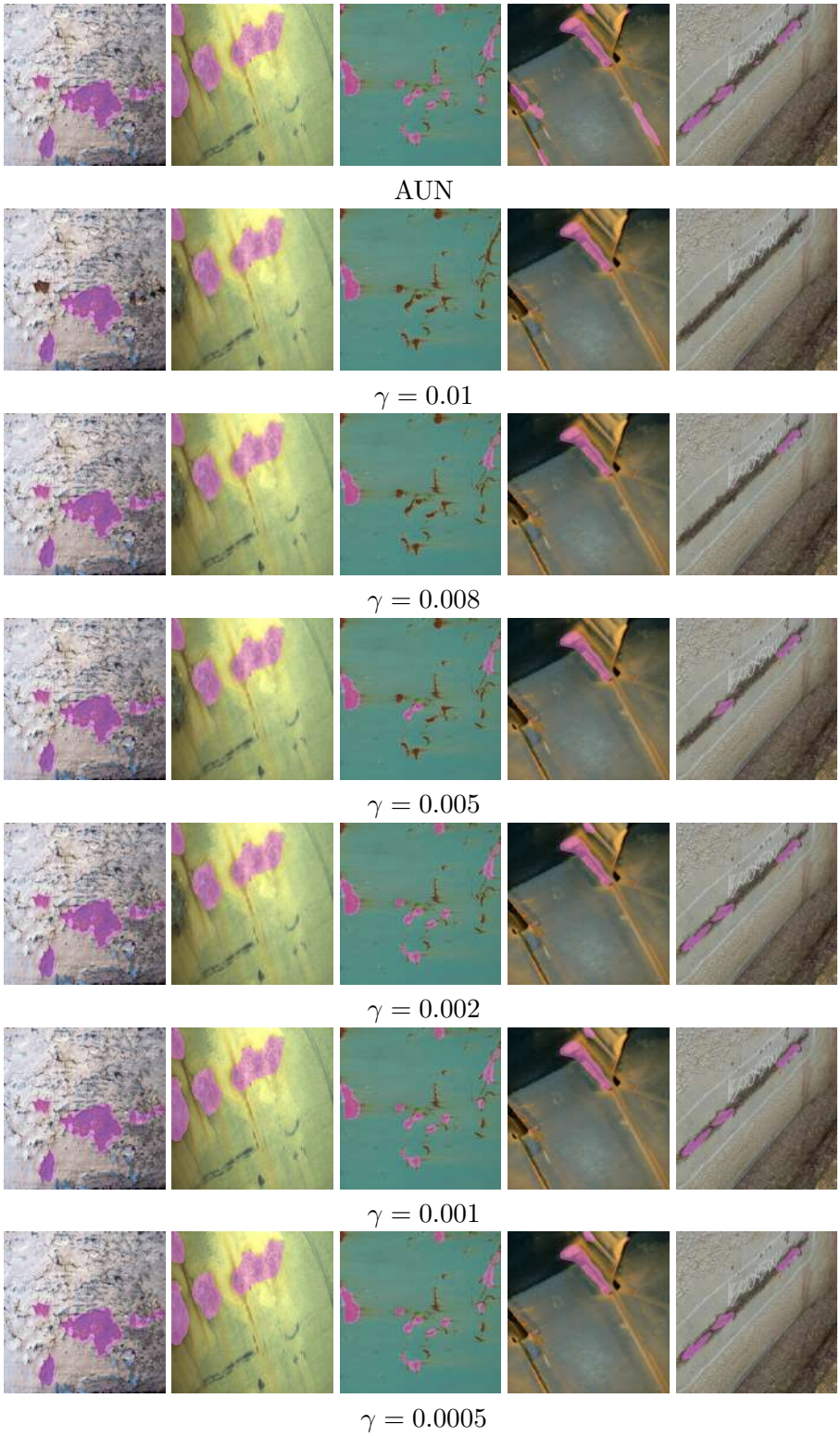
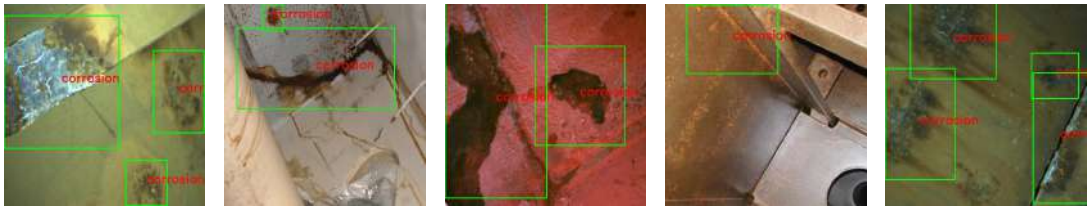
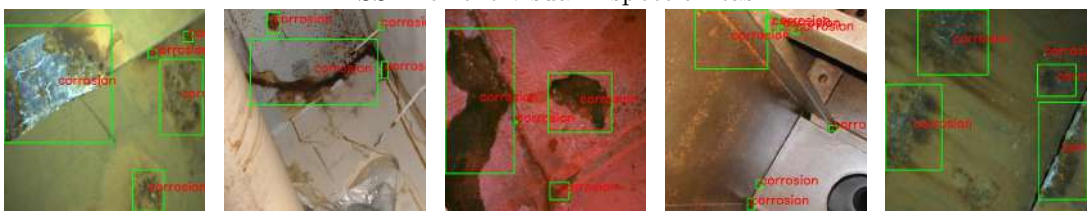


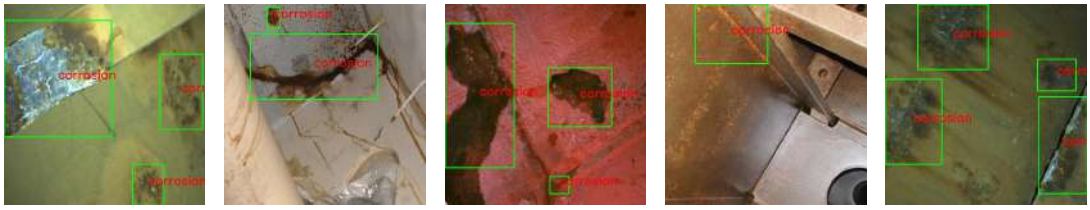
Figure 6.4: Qualitative comparison between AUN and the BBox-Seg intersection-with-threshold strategy for the visual inspection task and different values of γ.



FPSSD for the visual inspection task



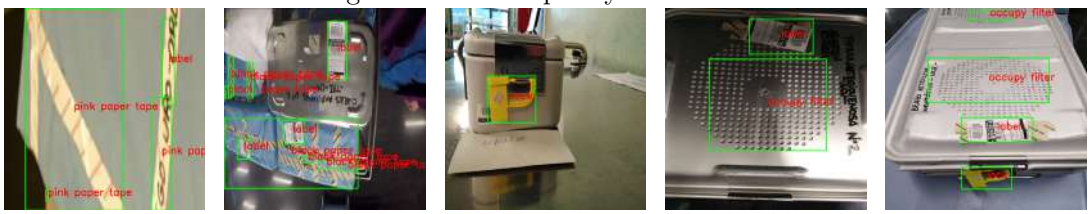
Seg-BBox for the visual inspection task



Seg-BBox after removing small BBoxes for the visual inspection task

FPSSD for the quality control task

Seg-BBox for the quality control task



Seg-BBox after removing small rectangles for the quality control task

Figure 6.5: Qualitative comparison between FPSSD and the Seg-BBox strategy [$\gamma_1 = 0.005 \times$ area(image)].

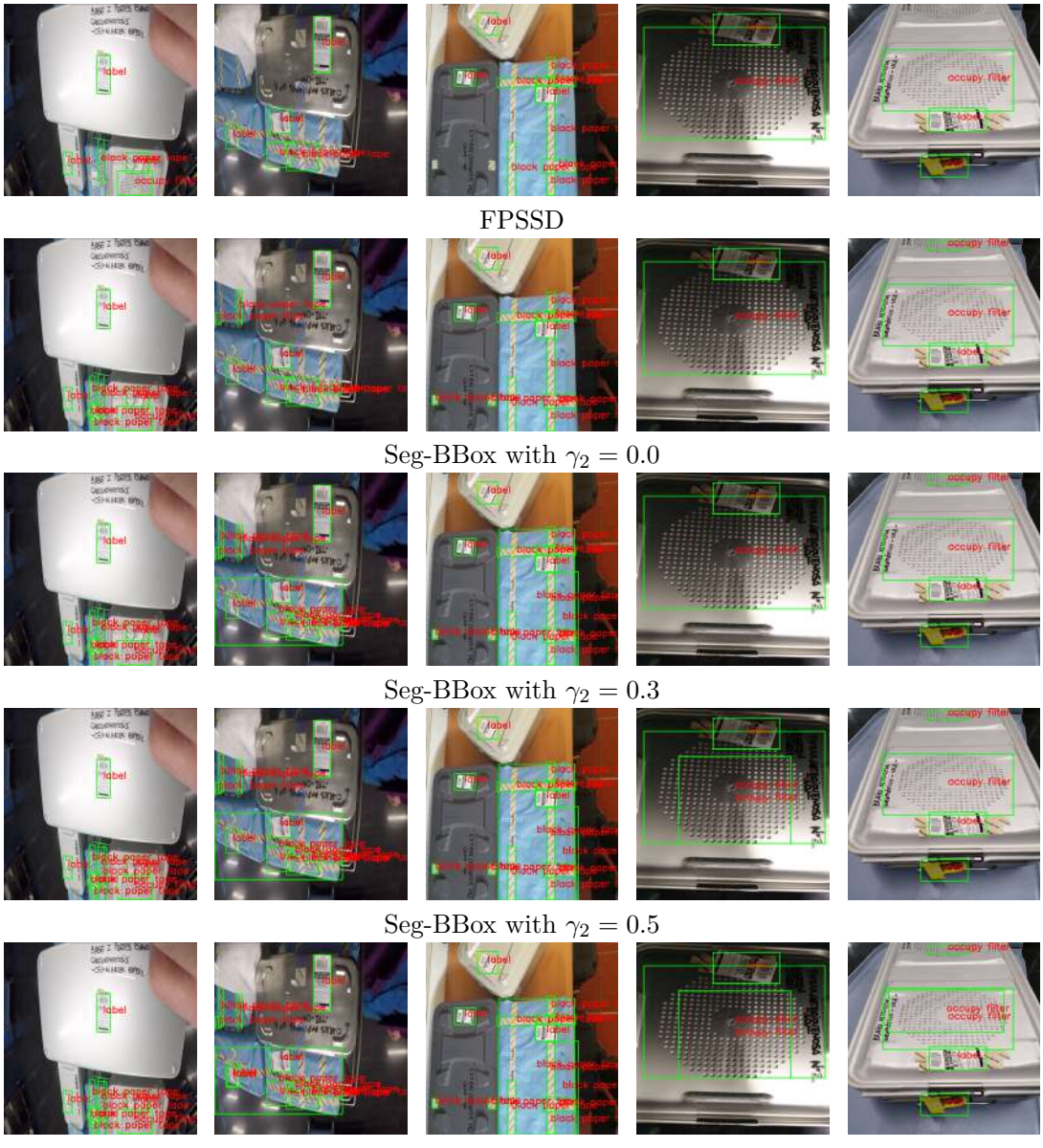


Figure 6.6: Qualitative comparison between FPSSD and the Seg-BBox strategy for different values of γ_2 [$\gamma_1 = 0.005 \times$ area(image)] and the quality control task.

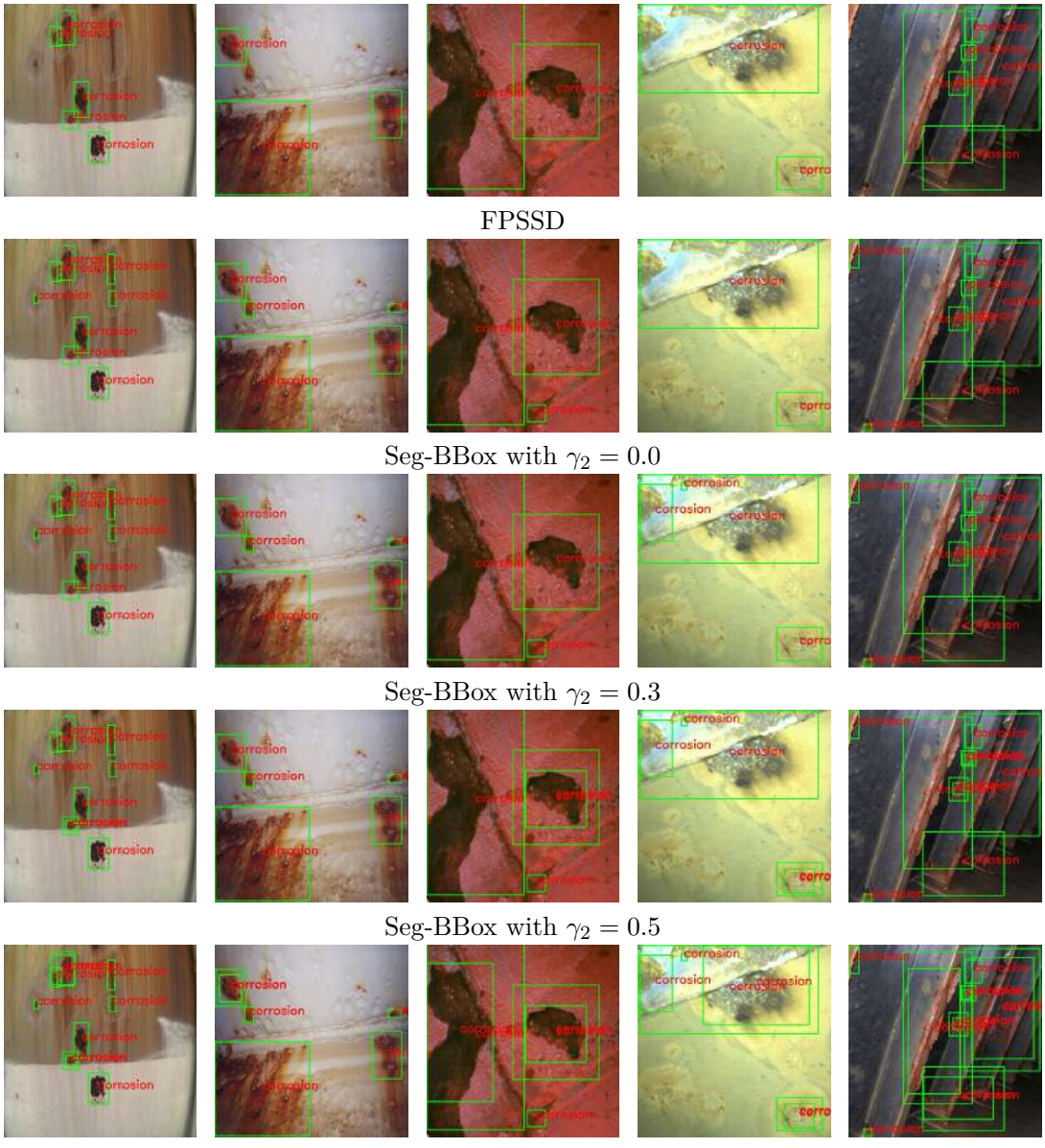


Figure 6.7: Qualitative comparison between FPSSD and the Seg-BBox strategy for different values of γ_2 [$\gamma_1 = 0.005 \times$ area(image)] and the visual inspection task.

Conclusions

7.1 Summary of the Thesis

In this dissertation, we have presented several DCNN-based solutions for target detection in digital images. They all have been assessed on two datasets related to two different industrial applications, namely visual inspection and quality control.

For a start, after the background exposed in Chapter 2, in Chapter 3 we have reviewed extensively the literature related with the different DCNN methodologies approached in this thesis. More classical methods addressing the two application cases that we have considered in this work have also been thoroughly reviewed.

Subsequently, in Chapter 4, we have proposed and evaluated a two-stage arbitrarily-oriented object detection approach making use of bounding boxes regression. The background of this work involves two prominent object detection approaches, namely Faster R-CNN [57] and SSD [30], that have been adapted for our two assessment tasks. Along a number of experiments, we have found that the existing approaches lack the ability to detect small objects, while, for some elongated targets, the detection results of un-oriented bounding boxes can be ambiguous and even inaccurate in some situations. As an improvement, in a first stage of the method, we propose the Feature Pyramid Single Shot Multi-box Detector (FPSSD) that aims at improving the detection performance for objects at different scales by means of a multi-scale feature fusion approach, taking inspiration from FPN [58]. On the other side, we have also addressed the configuration of the typical hyper-parameters of anchor-based bounding-box regression methods, such as the size and the aspect ratio of default boxes. In this regard, we have adopted a clustering-based strategy, which, using the data contained in the training sets, has allowed us to define prior bounding boxes from which regression can start on, to speed up convergence and improve the detection accuracy.

For the second stage of the bounding boxes detection method, we have designed and configured a lightweight neural network to infer rotated bounding boxes on the basis of the results of the first stage. As part of the design process, we have discussed and evaluated two different parameterizations for rotated bounding boxes: two terms (d_1, d_2) or three terms (d_1, d_2, h). The experimental results have shown that the two-term approach achieves better performance. To conclude with this, the network, though being simple, has shown to be effective to solve the regression problem that has been addressed.

In general, and according to all the experiments performed, the object detection approach has shown to be effective for both tasks, visual inspection, which involves irregular shapes, and quality control, where more than two classes and objects of very different shapes and sizes are involved.

Next, Chapter 5 has been devoted to a different approach for target identification in images, namely semantic segmentation. Firstly, an outstanding semantic segmentation approach, FCN [52], has been fine-tuned and tested on both tasks. Similarly to the object detection approach, we have rapidly learnt that semantic segmentation also faces the challenge of small-area objects detection. To solve the problem, we have considered and applied several loss functions to train the network, namely the Dice loss [208], the Focal loss [45] and cross-entropy. The Dice loss has achieved the highest performance on both tasks.

In the second part of Chapter 5, we have considered the large number of pixel-level annotations that are required to train properly the DCNN supporting a semantic segmentation solution, which is not only time-consuming but also demanding a significant amount of human effort. Therefore, a weakly-supervised semantic segmentation solution has been developed to mitigate the problem with the massive pixel-level annotations that are required by fully-supervised solutions. In this work, we have used U-Net [94] as segmentation network, which has been complemented with Attention Gates, in a way similar to Attention U-Net [103].

Considering the various forms of weak annotations, scribbles have been found quite convenient since they only require from the user to drag the cursor in the area where the targets lie, and use different colors to discriminate between categories. From the performance point of view, the kind of ground truth they provide has also shown to be useful for our purposes, as has been proved by the experimental results.

In this regard, since the scribble annotations only provide a few labelled pixels, it has been necessary to propagate the ground truth information available to give rise to proper trainings. Inspired by [125], we have adopted a methodology that makes use of an

over-segmentation of the input image in superpixels [35] to generate a pseudo-mask from where training can take place. Nevertheless, although the segmentation performance is not unacceptable using pseudo-masks, the segmentation results have shown to be still affected by the quality of the segmentation proposals. To solve this problem, we have designed a three-term loss function that includes the so-called Centroid loss term inspired by the K-means clustering algorithm. This term is intended to assist the training of the segmentation network and provide guidance on how to infer the correct categories.

Finally, according to the experimental results that have been reported in Chapter 5, our weakly-supervised segmentation approach based on Attention U-Net and a loss function including the Centroid loss term has been evaluated favourably on the two tasks that are considered in this thesis, with a significantly lower cost of image labeling. Moreover, under weak annotations of varying quality, the approach has managed to counteract the potential mistakes in the segmentation proposals and produce good-enough segmentations.

As an extension, in Chapter 6, we have considered how to combine bounding box regression and semantic segmentation by means of different strategies with the aim of jointly improving the target detection performance. Not surprisingly, the bounding boxes regression approach has not been able to bring significant improvement on the segmentation performance through the *BBox-Seg* strategies; even in some cases, the segmentation performance has worsen.

On the contrary, semantic segmentation has shown to be useful to effectively improve the detection performance of bounding boxes detection through the *Seg-BBox* strategy. According to our experiments, *Seg-BBox* can supplement the detection ability of the bounding boxes detector, particularly for small targets.

7.2 Future Work

After the different problems that have been considered in this thesis and the several developments which have been carried out, a number of improvements can be suggested, and thus can be considered as future work. They are enumerated in the following:

- *Develop a lighter object detection architecture.* Visual inspection and quality control applications can require performing inferences in real time and on embedded systems with limited computational resources. Some methodologies, including network pruning, network compression, specific architectures such as MobileNet [238], etc., have achieved promising performance which the aforementioned applications could benefit from.

- *Improve the oriented bounding boxes detection model.* The two stages of the network for oriented bounding-boxes detection are trained separately, since they actually operate independently. A single network able to directly regress oriented bounding boxes would certainly enhance the detector. On the other side, in the current model, the second stage depends on the output of the first stage. This means that improving the detection performance of the first stage will improve the global detection capability. Clearly, several ways of improvement are possible: adopt a faster and more accurate unoriented bounding boxes detector and/or adopt a more robust network for oriented bounding boxes regression.

- *Improve furture the small object detection capability.* Detecting small objects in cluttered, complex scenes has long been a challenge. Currently, there is already some work in this research direction, such as detecting human faces in noisy and dense scenes, locating some targets within satellite images, etc. The detection of small objects is still a challenge, and hence a good research direction for future, also applicable to the visual inspection and quality control problems.

- *Enhance the clustering/Centroid loss approach.* As discussed in Chapter 5, our approach has shown the potential of achieving good image segmentation performance from weak annotations. However, compared with the fully supervised approach, the detection performance still has a gap that can be shortened. Of particular interest for us is to explore further the idea of incorporating a clustering process into the detector, in particular for multi-category problems. Another way of improvement would be to explore other backbones, such FCN [52] or DeepLab [96], as base segmentation network for the weakly-supervised segmentation approach.

- *Explore other ways of combining bounding boxes-based and semantic segmentation-based detection methodologies.* The experiments on the combination of the two approaches has shown improvements of the joint detection performance, particularly regarding small targets, in comparison with bounding boxes-based regression alone. The design of a single network able to provide pixel-level classification and bounding boxes regression would be a research line to explore.

Bibliography

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson, 2018.

[2] Z. Wang, B. Ma, and Y. Zhu, "Review of Level Set in Image Segmentation," *Archives of Computational Methods in Engineering*, 2020.

[3] Y. Boykov and G. Funka-Lea, "Graph Cuts and Efficient N-D Image Segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006.

[4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Neural Information Processing Systems*, 2012, pp. 1097–1105.

[6] H. Golnabi and A. Asadpour, "Design and Application of Industrial Machine Vision Systems," *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 6, pp. 630–637, Dec. 2007.

[7] J. Beyerer, F. P. León, and C. Frese, *Machine Vision: Automated Visual Inspection: Theory, Practice and Applications*, 2016.

[8] D. Ponsa, "Quality Control of Aafety Belts by Machine Vision Inspection for Real-time Production," *Optical Engineering*, vol. 42, no. 4, p. 1114, Apr. 2003.

[9] S. Cubero, N. Aleixos, E. Moltó, J. Gómez-Sanchis, and J. Blasco, "Advances in Machine Vision Applications for Automatic Inspection and Quality Evaluation of Fruits and Vegetables," *Food and Bioprocess Technology*, vol. 4, no. 4, pp. 487–504, Jul. 2010.

[10] J. Wang, P. Fu, and R. X. Gao, "Machine Vision Intelligence for Product Defect Inspection Based on Deep Learning and Hough Transform," *Journal of Manufacturing Systems*, vol. 51, pp. 52–60, Apr. 2019.

[11] J. P. Yun, W. C. Shin, G. Koo, M. S. Kim, C. Lee, and S. J. Lee, "Automated Defect Inspection System for Metal Surfaces Based on Deep Learning and Data Augmentation," *Journal of Manufacturing Systems*, vol. 55, pp. 317–324, Apr. 2020.

[12] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 9, pp. 731–747, Nov. 2017.

[13] Y. Santur, M. Karakose, and E. Akin, "A New Rail Inspection Method Based on Deep Learning Using Laser Cameras," in *Proceedings of the International Artificial Intelligence and Data Processing Symposium*, Sep. 2017.

[14] D. Xu, C. Wen, and J. Liu, "Wind Turbine Blade Surface Inspection Based on Deep Learning and UAV-taken Images," *Journal of Renewable and Sustainable Energy*, vol. 11, no. 5, p. 053305, Sep. 2019.

[15] W. S. McCulloch and W. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity," *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[16] F. Rosenbaltt, "The Perceptron: a Perceiving and Recognizing Automation," *Cornell Aeronautical Laboratory*, 1957.

[17] G. E. Hinton, "Learning Distributed Representations of Concepts," in *Proceedings of the Annual Conference of the Cognitive Science Society*, vol. 1, 1986, p. 12.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[19] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the International Conference on Machine Learning*, 2015, pp. 448–456.

[20] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[23] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," University of Toronto, Tech. Rep., 2009.

[24] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects In Context," in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 740–755.

[26] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective Search for Object Recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[27] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[29] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2016, pp. 379–387.

[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 21–37.

[31] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies For Accurate Object Detection and Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[32] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[33] P. Krähenbühl and V. Koltun, "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials," in *Proceedings of the Advances in Neural Information Processing Systems*, 2011, pp. 109–117.

[34] Y. Boykov and M.-P. Jolly, "Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images," in *Proceedings IEEE International Conference on Computer Vision*, vol. 1, 2001, pp. 105–112.

[35] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-art Superpixel Methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[36] V. Jampani, D. Sun, M.-Y. Liu, M.-H. Yang, and J. Kautz, "Superpixel Sampling Networks," in *Proceedings of the European Conference on Computer Vision*, September 2018.

[37] S. Kwak, S. Hong, and B. Han, "Weakly Supervised Semantic Segmentation using Superpixel Pooling Network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[38] R. Gadde, V. Jampani, M. Kiefel, D. Kappler, and P. V. Gehler, "Superpixel Convolutional Networks Using Bilateral Inceptions," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 597–613.

[39] W. Zhao, Y. Fu, X. Wei, and H. Wang, "An Improved Image Semantic Segmentation Method Based on Superpixels and Conditional Random Fields," *Applied Sciences*, vol. 8, no. 5, p. 837, May 2018.

[40] K. Hornik, "Approximation Capabilities of Multilayer Feedforward Networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[41] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[42] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[43] S. Mallat, "Understanding Deep Convolutional Networks," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150203, 2016.

[44] S. Jadon, "A Survey of Loss Functions for Semantic Segmentation," in *Proceedings of the IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology*, oct 2020.

[45] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[46] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Jorge Cardoso, "Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2017, pp. 240–248.

[47] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, "Tversky Loss Function for Image Segmentation Using 3D Fully Convolutional Deep Networks," in *Proceedings of the International Workshop on Machine Learning in Medical Imaging*, 2017, pp. 379–387.

[48] S. R. Hashemi, S. S. M. Salehi, D. Erdogmus, S. P. Prabhu, S. K. Warfield, and A. Gholipour, "Asymmetric Loss Functions and Deep Densely-Connected Networks for Highly-Imbalanced Medical Image Segmentation: Application to Multiple Sclerosis Lesion Detection," *IEEE Access*, vol. 7, pp. 1721–1735, 2019.

[49] J. Ribera, D. Güera, Y. Chen, and E. Delp, "Weighted Hausdorff Distance: A Loss Function for Object Localization," *arXiv preprint arXiv:1806.07564*, vol. 2, 2018.

[50] H. Kervadec, J. Bouchtiba, C. Desrosiers, E. Granger, J. Dolz, and I. B. Ayed, "Boundary Loss for Highly Unbalanced Segmentation," in *Proceedings of the International Conference on Medical Imaging with Deep Learning*, 2019, pp. 285–296.

[51] S. A. Taghanaki, Y. Zheng, S. K. Zhou, B. Georgescu, P. Sharma, D. Xu, D. Comaniciu, and G. Hamarneh, "Combo Loss: Handling Input and Output Imbalance in Multi-Organ Segmentation," *arXiv preprint arXiv:1805*, 2018.

[52] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[53] M.-L. Zhang and Z.-H. Zhou, "A Review on Multi-label Learning Algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2013.

[54] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2014.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[56] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[57] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks," in *Proceedings of the Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[58] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[59] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[60] W. Ouyang, K. Wang, X. Zhu, and X. Wang, "Chained Cascade Network for Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1938–1946.

[61] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards Balanced Learning for Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 821–830.

[62] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7036–7045.

[63] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[64] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks," *arXiv preprint arXiv:1312.6229*, 2013.

[65] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7263–7271.

[66] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable Object Detection Using Deep Neural Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2147–2154.

[67] H. Law and J. Deng, "CornerNet: Detecting Objects as Paired Keypoints," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 734–750.

[68] A. Newell, K. Yang, and J. Deng, "Stacked Hourglass Networks for Human Pose Estimation," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 483–499.

[69] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint Triplets for Object Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6569–6578.

[70] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, last access: Novermber 2021.

[71] X. Yang, K. Fu, H. Sun, J. Yang, Z. Guo, M. Yan, T. Zhan, and S. Xian, "R2CNN++: Multi-dimensional Dttention Based Rotation Invariant Detector with Robust Anchor Strategy," *arXiv preprint arXiv:1811.07126*, vol. 2, p. 7, 2018.

[72] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue, "Arbitrary-oriented Scene Text Detection Via Rotation Proposals," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3111–3122, 2018.

[73] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo, "R2CNN: Rotational Region CNN for Orientation Robust Scene Text Detection," *arXiv preprint arXiv:1706.09579*, 2017.

[74] L. Deng, Y. Gong, Y. Lin, J. Shuai, X. Tu, Y. Zhang, Z. Ma, and M. Xie, "Detecting Multi-oriented Text with Corner-based Region Proposals," *Neurocomputing*, vol. 334, pp. 134–142, 2019.

[75] L. Liu, Z. Pan, and B. Lei, "Learning a Rotation Invariant Detector with Rotatable Bounding Box," *arXiv preprint arXiv:1711.09405*, 2017.

[76] Q. Yang, M. Cheng, W. Zhou, Y. Chen, M. Qiu, W. Lin, and W. Chu, "Inceptext: A New Inception-text Module with Deformable PSROI Pooling for Multi-oriented Scene Text Detection," *arXiv preprint arXiv:1805.01167*, 2018.

[77] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask Textspotter: An End-to-end Trainable Neural Network for Spotting Text with Arbitrary Shapes," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 67–83.

[78] M. Dickenson and L. Gueguen, "Rotated Rectangles for Symbolized Building Footprint Extraction." in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 225–228.

[79] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A Single-shot Oriented Scene Text Detector," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3676–3690, 2018.

[80] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "TextBoxes: A Fast Text Detector with a Single Deep Neural Network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[81] Y. Liu and L. Jin, "Deep Matching Prior Network: Toward Tighter Multi-oriented Text Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1962–1969.

[82] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[83] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "DOTA: A Large-scale Dataset for Object Detection in Aerial Images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3974–3983.

[84] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting Texts of Arbitrary Orientations in Natural Images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1083–1090.

[85] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, "ICDAR 2013 Robust Reading Competition," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2013, pp. 1484–1493.

[86] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "ICDAR 2015 Competition on Robust Reading," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2015, pp. 1156–1160.

[87] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep Direct Regression for Multi-Oriented Scene Text Detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 745–753.

[88] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single Shot Text Detector with Regional Attention," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3047–3055.

[89] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "EAST: An Efficient and Accurate Scene Text Detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[90] M. Liao, Z. Zhu, B. Shi, G.-s. Xia, and X. Bai, "Rotation-sensitive Regression for Oriented Scene Text Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5909–5918.

[91] S. Ruan, J. Lu, F. Xie, and Z. Jin, "A Novel Method for Fast Arbitrary-oriented Scene Text Detection," in *Proceedings of the Chinese Control and Decision Conference*, 2018, pp. 1652–1657.

[92] H. Noh, S. Hong, and B. Han, "Learning Deconvolution Network for Semantic Segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.

[93] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[94] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer-assisted Intervention*, 2015, pp. 234–241.

[95] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs," *arXiv preprint arXiv:1412.7062*, 2014.

[96] ——, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[97] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional Random Fields as Recurrent Neural Networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1529–1537.

[98] F. Yu and V. Koltun, "Multi-scale Context Aggregation by Dilated Convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[99] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A Deep Neural Network Architecture for Real-time Semantic Segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[100] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, "Efficient Piecewise Training of Deep Structured Models for Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3194–3203.

[101] G. Ghiasi and C. C. Fowlkes, "Laplacian Pyramid Reconstruction and Refinement for Semantic Segmentation," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 519–534.

[102] S. Lian, Z. Luo, Z. Zhong, X. Lin, S. Su, and S. Li, "Attention Guided U-Net for Accurate IRIS Segmentation," *Journal of Visual Communication and Image Representation*, vol. 56, pp. 296–304, 2018.

[103] O. Oktay, J. Schlemper, L. L. Folgoc, M. C. H. Lee, M. P. Heinrich, K. Misawa, K. Mori, S. G. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention U-Net: Learning Where to Look for the Pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[104] S. Li, M. Dong, G. Du, and X. Mu, "Attention Dense-U-Net for Automatic Breast Mass Segmentation in Digital Mammogram," *IEEE Access*, vol. 7, pp. 59 037–59 047, 2019.

[105] Z.-L. Ni, G.-B. Bian, X.-L. Xie, Z.-G. Hou, X.-H. Zhou, and Y.-J. Zhou, "RASNet: Segmentation for Tracking Surgical Instruments in Surgical Videos Using Refined Attention Segmentation Network," in *Proceedings of the IEEE Annual International Conference of the Engineering in Medicine and Biology Society*, 2019, pp. 5735–5738.

[106] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to Scale: Scale-aware Semantic Image Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3640–3649.

[107] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid Attention Network for Semantic Segmentation," *arXiv preprint arXiv:1805.10180*, 2018.

[108] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Proceedings of the Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[109] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic Segmentation Using Adversarial Networks," *arXiv preprint arXiv:1611.08408*, 2016.

[110] W. Zhu, X. Xiang, T. D. Tran, and X. Xie, "Adversarial Deep Structural Networks for Mammographic Mass Segmentation," *arXiv preprint arXiv:1612.05970*, 2016.

[111] P. Moeskops, M. Veta, M. W. Lafarge, K. A. Eppenhof, and J. P. Pluim, "Adversarial Training and Dilated Convolutions for Brain MRI Segmentation," in *Proceedings of the Deep learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, 2017, pp. 56–64.

[112] M. Rezaei, K. Harmuth, W. Gierke, T. Kellermeier, M. Fischer, H. Yang, and C. Meinel, "A Conditional Adversarial Network for Semantic Segmentation of Brain Tumor," in *Proceedings of the International MICCAI Brainlesion Workshop*, 2017, pp. 241–252.

[113] T. Neff, C. Payer, D. Stern, and M. Urschler, "Generative Adversarial Network based Synthesis for Supervised Medical Image Segmentation," in *Proceedings of the OAGM and ARW Joint Workshop*, 2017.

[114] Y. Zhang, L. Yang, J. Chen, M. Fredericksen, D. P. Hughes, and D. Z. Chen, "Deep Adversarial Networks for Biomedical Image Segmentation Utilizing Unannotated Images," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2017, pp. 408–416.

[115] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[116] Y. Xie, Z. Zhang, M. Sapkota, and L. Yang, "Spatial Clockwork Recurrent Neural Network for Muscle Perimysium Segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2016, pp. 185–193.

[117] J. Koutnik, K. Greff, F. Gomez, and J. Schmidhuber, "A Clockwork RNN," *arXiv preprint arXiv:1402.3511*, 2014.

[118] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci, and A. Courville, "ReSeg: A Recurrent Neural Network-based Model for Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 41–48.

[119] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. C. Courville, and Y. Bengio, "ReNet: A Recurrent Neural Network based Alternative to Convolutional Networks," *arXiv preprint arXiv:1505.00393*, 2015.

[120] Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, and L. Lin, "LSTM-CF: Unifying Context Modeling and Fusion with Lstms for RGB-D Scene Labeling," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 541–557.

[121] W. Bai, H. Suzuki, C. Qin, G. Tarroni, O. Oktay, P. M. Matthews, and D. Rueckert, "Recurrent Neural Networks for Aortic Image Sequence Segmentation with Sparse Annotations," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018, pp. 586–594.

[122] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," *arXiv preprint arXiv:1612.01105*, 2016.

[123] A. Chakravarty and J. Sivaswamy, "RACE-Net: a Recurrent Neural Network for Biomedical Image Segmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 3, pp. 1151–1162, 2018.

[124] H. Li, J. Li, X. Lin, and X. Qian, "Pancreas Segmentation via Spatial Context based U-Net and Bidirectional Lstm," *arXiv preprint arXiv:1903.00832*, 2019.

[125] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-supervised Convolutional Networks for Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3159–3167.

[126] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.

[127] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, "Object Region Mining with Adversarial Erasing: A Simple Classification to Semantic Segmentation Approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1568–1576.

[128] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. S. Huang, "Adversarial Complementary Learning for Weakly Supervised Object Localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1325–1334.

[129] S. Yang, Y. Kim, Y. Kim, and C. Kim, "Combinational Class Activation Maps for Weakly Supervised Object Localization," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2941–2949.

[130] A. Kolesnikov and C. H. Lampert, "Seed, Expand and Constrain: Three Principles for Weakly-Supervised Image Segmentation," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 695–711.

[131] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-Supervised Semantic Segmentation Network with Deep Seeded Region Growing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7014–7023.

[132] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, and J. Jiao, "Weakly Supervised Instance Segmentation Using Class Peak Response," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3791–3800.

[133] X. Zhang, Y. Wei, G. Kang, Y. Yang, and T. Huang, "Self-produced Guidance for Weakly-supervised Object Localization," in *Proceedings of the European Conference on Computer Vision* , 2018, pp. 597–613.

[134] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, "Simple Does It: Weakly Supervised Instance and Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 876–885.

[135] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut Interactive Foreground Extraction Using Iterated Graph Cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[136] X. Zhao, S. Liang, and Y. Wei, "Pseudo Mask Augmented Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4061–4070.

[137] R. Hu, P. Dollár, K. He, T. Darrell, and R. Girshick, "Learning to Segment Every Thing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4233–4241.

[138] C. Song, Y. Huang, W. Ouyang, and L. Wang, "Box-driven Class-wise Region Masking and Filling Rate Guided Loss for Weakly Supervised Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3136–3145.

[139] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, "What's the Point: Semantic Segmentation with Point Supervision," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 549–565.

[140] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari, "Extreme Clicking for Efficient Object Annotation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4930–4939.

[141] J. Xu, A. G. Schwing, and R. Urtasun, "Learning to Segment Under Various Forms of Weak Supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3781–3790.

[142] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers, "Normalized Cut Loss for Weakly-supervised CNN Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1818–1827.

[143] M. Tang, F. Perazzi, A. Djelouah, I. Ben Ayed, C. Schroers, and Y. Boykov, "On Regularized Losses for Weakly-supervised Cnn Segmentation," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 507–522.

[144] Y. Zhu, Y. Zhou, H. Xu, Q. Ye, D. Doermann, and J. Jiao, "Learning Instance Activation Maps for Weakly Supervised Instance Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3116–3125.

[145] J. Choe and H. Shim, "Attention-based Dropout Layer for Weakly Supervised Object Localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2219–2228.

[146] G. Papandreou, L.-C. Chen, K. Murphy, and A. Yuille, "Weakly- and Semi-supervised Learning of a DCNN for Semantic Image Segmentation," *arXiv preprint arXiv:1502.02734*.

[147] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep Extreme Cut: From Extreme Points to Object Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 616–625.

[148] S. A. Idris, F. A. Jafar, and S. Saffar, "Improving Visual Corrosion Inspection Accuracy with Image Enhancement Filters," in *Proceedings of the IEEE International Conference on Ubiquitous Robots and Ambient Intelligence*, 2015, pp. 129–132.

[149] A. K. Aijazi, L. Malaterre, M. L. Tazir, L. Trassoudaine, and P. Checchin, "Detecting and Analyzing Corrosion Spots on the Hull of Large Marine Vessels Using Colored 3D Lidar Point Clouds," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 3, no. 3, 2016.

[150] N. S. Roberts, "Corrosion Detection in Enclosed Enivironments Using Remote Systems," Ph.D. dissertation, Alfred University, 2016.

[151] M. P. Bento, F. de Medeiros, I. C. de Paula Jr, and G. Ramalho, "Image Processing Techniques Applied for Corrosion Damage Analysis," in *Proceedings of the Brazilian Symposium on Computer Graphics and Image Processing*, 2009.

[152] F. Bonnín-Pascual and A. Ortiz, "Detection of Cracks and Corrosion for Automated Vessels Visual Inspection," in *Proceedings of the Catalan Conference on Computer Vision*, 2010, pp. 111–120.

[153] J. Iivarinen, "Surface Defect Detection with Histogram-based Texture Features," in *Intelligent Robots and Computer Vision XIX: Algorithms, Techniques, and Active vision*, vol. 4197, 2000, pp. 140–145.

[154] S. Odemir, A. Baykut, R. Meylani, A. Erçil, and A. Ertuzun, "Comparative Evaluation of Texture Analysis Algorithms for Defect Inspection of Textile Products," in *Proceedings of the IEEE International Conference on Pattern Recognition*, vol. 2, 1998, pp. 1738–1740.

[155] F. Bonnín-Pascual and A. Ortiz, "Corrosion Detection for Automated Visual Inspection," in *Developments in Corrosion Protection.* InTech, 2014.

[156] C. Ünsalan and A. Erçil, "Automated Inspection of Steel Structures," *Recent Advances in Mechatronics*, pp. 468–480, 1999.

[157] K. Choi and S. Kim, "Morphological Analysis and Classification of Types of Surface Corrosion Damage by Digital Image Processing," *Corrosion Science*, vol. 47, no. 1, pp. 1–15, 2005.

[158] F. N. Medeiros, G. L. Ramalho, M. P. Bento, and L. C. Medeiros, "On the Evaluation of Texture and Color Features for Nondestructive Corrosion Detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, p. 817473, 2010.

[159] N. Tanaka and K. Uematsu, "A Crack Detection Method in Road Surface Images Using Morphology," *Machine Vision and Application*, vol. 98, pp. 17–19, 1998.

[160] H. Zheng, L. X. Kong, and S. Nahavandi, "Automatic Inspection of Metallic Surface Defects Using Genetic Algorithms," *Journal of Materials Processing Technology*, vol. 125, pp. 427–433, 2002.

[161] M. Yoshioka and S. Omatu, "Defect Detection Method Using Rotational Morphology," *Artificial Life and Robotics*, vol. 14, no. 1, pp. 20–23, 2009.

[162] M. R. Jahanshahi, S. F. Masri, C. W. Padgett, and G. S. Sukhatme, "An Innovative Methodology for Detection and Quantification of Cracks Through Incorporation of Depth Perception," *Machine Vision and Applications*, vol. 24, no. 2, pp. 227–241, 2013.

[163] W. Zhang, Z. Zhang, D. Qi, and Y. Liu, "Automatic Crack Detection and Classification Method for Subway Tunnel Safety Monitoring," *Sensors*, vol. 14, no. 10, pp. 19 307–19 328, 2014.

[164] G. Zhao, T. Wang, and J. Ye, "Anisotropic Clustering on Surfaces for Crack Extraction," *Machine Vision and Applications*, vol. 26, no. 5, pp. 675–688, 2015.

[165] B. R. Suresh, R. A. Fundakowski, T. S. Levitt, and J. E. Overland, "A Real-time Automated Visual Inspection System for Hot Steel Slabs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 563–572, 1983.

[166] S. H. Choi, J. P. Yun, B. Seo, Y. S. Park, and S. W. Kim, "Real-time Defects Detection Algorithm for High-speed Steel Bar in Coil," in *Proceedings of World Academy of Science, Engineering and Technology*, vol. 21, 2007, pp. 1307–6884.

[167] G. Wu, H. Kwak, S. Jang, K. Xu, and J. Xu, "Design of Online Surface Inspection System of Hot Rolled Strips," in *Proceedings of the IEEE International Conference on Automation and Logistics*, 2008, pp. 2291–2295.

[168] L. Weiwei, Y. Yunhui, L. Jun, Z. Yao, and S. Hongwei, "Automated On-line Fast Detection for Surface Defect of Steel Strip Based on Multivariate Discriminant Function," in *Proceedings of the International Symposium on Intelligent Information Technology Application*, vol. 2, 2008, pp. 493–497.

[169] J. Blackledge and D. Dubovitskiy, "A Surface Inspection Machine Vision System that Includes Fractal Texture Analysis," *Journal of Intelligent Systems*, vol. 3, no. 2, pp. 76–89, 2008.

[170] J. Cong, Y.-h. Yan, H.-a. Zhang, and J. Li, "Real-time Surface Defects Inspection of Steel Strip based on Difference Image," in *Proceedings of the International Symposium on Photoelectronic Detection and Imaging: Related Technologies and Applications*, vol. 6625, 2008, p. 66250W.

[171] D. Djukic and S. Spuzic, "Statistical Discriminator of Surface Defects on Hot Rolled Steel," *Proceedings of Image and Vision Computing New Zealand*, pp. 158–163, 2007.

[172] Y.-C. Liu, Y.-L. Hsu, Y.-N. Sun, S.-J. Tsai, C.-Y. Ho, and C.-M. Chen, "A Computer Vision System for Automatic Steel Surface Inspection," in *Proceedings of the IEEE Conference on Industrial Electronics and Applications*, 2010, pp. 1667–1670.

[173] W.-b. Li, C.-h. Lu, and J.-c. Zhang, "A Local Annular Contrast Based Real-time Inspection Algorithm for Steel Bar Surface Defects," *Applied Surface Science*, vol. 258, no. 16, pp. 6080–6086, 2012.

[174] P. Subirats, J. Dumoulin, V. Legeay, and D. Barba, "Automation of Pavement Surface Crack Detection Using the Continuous Wavelet Transform," in *Proceedings of the International Conference on Image Processing*, 2006, pp. 3037–3040.

[175] X.-y. Wu, K. Xu, and J.-w. Xu, "Application of Undecimated Wavelet Transform to Surface Defect Detection of Hot Rolled Steel Plates," in *Proceedings of the Congress on Image and Signal Processing*, vol. 4, 2008, pp. 528–532.

[176] C. Park and S. Won, "An Automated Web Surface Inspection for Hot Wire Rod Using Undecimated Wavelet Transform and Support Vector Machine," in *Proceedings of the Annual Conference of IEEE Industrial Electronics*, 2009, pp. 2411–2415.

[177] Y.-J. Jeon, J. P. Yun, D.-c. Choi, and S. W. Kim, "Defect Detection Algorithm for Corner Cracks in Steel Billet Using Discrete Wavelet Transform," in *Proceedings of the International Conference on Control, Automation and Systems*, 2009, pp. 2769–2773.

[178] D.-c. Choi, Y.-J. Jeon, J. P. Yun, S. W. Yun, and S. W. Kim, "An Algorithm for Detecting Seam Cracks in Steel Plates," *International Journal of Industrial and Manufacturing Engineering*, vol. 6, no. 12, pp. 2835–2838, 2012.

[179] S. Ghorai, A. Mukherjee, M. Gangadaran, and P. K. Dutta, "Automatic Defect Detection on Hot-rolled Flat Steel Products," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 3, pp. 612–621, 2012.

[180] G. Ji, Y. Zhu, and Y. Zhang, "The Corroded Defect Rating System of Coating Material Based on Computer Vision," in *Transactions on Edutainment VIII*, 2012, pp. 210–220.

[181] Y.-J. Jeon, D.-c. Choi, S. J. Lee, J. P. Yun, and S. W. Kim, "Defect Detection for Corner Cracks in Steel Billets Using a Wavelet Reconstruction Method," *Journal of the Optical Society of America A*, vol. 31, no. 2, pp. 227–237, 2014.

[182] F. Tsutsumi, H. Murata, T. Onoda, O. Oguri, and H. Tanaka, "Automatic Corrosion Estimation Using Galvanized Steel Images on Power Transmission Towers," in *Proceedings of the Transmission & Distribution Conference & Exposition: Asia and Pacific*, 2009, pp. 1–4.

[183] J. Iivarinen, J. Rauhamaa, and A. Visa, "Unsupervised Segmentation of Surface Defects," in *Proceedings of International Conference on Pattern Recognition*, vol. 4, 1996, pp. 356–360.

[184] M. R. Yazdchi, A. G. Mahyari, and A. Nazeri, "Detection and Classification of Surface Defects of Cold Rolling Mill Steel Using Morphology and Neural Network," in *Proceedings of the International Conference on Computational Intelligence for Modelling Control & Automation*, 2008, pp. 1071–1076.

[185] H. Jia, Y. L. Murphey, J. Shi, and T.-S. Chang, "An Intelligent Real-time Vision System for Surface Defect Detection," in *Proceedings of the International Conference on Pattern Recognition*, vol. 3, 2004, pp. 239–242.

[186] K. Agarwal, R. Shivpuri, Y. Zhu, T.-S. Chang, and H. Huang, "Process Knowledge based Multi-class Support Vector Classification (PK-MSVM) Approach for Surface Defects in Hot Rolling," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7251–7262, 2011.

[187] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[188] J. Zhao, Y. Yang, and G. Li, "The Cold Rolling Strip Surface Defect On-line Inspection System based on Machine Vision," in *Proceedings of the Pacific-Asia Conference on Circuits, Communications and System*, vol. 1, 2010, pp. 402–405.

[189] M. Yamana, H. Murata, T. Onoda, and T. Ohashi, "Development of System for Crossarm Reuse Judgment on the Basis of Classification of Rust Images Using Support Vector Machine," in *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence*, 2005, pp. 5–pp.

[190] K. Choi, K. Koo, and J. S. Lee, "Development of Defect Classification Algorithm for POSCO Rolling Strip Surface Inspection System," in *Proceedings of the IEEE SICE-ICASE International Joint Conference*, 2006, pp. 2499–2502.

[191] P. Caleb and M. Steuer, "Classification of Surface Defects on Hot Rolled Steel Using Adaptive Learning Methods," in *Proceedings of the International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, vol. 1, 2000, pp. 103–108.

[192] L. A. Martins, F. L. Pádua, and P. E. Almeida, "Automatic Detection of Surface Defects on Rolled Steel Using Computer Vision and Artificial Neural Networks," in *Proceedings of the IEEE Annual Conference on Industrial Electronics Society*, 2010, pp. 1081–1086.

[193] G. Wu, H. Zhang, X. Sun, J. Xu, and K. Xu, "A Bran-new Feature Extraction Method and Its Application to Surface Defect Recognition of Hot Rolled Strips," in *Proceedings of the IEEE International Conference on Automation and Logistics*, 2007, pp. 2069–2074.

[194] L. J. Olsson and S. Gruber, "Web Process Inspection Using Neural Classification of Scattering Light," *IEEE Transactions on Industrial Electronics*, vol. 40, no. 2, pp. 228–234, 1993.

[195] A. Ortiz, F. Bonnín-Pascual, and E. Garcia-Fidalgo, "Visual Inspection of Vessels by Means of a Micro-Aerial Vehicle: An Artificial Neural Network Approach for Corrosion Detection," in *Proceedings of the Iberian Robotics Conference*, 2016, pp. 223–234.

[196] A. Ortiz, K. Yao, F. Bonnín-Pascual, E. Garcia-Fidalgo, and J. P. Company-Corcoles, "New Steps towards the Integration of Robotic and Autonomous Systems in the Inspection of Vessel Holds," in *Proceedings of the Spanish National Robotics Conference*, 2018.

[197] A. Ortiz, F. Bonnín-Pascual, E. Garcia-Fidalgo, J. P. Company-Corcoles, and K. Yao, "Visual Inspection of Vessels Cargo Holds: Use of a Micro-Aerial Vehicle

as a Smart Assistant," in *International Workshop on Metrology for the Sea*, 2019, pp. 221–226.

[198] J.-K. Park, B.-K. Kwon, J.-H. Park, and D.-J. Kang, "Machine Learning-based Imaging System for Surface Defect Inspection," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 3, no. 3, pp. 303–310, 2016.

[199] L. Petricca, T. Moss, G. Figueroa, and S. Broen, "Corrosion Detection Using A.I : A Comparison of Standard Computer Vision Techniques and Deep Learning Model," in *Proceedings of the Computer Science & Information Technology*. Academy & Industry Research Collaboration Center (AIRCC), may 2016.

[200] R. Ren, T. Hung, and K. C. Tan, "A Generic Deep-learning-based Approach for Automated Surface Inspection," *IEEE Transactions on Cybernetics*, vol. 48, no. 3, pp. 929–940, 2017.

[201] R. Oullette, M. Browne, and K. Hirasawa, "Genetic Algorithm Optimization of a Convolutional Neural Network for Autonomous Crack Detection," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 1, 2004, pp. 516–521.

[202] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road Crack Detection Using Deep Convolutional Neural Network," in *Proceedings of the IEEE International Conference on Image Processing*, 2016, pp. 3708–3712.

[203] L. Yang, B. Li, W. Li, Z. Liu, G. Yang, and J. Xiao, "Deep Concrete Inspection Using Unmanned Aerial Vehicle Towards CSSC Database," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 24–28.

[204] Y.-J. Cha, W. Choi, and O. Büyüköztürk, "Deep Learning-based Crack Damage Detection Using Convolutional Neural Networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.

[205] M. Eisenbach, R. Stricker, K. Debes, and H.-M. Gross, "Crack Detection with an Interactive and Adaptive Video Inspection System," *Arbeitsgruppentagung Infrastrukturmanagement*, vol. 94, 2017.

[206] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout, "Steel Defect Classification with Max-pooling Convolutional Neural Networks," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2012, pp. 1–6.

[207] K. Peng and X. Zhang, "Classification Technology for Automatic Surface Defects Detection of Steel Strip based on Improved BP Algorithm," in *Proceedings of the International Conference on Natural Computation*, vol. 1, 2009, pp. 110–114.

[208] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," in *Proceedings of the International Conference on 3D Vision (3DV)*, 2016, pp. 565–571.

[209] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "A Deep Convolutional Activation Feature for Generic Visual Recognition," UC Berkeley & ICSI, Berkeley, CA, USA, Tech. Rep.

[210] K. Makantasis, E. Protopapadakis, A. Doulamis, N. Doulamis, and C. Loupos, "Deep Convolutional Neural Networks for Efficient Vision Based Tunnel Inspection," in *Proceedings of the IEEE International Conference on Intelligent Computer Communication and Processing*, 2015, pp. 335–342.

[211] E. Protopapadakis, A. Voulodimos, A. Doulamis, N. Doulamis, and T. Stathaki, "Automatic Crack Detection for Tunnel Inspection Using Deep Learning and Heuristic Image Post-processing," *Applied Intelligence*, vol. 49, no. 7, pp. 2793–2806, 2019.

[212] A. Shihavuddin, X. Chen, V. Fedorov, A. Nymark Christensen, N. Andre Brogaard Riis, K. Branner, A. Bjorholm Dahl, and R. Reinhold Paulsen, "Wind Turbine Surface Damage Detection by Deep Learning Aided Drone Inspection Analysis," *Energies*, vol. 12, no. 4, p. 676, 2019.

[213] M. Siegel and P. Gunatilake, "Remote Enhanced Visual Inspection of Aircraft by a Mobile Robot," in *Proceedings of the IEEE Workshop on Emerging Technologies, Intelligent Measurement and Virtual Systems for Instrumentation and Measurement*, 1998, pp. 49–58.

[214] F. Dupont, C. Odet, and M. Cartont, "Optimization of the Recognition of Defects in Flat Steel Products with the Cost Matrices Theory," *NDT & E International*, vol. 30, no. 1, pp. 3–10, 1997.

[215] S. Cateni, V. Colla, M. Vannucci, and A. Borselli, "Fuzzy Inference Systems Applied to Image Classification in the Industrial Field," in *Fuzzy Inference System: Theory and Applications*. InTech, 2012.

[216] A. Borselli, V. Colla, M. Vannucci, and M. Veroli, "A Fuzzy Inference System Applied to Defect Detection in Flat Steel Production," in *Proceedings of the International Conference on Fuzzy Systems*, 2010, pp. 1–6.

[217] B. Shi, X. Bai, and S. Belongie, "Detecting Oriented Text in Natural Images by Linking Segments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2550–2558.

[218] Y. Dai, Z. Huang, Y. Gao, Y. Xu, K. Chen, J. Guo, and W. Qiu, "Fused Text Segmentation Networks for Multi-oriented Scene Text Detection," in *Proceedings of the International Conference on Pattern Recognition*, 2018, pp. 3604–3609.

[219] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented Scene Text Detection Via Corner Localization and Region Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7553–7563.

[220] S. Qin and R. Manduchi, "Cascaded Segmentation-detection Networks for Word-level Text Spotting," in *Proceedings of the IEEE International Conference on Document Analysis and Recognition*, vol. 1, 2017, pp. 1275–1282.

[221] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic Data for Text Localisation in Natural Images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2315–2324.

[222] Z. Li and F. Zhou, "FSSD: Feature Fusion Single Shot Multibox Detector," *arXiv preprint arXiv:1712.00960*, 2017.

[223] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional Single Shot Detector," *arXiv preprint arXiv:1701.06659*, 2017.

[224] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten Digit Recognition with a Back-Propagation Network," *Advances in Neural Information Processing Systems*, vol. 2, 1989.

[225] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[226] S. Bell, P. Upchurch, N. Snavely, and K. Bala, "OpenSurfaces: A Richly Annotated Catalog of Surface Appearance," *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 1–17, 2013.

[227] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and Semi-supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1742–1750.

[228] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all You Need," in *Proceedings of the Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[229] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What Does Bert Look at? An Analysis of BERT's Attention," *arXiv preprint arXiv:1906.04341*, 2019.

[230] S. Serrano and N. A. Smith, "Is Attention Interpretable?" *arXiv preprint arXiv:1906.03731*, 2019.

[231] S. Jain and B. C. Wallace, "Attention is Not Explanation," *arXiv preprint arXiv:1902.10186*, 2019.

[232] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[233] S. Jetley, N. A. Lord, N. Lee, and P. H. Torr, "Learn to Pay Attention," *arXiv preprint arXiv:1804.02391*, 2018.

[234] A. Sinha and J. Dolz, "Multi-Scale Self-Guided Attention for Medical Image Segmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 1, pp. 121–130, jan 2021.

[235] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A Discriminative Feature Learning Approach for Deep Face Recognition," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 499–515.

[236] X. Peng, I. W. Tsang, J. T. Zhou, and H. Zhu, "K-Means Net: When K-Means Meets Differentiable Programming," *arXiv preprint arXiv:1808.07292*, 2018.

[237] H. Kervadec, J. Dolz, M. Tang, E. Granger, Y. Boykov, and I. B. Ayed, "Constrained-CNN Losses for Weakly Supervised Segmentation," *Medical Image Analysis*, vol. 54, pp. 88–99, 2019.

[238] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.