People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Shahid Hama Lakhdar - El Oued
Computer Science Department

Doctorate science thesis in Computer Science

Object Detection and Identification Based on Contour Segmentation

By
Sana Sahar Guia

Defended on November 06, 2024. Examination Committee :
Dr. Abdelkamel Ben Ali (President)
Pr. Abdelkader Laouid (Supervisor)
Dr. Youcef Elmir (Examiner)
Dr. Hichem Rahab (Examiner)
Dr. Essaid Kadri (Examiner)
Dr. Mohammed Anouar Naoui (Examiner)

Academic year: 2024/2025

République Algérienne Démocratique Et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

---

Université Shahid Hama Lakhdar - El Oued
Département D'Informatique

Thèse pour l'obtention du grade de Docteur Science En Informatique

---

Détection et Identification d'Objets Fondée sur la Segmentation par Contour

---

Présenté par
Sana Sahar Guia

Soutenue le 06 November 2024, Devant le jury :

Dr. Abdelkamel Ben Ali (Président)
Pr. Abdelkader Laouid (Encadreur)
Dr. Youcef Elmir (Examinateur)
Dr. Hichem Rahab (Examinateur)
Dr. Essaid Kadri (Examinateur)
Dr. Mohammed Anouar Naoui (Examinateur)

Année Universitaire : 2024/2025

Acknowledgments

First and foremost, I thank ALLAH for granting me the strength and patience needed to complete this work.

I would like to express my deep gratitude to my thesis director, Doctor Abdelkader Laouid, for guiding me from the beginning and helping me take the most efficient path to achieve my goal. His knowledge, advice, patience, availability, and support throughout these three years have been invaluable.

My warmest thanks also go to my thesis co-supervisor, Dr. Amna Eleyan, a Programme Support Tutor at the University of Manchester, and to Mr. Mohammad Hammoudeh, a Professor at King Fahd University of Petroleum and Minerals, for their constant guidance and mentorship.

I extend my heartfelt thanks to Dr. Mohammed Amine Yagoub and Dr. Mostafa Kara, both from the University of Eloued, for their assistance and support.

I would also like to express my sincere thanks to all the members of the jury for honoring me with their agreement to evaluate this thesis and for their interest in my work.

I sincerely thank all those whose encouragement contributed to the completion of my doctoral thesis.

Finally, my deepest gratitude goes to my loved ones, without whom I could never have reached this milestone. I think of my parents, my family, my brothers, and my sister.

Sana Sahar Guia

Contents

List of Figures

List of Tables

Abstract

Object Detection and Identification Based on Contour Segmentation

Artificial intelligence has been applied across various domains. Despite its advantages, there are significant limitations that hinder the implementation of artificial intelligence techniques in specific fields and locations. Major concerns include accuracy, the poor quality of gathered data, and processing time, particularly when deploying machine learning techniques on low-end smart devices.

In the field of computer vision, advancements in multimedia and visual computing technologies have the potential to revolutionize human life. Many applications utilize captured images from autonomous entities as data sources for various purposes. These images must be interpreted to extract information about their external environment. Researchers in this domain face challenges such as detecting and interpreting the context of these images.

The proposed approach introduces a new pipeline that reduces the size of the image in both the input and intermediate layers of a deep learning model by merging pixels based on their standard deviation values rather than processing the entire image. Experimental results demonstrate that this technique significantly enhances the performance of existing text detection methods, especially in challenging scenarios involving low-end IoT devices that offer low contrast or noisy backgrounds. Compared to other techniques, the proposed method can be effectively used for object detection in IoT-gathered multimedia data, achieving reasonable accuracy with short computation times.

Keywords: Object detection, Image segmentation, deep learning, Standard deviation, IoT Image Processing

Résumé

Détection et Identification d'Objets Fondée sur la Segmentation par Contour

L'intelligence artificielle a été appliquée dans divers domaines. Malgré ses avantages, il existe des limitations significatives qui entravent l'implémentation des techniques d'intelligence artificielle dans certains domaines et endroits spécifiques. Les principales préoccupations incluent la précision, la mauvaise qualité des données recueillies et le temps de traitement, en particulier lors du déploiement de techniques d'apprentissage automatique sur des appareils intelligents de bas de gamme.

Dans le domaine de la vision par ordinateur, les avancées en matière de multimédia et de calcul visuel ont le potentiel de révolutionner la vie humaine. De nombreuses applications utilisent des images capturées par des entités autonomes comme sources de données pour divers objectifs. Ces images doivent être interprétées pour extraire des informations sur leur environnement extérieur. Les chercheurs de ce domaine sont confrontés à des défis tels que la détection et l'interprétation du contexte des images.

L'approche proposée introduit un nouveau pipeline qui réduit la taille de l'image dans les couches d'entrée et intermédiaires d'un modèle d'apprentissage profond en fusionnant les pixels basés sur leurs valeurs d'écart type au lieu de traiter l'image entière. Les résultats expérimentaux montrent que cette technique améliore significativement les performances des méthodes de détection de texte existantes, notamment dans des scénarios difficiles impliquant des appareils IoT de bas de gamme offrant un faible contraste ou des arrière-plans bruyants. Comparée à d'autres techniques, la méthode proposée peut être efficacement utilisée pour la détection d'objets dans les données multimédias recueillies par des appareils IoT, atteignant une précision raisonnable avec un temps de calcul réduit.

Mots clés: Détection d'objets, Segmentation d'image, Apprentissage profond, Equart type, traitement d'images IoT.

ملـخـص

الكشف عن الكائنات وتحديدها بناءً على تجزئة الصور باستعمال الحواف

تم تطبيق الذكاء الاصطناعي في عدة مجالات. على الرغم من مزايا استخدام تقنيات الذكاء الاصطناعي، إلا أن هناك بعض القيود الهامة التي تعيق تنفيذها في مجالات وأماكن محددة. تعد الدقة، وسوء جودة البيانات المجمعة، ووقت المعالجة من أهم القضايا التي تواجه تنفيذ تقنيات التعلم الآلي، خاصة على الأجهزة الذكية منخفضة الأداء.

في مجال الرؤية الحاسوبية، تتيح التطورات في الوسائط المتعددة والحوسبة البصرية إمكانيات جديدة تغير حياة الإنسان بشكل كبير. تستغل العديد من التطبيقات الصور الملتقطة بواسطة الكيانات الذاتية كبيانات لتحقيق أهداف مختلفة. تحتاج هذه الصور إلى تفسير لاستخلاص معلومات عن بيئتها الخارجية. يواجه الباحثون في هذا المجال تحديات مثل كيفية اكتشاف وتفسير سياق الصور.

تقترح الأطروحة نهجًا جديدًا يقلل حجم الصورة في الطبقات المدخلة والوسطى لنموذج التعلم العميق بناءً على دمج البكسلات باستخدام قيم الانحراف المعياري بدلاً من معالجة الصورة بأكملها. تُثبت النتائج التجريبية أن هذه التقنية تحسن بشكل كبير أداء أساليب اكتشاف النصوص الحالية، خاصة في السيناريوهات الصعبة مثل استخدام أجهزة إنترنت الأشياء منخفضة الأداء التي تقدم تباينًا منخفضًا أو خلفيات مشوشة. بالمقارنة مع التقنيات الأخرى، يمكن استخدام التقنية المقترحة بشكل فعال لاكتشاف الأجسام في بيانات الوسائط المتعددة التي تجمعها أجهزة إنترنت الأشياء، مما يحقق دقة معقولة في وقت حسابي قصير.

الكلمات المفتاحية: اكتشاف الكائنات، تجزئة الصور، التعلم العميق، الانحراف المعياري، معالجة الصور في إنترنت الأشياء.

General Introduction

Introduction

Nowadays, Image analysis has become increasingly important in a wide variety of applications, especially in such areas where the automatic searching in images is a challenge.

Object detection, segmentation, and identification are the most fundamental yet challenging problems in image processing and computer vision community. Their applications and use reside in many domains such as robotics, image processing, recognition, and others. Detecting an object in an image is an operation that aims to identify the presence of various individual objects in a given image or video from distinguishing the region of the object from the background according to a defined criterion. Several techniques in the literature could be found to ensure this operation, which is the first step towards object recognition.

On the other hand the Internet of Things (IoT) plays a significant role in the domain of object detection, further emphasizing the importance of this technology in today's digital landscape.

IoT devices, which encompass a vast array of sensors, cameras, and other data-gathering tools, generate a massive amount of multimedia data. This data comes from various sources, including surveillance cameras, environmental sensors, smart appliances, and wearable devices, among others.

With the proliferation of IoT devices, there's an unprecedented volume and diversity of information being generated continuously. This influx of data presents both challenges and opportunities, particularly in the realm of object detection.

Object detection algorithms can be deployed on IoT devices themselves or on edge computing platforms to analyze the data locally, reducing latency and bandwidth usage. This capability is crucial in applications where real-time detection and response are essential, such as in smart cities for traffic monitoring, in industrial settings for quality control, or in healthcare for patient monitoring.

Context

General context Computer vision has indeed emerged as a major research topic within computer science, driven by advancements in artificial intelligence and deep learning. While artificial intelligence (AI) has shown tremendous potential across various domains, there are indeed several limitations that can hinder its implementation, especially in certain contexts such as low-end smart devices. Some of the key limitations include:

1. Accuracy: The accuracy of AI models heavily depends on the quality and quantity of training data available. In domains where data is scarce or noisy, achieving high accuracy can be challenging. Additionally, AI models may struggle with generalizing to unseen data or adapting to changing conditions, leading to potential inaccuracies in real-world applications.

2. Data Quality: Poor quality or biased data can significantly impact the performance of AI models. Biases in training data can lead to unfair or discriminatory outcomes, especially in sensitive domains like healthcare or criminal justice. Ensuring high-quality and diverse training data is essential for mitigating these issues.

3. Processing Time: AI algorithms, particularly deep learning models, can be computationally intensive and require significant processing power and memory resources. This poses a challenge for implementation on low-end smart devices with limited computational capabilities. Real-time or near-real-time inference may be difficult to achieve in such constrained environments.

4. Resource Constraints: Low-end smart devices often have limited memory, storage, and power resources, which can pose constraints on the size and complexity of AI models that can be deployed. Optimizing models for resource efficiency while maintaining acceptable performance becomes crucial in such scenarios.

Specific context Object detection, segmentation, and identification lay the foundation for many advanced image processing and computer vision tasks.

Object detection involves locating instances of objects within images or videos, typically by creating bounding boxes around them. This process is essential for tasks like autonomous driving, surveillance, medical imaging, and industrial automation.

Segmentation, on the other hand, goes a step further by precisely delineating the boundaries of objects, assigning each pixel to a specific class or object category.

This level of detail is crucial in medical image analysis, satellite imaging, and quality control in manufacturing.

Identification involves recognizing and labeling specific objects or instances within an image. This could range from identifying different species of plants in agriculture to recognizing faces in security systems.

Deep learning, particularly convolutional neural networks (CNNs), has revolutionized object detection and image analysis by enabling end-to-end learning from raw pixel data to object detection and classification.

As technology continues to evolve, we can expect even more sophisticated methods and applications of image analysis and object detection across various domains.

Furthermore, integrating object detection with IoT devices enables a wide range of applications, including smart retail (tracking inventory and analyzing customer behavior), environmental monitoring (detecting pollution or changes in wildlife habitats), and home automation (recognizing occupants and adjusting settings accordingly).

The synergy between IoT and object detection not only underscores the abundance of multimedia data but also presents new avenues for innovation and application across various domains.

Motivation

A crucial aspect of deep learning-based approaches in object detection is striking the right balance between accuracy and speed. While Convolutional Neural Networks (CNNs) have demonstrated remarkable effectiveness in object detection tasks by automatically learning features from data, achieving a balance between accuracy and speed remains a fundamental goal.

In object detection systems, especially those deployed in real-time applications or on resource-constrained devices, both accuracy and speed are critical factors:

- Accuracy: Accuracy refers to how well the object detection system can correctly identify and localize objects in an image or video. Higher accuracy ensures reliable performance, especially in tasks where precision is paramount, such as medical imaging or autonomous driving.
- Speed: Speed, on the other hand, refers to the efficiency of the object detection system in processing input data and generating predictions. Faster processing times are essential for real-time applications like video surveillance, where timely detection and response are crucial.

Research question

When designing object detection systems, these considerations are crucial, especially in real-world scenarios where factors like model complexity, real-time constraints, and resource constraints play significant roles:

1. Model Complexity: Deep learning models, especially deep neural networks like CNNs, may offer high accuracy by learning intricate features from data. However, they often come with a high computational cost due to their complexity. This can be challenging for deployment on devices with limited processing power and memory, such as low-end IoT devices or edge computing platforms.

2. Real-Time Constraints: Many applications, such as autonomous vehicles, video surveillance, and robotics, require low-latency predictions for timely decision-making. High computational complexity can lead to longer inference times, which may not be feasible in real-time scenarios. Achieving real-time performance while maintaining accuracy is crucial for the success of these applications.

3. Resource Constraints: Low-end devices, including IoT devices and embedded systems, often have limited computational resources, memory, and power. Deploying complex deep learning models on such devices may not be practical due to resource constraints. Optimizing models for resource efficiency while preserving accuracy becomes essential in these scenarios.

Objective

Detecting and segmenting objects in natural scenes has attracted a lot of interest in computer vision. Despite many models have been proposed and several applications have emerged, yet a deep understanding of achievements and issues is lacking.

The primary goal of this thesis is to achieve a delicate balance between accuracy and speed in the development of object extraction systems. The significant contribution lies in addressing this balance, ensuring that these systems not only maintain high accuracy but also operate efficiently in terms of speed. The purpose of the thesis is to tackle the challenges posed by real-time constraints, resource limitations, and model complexity. To address these challenges, we propose a novel text detection approach.

In addition to achieving a delicate balance between accuracy and speed, this thesis aims to improve image object detection by reducing image size while preserving essential features. The proposed reduction technique enhances model training efficiency without sacrificing accuracy, thus addressing the challenges posed by real-time constraints, resource limitations, and model complexity.

Contribution

This thesis contributes to the ongoing efforts in advancing object detection techniques by introducing a novel pre-treatment technique that addresses important challenges in model training and performance, particularly in the context of low-end IoT devices. It would be fascinating to see how this approach evolves and potentially influences future research and development in the field of computer vision.

Our work presents several distinct advantages that set it apart from existing literature:

1. Reduced Processing Time : - Our approach significantly diminishes the computational overhead typically associated with fine-tuning or adapting these models to specific tasks. - This leads to faster object detection, making it suitable for real-time or resource-constrained applications.
2. Customization and Adaptability: - Our model offers greater flexibility in adapting to various domains and datasets. - Researchers and practitioners can tailor the algorithm more effectively to their specific needs. - This adaptability ensures better performance across diverse scenarios.
3. Competitive Precision Rates: - Our approach maintains competitive precision rates. - Ensuring that the accuracy of object detection is not compromised while benefiting from reduced processing time and enhanced adaptability.

Presentation Order

This thesis contains four chapters. The general introduction presents the context and the research problem. It identifies the motivations and objectives of the thesis and also presents the main contributions.

Chapter 1 introduces the Deep Learning approach employed in computer vision, focusing on Convolutional Neural Networks (CNNs) for tasks such as semantic

segmentation and object detection. CNNs are particularly well-suited for these tasks due to their ability to automatically learn hierarchical features from raw input data.

Chapter 2 presents the color divergence technique utilized for image segmentation and object detection. This method leverages color information to segment images and detect objects within them.

In Chapter 3, the focus is on the application of the σNet approach for object detection of text in natural scene images. The chapter delves into the technical details of the σNet approach, its architecture, training process, and performance evaluation.

The chapter 4 addresses the multi-class challenge in colorectal cancer detection using a σNet with a pre-trained convolutional neural network and a spatial attention module.

Finally, the general conclusion consolidates all the insights and contributions gained throughout the thesis, and the research's perspectives and future directions are provided.

List of publications

Journal papers:

1. Co-Simulation of Multiple Vehicle Routing Problem Models. Future Internet 2022.[2]

   Authors Sana Sahar Guia, Abdelkader Laouid, Mohammad Hammoudeh, Ahcène Bounceur, Mai Alfawair and Amna Eleyan.

2. Intelligent Image Text ComputerDetection via Pixel Standard Deviation Representation. Systems Science and Engineering 2024.[3]

   Authors Sana Sahar Guia, Abdelkader Laouid, Mohammad Hammoudeh and Mostafa Kara.

Conference papers:

1. A Salient Object Detection Technique Based on Color Divergence. ICFNDS '20: The 4th International Conference on Future Networks and Distributed Systems - St. Petersburg Russian Federation, November, 2020 [4]

   Authors: Sana Sahar Guia, Abdelkader Laouid, Reinhardt Euler, Mohammed Amine Yagoub, Ahcène Bounceur, Mohammad Hammoudeh.

2. Categorization of Digital Pathology Image using Deep Learning model. The 7th International Conference on Future Networks and Distributed Systems. Dubai UEA, December 2023 [5]

   Authors: Sana Sahar Guia, Abdelkader Laouid, Khaled Chait.

Chapter 1

Preliminary knowledge

1.1 Introduction

The Deep Learning approach in computer vision predominantly utilizes Convolutional Neural Networks (CNNs). CNNs have revolutionized computer vision tasks, including semantic segmentation and object detection, due to their ability to learn spatial hierarchies and capture local and global context. The content within this chapter focuses on the foundational knowledge of this thesis. First, we describe the principles of Convolutional Neural Networks (CNNs). Then, we clarify the application and implementation of encoder-decoder networks in the context of semantic segmentation. Finally, we introduce the task of object detection.

1.2 Convolutional Neural Networks CNNs

1.2.1 Overview

The cnn is a sophisticated deep learning method commonly employed for detecting objects. Initially introduced by Fukushima in 1980 [6] and later refined by LeCun [7], CNNs are structured to handle data represented as multi-dimensional arrays. They incorporate four fundamental concepts that exploit the characteristics of natural signals: localized connectivity, weight sharing, pooling strategies, and the implementation of numerous layers [8] The architecture of a typical Convolutional Neural Network (CNN) model is composed of a series of sequential layers detailed as follows :

- Convolutional layers: These layers are crucial for extracting features. Early layers detect simple features such as edges and corners, while subsequent layers identify complex features like structures and objects by integrating

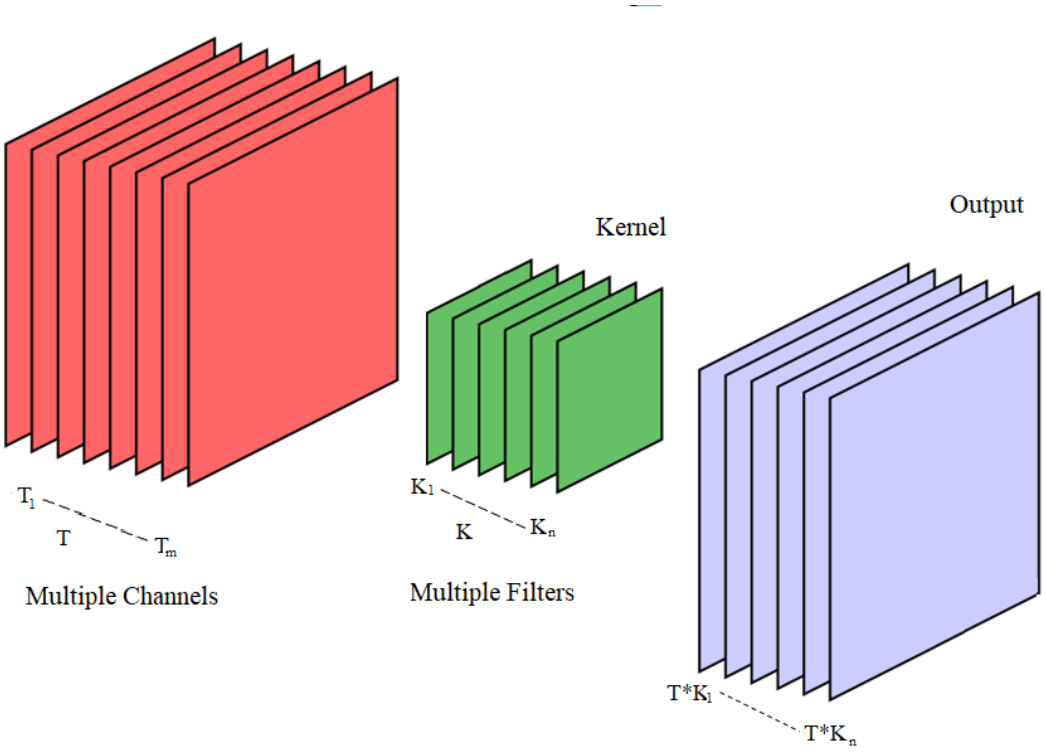


Figure 1.1: Convolution with multiple filters from multiple channels

the simpler ones. Each neuron within a convolutional layer connects to a small region of the preceding layer's feature maps via a kernel set known as a filter bank as shown in figure 1.1. The local weighted sum produced is then processed through a nonlinear function, commonly a rectified linear unit (ReLU). Neurons in the same feature map utilize identical filter banks, whereas different feature maps employ distinct ones. The equation 1.1 describes the convolution operation between a multi-channel input tensor T and a filter kernel K_l.

$$F(x, y, l) = (T * K_l)(x, y, l) = \sum_m \sum_i \sum_j T(x - i, y - j, m)\ K(i, j, l) \quad (1.1)$$

- Pooling layers: The purpose of pooling layers is to condense the data representation and achieve robustness against minor positional changes and distortions. Positioned typically between convolutional layers, each pooling layer's feature map links to the corresponding feature map of the preceding convolutional layer. A standard pooling operation calculates the maximum

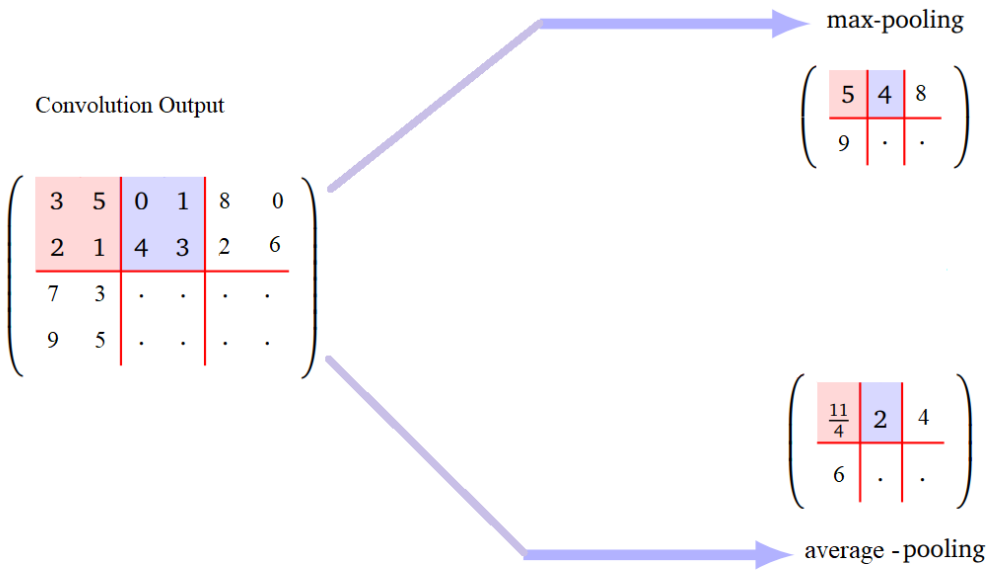


Figure 1.2: Max and Average Pooling example

value within a local cluster of neurons in a single feature map. Figure 1.2 illustrates an example of max and average pooling process.

- Nonlinear Activation Function : The activation functions are essential in neural networks as they establish a non-linear relationship between the input and output. This non-linearity allows the network to handle complex, non-linear data. Choosing an appropriate activation function is crucial for enhancing network performance [9]. Widely used activation functions are the Sigmoid and Rectified Linear Unit (ReLU) [10], each contributing uniquely to the network's learning capabilities.

- Fully connected layers: These layers are generally found near the end of the network and serve to synthesize the information from the preceding layers, aiding in the final decision-making process. Figure 1.3 shows different layers of the CNN, including the convolution layer, pooling layer, and fully connected layer.

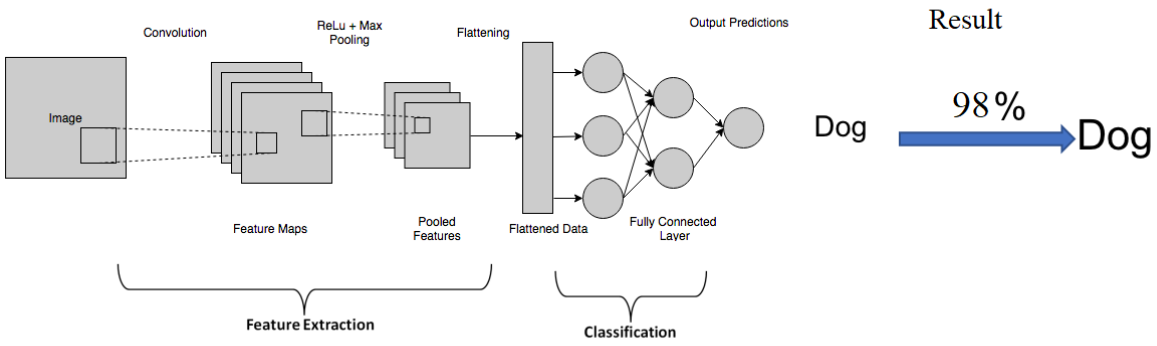


Figure 1.3: CNN example [1]

1.2.2 Applications of CNNs

1.2.2.1 Image classification with CNNs

Feature extraction is a crucial step in image classification. It involves transforming the raw data of the image into a set of features that can be effectively used by a classifier to distinguish between different object categories [11].

There are two main approaches to feature extraction:

- Manual Feature Extraction: This involves selecting and designing features based on domain knowledge. For example, color, texture, and shape descriptors are manually crafted and extracted from the image.
- Feature Learning: This approach relies on machine learning algorithms to automatically discover the representations needed for classification or other tasks.

The advent of Deep Convolutional Neural Networks (DCNNs) has revolutionized the field of image classification [12, 13, 14, 15]. Deep learning, particularly Convolutional Neural Networks (CNNs), is a popular method for feature learning. CNNs can automatically learn hierarchical feature representations from the data. The key factors contributing to the success of DCNNs include:

- Hierarchical Feature Learning: DCNNs learn multiple layers of features automatically, from simple to complex, capturing various aspects of the visual content.

- Shared Weights Architecture: This reduces the number of parameters in the network, making it efficient and less prone to overfitting.
- Non-linearity: The use of non-linear activation functions like ReLU helps the network learn complex patterns.
- Pooling Layers: These reduce the spatial size of the representation, making the network invariant to small translations of the input.
- End-to-End Training: DCNNs can be trained in an end-to-end manner with backpropagation, allowing the network to adjust its filters to improve the task performance

1.2.2.2 Bounding box object detection using CNNs

Convolutional Neural Networks (CNNs) can be effectively used for bounding box object detection. This process involves training a CNN to not only classify objects within an image but also to localize them by predicting the coordinates of bounding boxes around each object. Techniques like bounding box regression [16, 17] are commonly used for this purpose. Additionally, architectures such as YOLO (You Only Look Once) have been specifically designed to perform object detection using CNNs by simultaneously predicting class probabilities and bounding box coordinates

1.2.2.3 Part and key point prediction with CNNs

Keypoint detection is a critical preprocessing step in various computer vision tasks involving faces. It is essential for creating face models and plays a significant role in the processes of recognition and identity verification [18]. Moreover, in the realm of 3D shape segmentation and keypoint prediction, it facilitates the forecasting of vertex functions, including part segments or keypoints, even on irregular data structures [19]. Convolutional Neural Networks (CNNs) are highly effective for part and keypoint prediction tasks. These tasks involve identifying specific parts of objects or keypoints, which are significant points of interest within an image. CNNs can learn to recognize these features through training on labeled datasets [20, 21].

1.2.2.4 Local correspondence using CNNs

Establishing reliable dense correspondences between two images is a complex task, made difficult by significant differences in appearance among corresponding elements and the confusion caused by repetitive patterns. This process is essential

for various computer vision applications, such as 3D reconstruction and visual localization [22].

Convolutional Neural Networks (CNNs) are increasingly being used to establish local correspondence between images. This involves identifying points in one image that correspond to points in another, despite variations in scale, rotation, and lighting [23, 24]

1.2.3 Advanced Architecture Design for Artificial Neural Networks (ANNs)

One kind of machine learning model that draws inspiration from the composition and operations of the human brain is called an artificial neural network (ANN). They are made up of layers of interconnected nodes, or neurons. These networks have the ability to extract intricate patterns from data [25]. CNNs are a specific type of ANN designed to process image data. In general, ANNs are a flexible tool for many machine learning applications, such as time series analysis, audio recognition, and natural language processing.

1.2.3.1 Widespread CNNs models

AlexNet is renowned as a pioneering Convolutional Neural Network (CNN) and made a significant impact by participating in the 2012 ImageNet Large Scale Visual Recognition Challenge [26].

Following the success of AlexNet, a series of influential CNNs were developed. Network-In-Network (NIN) [27] demonstrated state-of-the-art classification performances on various datasets. VGG-Net [13] consists of very deep convolutional networks that achieve state-of-the-art performance in the ImageNet Challenge 2014. GoogLeNet [28] introduced the inception module, which allows for more efficient computation and deeper architectures without a significant increase in the number of parameters. ResNet [26] utilized residual connections to enable the training of very deep networks by addressing the vanishing gradient problem. DenseNet [29] connected each layer to every other layer in a feed-forward fashion to ensure maximum information flow between layers in the network.

These networks have been introduced one after another, contributing to the rapid evolution and remarkable achievements of CNNs in the realm of computer vision [30]. They have set the stage for the development of even more sophisticated and efficient architectures in the years that followed.

1.2.3.2 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a class of neural networks designed to handle sequential data, such as time series, natural language, and speech. RNNs process input sequences one step at a time while maintaining hidden states that capture context, allowing them to remember past inputs and use that information to make predictions about future outputs. This feedback loop enables information to persist across different time steps [31].

However, traditional RNNs suffer from the vanishing gradient problem, which limits their ability to capture long-term dependencies.

Long Short-Term Memory (LSTM) networks address this issue by introducing gating mechanisms. LSTMs consist of four interacting layers: the input gate, forget gate, cell state, and output gate. These gates allow LSTMs to learn long-term dependencies and prevent both decaying and exploding gradients.

Gated Recurrent Units (GRUs) are similar to LSTMs but feature a simplified architecture. They combine the input and forget gates into a single update gate, making GRUs computationally efficient while maintaining strong performance across various tasks [32].

While CNNs excel at grid-structured data like images, RNNs handle sequential data with memory. Understanding their strengths and limitations helps choose the right architecture for specific tasks

1.2.3.3 Encoder-Decoder Models

An encoder-decoder is a type of neural network architecture used for sequence-to-sequence learning. It consists of two main components: the encoder and the decoder [33].

- Encoder: The encoder processes an input sequence (such as text, images, or other data) and extracts essential features. It creates a concise representation (often a fixed-length vector) that captures the relevant information from the input. The encoder's role is to encode the input data into a meaningful format.
- Decoder: The decoder takes the encoded representation (output by the encoder) and generates an output sequence. It acts as a conditional language model, predicting the subsequent tokens in the target sequence.

Encoder-decoder models are versatile and find applications beyond translation, including:

- Machine Translation: Encoder-decoder models are widely used for translating text from one language to another.
- Image Segmentation: They can also be applied to segment images into meaningful regions [34].
- Text Summarization: Encoder-decoder architectures help summarize long articles into concise summaries.

These models have significantly advanced various fields by providing a robust framework for processing and generating sequential data.

1.2.3.4 Attention mechanism

The attention mechanism in neural networks is inspired by the human visual system's ability to focus on specific parts of the visual field. In the context of Convolutional Neural Networks (CNNs), attention mechanisms can significantly enhance model performance by directing the network's focus to the most informative elements of the input. Channel attention mechanisms assess the importance of each channel (feature map) to highlight more pertinent features. Conversely, spatial attention mechanisms prioritize different spatial locations within the feature maps. Both types of attention can be seamlessly integrated into existing CNN architectures, serving as plug-and-play modules that sharpen the network's focus. Such selective attention enables the model to make more accurate predictions by allocating computational resources to the most prominent features , improving performance on tasks like image captioning and visual question answering. They are versatile improvements that can be integrated into various CNN architectures to improve their ability to capture long-range feature interactions and boost representational power [35]

1.2.3.5 Generative Adversial Networks (GANs)

Generative Adversarial Networks (GANs) are a powerful class of neural networks that can learn to generate new data that is similar to the input data, without needing extensively labeled training data. The two networks, the generator and the discriminator, work in tandem where the generator creates data and the discriminator evaluates it. Through this adversarial process, GANs can learn robust representations. The applications like image synthesis, semantic image editing, style transfer, image super-resolution, and classification, showcase the versatility of GANs. They have become a significant tool in the field of machine learning for their ability to generate high-quality, realistic images and their potential in unsupervised learning tasks [36].

1.3 Exploring Fully convolutional networks FCNs

Fully Convolutional Networks (FCNs) are a pivotal architecture in the field of computer vision. They streamline the process of learning dense predictions and are efficient both in terms of computational resources and performance, thus avoiding the need for patchwise training. This method eliminates the need for complex pre- and post-processing steps such as superpixels, proposals, and refinements by random fields or local classifiers. It also allows for end-to-end training without the need for supervised pre-training on small convnets [37]. This approach has been transformative for tasks that require an understanding of the image at a pixel level, like semantic segmentation, and has set a new standard for efficiency and effectiveness in deep learning architectures for vision.

1.3.1 Downsampling and upsampling

FCNs are composed entirely of locally connected layers such as convolution, pooling, and upsampling layers [37, 38]. They omit dense layers, which reduces the number of parameters and speeds up training.

1.3.1.1 Recap

Downsampling and upsampling are two fundamental processes in signal processing, particularly in the context of digital signal processing (DSP) [39].

- Downsampling, also known as decimation, is the process of reducing the sample rate of a signal by removing some of the sample points. Downsampling by a factor of N can be described in equation 1.2 as follows:

$$y[n] = x[Nn] \tag{1.2}$$

  Where $y[n]$ is the downsampled signal and $x[n]$ is the original signal.

- Upsampling, also known as interpolation, is the process of increasing the sample rate of a signal. This is done by inserting new sample points between the existing samples. Mathematically, upsampling by a factor of N can be represented in equation 1.3 as follows:

$$y[n] = \begin{cases} x[\frac{n}{N}], & \text{if } N \text{ divides } n \\ 0, & \text{otherwise} \end{cases} \tag{1.3}$$

  Where $y[n]$ is the upsampled signal and $x[n]$ is the original signal.

These operations are crucial in various applications such as audio processing and multirate signal processing. Precisely, downsampling and upsampling are crucial techniques in image processing [40].

- Downsampling reduces the resolution of an image, which can be beneficial for decreasing file size and saving storage space. It's often used when high-resolution images are not necessary, or when bandwidth for transmission is limited. The challenge with downsampling is to minimize the loss of important information from the original image.
- Upsampling, on the other hand, increases the resolution of an image. This might be used for printing purposes, where a higher resolution is required, or for zooming into an image while maintaining visual quality. The main challenge with upsampling is to avoid artifacts like blurring or aliasing, which can occur when new pixel values are interpolated .

1.3.1.2 Downsampling and Upsampling in FCN

In the context of neural networks, particularly Convolutional Neural Networks (CNNs), downsampling and upsampling are techniques used to manipulate the spatial resolution of feature maps [37].

- Downsampling (contracting path or encoder)in FCN is used to reduce the spatial dimensions of the input image as it progresses through the network, which can help to make the network more computationally efficient and reduce overfitting by providing an abstracted form of the features. This is typically achieved through pooling layers, such as max pooling or average pooling, or by using convolutions with a stride greater than one. The downsampling path helps the network to extract and interpret the context of the input image, allowing it to understand the larger structures and patterns present.
- Upsampling (expanding path or decoder) is used to increase the spatial dimensions of the feature maps as they move up the network towards the output layer. This is necessary to produce a detailed, pixel-wise prediction that matches the original input size, which is crucial for tasks like semantic segmentation. Upsampling in FCNs is often performed using transposed convolutions (sometimes called deconvolutions), unpooling, or interpolation methods.

As illustrated in figure 1.4, the principle consists of first performing max pooling on the input feature map, reducing its size. Then, upsampling the pooled feature map

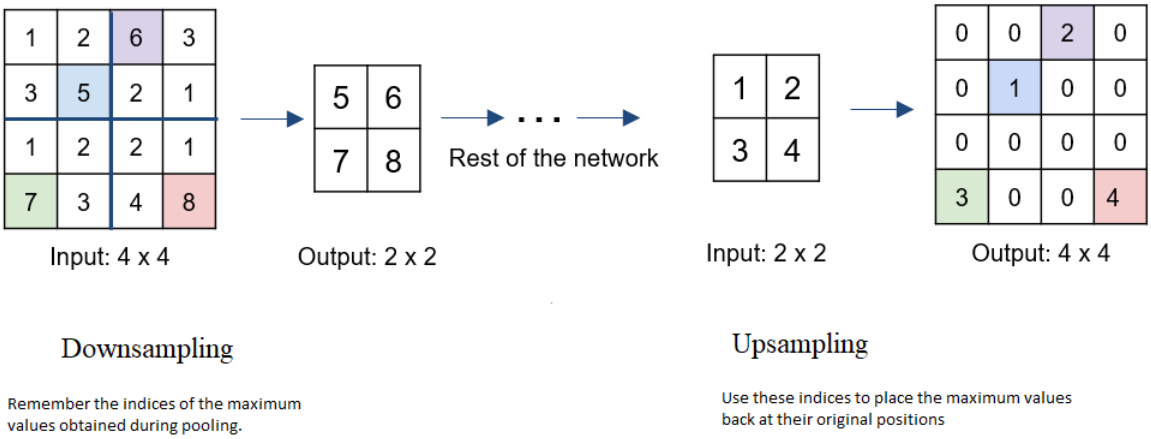


Figure 1.4: Downsampling and Upsampling in FCN

back to the original size. Both downsampling and upsampling play a critical role in the architecture of CNNs, especially in models designed for tasks that require detailed spatial understanding, such as image segmentation, super-resolution, and generative models.

1.3.2 Skip connection

Skip connections, also known as shortcut connections, are a neural network architecture technique that allows the output of one layer to skip over one or more layers and be added directly to a later layer's input. This approach helps to mitigate the vanishing gradient problem, which can occur in deep neural networks, making it difficult to train them effectively.

In essence, skip connections create an alternative path for the gradient during backpropagation, allowing for the training of deeper networks.

The main benefits of skip connections include:

- Easing the training of deep networks: By providing a direct path for the flow of gradients, skip connections help to maintain the strength of the gradient even in deep layers.
- Improving accuracy: Networks with skip connections tend to outperform those without them, as they can learn more complex functions.

- Enabling deeper architectures: Skip connections allow for the construction of very deep networks without the degradation of performance typically associated with increased depth

1.3.2.1 Skip connection in CNN

Skip connections, also known as residual connections, are a neural network architecture component that allows the output of one layer to be fed to non-adjacent layers, typically in a Convolutional Neural Network (CNN). They are a key feature in architectures like ResNet and DenseNet, which have shown significant improvements in training deep networks.

The primary purpose of skip connections is to mitigate the vanishing gradient problem, which can occur in deep networks as gradients are backpropagated through many layers. By providing an alternative shortcut path for the gradient flow, skip connections help preserve the gradient from earlier layers, making it easier to train deeper networks.

In a ResNet, skip connections work by performing element-wise addition of the input to the output of a block of layers. This means that the identity of the input is preserved and added to the learned representation, which theoretically allows the network to learn modifications to the identity rather than the entire transformation [26].

In DenseNet, skip connections are implemented through concatenation, where each layer receives the feature maps of all preceding layers as input. This encourages feature reuse throughout the network and leads to more efficient learning [29].

These connections are not just a workaround for the vanishing gradient problem; they also encourage feature reuse and lead to more interpretable layer activations, as each layer has access to the raw input data as well as the deep features.

1.3.2.2 Skip connection in FCN

Skip connections in Fully Convolutional Networks (FCNs) are a technique used to combine low-level feature information with high-level features to improve the performance of semantic segmentation tasks. In standard FCNs, long skip connections are typically used to concatenate features from the contracting path (encoder) to the expanding path (decoder) to recover spatial information lost during downsampling.

The idea is that by adding these skip connections, the network can use the detailed spatial information from earlier layers to refine the output. This is particularly

useful for tasks where precise localization is required, such as in biomedical image segmentation [41].

Moreover, some studies have extended FCNs by adding short skip connections, similar to those introduced in residual networks (ResNets), to build very deep FCNs. These short skip connections help maintain the gradient flow, allowing for effective training of deeper network architectures [42].

These techniques help the network to better localize and delineate the boundaries of objects within an image, leading to more accurate segmentation results.

FCNs employ skip connections to combine the high-level, semantic information from the deeper layers with the detailed, spatial information from the earlier layers. This helps to recover the fine-grained spatial information that may be lost during the downsampling process FCNs utilize skip connections to combine semantic information from deep layers with appearance information from shallow layers for detailed segmentations

1.3.3 Overall architecture of FCN

The network includes a downsampling path to extract context and an upsampling path for precise localization

In CNNs, downsampling is often achieved through pooling layers or strided convolutions, which reduce the spatial dimensions of the feature maps. Upsampling in CNNs might not be common unless the network is specifically designed for tasks that require increasing the resolution, such as in some segmentation or super-resolution tasks.

FCNs, on the other hand, are a specific type of CNN that are designed for pixel-wise tasks like semantic segmentation. In FCNs, downsampling is used to capture context and reduce the resolution, while upsampling is used to restore the resolution for precise localization. FCNs utilize skip connections to combine coarse, high-level features with fine, low-level features during the upsampling process.

The key difference lies in the purpose and the architecture of the networks. While traditional CNNs may not always include upsampling, FCNs are characterized by their use of upsampling and skip connections to produce outputs that are the same size as the input.

These operations enable FCNs to perform semantic segmentation, where each pixel of the input image is classified into a predefined category, making them powerful tools for tasks that require detailed, pixel-level understanding of the image [43]. Figure 1.5 illustrate the overall architecture of FCN.

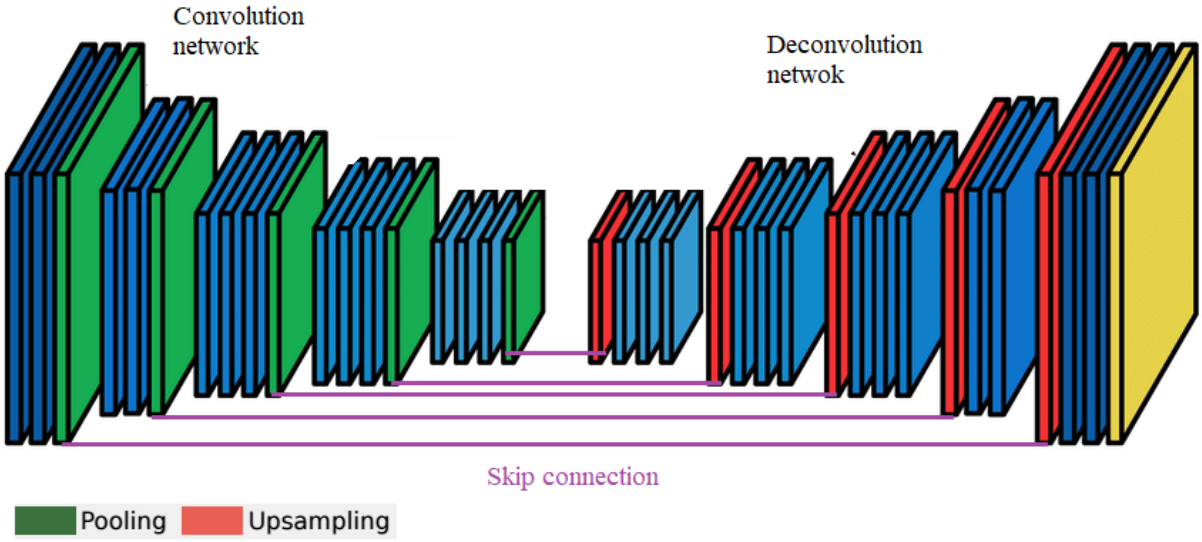


Figure 1.5: Fully Convolutional Networks's Overall architecture

1.4 Semantic segmentation

1.4.1 Image segmentation

Image segmentation is pivotal in computer vision and image processing, with its extensive applications spanning diverse sectors [44]. It is a method in computer vision that partitions a digital image into distinct segments [45], simplifying analysis and processing tasks. This technique is instrumental in identifying objects and demarcating boundaries within images[46].

1.4.2 Typical image segmentation techniques

There are various methods for image segmentation, including thresholding, clustering, edge detection. These methods analyze the visual features of each pixel, like color or brightness, to identify object boundaries and background regions.

1.4.2.1 Histograms

Histograms in the context of image processing are graphical representations that plot the frequency of pixel values found in an image.

Histogram-based methods are recognized for their efficiency in image segmentation, as they generally require only a single pass through the image's pixels. The

process involves creating a histogram from all the pixels, utilizing the peaks and valleys to identify clusters within the image. Both color and intensity serve as viable metrics for this analysis [46].

To enhance this method, the histogram analysis is recursively applied to the identified clusters, subdividing them into increasingly smaller groups. This recursive process continues until it no longer produces distinct clusters.

However, a notable limitation of histogram-based segmentation is the potential difficulty in pinpointing significant peaks and valleys within the histogram, which are critical for accurate cluster identification

1.4.2.2 Thresholding

Thresholding methods are a fundamental image segmentation technique that converts grayscale images into binary images. This is achieved by classifying each pixel according to whether its intensity level is higher or lower than a specified threshold value. The result is a binary image where pixels are marked as either foreground or background, facilitating the identification of objects or features within the image.

Otsu's method is a popular algorithm used in thresholding for automatically determining the optimal threshold value. It works by minimizing the intra-class variance, which is the variance within the same class, or equivalently, by maximizing the inter-class variance, which is the variance between different classes. This method is particularly effective when the histogram of the image has a bimodal distribution, meaning it has two distinct peaks or classes of pixel values [47].

1.4.2.3 Edge detection

Edge detection is a fundamental component of image segmentation, crucial for identifying the contours that define regions within an image through the detection of sharp changes in brightness or contrast. These techniques are predicated on the idea that object edges within an image are indicative of significant variations in pixel intensity. The intrinsic link between region boundaries and edges is what makes edge detection a key element in the development of various segmentation techniques.

Among the common algorithms for edge detection are the Sobel, Prewitt, and Canny methods. Each of these employs a unique approach to calculate gradients and edges, enabling the effective identification of object boundaries in different types of images.

1.4.2.4 Region based segmentation

Region-growing algorithms are indeed a class of image segmentation techniques that start with one or more 'seed pixels' and expand the region by aggregating neighboring pixels that have similar properties. These algorithms can be:

- Agglomerative: This approach begins with individual pixels or small regions and merges them into larger regions based on predefined criteria, such as intensity or color similarity.
- Divisive: In contrast, this method starts with the whole image and divides it into smaller segments by detecting dissimilarities or changes in pixel characteristics.

The choice between agglomerative and divisive approaches depends on the specific requirements of the application and the characteristics of the image being processed. Region-growing methods are particularly useful for images where regions of interest are not clearly defined by edges, or when the regions have uniform or gradually varying intensities

1.4.2.5 Watershelds

The watershed transformation is an image segmentation technique that treats the gradient magnitude of an image as if it were a topographic surface. Pixels with the highest gradient magnitude intensities (GMIs) are akin to the peaks of a mountain range, forming the watershed lines that delineate the boundaries of different regions. When 'water' is metaphorically poured onto any pixel within these boundaries, it flows downhill to the nearest local intensity minimum (LIM). The collection of pixels that drain into the same LIM forms what is known as a catchment basin, which corresponds to a distinct segment in the image.

This method is particularly effective for separating touching objects in an image and is widely used in various applications, such as medical imaging for separating overlapping cells or nuclei.

1.4.2.6 Clustering methods

Clustering algorithms, a type of unsupervised learning method, categorize visual data into groups or 'clusters' of pixels that share similar characteristics. K-means clustering is a widely-used variant where 'k' represents the number of desired clusters. The process involves:

1. Plotting pixel values as data points in a feature space.
2. Selecting 'k' random points as initial cluster centers or 'centroids'.
3. Assigning each pixel to the cluster with the closest centroid.
4. Updating the position of each centroid to the average of all pixels in the cluster.
5. Repeating the assignment and update steps until the centroids no longer change significantly, indicating that the clusters are stable.

This iterative process organizes pixels into clusters based on similarity, which is often measured by Euclidean distance in the feature space. The result is a segmentation of the image into distinct regions that can be used for further analysis or processing tasks.

1.4.3 Deep learning image segmentation tasks

The distinction between different types of image segmentation tasks lies in how they handle semantic classes—specifically, the categories to which individual pixels belong.

In computer vision, there are three groups of images segmentation tasks:

1.4.3.1 Semantic segmentation

Semantic segmentation is a fundamental task in computer vision. It involves assigning a semantic class label to each pixel in an image, categorizing pixels based on their meaning or content. The goal is to understand the scene structure by identifying different regions corresponding to specific objects or classes[48]. Unlike object classification, which annotates the entire image with one or more semantic labels, semantic segmentation operates at a finer granularity—labeling individual pixels. Meanwhile, object detection focuses on locating target objects, while semantic segmentation provides richer information about the scene's composition [49].

The significance of semantic segmentation lies in the following points:

- **Pixel-Level Understanding** By providing category information at the pixel level, semantic segmentation enables fine-grained analysis of images. It allows systems to discern intricate details within the scene.

- **Spatial Context** Semantic segmentation helps intelligent systems understand the spatial positions of objects and their relationships. This context is crucial for tasks like navigation, scene understanding, and object interaction.
- **Critical Judgments** Systems can make informed decisions based on pixel-level semantics. Whether it's avoiding obstacles, identifying regions of interest, or enhancing visual perception, semantic segmentation plays a pivotal role.

Semantic segmentation empowers intelligent systems to perceive images at a detailed level, making it a cornerstone of modern computer

1.4.3.2 Instance segmentation

Instance segmentation is a computer vision task driven by deep learning. Its goal is to predict the precise pixel-wise boundaries for each individual object instance within an image. Unlike conventional object detection algorithms, instance segmentation provides more detailed and sophisticated output by distinguishing separate instances of the same object class.

Semantic segmentation aims to achieve fine inference by predicting labels for each pixel in an image. Each pixel receives a class label based on the object or region it belongs to. Taking this direction further, instance segmentation provides distinct labels for separate instances of objects belonging to the same class. In essence, instance segmentation can be defined as the task of finding a simultaneous solution to both object detection and semantic segmentation [50]. So, instance segmentation provides detailed boundaries, while semantic segmentation focuses on pixel-level labeling. Both techniques contribute to our understanding of visual content.

1.4.3.3 Panoptic segmentation

In computer vision, Panoptic segmentation is a hybrid method that bridges the gap between semantic segmentation and instance segmentation. It aims to provide a unified view of segmentation rather than treating these tasks separately. panoptic segmentation involves three key steps:

1. Separating Objects: Each object in the image is broken down into individual parts, which are independent of each other.
2. Color Labeling: These separated parts are then painted with different colors (i.e., labeled).

3. Object Classification: Finally, the system classifies the objects.

Panoptic segmentation, introduced by Alexander Kirillov and team in 2018 [51], represents a significant advancement in computer vision and provides a unified view of segmentation, considering both semantics and individual instances. By combining the strengths of semantic and instance segmentation, panoptic segmentation offers a richer understanding of visual scenes. Consequently, panoptic segmentation paints a comprehensive picture, capturing both the forest (semantic context) and the trees (individual instances)

1.4.4 Deep learning semantic segmentation models

1.4.4.1 FCNs

Long et al [37]. adapted well-known CNN architectures like VGG16 and GoogLeNet by removing fully-connected layers. As a result, the modified networks output spatial segmentation maps directly.

1.4.4.2 Medical Image Segmentation UNET

U-Net, an extension of the Fully Convolutional Neural Network (FCN) architecture, addresses data loss during downsampling by incorporating skip connections. These connections selectively bypass certain convolutional layers, allowing the network to retain more detailed information as data and gradients flow through it. The name "U-Net" originates from the visual arrangement of its layers, which resembles the letter "U." This innovative design has proven effective for tasks like image segmentation, particularly in medical imaging applications[52].

1.4.4.3 Multiscale and Pyramid Network Based Models PSPNET

Zhao et al.[53] introduced the Pyramid Scene Parsing Network (PSPNet), a remarkable model specifically designed for semantic image segmentation. The PSPNet aims to enhance scene parsing by effectively capturing global context information. Its innovative design leverages pyramid pooling and context aggregation, resulting in state-of-the-art performance on various datasets. Notably, PSPNet secured the top position in the ImageNet scene parsing challenge 2016, as well as the PASCAL VOC 2012 benchmark and the Cityscapes benchmark [44].

1.4.4.4 R-CNN based model

Mask R-CNN (Mask Region-based Convolutional Neural Network) is a powerful model for instance segmentation. It seamlessly combines a region proposal network (RPN), which generates bounding boxes for potential instances, with an FCN-based "mask head" that produces detailed segmentation masks within each confirmed bounding box. Mask R-CNN excels in distinguishing individual object instances within an image, making it a go-to choice for various computer vision tasks[54].

1.4.4.5 Transformers

ViTs (Vision Transformer) are directly influenced by the architecture of transformer models like GPT (used for natural language understanding) and BERT. ViTs have excelled in various computer vision tasks like segmentation. Vision Transformers leverage attention mechanisms to achieve impressive performance in computer vision, demonstrating their versatility [55].

1.4.5 Practical applications

Image segmentation is a fundamental technique in computer vision with diverse practical applications.

1.4.5.1 Medical Imaging:

Image segmentation plays a critical role in medical imaging, including:

- Tumor Detection: Identifying the exact location and size of tumors using segmented images.
- Brain Segmentation: Precise delineation of brain structures for diagnosis and surgical planning.
- Disease Diagnosis: Assisting doctors in analyzing medical images for various conditions[56, 57]

1.4.5.2 Satellite Imaging:

Both semantic and instance segmentation automate the identification of different terrain and topographical features in satellite images. Applications include environmental monitoring, disaster response, and urban planning.

1.4.5.3 Smart Cities:

Image segmentation powers real-time tasks in smart cities:

- Traffic Monitoring: Analyzing traffic flow and congestion.
- Surveillance: Identifying incidents and enhancing safety.

1.4.5.4 Self driving vehicles

Image segmentation is crucial for self-driving cars:

- Obstacle Avoidance: Identifying pedestrians [58, 59], other vehicles, and obstacles.
- traffic signs: Ensuring safe navigation by recognizing lanes and road markings.

1.4.5.5 Manufacturing:

Beyond robotics, image segmentation is used for:

- Product Sorting: Efficiently categorizing items on production lines.
- Defect Detection: Identifying flaws in manufactured products.

1.4.5.6 Agriculture:

Image segmentation helps farmers:

- Estimate crop yields: By analyzing plant regions.
- Detect weeds: Identifying unwanted vegetation for targeted removal

1.4.5.7 Other Applications:

- Face Recognition, Fingerprint Recognition, and Iris Recognition rely on accurate segmentation.
- Airport Security: Detecting prohibited items at checkpoints.
- Traffic Control Systems: Optimizing traffic flow.
- Video Surveillance: Enhancing security through object tracking.

1.5 Object detection

Object detection is a fascinating field in computer vision and image processing. from early rule-based methods to today's deep learning-driven approaches, object detection has come a long way [60]. It involves identifying and locating specific objects within images or videos. The primary goal is to detect instances of semantic objects belonging to certain classes (such as humans, buildings, or cars).

1.5.1 Background of Object Detection

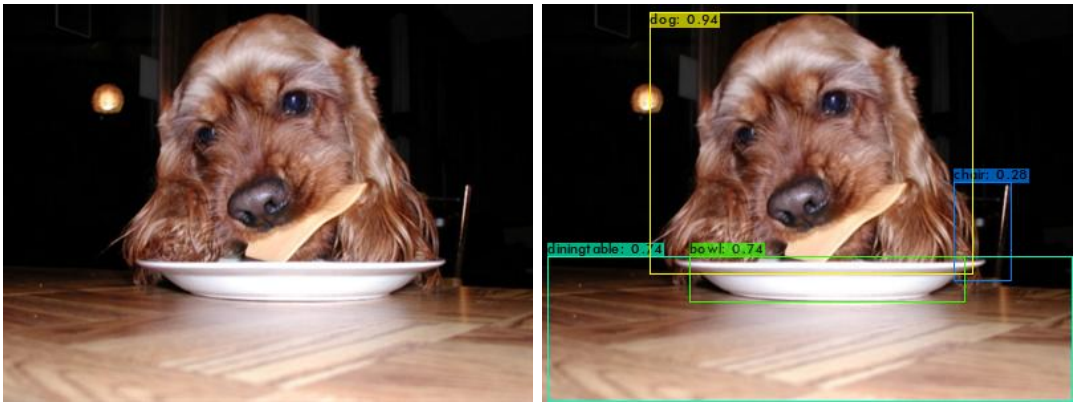
1.5.1.1 Task Description:

Object detection identifies the precise bounding boxes that enclose each object within an image. These bounding boxes define the spatial extent of the detected objects. Beyond localization, object detection also involves classifying these objects. Object detectors are typically trained on large datasets with annotated class labels. These labels define the categories of objects the model can recognize. Each detected object is assigned a class label based on its category as mentioned in figure 1.6(b). The figure 1.6 is an example showing the detected object from figure1.6(a).

1.5.1.2 Challenges:

The challenges faced in object detection are:

- **Scale Variation:** Objects can appear at different scales within an image. Detecting both tiny objects (like small insects) and large objects (like buildings) is essential. Methods like feature pyramids and multi-scale anchors address this challenge.



(a) The original image. (b) The output image after detecting objects.

Figure 1.6: Localization and classification objects within an image

- **Occlusion:** Partial occlusion occurs when objects are hidden by other objects or structures. Occluded objects may only reveal a portion of their true appearance. Handling occlusion requires robust features and context-aware models.
- **Complex Backgrounds:** Distinguishing objects from cluttered backgrounds is critical. Background elements (trees, walls, etc.) can interfere with accurate detection. Techniques like context modeling and attention mechanisms help focus on relevant regions. overcoming these challenges leads to more accurate and reliable object detection systems

Moreover, the challenges encountered in Classification are Fine-Grained Categories: Some applications require distinguishing between similar objects (e.g., different bird species). Imbalanced Data: Some classes may have fewer examples, affecting model performance. Context Matters: The context of an object can influence its classification (e.g., a "car" near a "road").

1.5.1.3 The Process of Detecting Objects

Detecting objects involves training object detection models. This entails creating a neural network that learns features from images. The network learns by analyzing images of objects in various scenarios, encompassing different backgrounds and angles, each paired with corresponding labels indicating the object's location [61].

In the following more detailed step-by-step process on how object detection models are typically trained:

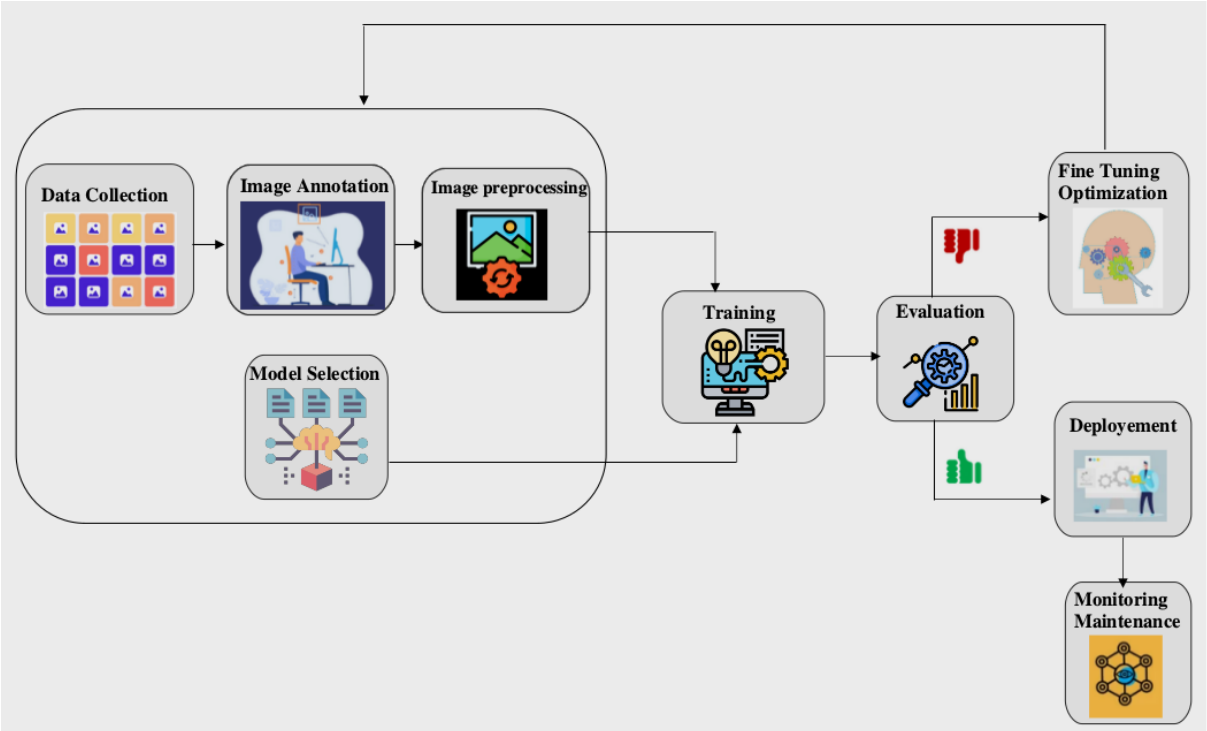


Figure 1.7: The process of training object detection models

- **Data Collection**: The first step is to gather a large dataset of images that contain the objects. These images should cover various scenarios, backgrounds, lighting conditions, and object orientations to ensure robustness.
- **Annotation**: Each image in the dataset needs to be annotated with bounding boxes indicating the location of objects of interest and their corresponding labels. This process can be time-consuming and may require manual effort, although there are some tools and services available to assist with annotation.
- **Preprocessing:** The annotated images are then preprocessed to prepare them for training. This may involve resizing images to a consistent size, normalizing pixel values, and augmenting the data with transformations such as rotation, flipping, and adjusting brightness or contrast. Data augmentation helps improve the model's generalization ability and robustness.
- **Model Selection:** Build a suitable object detection model architecture based on specific requirements, such as accuracy, speed, and hardware constraints. Common models include Faster R-CNN, YOLO (You Only Look Once), SSD (Single Shot Multibox Detector), and their variants.
- **Training:** Train the object detection model on the annotated dataset. During training, the model learns to predict bounding boxes and class labels for

objects within images by minimizing a loss function that measures the discrepancy between predicted and ground-truth annotations. This typically involves iterating over the dataset multiple times (epochs) until the model converges to a satisfactory level of performance.

- **Evaluation:** After training, evaluate the performance of the trained model on a separate validation or test dataset. Metrics such as precision, recall, and mean average precision (mAP) are commonly used to quantify the model's accuracy and robustness.
- **Fine-tuning and Optimization:** Depending on the evaluation results, you may need to fine-tune hyperparameters, adjust the model architecture, or collect additional data to further improve performance. Optimization techniques such as pruning, quantization, and model compression may also be applied to reduce model size and inference latency for deployment on resource-constrained devices.
- **Deployment:** Once satisfied with the performance, the trained object detection model can be deployed for inference on new, unseen data. This may involve integrating the model into a larger application or system, optimizing inference speed, and ensuring compatibility with the target deployment environment.
- **Monitoring and Maintenance:** Continuously monitor the deployed model's performance and periodically retrain or update it with new data to adapt to changes in the environment or data distribution over time. Regular maintenance is essential to ensure that the model remains effective and reliable in real-world applications.

Object detection is a powerful tool for various real-world scenarios, and its accuracy and efficiency continue to improve with advancements in deep learning and computer vision techniques [62]. The figure 1.7 illustrate these different steps of training object detection models.

1.5.2 Neural Network Methods and Techniques

Object detection algorithms often utilize machine learning or deep learning techniques to identify and locate objects within images or video streams. These algorithms learn to recognize patterns and features in the input data, allowing them to detect and classify objects accurately.

Machine learning-based object detection algorithms usually involve traditional computer vision techniques combined with handcrafted features and classifiers.

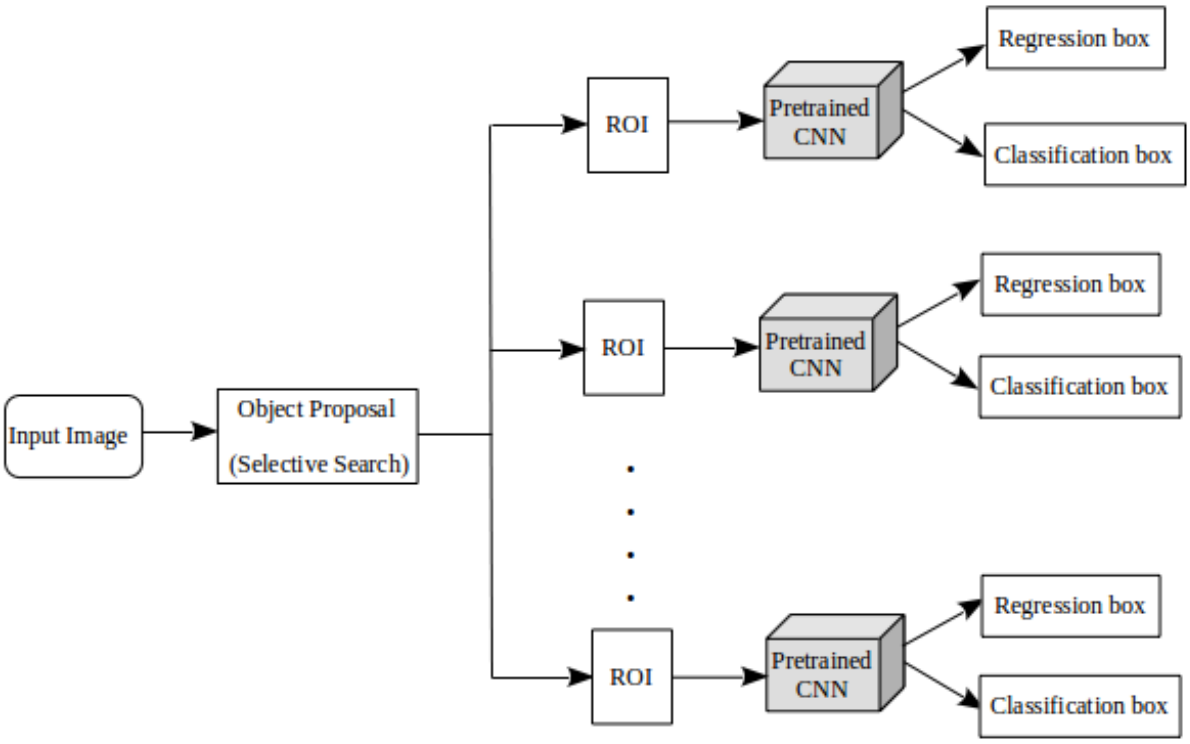


Figure 1.8: An overview of the architecture of the R-CNN model.

However, deep learning has emerged as a powerful approach for object detection due to its ability to automatically learn relevant features directly from the data. Two distinct approaches commonly used in object detection tasks[63] :

1.5.2.1 Region-Based Detection Framework

This framework typically involves two-stage object detection methods. In the first stage, the algorithm generates a set of region proposals or candidate bounding boxes that potentially contain objects. These proposals are then refined and classified in the second stage to determine the presence and location of objects within the proposed regions. Examples of region-based detection frameworks include Faster R-CNN (Region-based Convolutional Neural Network) and R-CNN (Region-based Convolutional Neural Network).

R-CNNs (Region-Based Convolutional Neural Networks) [64, 65] are a family of techniques designed for object localization and recognition tasks. In the following, it is described how they work:

- **Region Proposal:** Initially, the image is divided into smaller regions (proposals) using selective search or other methods.
- **Feature Extraction:** Each region proposal is passed through a convolutional neural network (CNN) to extract features.
- **Classification and Localization:** These features are used for both classification (identifying the object category) and localization (drawing bounding boxes around the object).
- **Non-Maximum Suppression:** To avoid duplicate detections, non-maximum suppression is applied to select the most confident bounding boxes.

The following figure 1.8 explains the principle of R-CNNs object detection method

1.5.2.2 Predictive Modeling Framework

This framework typically involves single-stage object detection methods. In this approach, the algorithm directly predicts the class labels and bounding box coordinates for objects in a single step, without the need for explicit region proposal generation. Predictive modeling frameworks are often faster but may sacrifice some accuracy compared to region-based approaches. Examples of predictive modeling frameworks include YOLO (You Only Look Once) and SSD (Single Shot Multibox Detector) [66].

YOLO (You Only Look Once) is a famous family of techniques for object recognition. Figure 1.9 illustrates model architecture of YOLO series network shown as the backbone, neck, and detect (head). In the following Key features of this technique [67, 68]:

- **Speed:** YOLO prioritizes real-time performance.
- **Single Pass:** It processes the entire image in a single forward pass through the neural network.
- **Grid-Based Prediction:** YOLO divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell.

1.5.3 Salient Object Detection

Generic object detection typically involves predicting bounding boxes around objects of interest within an image, often using techniques like bounding box regression. On the other hand, salient object detection is a captivating field of research

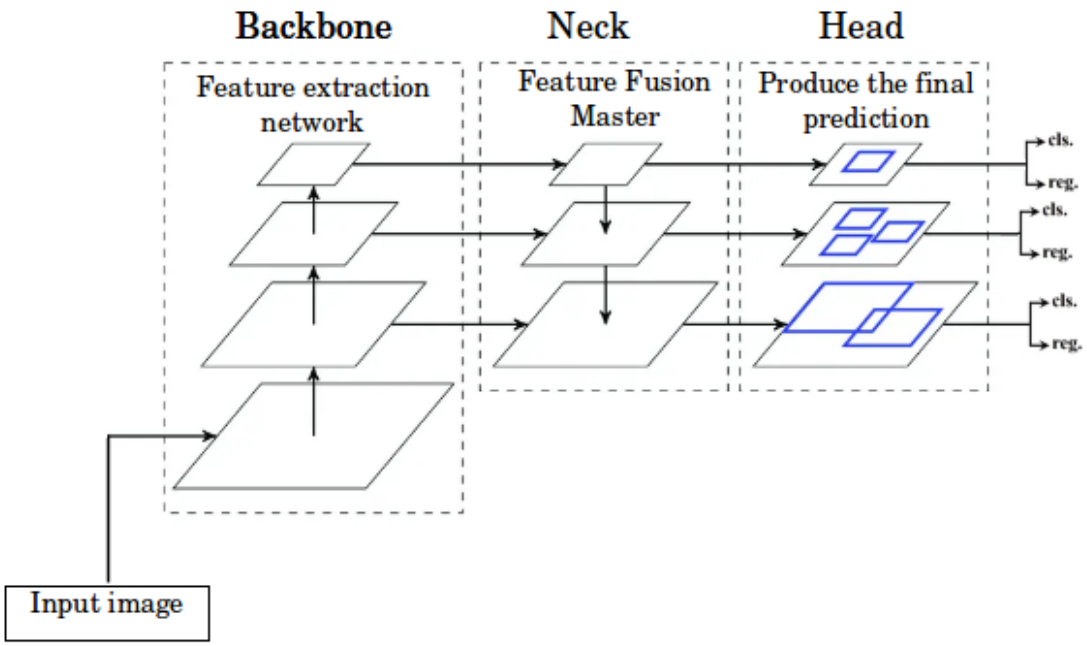


Figure 1.9: The YOLO series network's basic architecture.

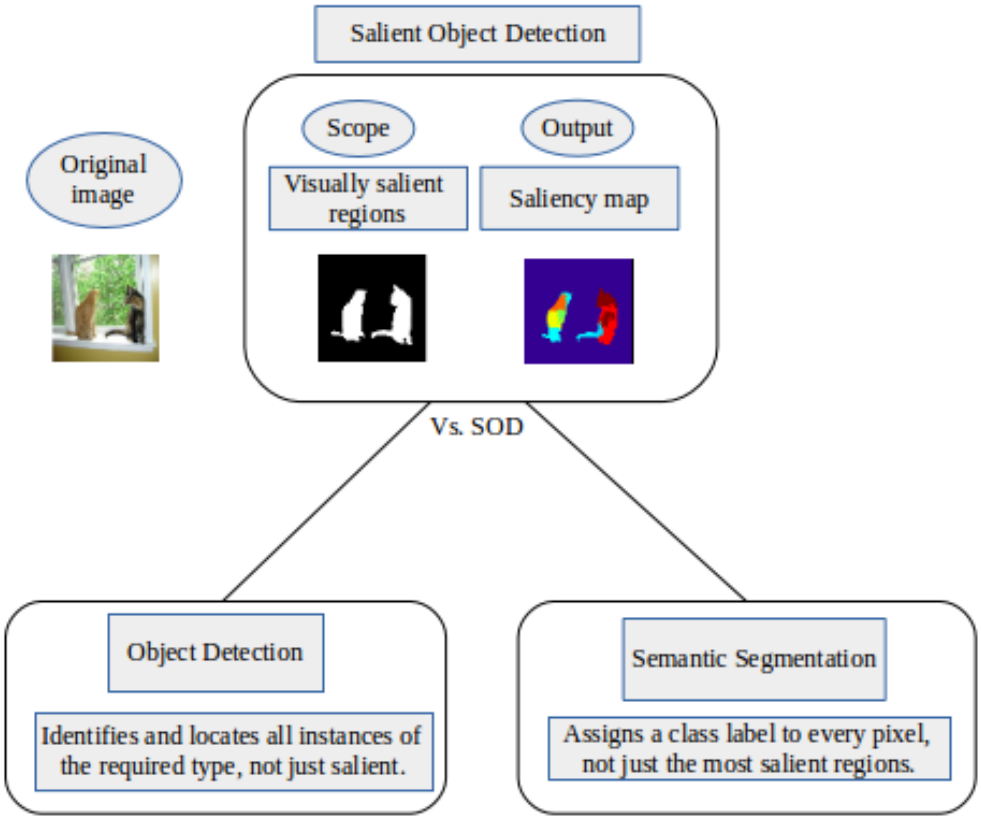


Figure 1.10: The overview of SOD Vs. object detection and semantic segmentation.

in computer vision that seeks to emulate the human visual attention mechanism. This process involves identifying the most visually prominent or salient object(s) within an image by highlighting objects or regions that are most likely to draw human attention. [63, 69]. Figure 1.10 presents an overview of SOD in comparison to semantic segmentation and object detection.

Salient object detection or segmentation commonly involves two main stages:

- Salient Object Detection: This stage involves identifying the most salient or visually striking object(s) in the image. This can be done using various techniques such as contrast enhancement, saliency maps, or deep learning-based approaches.
- Salient Object Segmentation: Once the salient object(s) are detected, the next stage is to accurately segment the region corresponding to those object(s) from the background. This can be achieved through methods like

pixel-level segmentation, where each pixel is classified as belonging to the salient object or the background.

1.5.3.1 Ispiration from Human Visual Attention

SOD is inspired by both bottom-up and top-down saliency mechanisms observed in the human visual system [70].

- Bottom-Up Saliency: This aspect of SOD is akin to the natural way humans are drawn to visually distinct regions in an image, such as areas with high contrast, unique textures, or intense colors. Bottom-up saliency is primarily driven by low-level image features and does not require prior knowledge or context.
- Top-Down Saliency: In contrast, top-down saliency incorporates higher-level cognitive factors such as knowledge, expectations, and task-specific goals into the saliency detection process. This aspect of SOD involves identifying regions that are relevant to specific objects or categories based on contextual information and task requirements.

Both bottom-up and top-down approaches to SOD face unique challenges. These challenges highlight the complexity of emulating human visual attention mechanisms in computational models. Addressing these challenges is crucial for advancing the field of SOD and improving the accuracy and reliability of saliency detection algorithms :

1. Bottom-Up SOD Challenges:
   - Large Appearance Variation:
     - Unconstrained object categories exhibit diverse appearances.
     - Handling variations in lighting, scale, pose, and occlusion is challenging.
   - Contextual Complexity:
     - Objects interact with complex backgrounds.
     - Separating salient objects from cluttered scenes is nontrivial.
   - Foreground-Background Ambiguity:
     - Distinguishing between foreground (salient) and background (non-salient) regions can be ambiguous.
     - Boundaries may blur due to contextual cues.

- Data Imbalance:
  - An imbalance between salient and non-salient regions affects model training.
  - Ensuring sufficient examples of both classes is crucial.

2. Top-Down SOD Challenges:

   - Semantic Association:
     - Associating desired visual stimuli (usually at a semantic level) with corresponding regions.
     - Balancing cognitive cues (knowledge, expectations, reward) and visual features.
   - Task-Specific Guidance:
     - Determining which objects or regions are relevant based on specific tasks.
     - Incorporating top-down cues (e.g., task instructions) effectively.
   - Object Category Determination:
     - Identifying regions related to specific object categories.
     - Cognitive factors (knowledge, expectations) play a role.
   - Spatial Localization:
     - Precisely localizing the desired visual stimulus.
     - Ensuring alignment with semantic cues.

By overcoming these hurdles, researchers can pave the way for more effective applications of SOD in various domains, ultimately enhancing our understanding of visual attention processes.

1.5.3.2 Benefits and Applications

The benefits and applications of salient object detection (SOD), demonstrate its versatility and utility in various domains of computer vision [71, 72, 73, 74] :

1. Object Highlighting and Segmentation:

   - Benefit: Salient object detection identifies the most visually significant regions within an image. It highlights objects or regions that naturally draw human attention.

- Applications: Content-Aware Image Editing: Enhance or manipulate specific objects while preserving context. Object Segmentation: Accurate segmentation aids in further analysis and understanding.

2. Visual Attention Modeling:
   - Benefit:
     - Understanding where humans naturally look in an image.
     - Provides insights into cognitive processes and attention mechanisms.
   - Applications:
     - Human-Computer Interaction: Design interfaces that align with human gaze patterns.
     - Advertising Optimization: Place critical content where users are likely to focus.
3. Scene Understanding and Recognition:
   - Benefit:
     - Salient regions often correspond to informative parts of a scene.
     - Helps in scene understanding and context-aware recognition.
   - Applications:
     - Scene Understanding: Identify relevant objects and their relationships.
     - Visual Search Engines: Improve retrieval by emphasizing salient regions.
4. Content-Based Image Retrieval:
   - Benefit: Saliency maps guide retrieval by emphasizing relevant regions. Enhances search accuracy and relevance.
   - Applications: Image Databases: Retrieve images based on content similarity. Reverse Image Search: Find similar images online.
5. Attention-Driven Robotics and Autonomous Systems:
   - Benefit:
     - Robots and autonomous vehicles can focus on relevant parts of the environment.
     - Efficiently allocate computational resources.
   - Applications:
     - Robot Navigation: Prioritize relevant visual cues.

- Autonomous Vehicles: Detect pedestrians, obstacles, and road signs.

6. Visual Aesthetics and Design:

   - Benefit:
     - Salient regions contribute to image aesthetics.
     - Aids in photo composition and design.
   - Applications:
     - Photography: Compose visually appealing shots.
     - Graphic Design: Emphasize key elements in layouts.

Salient object detection plays a crucial role in various applications, ranging from enhancing image editing capabilities to improving the efficiency and safety of autonomous systems.

1.6 Conclusion

The objective of this chapter was to provide an overview of several key concepts introduced to serve the overarching purpose of this thesis. We began by discussing CNN (Convolutional Neural Networks), a widely used deep learning architecture for various computer vision tasks. Following that, we delved into image segmentation, a process involving the division of an image into multiple segments or regions to simplify its representation or enhance its analytical significance. This technique is commonly applied in tasks such as object identification within images or the delineation of regions of interest.

Subsequently, we explored the principle of object detection. Unlike image classification, which predicts the presence of objects in an image, object detection also furnishes information about the precise location of each object, typically through bounding boxes. The aim of this exploration is to establish a foundation for comprehending the techniques and methodologies utilized in the analysis and interpretation of visual data. These insights will be further leveraged in subsequent chapters to elucidate the proposed approach for addressing object detection challenges.

Chapter 2

A Salient Object Detection Technique Based on Color Divergence

This chapter consists of the article: "A Salient Object Detection Technique Based on Color Divergence, ICFNDS '20: The 4th International Conference on Future Networks and Distributed Systems - St. Petersburg Russian Federation, November 2020."

2.1 Introduction

In view of the constraints and problems brought up by the artificial vision system, salient object detection has raised an increasing amount of interest in computer vision which is applied to a wide range of applications for further processing in computer vision and image processing, such as object detection [75], and semantic segmentation [76]. In fact, object detection may offer many benefits in several research domains [77, 78, 79, 80]

SOD (Salient Object Detection) tries to simulate the visual attention mechanism to highlight objects that attract the attention from each given image. This is inspired by cognitive studies of visual attention [81], where the researchers proposed intensive approaches to better detect and segment the most visually distinctive object in natural scenes. Classical methods [82, 83, 84] are based on low-level features and cues to obtain salient object regions in an image. They suffer from several deficiencies which limit their ability for processing images with complex scenes. Recently, fully-supervised approaches based on deep Convolutional Neural Networks (CNNs) have had a significant improvement for salient object detection, and have shown superior performance over traditional solutions which is obtained by powerful representation methods of deep learning techniques. SOD depends

principally on the number of training images which contain manually annotated salient objects.

Despite the great improvement brought by (CNNs), still several improvements could be attained such as reducing time consumption, raising the accuracy and improving the quality of saliency maps by implementing a reliable detection technique of salient objects.

In this paper we will propose a salient object detection method which is based on color divergence of the RGB cube representation. The main contribution of this research is observed in the development of the model which considers both the Euclidean distance matrix, and the standard deviation vector.

2.2 Related work

Nowadays, object detection including salient object detection is one of the most fundamental yet challenging problems in the computer vision community [69]. The set of saliency detection methods are extremely rich. With the great success of deep learning technologies in computer vision, several deep learning based SOD methods have emerged since 2015 [85] and a series of algorithms have been published to learn the saliency representations from large amounts of data [86]. The authors of this paper focus on some representative work having great practical importance to improve the performance of SOD.

Specifically, Li et al. [87] introduce an end-to-end deep contrast network with two complementary components: a fully convolutional stream and a segment-wise spatial pooling. In Kuen et al. [88], a recurrent attention and convolutional-deconvolutional networks are used to perform saliency progressive refinement from exploiting spatial transformer and recurrent network units. Kruthiventi et al. in [89] consider both fixation prediction and segmentation of salient objects in a unified network. However, these models still fail in several cases including scenes with transparent objects, low contrast between foreground and background, and complex backgrounds.

In order to adress these problems, Zhang et al. [90] detect salient objects by using saliency cues at different levels through fully convolutional neural networks and multi-level fusion mechanisms. Hou et al.[81] suggest adding a series of short connections to the Holistically-Nested Edge Detector (HED) architecture [91] to combine different side outputs from deeper layers to shallower ones to better detect the salient objects/regions. Pingping Zhang et al. [92] aggregate a multi-level features framework at multiple resolutions. A boundary refinement is used at each aggregated feature before using the fused prediction as a final saliency map.

However, salient results in these methods attend to contain many non-salient objects and suffer from the loss of certain details of the salient object when directly merging several level features. Inspired by these observations, Xiaowei Hu et al. [93] introduced the Fully Convolutional neural Network (FCN) with recurrently aggregated deep features for salient object detection. They use the built-in side features to refine the features of each layer of the FCN and such processes are repeated to gradually produce finer salience predictions. Another improvement proposed by Xin Li et al. in [94], comes from creating a two-branch Contour-to-Saliency Network (C2S-Net) for salient object detection. With the capability of combining both saliency and contour knowledge, some enhanced results are obtained by inserting an SOD branch to a pre-trained contour detection model.

All these models use context regions holistically to construct contextual features. Even with their robustness, they still need to get the most accurate results. Recently, in [95] Nian et al. formulate a Pixel-wise Contextual Attention Network (PiCANet) within a U-Net architecture [52] to attend global and local context regions for each pixel. The work of [96] expands the potentials of pooling in convolutional neural networks based on the feature pyramid networks (FPNs)[97]. Yuzhu Ji et al. in [98] solve the problem of salient object detection by fusing multiple saliency cues and objectness based on a graph model bottom-up method. The work of [99] presents a simplified end-to-end method based on a saliency regression network which runs as a compact pipeline to directly output a full resolution saliency map (image-to-image prediction). Despite the good performance obtained by these approaches, there is still a great room for improvement. To this end, we will use the color divergence technique to be discussed in the next section.

2.3 Background and contribution

Image segmentation is a process that subdivides an image into its constituent parts or objects [77, 100]. Several general-purpose algorithms and techniques have been developed for image segmentation because there is no prominent solution for the image segmentation problem. Regions of a digital image whose pixels are characterized by color homogeneity can be interpreted as constituting parts of the objects present in the image. However, if a 24-bit true color image is considered, where the number of possible different colors may reach 16 millions, the number of perfectly homogeneous regions in the image would most possibly be noticeably larger than the number of objects perceived in the image by a human observer. In fact, the human visual system is able to distinguish a reasonably large number of colors. In visual computing, only a group of colors with similar tonality could define the objects since few colors are often sufficient for understanding and interpreting the image in question.

2.4 Philosophical enclosure of the assumption

Color image clustering of natural scenes is widely studied to address numerous applications in computer vision such as image segmentation and SOD. The combination of different features for detecting salient regions could be a good solution because manually modelling an interaction method to integrate inherently different saliency features is a challenging problem. A good integration method could be developed to address this problem for adopting the Convolutional Neural Network mechanism to train a saliency map integration model.

This paper aims to introduce a new clustering methodology that will classify the distribution of an image's pixels in sets of clusters to efficiently show the salient object upon this image. The proposed assumption should be conceptually simple in term of complexity in order to provide a short computation time. Furthermore, the proposed technique should take into consideration the limitations of the previously proposed algorithms, since many algorithms are utilized to detect the salient objects. The proposed clustering technique focuses on the standard deviation to detect salient objects using an optimized clustering process.

2.5 The proposed Color Divergence Technique (CDT) to detect salient objects

This work presents a salient object detection technique based on color divergence using an RGB cube representation. An RGB cube represents colors of an image in three dimensions, Red, Green and Blue. The standard deviation is used with the aim to discover classes of color homogeneity in close location, i.e., that are very likely to represent salient objects. Specifically, we first use the Euclidean distance matrix between each RGB color pixel of the original image and the eight vertices of the RGB cube. For each pixel, eight values (distances between this point and eight vertices of the cube) will be computed. The ninth value is the standard deviation of the eight values computed. This computation will reduce three pixel values of an image to one single value *sdv*. Based on the desired salient objects' color, we classify *sdv* in order to extract the salient objects. To this end, we affect a coefficient to each computed *sdv* as a function of the desired salient objects' color. Then we apply different type of coefficients to obtain a set of matrices, where for each matrix we create the standard deviation vector with the aim to use it for image classification. Finally, we get an approximate saliency map after the fusion of the results of all type coefficient classifications.

We observe that the difficulty to detect the salient objects or to draw the boundary of an object depends on the quality and complexity of the image. In this work,

we will redistribute the RGB colors of each pixel of a given image in a 3-D cube following the three values (Red, Green, and Blue). With the aim to extract the salient objects, the proposed technique clusters these scattered 3-D cube points. The principle is that instead of exploring million of points to detect the salient objects the clustering process minimizes this number depending on the number of proposed clusters. Therefore, the image segmentation in this contribution will be reached using the clustering process. Each segment will present an object and the objects that reside in the far region will be presented as salient objects.

2.5.1 RGB cube classification

The goal of RGB cube representation is to show the classes of color homogeneity in close location that are very likely to represent objects. The proposed technique focuses on the idea of representing the RGB color cube of an image provided as input. We assume that each object color resides in a separate region in the 3-D cube depending on the object color. The proposed technique focuses on the principle that detecting an object is a process that starts from unifying the colors of each object. This procedure will reduce the number of similar colors to one color in such a way as to provide a new version of the original image while preserving the image features.

Image segmentation based on a distance vector:

The RGB color space is characterized by a cube with R, G, and B axis, and colors are interpreted as three-dimensional ($3D$) vectors, with each vector element having an 8-bit dynamic range. Each of the three edges of the cube has length 256, since each color component may assume any value in the range [0, 255] as shown in Figure 2.1.

The representation of an image as an RGB color cube is used to unify similar pixels, i.e., clustering the similar points that will represent an object. To this end, the distances between each pixel and the eight cube vertices are computed using Equation 3.1. The distance vector of a given pixel p is defined as $D_p = [d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7]$, where d_i corresponds to the distance between the pixel p and the vertex $i (i = 0 \dots 7)$ of the cube which is computed as follows:

$$d_i = \sqrt{r_i^2 + g_i^2 + b_i^2} \tag{2.1}$$

where r_i, g_i and b_i represent, respectively, the red, green and blue values of the pixel p depending on the vertex i.

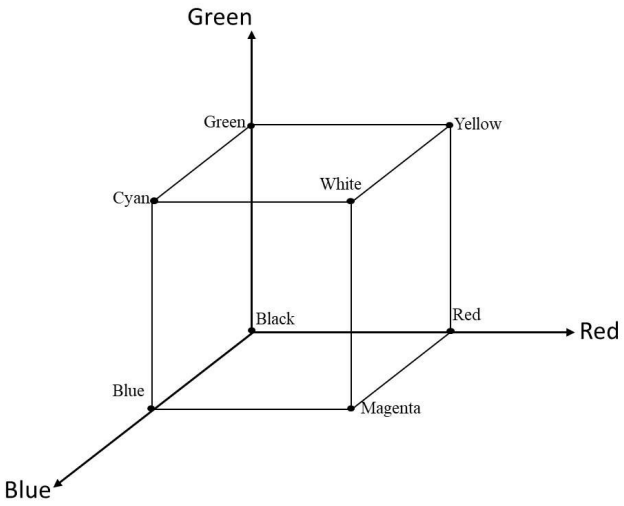


Figure 2.1: A 3D cube geometrically representing the RGB color space

The obtained values D_p of all pixels will be presented in a table with the aim to segment the input image by representing similar colors by a single color. Therefore, the segmentation process orders the table rows in such a way that every similar color appears on the consecutive rows. Sorting the obtained table following distances d_i may offer a preliminary result. This ordering will consider all pixel points that have the same d_i value as similar pixels, e.g., the blue and red pixels having the same value compared to d_0. Therefore, the standard deviation is used to distinguish the pixels having the same d_i.

We define the standard deviation vector sdv which will be used to order the table rows. A column will be inserted into the table that contains the standard deviation of the distance vector D_p.

$$sdv_p = \frac{\sqrt{\sum(d_i - averageD_p)^2}}{8} \tag{2.2}$$

where d_i corresponds to the distance of the pixel p and the vertices $i(i = 0 \dots 7)$ of the cube. $averageD_p$ is the average of the distance vector D_p.

In fact, the standard deviation may provide more accuracy by distinguishing further pixel points of the input image. However, we may observe that there are some symmetric pixels that have the same sdv_p, i.e., the pixels that have the same standard deviation with different d_i values. Mathematically, two pixels are symmetric if the following condition is verified:

$$\exists i \neq j \; d_i = d_j \quad | \quad \forall i, j \in \{0, \dots, 7\}$$

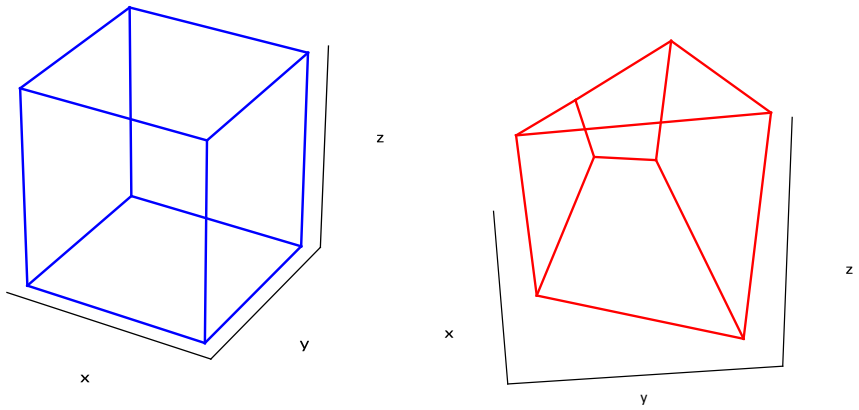


Figure 2.2: A 3D regular and irregular cube geometrically representing the pixel standard deviations

We define the coefficient f to separate these symmetric pixels. The cube that presents the colors will be presented in a new cube with irregular edges. Hence, we assign a different value of f for each cube vertex. The values of the d_i are computed as follows:

$$d_i = f_i * \sqrt{r_i^2 + g_i^2 + b_i^2} \tag{2.3}$$

where f_i is used to ensure that the symmetric pixels will not have the same standard deviation and will not belong to the same class, thereby clearly separating the different objects. We note that the coefficient f separates the symmetric values only when the following condition is verified: $i \neq j \Rightarrow f_i \neq f_j$. Figure 2.2 shows an example of an irregular cube with $f_i = i$.

The eight values of f should be assigned carefully because their role is to separate the image's objects and detect the salient objects. Therefore, we assign to the image background color the minimal value of f and to the most salient object the maximal value. By default, we assign the minimal value to the black color and the maximal value to the white color. The potential of these coefficients could be observed when detecting salient objects having a dominant color, so we can assign the maximal value to the coefficient of the dominant color of the salient object.

2.5.2 Segmentation and salient object detection

To detect the salient objects, we use a table which contains the value of the standard deviation sdv_p and the pixel values of the input image. The table will be sorted in an ascending order following the standard deviation sdv_p. Then we divide it into classes where the pixels of the same class will represent a class of objects. The last class represents the most salient objects that exist in the input

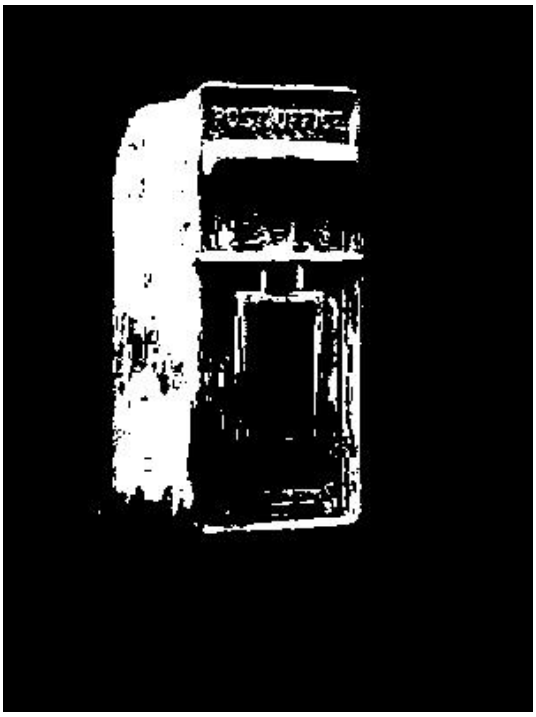
image. In this paper, we segment the image into two classes in order to obtain several saliency maps by applying different values of coefficients to the Euclidean distance matrix. Hence, we will create for each type of coefficients a standard deviation table and each standard deviation table is divided into two classes that will represent the saliency map.

2.6 Implementation and results

Evaluating the performance of the proposed clustering algorithm is pertinent for showing its performance mainly in terms of execution speed and accuracy. To this end, we will test it on real images and see what it extracts. In fact, it is difficult to judge the quality of the obtained results without comparing it with other algorithms. In this section, we present the carried out experiments in order to see them and evaluate our classification technique compared with others. The objective of these experiments is in particular to evaluate the ability of the algorithm to correctly detect and extract the most salient objects based on such already predefined colors. Figure 2.3 shows examples of the color divergence technique for detecting salient objects. The evaluation of the proposed CDT is often done by applying the technique to subsets of benchmarks. We conducted our experiments with the different algorithms containing a two-dimensional vision. For all the experiments that we have carried out, we have varied the parameters processing time and accuracy. We have used random images with different levels of difficulty to detect salient objects because our goal in this study is to compare the proposed technique in various different cases. Table 2.1 shows the first obtained results.

The processed MSRA-B dataset [101] contains a diversity of 5000 images widely used in the salient object detection community. This dataset is treated by a Python program to visualize the processing time and accuracy results. The results obtained by applying several executions are presented in Table 2.1. For pixel accuracy that evaluates the precision of detected salient objects, we evaluated $accuracy = (TP + TN)/(TP + TN + FP + FN)$ which is the percent of pixels in the image which were correctly classified. A true positive represents a pixel that is correctly predicted to belong to the salient object (according to the ground truth) whereas a true negative represents a pixel that is correctly identified as not belonging to the salient object .

(a) The original image for detecting salient red objects.



(b) The output image after detecting salient red objects.

(c) The input image for detecting salient black and white objects.

(d) The output image after detecting salient black and white objects.

Figure 2.3: An example showing some detected salient objects based on CDT

Technique	**Accuracy %**	**Processing time (ms)**
Our CDT	0.785	4.202
SVR	0.662	6.342
DSR_D	0.801	8.127

Table 2.1: The processing time and the accuracy.

2.7 Conclusion and future work

A novel color classification technique is proposed in this paper to efficiently detect salient objects using color divergence and RGB cube representation. We have applied RGB cube classification focusing on a distance vector and standard deviation values for the clustering of RGB pixels. Our method achieves a comparable performance in term of processing time in compact salient object detection systems. Our technique is efficiently capable of identifying salient colored objects in both simple and difficult cases, which helps to enhance the deep learning process for the object identification phase. Furthermore, a simple salient object detection process and a fast detection speed may offer further advantages for image processing in computer vision and artificial intelligence domains. As future work, we will check its performance by implementing it for object identification using different deep learning methods.

Chapter 3

Pixel and tensor standard deviation representation

This chapter introduces a new pipeline approach dedicated to image text detection, which reduces the size of the input and intermediate layers in a deep learning model by merging feature maps based on standard deviation values.

3.1 Introduction

Computer vision in general, in recent years, has become one of the major computer science research topics, where efficient text detection in multimedia content is one of the major challenges. Advancements in artificial intelligence have significantly improved various facets of computer vision, allowing for the automation of numerous tasks. In the realm of image classification, various machine-learning algorithms are frequently used [102]. Deep Learning (DL) stands out as a subset of these algorithms, with one of its key strengths being its capacity to automatically extract pertinent features from the data, thereby diminishing the necessity for manual feature selection [103]. With the rapid advancement of mobile computing devices and the massive proliferation of smart devices[104], images containing text data are being collected more readily, conveniently, and efficiently. Therefore, recognizing text within an image has become an active research topic in computer vision. The texts in the image often contain rich semantic information of high-level importance that helps in analysis, guidance, and understanding of the corresponding environment. Therefore, detecting and recognizing texts in images have received increasing attention in several applications[100], including automatic navigation, image retrieval, human-computer interaction, security, data integrity [105, 106, 107, 108, 109] , etc. For example, considering the application of autonomous vehicles, a field rapidly gaining momentum. These self-driving cars navigate complex urban environments, relying on many sensors and cameras to perceive their surroundings. In this context, accurate text detection can be a

game-changer. The extraction and analysis of text from images and videos have emerged as focal points within multimedia understanding systems. Numerous methods have been put forward for extracting text from visual media, and many studies have offered comprehensive reviews [110, 111, 112, 113] on this subject. Text recognition and extraction address two primary challenges when locating text within a target image. The first challenge is identifying the geometric representation that encapsulates the text's position, which may take the form of a contour, such as a square, rectangle, oriented rectangle, or quadrilateral. The second task is to transform text-containing regions within images into machine-readable strings. Hence, the first crucial step in text identification is text discovery [114]. Text within an image can be classified into two categories: scene text, which is inherent in the image at the time of capture, and artificial text, which is introduced into the image during post-production processes [115]. On the other hand, the Internet of Things's significance in the text detection domain cannot be overstated. In today's digital information, a substantial portion of multimedia data emanates from IoT devices and embedded systems, underscoring the sheer volume and diversity of information generated. Integrating IoT into various applications underscores the pivotal role of text detection in computer vision. This paper addresses IoT-generated multimedia data's critical challenges, focusing on enhancing text detection methodologies to extract valuable information efficiently. IoT-generated data containing embedded text is indispensable for streamlining shipping information processing in logistics and supply chain management domains. Similarly, in the realm of smart cities, the utilization of IoT is instrumental in text detection applications such as smart parking systems, enabling effective traffic management through license plate and signage recognition.

This paper's primary target is to propose novel approaches tailored to the intricacies of IoT-generated multimedia data. Specifically, our research endeavors to enhance text detection algorithms to cope with the unique challenges of variable image quality, diverse orientations, and complex environments inherent in IoT-driven scenarios. We aim to contribute significantly to advancing robust and efficient text detection methods within IoT frameworks by addressing these challenges.

However, a pivotal challenge that emerges is the often poor quality of text data extracted from these sources. Text detection in this context necessitates robust solutions capable of real-time processing and maintaining high levels of accuracy. In response to this challenge, researchers and engineers have proposed a spectrum of innovative solutions designed to address the multifaceted concerns surrounding poor-quality data. These solutions not only enhance the accuracy of text detection but also ensure that it operates seamlessly in real-time scenarios, ultimately facilitating the effective utilization of IoT-generated multimedia data in various applications. The following points, among others, render the detection task more difficult [116]: 1) The absence of prior information regarding the text's

location within the image presents a significant challenge, particularly when the text is distributed throughout the scene. Knowledge of factors such as the number of text lines, line spacing, and word count can significantly streamline the process, especially in the case of scanned documents. Consequently, the lack of such pre-established formatting rules makes direct segmentation implementation more challenging in these image types. 1) Text within an image can appear in many sizes, fonts, and orientations. It may even include embossed or uniquely designed characters, such as calligraphy logos, presentation slides, or messages displayed on a digital bulletin board. Recognizing text with such distinctive and non-traditional appearances poses a significant challenge. 2) The quality may differ from one photo to another, depending on the digital device the image was taken with, which may in several cases be poor. 3) Images often contain numerous patterns resembling letters, set against complex backgrounds, where certain unfamiliar objects like icons, windows, and leaves may closely resemble letters and words. Various other elements may be interwoven with the text, resulting in new styles and representations. Therefore, one of the critical sources of information in the image and video is the inside text. For instance, the caption text may explain information about where and when the events in the video occur and possibly who participated in the events, and the banner text is used as a visual indicator for navigation and notification in the scene. Additionally, among the different types of objects that appear in the videos, the text that contains abundant semantic information plays an important role in many practical applications, such as video annotation and multimedia retrieval [117]. The accuracy of text extraction is critical for the reliability and effectiveness of these applications, while faster processing times enhance efficiency and user experience. Achieving the right balance between accuracy and speed is often a key consideration in developing text extraction systems. Figure 1 shows an overview of text detection in images in the literature. In the pre-processing phase of image text detection, particularly in practical real-world applications, substantial efforts have been poured into the pursuit of swift and resilient algorithms [118, 119, 120, 121]. The most important, among other challenges, is the imperative for outstanding performance, which includes achieving an acceptable processing time and a high recall rate while maintaining a low false alarm rate. These requested efficiencies must be accomplished while accommodating an intricate web of variables. These variables encompass the intricacies of font size, font color, textual orientation, linguistic diversity, and the often perplexing intricacies of diverse backgrounds. Nonetheless, a further challenge lies in the arduous task of distinguishing text from an eclectic array of text-like elements, ranging from leaves and window curtains to generic textures. Adding to the complexity is the ever-evolving nature of text patterns, as they meander through variations in font sizes, colors, and languages. To complicate matters further, the pervasive influence of noise and the capricious nature of image encoding and decoding procedures conspire to undermine the integrity and quality of the text. Moreover, the factor of treatment time emerges as a crucial consideration in

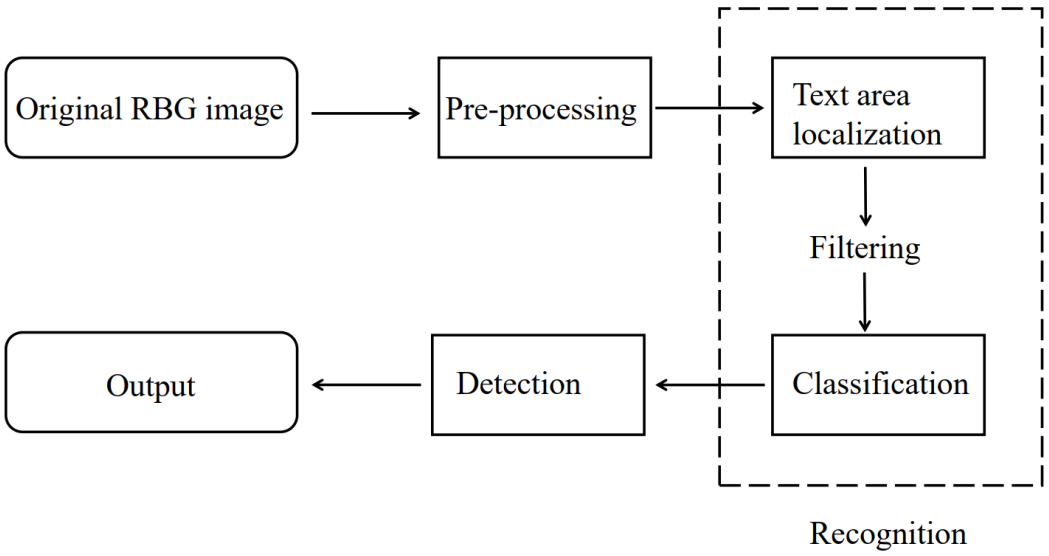


Figure 3.1: A comprehensive vision of text detection in images

this landscape. Swift and robust text detection algorithms must also account for timely processing to meet the demands of real-world applications. The pertinence of time in text detection further underscores the complexity of this multifaceted challenge.

Compared with proposed approaches, recent text detection algorithms often face the following challenges:

(1) Text Orientation: Text can be written in different orientations, such as horizontal, vertical, and diagonal. Detecting text in non-horizontal orientations can be challenging for text detection algorithms.

(2) Font Size and Style: Text detection algorithms can struggle with recognizing text with a small font size, written in a different style, or complex structure.

(3) Language and Script: Text detection algorithms may be trained in specific languages or scripts and cannot detect text in other languages or scripts.

(4) Background and Contrast: Text detection algorithms can have difficulty detecting text if the background is complex or if the contrast between the text and the background is low.

(5) Image Quality: The image quality can significantly affect text detection algorithms' performance. Low-quality images or images with noise or distortion can make it difficult for the algorithms to detect text accurately.

(6) Computational Resources: Text detection algorithms can be computationally intensive and may require significant computational resources to operate efficiently.

This can be a limitation in some scenarios where computational resources are limited.

The primary goal of this paper is to achieve the right balance between accuracy and speed in developing text extraction systems, so the significant contribution lies in addressing this balance, ensuring that text extraction systems not only maintain high accuracy but also operate efficiently in terms of speed. The purpose is to address the three last challenges, where we propose a text detection approach focusing on the pre-processing phase of Figure 1. To reach this goal, the proposed approach considers the values of the computed σ instead of the whole pixels' image while preserving its features to reduce the data size by reducing both the input image channels and the tensor channels of the intermediate layer of the deep learning model. We run a deep learning model without pre-trained models to reduce the computational time speed and increase the accuracy compared with original RGB images. To improve the accuracy, we introduce σ in different stages of the proposed pipeline approach. The rest of the paper is organized as follows: Section 2 presents a background and some related works. In Section 3, the proposed approach is explained. Section 4 shows the experimental evaluation and results. The manuscript is concluded in Section 5.

3.2 Background and Related Work

This section presents an overview with an analysis of text detection techniques of recent and relevant work. The discussed text detection techniques in this section will be reviewed depending on their operating modes and characteristics. We organize our review by classifying text detection techniques into traditional Machine Learning-based (ML-based) approaches and deep learning-based approaches. Machine learning is a powerful tool that aids in a multitude of tasks, particularly in prediction. Its versatility lies in exploring various methods and algorithms, allowing for effective analysis and interpretation of data [122]. However, traditional ML-based approaches rely on handcrafted features and classifiers. These features could include gradient-based features, edge detection, and texture analysis. The common classifiers used in traditional ML-based approaches are Support Vector Machines (SVM), decision trees, and random forests. These techniques prove their good performance in several applications, which explains their wide uses. However, these approaches have some critical limitations, such as their dependency on the quality of handcrafted features, which could be sensitive to lighting conditions, and their inability to generalize well to unseen data. Deep learning-based approaches in text detection are often based on neural networks that can learn features automatically from data. Convolutional Neural Networks (CNNs) proved to be effective in text detection tasks. CNNs can learn hierarchical representations

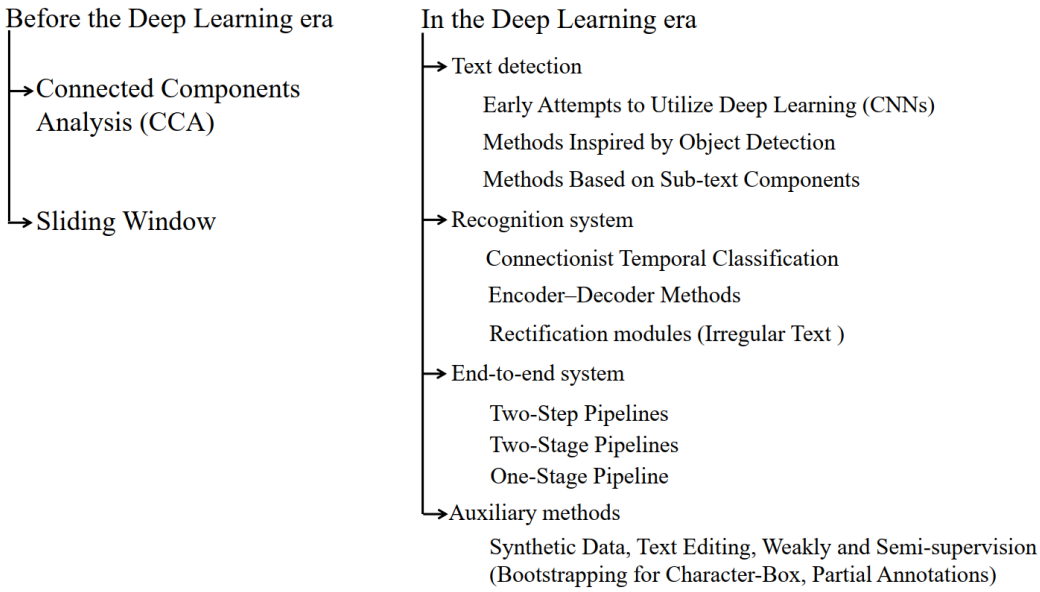


Figure 3.2: Text Detection Taxonomy

of visual features, enabling them to detect text in images with varying orientations, fonts, and sizes. Several deep-learning architectures have been proposed for text detection, including Faster R-CNN, YOLO, and SSD. These architectures exhibited outstanding performance on various text detection benchmarks.

3.2.1 A Taxonomy of Text Detection Techniques

The main characteristic of text detection is the features' design and testing. Deep learning incorporates many modern methods and enables researchers to avoid the cumbersome work of designing and testing handcrafted features. To provide a comprehensive investigation of the existing techniques for text detection, we classify existing techniques into two categories [123], before deep learning and in deep learning, as shown in 3.2.

1. Before deep learning: The text detection methods relied on two main techniques: a) Connected Components Analysis, which extracts the candidate components through various methods such as color grouping and then filters the non-text components using manually designed rules, as well as automatically trained on handcrafted features. b) Sliding Window where each window is specified as text areas. Those deemed positive are then grouped into text regions.

2. In deep learning: Deep learning techniques automatically learn features from the input images. In this era, researchers no longer need to design or test

handcrafted features, as deep learning models can independently learn the most relevant features for text detection. There are four main known deep learning types: (1) Text detection to localize text in natural images. This method went through several stages, starting with early attempts to utilize deep learning, which uses CNNs passing by methods inspired by object detection and methods based on sub-text components. (2) Recognition strategy that transforms the content of the detected text regions into linguistic characters. This strategy contains several methods, including connectionist temporal classification, encoder-decoder methods, and rectification modules for Irregular text. (3) End-to-end strategy, where text detection and recognition are performed in one unified pipeline. Several techniques can be cited, such as two-step, two-stage, and one-stage pipelines. (4) Auxiliary methods, where these kinds of approaches aim to support the principal task of text detection and recognition. Among these methods, we can mention synthetic data, text editing, and weakly and semi-supervision (bootstrapping for character-box, partial annotations).

3.2.2 Related Work

Advanced technology offers the possibility to place modern camera systems in different places, such as mobile phones, surveillance systems, and autonomous cars, generating high-quality images at a lower cost. The huge amount of generated images and videos increases the demand for the systems to read and interpret these images. In fact, the rapid development of Deep Neural Networks (DNN) has made significant progress in detecting text in scene images, which explains why recently several works have been released using DNN. Scene text detection can be included taxonomically under general object detection. The generic object detection methods' designs inspire many scene text detection algorithms. However, scene text discovery has different characteristics and challenges that require independent methodologies and solutions. Huang et al. [124] used CNNs to classify local image patches into text and non-text classes. The authors proposed scooping such image patches by using MSER features. Text Flow [125] applied CNNs to the whole images in a fully convolutional approach to detect characters. In [126], a convolutional neural network is utilized to determine if a pixel in the input image belongs to characters if it is inside the text region and the text orientation around the pixel. Hereafter, they considered the connected positive responses as detected characters or text regions. Lei et al. [127] presented a text detection technique belonging to the Connected Component (CC) based methods that divide text detection into two sub-issues: CC generation and text/non-text classification. The proposed technique uses neural networks and Color-Enhanced Contrasting external regions (CER). The authors of [128] presented text recognition in natural scenes. Two steps were performed to achieve the goal. Firstly,

detecting the text area which is an important function in the general performance of the OCR engine. The second step translates the detected text area using the accurate and open-sourced OCR engine Tesseract V5. In their work, the authors found that minor lighting and text angle differences resulted in significant text recognition errors. Sayahi et al. in [115] proposed a technique for detecting text in the scene file. Using neural networks and wavelet transformation to classify pixels into text and non-text areas. Yan et al. [129] presented a Uyghur language text detection system in complex background images for intelligent vehicles. They proposed a channel-enhanced maximally stable extremal regions (MSERs) algorithm to detect component candidates. Using the ICDAR 2003 database, Shehzad et al. [130] proposed a text detector that uses a small set of heterogeneous features spatially (choosing the Likelihood Ratio Test (LRT) as a weak classifier) combined to build an extensive set of features; the text segment is equal to 32×16 pixels. They presented a modified AdaBoost technique named CAdaBoost that considers the complexity of the weak classifiers at the feature selection step. The authors used a neural network to learn the localization rules automatically for text localization. As an application of Intelligent Transportation Systems (ITS), the authors in [131] proposed an approach to detect traffic boards in street-level images to recognize their information. After applying blue-and-white segmentation, the technique focuses on extracting local descriptors at some key points of interest. These images are then represented as a bag of visible words and categorized using Naïve Bayes vector machines. Finally, the images in which a pass-through was detected are considered to apply the text-detection method to it. Using the reverse geocoding service, a language model based on a dynamic dictionary for a limited geographical area is adapted to automatically read and save the information shown in the panels. Non-Latin families of cursive scripts like Hindi, Arabic, and Chinese pose a challenge for text detection from natural scene images. Syed and Muhammad [132] presented a technique to detect, predict orientation, and recognize Urdu ligatures in images. The authors have used the FasterRCNN algorithm and CNNs such as Resnet50 and Googlenet on images of size 320×240 pixels for detection and localization. In [133], the authors introduce an efficient pipeline that combines simplicity with effectiveness, delivering swift and precise text detection in natural scenes. This approach directly anticipates words or text lines with arbitrary orientations and quadrilateral shapes, utilizing a single neural network. The authors proposed a Differentiable Binarization (DB) module in [134] to perform the segmentation network's binarization process. In [135], the authors incorporated a DB module and an Adaptive Scale Fusion (ASF) module into the segmentation network, improving the network's performance by fusing multi-scale features. Optimized along with a DB module, a segmentation network can adaptively set the thresholds for binarization, simplifying the post-processing and enhancing the performance of text detection. A new approach is presented in [136] for text detection that combines a text proposal model with a deep relational reasoning network using a local graph and a Graph Convolutional

Network (GCN). The resulting network is end-to-end trainable, allowing for efficient and accurate arbitrary shape text detection. The authors of [137] propose a novel approach for representing arbitrary-shaped text contours using the Fourier Contour Embedding (FCE) method, which converts text instances into compact signatures in the Fourier domain. Based on FCE, the FCENet is developed with a backbone, feature pyramid networks, and post-processing with the Inverse Fourier Transformation and Non-Maximum Suppression, enabling end-to-end training for arbitrary-shaped text detection. The proposed FCE method is accurate and robust in fitting highly-curved text contours, and FCENet achieves good generalization for detecting arbitrary-shaped text in scenes. A Mask R-CNN-based approach for text detection that can detect multi-oriented and curved text in natural scene images is described in [138]. To enhance the feature representation ability of Mask R-CNN, the authors proposed to use the Pyramid Attention Network (PAN) as a new backbone network; PAN can more effectively suppress false alarms caused by text-like backgrounds. The authors of [139] proposed an efficient and accurate arbitrary-shaped text detector called the Pixel Aggregation Network. It consists of a low computational-cost segmentation head and a learnable post-processing module. The segmentation head comprises a Feature Pyramid Enhancement Module (FPEM) and a Feature Fusion Module (FFM), introducing multi-level information to guide better segmentation. The FFM gathers features from FPEMs of different depths for final segmentation. Pixel Aggregation (PA) implements the learnable post-processing, which precisely aggregates text pixels using predicted similarity vectors. The proposed method achieves high accuracy on various benchmark datasets while being computationally efficient In [140], the authors proposed a Progressive Scale Expansion Network (PSENet) to enable the accurate detection of text instances with arbitrary shapes. PSENet achieves this by generating kernels of different scales for each text instance and gradually expanding the most small-scale kernel to cover the complete shape of the text instance. Because the minimal scale kernels have large geometrical margins between them, this method effectively splits closely spaced text instances, making it easier to use segmentation-based methods for detecting text instances with arbitrary shapes. This paper proposes a text detection technique based on the channel-reducing step. The idea is based on the standard deviation value extracted from both RGB images to generate new images with clear objects and tensors of intermediate layers of the deep learning model. We implement the algorithm of the released work in [4] to reach this goal. Then, we use these newly generated images instead of the original images in the learning process. The reason for using this procedure is that in this proposed σNet, we apply the standard deviation over the RGB images in the σ step to avoid using any pre-trained model. The geometric representation of text positions is foundational in various domains, ensuring accurate and efficient transformation of text-containing regions into machine-readable formats. Several examples and case studies illustrate the significance of these processes. In scene text recognition [141], geometric representation aids in capturing the spatial layout of text through

methods like bounding boxes or quadrilateral shapes. This enhances accuracy in recognizing text within natural scenes. Geometric representation is crucial for aiding visually impaired individuals. By transforming text into machine-readable strings, systems can provide valuable assistance in interpreting and interacting with visual content. In autonomous driving systems, geometric representation is employed for sign recognition [142]. This enhances the understanding of visual content, contributing to safer and more efficient transportation.

While numerous existing methods in the field of text detection in images primarily rely on pre-trained deep learning models, the proposed approach takes a distinctive and innovative path by eschewing the use of pre-trained models. This decision carries several distinct advantages that set our work apart from the existing literature:

- Reduced Processing Time: By avoiding the need for pre-trained models, our approach significantly reduces the computational overhead typically associated with fine-tuning or adapting these models to specific tasks. This translates into faster text detection, making it well-suited for real-time or resource-constrained applications. Table 1 illustrates the use of a pre-trained model by the most relevant research. Our approach uses no pre-trained model to reduce computational load and increase processing speed.
- Customization and Adaptability: Our model's independence from pre-trained models allows for greater flexibility in adapting to various domains and datasets. Researchers and practitioners can tailor the algorithm more effectively to their specific needs, enabling better performance across various scenarios.
- Competitive Precision Rates: Our approach maintains competitive precision rates despite not relying on pre-trained models.

Method	Pretrained model
DB-RenNet [33]	ImageNet
DBNet [34]	Synthtext (model— lg)
DRRG [35]	ImageNet
DFCENet [36]	ImageNet
σNet (proposed model)	Without pre-trained model

Table 3.1: Text Detection results using different pre-trained models

3.3 Text Detection Pipeline Approach Specifications

We present a novel approach to image text detection that achieves state-of-the-art limits while significantly reducing the pipeline's computational complexity. This work focuses on the initial idea proposed in [4] to define the σ step. The present paper aims to adjust the adequate regions depending on the text detection in automation and robotics domains [143], such as automatic driving, visual search, and robot navigation, to quickly recover precious and meaningful textual information in image scenes.

There are some specifications in each multimedia domain, such as the background and the salient object colors. The best way to detect the salient objects is by eliminating the background and unnecessary objects. Therefore, if the user application area is well-defined, it will be easy to detect and extract the existing texts by considering them salient objects in this work.

So, the pipeline uses the standard deviation principle (σ step) in several pipeline stages. The first step applied to the original image reduces the three RGB channels to one, as shown in Equations 3.1 and 3.2.

Algorithm 1 illustrates in detail these steps, where for each pixel, we compute the distance to (0,0,0) and extract its σ value. In the end, Algorithm 1 returns the value of σ of each pixel depending to its distance to the RGB cube corner.

The binary semantic segmentation established by the fully convolutional network (FCN) architecture upgraded with the σ step called σNet in the context of text detection and extraction. The result of the deep learning model is a saliency map that is thresholded to obtain the binary image.

Finally, we use Connected Component Analysis (CCA) to generate a bounding polygon that encloses each text region. The first published idea in [4] applies the σ step on only the input RGB three-channel image without using the deep learning model.

Figure 3.3 presents the proposed approach by showing the role of the standard deviation image in the segmentation process, where for the general tensor T of n channels, we follow a consistent process for each group of three consecutive channels T_i within T with i ranges from 1 to $n/3$. As result, we obtain a single-channel tensor σ_i from the three channels of tensor T_i.

This method allows us to preserve the most pertinent information in the image or tensor while significantly reducing the computational complexity of subsequent operations. Text in scene images could be in random places, sizes, and colors. Furthermore, texts in images could be multi-oriented and curved or vertically.

Our approach has two main contributions based on [4]:

- First, it reduces the number of channels in the image sensor using the standard deviation of the image pixels [4], enabling us to achieve better performance with fewer computational resources.
- Second, it uses a deep learning model for text detection that leverages both downsampling and upsampling operations and skip connections, with a standard deviation of intermediate tensors, to preserve high-resolution information and improve the accuracy of the final output.

The proposed approach is designed to be a fast and light deep-learning text detection model.

3.3.1 Reducing the Channel Step (σ step)

A wide variety of machine learning and computer vision applications have used the clustering method as an unsupervised algorithm [144]. Clustering as an unsupervised machine learning method has great utility certainly in data science [145]. It groups a subset of the data set characterized by most similar features into the same cluster. A subset of dataset interclusters has the most dissimilarity features [146]. In unsupervised learning, unlabeled datasets are provided for training, and unsupervised algorithms can find clusters of data samples based on the notion of distance or another similarity measure [147]. The color divergence approach is a novel clustering method used for image segmentation and object detection that will be exploited in this work [4].

We name the standard deviation step of an image (RGB image defined as three channels) the tensor T of n channels with $n \geq 3$ as a generalization. In our study case of a tensor T, we take each of three successive channels T_i of T for $i = \{1, \ldots, [n/3]\}$ and apply Algorithm 1, which resumes the idea of the σ step. The output of Algorithm 1 will be used instead of using the three values of RGB pixel, where the standard deviation will keep the feature of the computed pixel and reduce its size from three to one value. To understand the secret behind using standard deviation, we discuss our study case by starting with the cube representation of the three channels tensor T_i. In this case, we define the cube C with eight vertices $j(j = 0 \ldots 7)$ in such a way that encloses the tensor. In this representation, we consider each element $t_i k$ for $k = \{1, \ldots, w \times h\}$ of T_i as have three coordinates (x_k, y_k, z_k) with w and h is the width and the height of the channel's tensor:

- x_k corresponding to the value of the element in the first channel

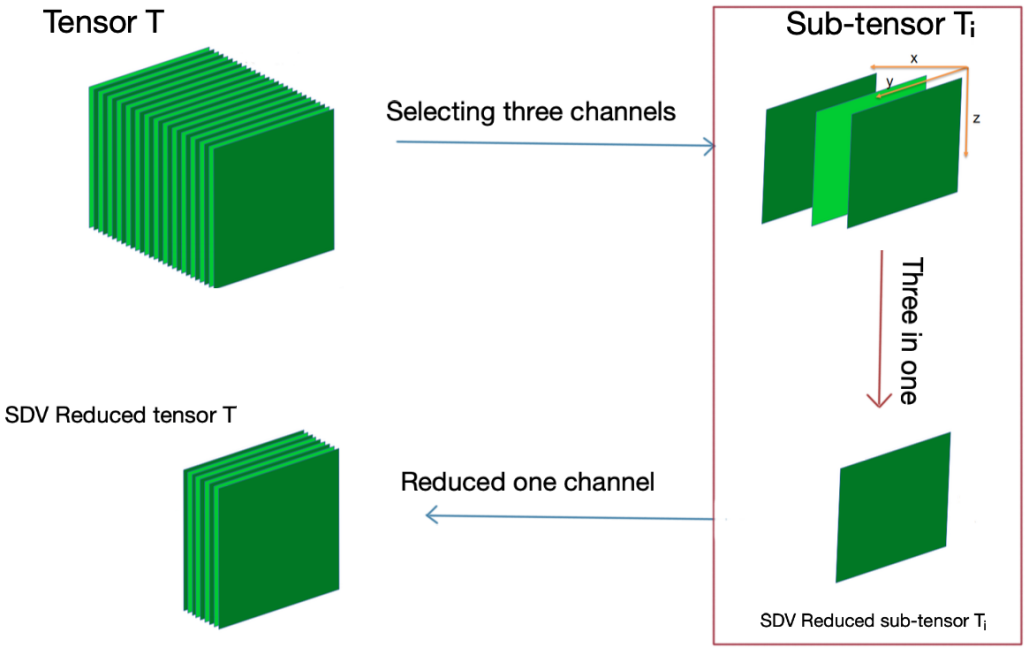


Figure 3.3: Illustration of Channel Reduction Principle

- y_k corresponding to the value of the element in the second channel
- z_k corresponding to the value of the element in the third channel

Then, we compute the Euclidean distance between each element of T_i and the eight vertexes of the cube $j(j = 0 \dots 7)$ as shown in Figure 3.3. The cube representation of three channels tensor T_i involves calculating the distance between each element t_{ik} of T_i and the cube's eight vertices using Equation 3.1. A distance vector $D_{tik} = [d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7]$ is defined for each element t_{ik}, where d_j is the distance between the element t_{ik} and vertex j of the cube. The distance is calculated using the formula

$$d_{jk} = \sqrt{x_{jk}^2 + y_{jk}^2 + z_{jk}^2} \tag{3.1}$$

where x_{jk}, y_{jk}, and z_{jk} are the difference between the element t_{ik} of the tensor and the corresponding vertex j. We will use Equation 3.2 to calculate the standard deviation vector sdv, which can enhance the accuracy of the outputs.

$$sdv_{tik} = \frac{\sqrt{\sum (d_{jk} - averageD_{tik})^2}}{8} \tag{3.2}$$

The distance values d_{jk} represent the distance between the element t_{ik} and the vertices $j(j = 0 \dots 7)$ of the cube. $averageD_ti$ denotes the element average distance vector D_{tik}.

Algorithm 1 Standard Deviation step.

Require: T_i, C;
Ensure: σ_i;
1: **for** ($j \leftarrow 0$ to 7) **do**
2: $\quad C \leftarrow C + vertexe_j$;
3: **end for**
4: **for** ($k \leftarrow 0$ to $w \times h$) **do**
5: $\quad$ **for** ($j \leftarrow 0$ to 7) **do**
6: $\quad\quad d_{jk} \leftarrow \sqrt{x_{jk}^2 + y_{jk}^2 + z_{jk}^2}$;
7: $\quad$ **end for**
8: $\quad \sigma_{tik} = \frac{\sqrt{\sum (d_{jk} - averageD_{tik})^2}}{8}$
9: $\quad \sigma_i[k] \leftarrow \sigma_{tik}$
10: **end for**
11: **return** σ_i;

Finally, from the three channels of the tensor T_i we obtain one channel Sdv_i tensor each element correspond to sdv_{tik} for $k = \{1, \ldots, w \times h\}$

The σ step is applied first to the input RGB three-channel image to reduce it to one σ channel, as illustrated in Figure 3.3. After that, it is applied to the tensor in the intermediate layer of the deep learning model. This method enables us to retain the most relevant information in the image or the tensor while significantly reducing the computational complexity of subsequent operations.

3.3.2 Deep Learning Text Detection Model

Segmentation-based approaches are commonly employed in scene text detection [134]. The significance of image classification and segmentation in object detection cannot be overstated, particularly in applications like autonomous driving. In addition to their role in object detection, these techniques are pivotal in advancing text detection and recognition systems, which are vital components of autonomous vehicles and blind assistance systems [148].

In digital image processing and analysis, image segmentation stands out as a fundamental and extensively used method. It facilitates the division of an image into multiple segments or regions, typically based on the distinctive characteristics of the pixels. This process offers several advantages, including the separation of foreground from background elements and the grouping of pixel regions with similarities in color or shape, ultimately enhancing the analysis and understanding of the image.

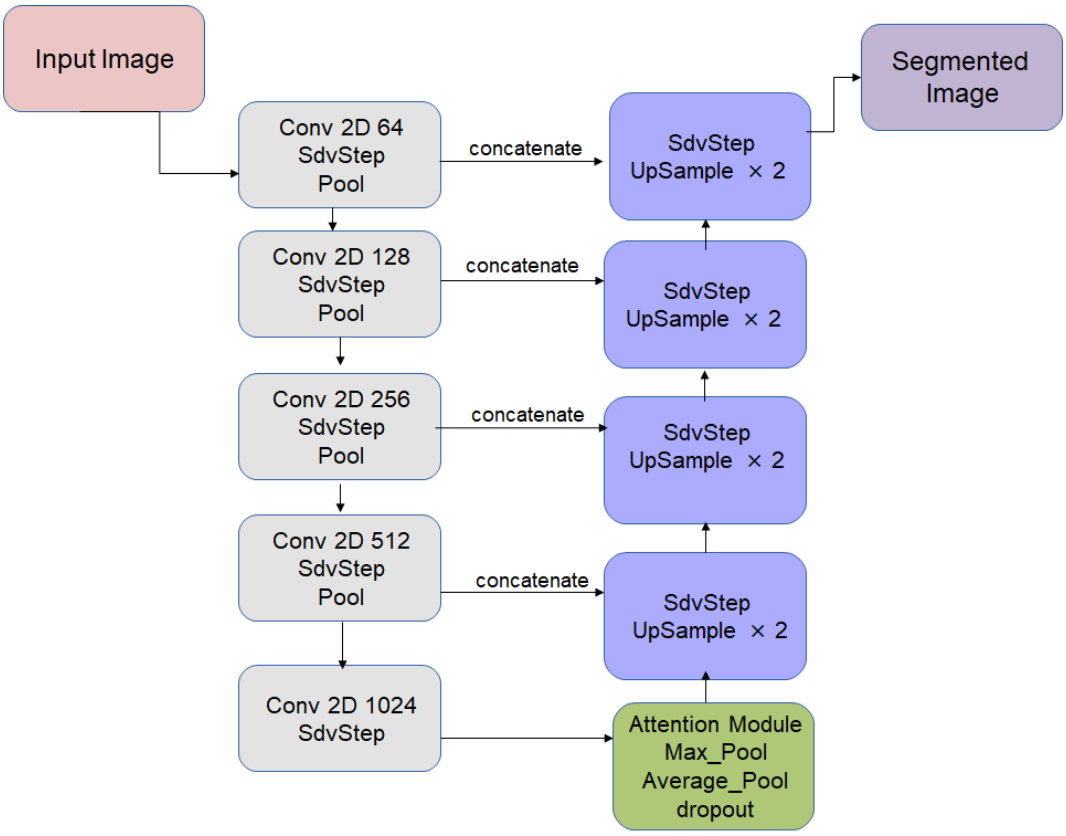


Figure 3.4: An overall overview of the proposed text detection approach

In the context of our research, image segmentation is a pivotal tool for achieving our text detection objectives. In order to enhance the accuracy of the regression method and introduce greater flexibility in the field of text object detection within scene images, we present a novel approach that involves harnessing the standard deviation matrix derived from the original image as the input to the FCN. In this stage, the proposed FCN incorporates the σ step into the tensors at various intermediate layers of the model.

Certain challenges could appear when using FCN, like the extensive variability in the sizes of word regions in scene images, where effectively detecting these regions requires a strategic adaptation. Recognizing the presence of larger words demands the extraction of features from the later stages of the neural network, while accurately delineating the geometry of smaller word regions relies on the availability of low-level information in the early stages. Consequently, our network must be capable of leveraging features at different levels to address these diverse requirements. To overcome this challenge, we embrace the concept of the U-shape architecture [52]. This approach facilitates the gradual fusion of feature maps while maintaining the compactness of the upsampling branches. This results in utilizing

features at different levels while minimizing computational overhead effectively.

The following steps show the pipeline details of the proposed method:

- **σ of intermediate channels in hidden Layers :** A fully convolutional network (FCN) is a type of neural network architecture commonly used in computer vision tasks such as semantic segmentation. The FCN architecture takes an input image and outputs a pixel-wise prediction of the class label for each pixel. This architecture with skip connection follows the ordinary architecture of a convolution neural network as the downsampling step and up-convolution as an upsampling step.

  In the first phase, we apply the standard deviation principle to the original three-channel image, which reduces it to a tensor of one channel, to preserve high-resolution information and improve the accuracy of the final output. Then, in the second phase, we develop a deep learning model σNet as shown in Figure 3.4, for text detection that leverages both downsampling and upsampling operations and skip connections. We apply the σ method in the downsampling step of our deep learning model, which consists of a contracting path defined by a sequence of operations that increase the number of feature channels. We reduce the number of feature maps using the standard deviation values, resulting in a tensor of $n/3$ channels instead of n channels. The upsampling step consists of an expanding path defined by the transposed 2D convolution (up-convolution) of the feature map that reduces the number of feature channels to half. Applying the standard deviation step between expanding paths also reduces the n's tensor channels to $n/3$. We introduce skip-connections between the downsampling and upsampling steps, which preserve and combine high-resolution information from previous layers.

  This approach enables our deep learning model to capture better the spatial relationships between different elements in the image, resulting in more accurate text detection.

- **Downsampling step:** It consists of a contracting path defined by a sequence of operations of two 3×3 convolutions with the same padding, each accompanied by the activation function rectified linear unit (ReLU) and the application of σ method which reduce the number of the feature map using the standard deviation, after that performing 2×2 max pooling with stride 2.

  At each step, the number of feature channels is increased by double, then decreased by a third using σ process until the fifth step, we add the attention module instead of 2×2 max pooling before the upsampling step

- **Attention module:** In this step we apply both the 2×2 max and average pooling to the output of Sdv map of the fifth step, we concatenate the result

of max and average pooling then we use the dropout regularization to avoid overfitting and improve the generalization ability of deep neural network.

- **Upsampling step:** It consists of an expanding path defined by the transposed 2D convolution (up-convolution) of the feature map followed by a skip connection and then by a 2×2 convolution that reduces the number of feature channels to half; after that, we add the σ process, which reduces the n's channel tensor to a third.

- **Skip connection** After each transposed convolution in the expanding path, the image is concatenated with the corresponding image result of max pooling from the contracting path. Skip connections preserve and combine high-resolution information from previous layers.

Unlike conventional deep-learning text detection models, the proposed method does not require feature extraction layer transfer learning. Therefore, the proposed method can be implemented with a relatively small memory and low computational complexity.

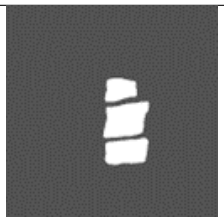
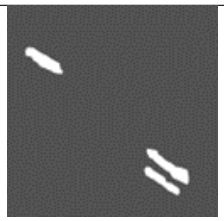
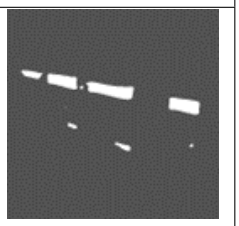
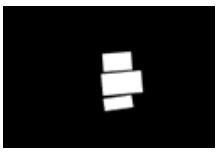
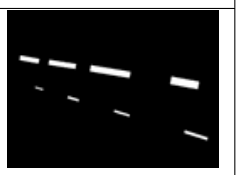
3.4 Experimental Evaluation and Results

This section gives the experimental setup and the three public datasets used to evaluate the proposed algorithm. Table 3.2 shows the result of the different steps of the pipeline, beginning with the original images through the standard deviation step and the output of the deep learning model. Compared with the ground truth, we remark that the result of the σNet is near to the ground truth and gives us an enhanced result in extracting text from scene images. While the σNet framework for text detection demonstrates significant advancements, it's important to acknowledge its limitations and areas for improvement. One notable weakness is its vulnerability to scenarios where text regions are closely positioned or intersecting.

3.4.1 Technical details

We conduct 100 epochs of fine-tuning the models on real-world datasets. The training batch size is set to 8. The model is compiled with a custom loss function related to the Dice coefficient, which is commonly used in image segmentation tasks, and uses the Adam optimizer with a learning rate of 1e-4 for training the model. We harness computational complexity by leveraging Google Colab's GPU resources for our experiments. Specifically, we utilize the T4 GPU, which boasts

Table 3.2: σ images detection compared with the ground and predicted images

Original Image				
σ step				
Predicted result				
Ground truth				
final result				

2,560 CUDA cores—parallel processing units designed for efficient parallel computation. This GPU comes equipped with 16 GB of GPU RAM, allowing us to work seamlessly with large datasets and complex models.

3.4.2 Datasets

- ICDAR 2015 dataset [149] includes 1000 training images and 500 testing images; Google Glasses captured this dataset with a resolution of 720 $\times$ 1280 and annotated the text instances at the word level.
- Total-Text dataset Total-Text [150] dataset is a collection of text data in various shapes, such as horizontal, multioriented, and curved. The dataset

consists of 1255 training images and 300 testing images. Each text instance in the dataset is labeled at the word level, meaning individual words within the text instances are annotated and labeled for training and evaluation purposes.

- MSRA-TD500 dataset [151] contains text data in English and Chinese. It comprises a total of 300 training images and 200 testing images. The text instances within the dataset are annotated at the text-line level. Additionally, we augment the dataset with an extra 400 training images from the HUST-TR400 dataset [152]

3.4.3 Dice Coefficient Metric

In the analysis, we evaluate the proposed approach using the dice coefficient metric. The σNet text detection pipeline incorporates a standard deviation step. To evaluate its performance, we compare the performance of σNet with a similar semantic segmentation model. The evaluation results, shown in Table 3.3, indicate the dice coefficient values for both σNet and the baseline RGB model on three public datasets. The dice coefficient is a commonly used metric for evaluating the similarity between two sets. In the context of text detection, it measures the overlap between the predicted text regions and the ground truth regions. A higher dice coefficient indicates a better performance of the text detection model. Table 3.3 presents the evaluation results of the MSRA-TD500 dataset and their corresponding variety in terms of the σ step in the dataset and model.

- "rgb fcn" is the proposed fcn model without σ step.
- "σ fcn" is the proposed fcn model with σ step applied only in the input image.
- "σNet " is the proposed model with σ step

For each dataset, we provide the size of the training set and test set, denoted as" training size" and" test size" respectively.

As a result, the standard deviation step in the σNet text detection pipeline significantly improves the dice coefficient values, thereby enhancing the model's performance on various public datasets as shown in Table 3.4 and 3.5.

This suggests that utilizing a one-channel input image with the standard deviation step effectively enhances the accuracy of the text detection model.

Table 3.3: Dice coefficient evaluation metric for semantic segmentation of MSRA-TD500 dataset .

Model	**Training set**	**Test set**	**Training size**	**Test size**	**Dice_coeff**
rgb fcn	$TD500TR400$	$TD500Test$	700	200	0.85
σ fcn	$\sigma TD500TR400$	$\sigma TD500Test$	700	200	0.94
σNet	$\sigma TD500TR400$	$\sigma TD500Test$	700	200	0.97

Table 3.4: Dice coefficient evaluation metric for semantic segmentation of ICDAR2015 dataset.

Model	**Training set**	**Test set**	**Training size**	**Test size**	**Dice_coeff**
rgb fcn	$ICDARtrain$	$ICDARtest$	1000	500	0.95
σNet	$\sigma ICDARtrain$	$\sigma ICDARtest$	1000	500	0.97

3.4.4 Precision and Recall

Among the existing methods, σNet is the only approach with no pre-trained model. This indicates that σNet does not benefit from pre-existing knowledge and instead learns directly from the provided data. Table 3.6 and 3.7 illustrate the precision and recall values of semantic segmentation with and without σ step.

3.4.5 Comparison with pre-trained models

We conduct comparative analysis between our proposed approach without pre-trained models and with the pretrained model used in [134, 136, 137] on the MSRA-TD500 dataset. In this study, we employed a cloud-based NVIDIA T4 GPU for the training and testing of our proposed model, while other studies utilized performance-focused hardware configurations. It is important to note that due to the inherent differences in hardware architecture, design, and optimization, we faced challenges when attempting to directly compare the results of other studies. Therefore, we implemented the proposed approach by incorporating a pre-trained ResNet-50 model which has been trained on the ImageNet dataset, as the feature extraction channel. Table 8 presents the evaluation results on the MSRA-TD500 dataset, where "Resnet fcn" is the fcn model with pre-trained Resnet-50 on ImageNet used in the contracting path of the fcn (feature extraction).

The comparative analysis conducted on the MSRA-TD500 dataset aimed to showcase the efficacy of the proposed approach. Table 8 presents the precision and recall results of the semantic segmentation phase. In contrast, Table 9 displays the precision and recall results of the final phase, which involves the prediction

Table 3.5: Dice coefficient evaluation metric for semantic segmentation of Total text dataset.

Model	**Training set**	**Test set**	**Training size**	**Test size**	**Dice_coeff**
σNet	$\sigma TotalTextTrain$	$\sigma TotalTextTest$	1254	300	0.97

Table 3.6: Precision and Recall of text detection results of MSRA-TD500 dataset.

Model	**Training set**	**Test set**	**Precision**	**Recall**
rgb fcn	$TD500TR400$	$TD500Test$	0.14	0.082
σ fcn	$\sigma TD500TR400$	$\sigma TD500Test$	0.85	0.77
σNet	$\sigma TD500TR400$	$\sigma TD500Test$	0.88	0.79

of bounding boxes for the text regions after segmentation. Table 9 delineates the performance improvements achieved by the σNet algorithm in contrast to existing methodologies in text detection. Analyzing the precision and recall metrics on the MSRA-TD500 dataset provides a comprehensive view of the algorithmic advancements.

The σNet algorithm demonstrates substantial progress in both precision (93.8) and recall (85.6), outperforming contemporary research endeavors. This outcome underscores its heightened effectiveness in accurately identifying text instances compared to prior state-of-the-art models. The notable improvement in precision indicates the algorithm's capability to limit false positives, while the enhanced recall signifies its proficiency in capturing a larger proportion of actual text instances. The significantly higher precision and recall values consolidate σNet as a frontrunner, showcasing its potential for advancing text detection methodologies. These results substantiate the compelling performance of the σNet algorithm, positioning it as a noteworthy advancement in text detection research, promising higher accuracy and reliability in practical applications.

3.4.6 A speed assessment comparative

A comparative analysis between the proposed model and the East text detection model [133] is conducted, emphasizing speed assessment as the focal point of comparison. Table 10 presents the result of speed comparison analysis.

We deployed the East [133] model to ascertain its performance metrics. The East model achieved an FPS of 1.44 on our system. We observe that the proposed model achieved a significantly higher FPS of 2.37, showcasing markedly superior

Table 3.7: Precision and Recall for semantic segmentation of ICDAR2015 dataset

Model	**Training set**	**Test set**	**Precision**	**Recall**
rgb fcn	$ICDAR2015Train$	$ICDAR2015Test$	0.75	0.32
σNet	$\sigma ICDAR2015Train$	$\sigma ICDAR2015Test$	0.93	0.72

Table 3.8: Precision and Recall of text detection results of MSRA-TD500 dataset.

Model	**Training set**	**Test set**	**Dice_coeff**	**Precision**	**Recall**
$Resnetfcn$	$TD500TR400$	$TD500Test$	0.92	0.88	0.30
σNet	$\sigma TD500TR400$	$\sigma TD500Test$	0.97	0.88	0.79

performance in terms of speed. This notable disparity in processing speed underscores the efficiency gains inherent in our proposed model compared to the established East text detection model. In scenarios where text strongly mimics the background, the effectiveness of the proposed text detection methodology may falter, introducing challenges in precisely isolating individual text components. This particular scenario highlights a critical aspect where the accuracy of the detection framework faces potential compromises. We observe that when text strongly resembles the background, the framework's efficiency may diminish, impacting detection accuracy. These weaknesses underline the need for further refinement in handling intricate text layouts, enhancing adaptability to varied backgrounds, and optimizing precision efficiency for broader applicability. These observations underscore the imperativeness of further refinement in addressing intricate text layouts. Enhancements should focus on augmenting the adaptability of the framework, particularly in scenarios with diverse and complex backgrounds.

3.5 Conclusion and Future Work

In conclusion, image text detection plays a crucial role in computer vision, but it remains a challenging task. This article presented a novel technique that enhances the accuracy of text detection by incorporating standard deviation values into the learning process. The experimental results clearly demonstrated the effectiveness of this technique in improving the performance of existing text detection methods, especially in challenging scenarios with low contrast or noisy backgrounds. The proposed technique holds great potential for improving text detection accuracy in various applications, including document analysis, scene text recognition,

Model	Precision	Recall
East model[33]	87.2	75.3
DB-ResNet[34]	91.5	79.2
DBNet[35]	91.5	83.3
DRRG [36]	88.05	82.30
σNet	93.8	85.6

Table 3.9: Precision and Recall of text detection results of MSRA-TD500 dataset

Model	Training set	Test set	FPS
East model[133]	TD500TR400	TD500 Test	1.44
σNet	$\sigma TD500TR400$	$\sigma TD500Test$	2.37

Table 3.10: Comparison of Speed Assessments with the East Model

and optical character recognition (OCR). By leveraging standard deviation values, the technique can better handle image quality variations and enhance text detection algorithms' robustness. The proposed technique opens up opportunities for its application in the context of IoT and low-end devices. Using standard deviation values can provide a lightweight and efficient solution for text detection on resource-constrained devices, making it suitable for deployment in scenarios where computational resources are limited. This is particularly relevant in IoT applications, where devices with limited processing power and memory are commonly used. Further future research can investigate the use of this technique in other computer vision tasks and explore its potential for real-world applications. However, it is vital to acknowledge the limitations of this approach. The proposed technique may encounter challenges when dealing with the closest text regions and multiple intersections of texts. Furthermore, in scenarios where text closely resembles the background, its efficiency may be compromised.

3.5.1 Future Research

The proposed approach not only holds promise but also opens doors to numerous opportunities for expansion. Its applicability in IoT and low-end devices stands as an intriguing prospect. Using standard deviation values can provide a lightweight and efficient solution for text detection, making it particularly fitting for resource-constrained devices often found in IoT applications. These devices, with limited processing power and memory, could benefit greatly from the enhanced accuracy and efficiency this technique offers. The path ahead beckons further exploration,

with future research avenues delving into its potential in other computer vision tasks and its applicability in a diverse array of real-world scenarios.

Chapter 4

Advances in the standard deviation detector method

This chapter consists of the article: "Categorization of Digital Pathology Image using Deep Learning model, ICFNDS '23 The 7th International Conference on Future Networks and Distributed Systems, Dubai, United Arab Emirates, December, 2023.

4.1 Introduction

Computer-Aided Detection and Diagnosis (CAD) is a dynamic and rapidly expanding research domain. The practicality of CAD, especially when coupled with cutting-edge deep learning technology within the artificial intelligence (AI) sphere, has significantly alleviated the persistent challenge of manual data interpretation within traditional CAD systems.

Colorectal carcinoma ranks as a major contributor to cancer-related mortality in the United States, securing the third position in terms of frequency among men, trailing only prostate and lung/bronchus cancers. Similarly, among women, it holds the third spot, following breast and lung/bronchus cancers [153]. Globally, colorectal cancer poses a substantial public health threat due to its widespread occurrence across many nations. This malignancy primarily targets the colon or rectum, often imposing profound adverse effects on individuals' quality of life.

In digital pathology, accurately identifying distinct tissue types within histological images is paramount, particularly for analyzing colorectal cancer specimens. Texture analysis has traditionally been widely adopted for quantifying the tumor-to-stroma ratio in histological samples. However, it is imperative to acknowledge that histological images frequently encompass more than just two tissue types, and prior research efforts aimed at addressing the multi-class challenge in colorectal cancer detection have been notably constrained [154].

To address these challenges and advance our understanding, there is a growing focus on enhancing the capabilities of digital pathology tools and deepening our comprehension of this intricate ailment. This pursuit involves harnessing the power of machine learning algorithms and artificial intelligence technologies [155].

Traditional machine learning algorithms and deep learning algorithms are the two types of AI technologies frequently employed in medical imaging. Combining AI and security including privacy [156, 157] and confidentiality [158, 159] in innovative solutions can give more efficient results.

The Convolutional Neural Network (CNN) represents a cutting-edge model in the realm of deep learning, particularly within the subfield of machine learning, where the computer autonomously extracts crucial features from raw input data. When computational resources are limited or inaccessible, using pre-trained CNNs and transfer learning mechanisms presents a viable alternative. Furthermore, integrating attention mechanisms has yielded promising results in the realms of diagnosis and categorization. To enhance the accuracy and effectiveness of classification and detection in the numerous histological images associated with colorectal cancer, this thesis proposes a novel approach. This approach combines a pre-trained CNN, specifically ResNet-50, serving as the contracting path within the fully convolutional network (FCN). This is augmented with a spatial attention module, followed by the generation of a head map composed of fully connected layers, and culminating in the application of a softmax classifier.

4.2 Related work

Several studies presented by *Timo Gaiser* and *Jakob Nikolas Kather* with their colleagues on the classification of digital pathological images using deep learning have been done, where one of the most important steps in the development of digital pathology is the recognition of different tissue types within histological images. This study showed 5,000 histological images of human colorectal cancer tissues, including eight different forms of cancer. This dataset assessed how well different texture descriptors and classifiers perform in classification tasks. With an accuracy of 87.4% across eight classes, this advance has increased the cutting-edge accuracy for distinguishing tumor stroma from 96.9% to 98.6% while also setting a new standard for differentiating multiple tissue classes.[160].

A study proposed by [161] uses deep learning to classify digital disease images, where histopathology (or tissue microscopy) is an important diagnostic process for colorectal cancer. Pathologists often analyze the histology to determine the various tissue types, paying close attention to the tumor-to-stroma ratio. Deep transfer learning is used in this study to classify tissue samples for colorectal cancer

automatically. This technique employs a one-cycle strategy for discriminating fine-tuning and structure-preserving color normalization to get better outcomes. The study also offers visual justifications for the tissue categorization conclusions the deep neural network reached. The research also investigates SqueezeNet's memory-efficient architecture for deployment on resource-constrained devices, resulting in an impressive test accuracy of 96.2%

4.3 Digital Pathology Image Specification

Colorectal cancer is the third most prevalent form of cancer globally, constituting about 10% of all cancer diagnoses, and is the second leading contributor to cancer-related fatalities worldwide. This disease primarily impacts older adults, most manifesting in individuals aged 50 and older.[162]. Digital pathology involves the analysis of high-resolution medical images, and deep learning models have shown remarkable success in this field, particularly for tasks like image classification, object detection, and segmentation [163, 164]. ResNet Networks excellently performed in image classification tasks among their greatest available solutions [26]. Also, Fully Convolutional Networks (FCNs) showcase the effectiveness in understanding the content of images, which is fundamental to image classification[165].

The proposed classification approach 4.1 centers around the Standard Deviation (SDV) dimensionality reduction over the Fully Convolutional Network (FCN), followed by the design of a head map for image classification.

4.3.1 Architectural model design

A fully convolutional network is a type of neural network architecture with downloading and upsampling layers inside the network used primarily for tasks like semantic segmentation, image-to-image translation, and dense prediction. Within the contracting path of the FCN, we incorporate ResNet-50. This ResNet-50 module handles image encoding, leading to a substantial reduction in dimensionality and effectively capturing essential features at bottleneck stages. Then, before the decoding path of the FCN, we add the spatial attention module [166] used to weight or emphasize different spatial regions of the input feature map. In the attention module, the output of the ResNet-50 is downsampled using both max pooling and average pooling with a 2×2 kernel size; the results of the max pooling and average pooling operations are concatenated. This technique improves the performance of neural networks in various computer vision tasks.

The expanding path involves several steps. It starts with a 2×2 upsampling operation, which is carried out by applying transposed 2D convolution to the

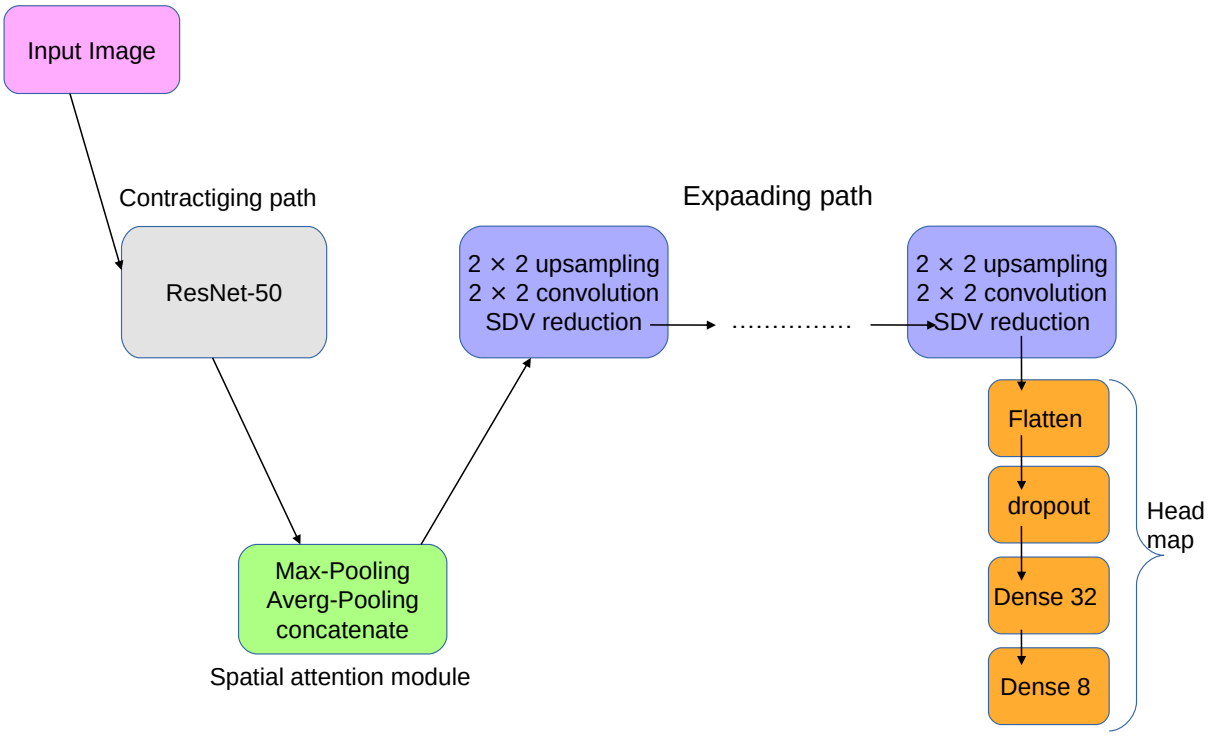


Figure 4.1: The Proposed model design

output of the spatial attention module. This is followed by a 2×2 convolution, a widely used technique for reducing the number of feature channels by half. This convolution helps manage the dimensionality of feature maps.

After the dimensionality reduction with the 2×2 convolution, we further reduce the tensor's channel count to one-third using a process known SDV. This sequential reduction of the feature map dimensions aims to optimize and streamline the data representation.

We carry out a repetitive cycle of two key operations: upsampling and channel reduction. We iterate through these steps until the feature maps return to the dimensions of the original input image. This iterative procedure is common in the decoding phase of numerous convolutional neural network designs, particularly in applications like image segmentation and image generation. In such tasks, the objective is to restore intricate details and spatial information from feature maps at lower resolutions.

4.3.2 SDV dimensionnality reduction

For a general tensor T, We adhere to a uniform procedure for every set of three consecutive channels T_i within the group T, where i spans from 1 to $[n/3]$.

To initiate this process, we represent the cube representation of three-channel tensor T_i, denoted as C, containing eight vertices labeled from 0 to 7, enveloping the tensor. In this representation, we treat each element t_{ik}, where k ranges from 1 to $w \times h$ within T_i, as having three coordinates: (x_k, y_k, z_k). Here, w and h represent the width and height of the channel's tensor, respectively.

- x_k corresponds to the value of the element in the first channel. - y_k corresponds to the value of the element in the second channel. - z_k corresponds to the value of the element in the third channel.

We then compute the Euclidean distance between each element of T_i and the eight vertices of the cube C (j where j ranges from 0 to 7). This cube representation involves calculating the distance between each element t_{ik} of T_i and the cube's eight vertices. For each element t_{ik}, a distance vector $D_{tik} = [d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7]$ is defined, where d_j represents the distance between the element t_{ik} and vertex j of the cube. The distance is computed using the formula:

$$d_{jk} = \sqrt{x_{jk}^2 + y_{jk}^2 + z_{jk}^2}$$

Here, x_{jk}, y_{jk}, and z_{jk} denote the differences between the element t_{ik} of the tensor and the corresponding vertex j.

We then use the formula:

$$sdv_{tik} = \frac{\sqrt{\sum (d_{jk} - \text{averageD}_{tik})^2}}{8}$$

to calculate the standard deviation vector sdv, which enhances the accuracy of the outputs. In this formula, the distance values d_{jk} represent the distances between the element t_{ik} and the vertices j of the cube, and averageD_{tik} is the average distance vector for element t_{ik}.

Finally, from the three channels of tensor T_i, we obtain a single-channel tensor Sdv_i, where each element corresponds to sdv_{tik} for k ranging from 1 to $w \times h$. This approach enables the retention of crucial information within the image or tensor while substantially decreasing the computational complexity of subsequent operations.

4.3.3 Head Part

We meticulously curate a specific set of network layers to ensure precise image categorization and address potential challenges effectively. It is crucial to emphasize that the Head Part of the network entrustes the classification task. The Head Part comprises a total of fourteen layers, each distinguished by its unique attributes:

1. A flattened layer responsible for processing the Fully Convolutional Network (FCN) output.
2. A Dropout Layer with a dropout rate of 0.5, strategically incorporated to enhance regularization and prevent overfitting.
3. A Dense Layer that uses the Rectified Linear Unit (ReLU) activation function, augmenting the network's capacity to capture intricate patterns and features.
4. Another Dense Layer with eight units employs the softmax activation function. This layer is pivotal in the final categorization process, yielding the classification output.

To summarize, our approach encompasses the following key stages: We kick off by constructing the encoding path of the Fully Convolutional Network (FCN) using the Imagenet dataset as the fundamental building block. This encoding path leverages the power of ResNet-50. Next, as we progress toward the decoding path of the FCN, we introduce an attention module that enhances the network's ability

to focus on relevant features. The decoding path includes an essential upsampling step achieved through transposed 2D convolutions applied to the feature map. The 2x2 convolution operation and SDV dimensionality reduction follow the upsampling step. Lastly, we incorporate the head map, a key component responsible for the classification task within the network. This component plays a central role in ensuring accurate classification.

4.4 Discussion and Results

This section provides details on the experimental configuration and the publicly available datasets employed for assessing the proposed algorithm.

4.4.1 Description of the dataset

The experimental setup involves the digital capture of histological images using an Aperio ScanScope at 20x magnification, with images being RGB and having a pixel size of 0.495 μm. The histological samples are sourced from the pathology collection of the Institute of Pathology, University Medical Center Mannheim, Heidelberg University, Germany, specifically depicting formalin-fixed paraffin-embedded human colorectal adenocarcinomas (primary tumors) as detailed in a cited reference [167].

Two public datasets are mentioned for evaluating the proposed algorithm:

- Kather-texture-5000 dataset: - This dataset consists of 5000 histology pictures. - Each picture measures 150 by 150 pixels (74 by 74 μm). - The images are categorized into eight tissue groups, with the folder name indicating the tissue group for each photograph.
- Kather-texture-10 dataset: - This dataset contains 10 larger histology images. - Each image is sized at 5000 $\times$ 5000 pixels. - The larger images may display multiple tissue types in each picture.

These datasets are likely used to assess and validate the performance of the proposed algorithm in the context of analyzing histological images.

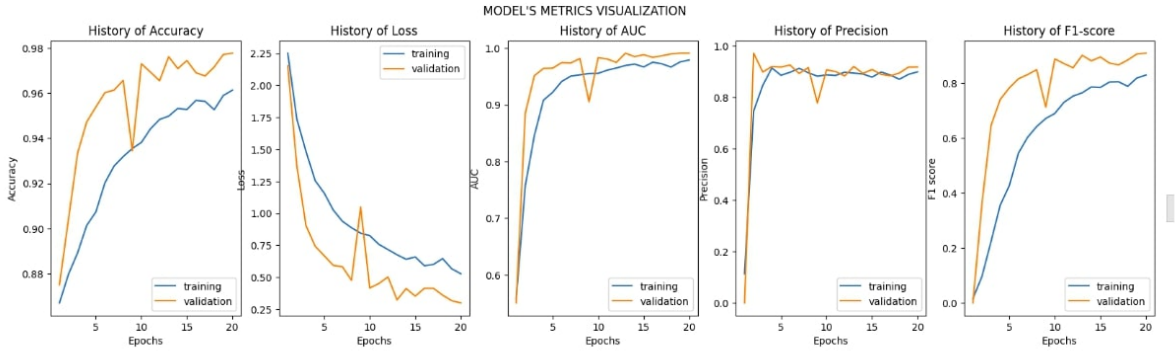


Figure 4.2: A summary of the metrics' performance history

4.4.2 Experiments and results

The model performed well after being trained across a number of epochs, as can be seen in the figure 4.2 below.

The figure 4.3 suggest that the proposed algorithm for the multi-class classification of histological images for embedded human colorectal adenocarcinomas (primary tumors) is performing very well. We break down the interpretation for each metric as follows :

- Test Loss (0.2699): - The loss is relatively low, indicating that the model is effectively minimizing errors during training. This suggests that the model is learning the patterns in the data.
- Test Accuracy (0.9801): - The high accuracy (98%) indicates that the model correctly predicts the classes for the majority of instances in the test set, demonstrating its effectiveness in multi-class classification.
- Test Precision (0.9254): - Precision is a measure of how many of the predicted positive instances are actually positive. In this case, the average precision across all classes is approximately 92.5%, suggesting that the model has a high precision for each class.
- Test Recall (0.9143): - Recall, or sensitivity, is a measure of how many actual positive instances the model correctly identifies. With an average recall of about 91.4%, the model is capturing a high proportion of positive instances for each class.

```
Test Loss: 0.26999497413635254
Test Accuracy: 0.9800724387168884
Test Precision: 0.9254278540611267
Test Recall: 0.9142512083053589
Test AUC: 0.992307186126709
Test F1 Score: 0.919860303401947
```

Figure 4.3: The result of the evaluation of the proposed model

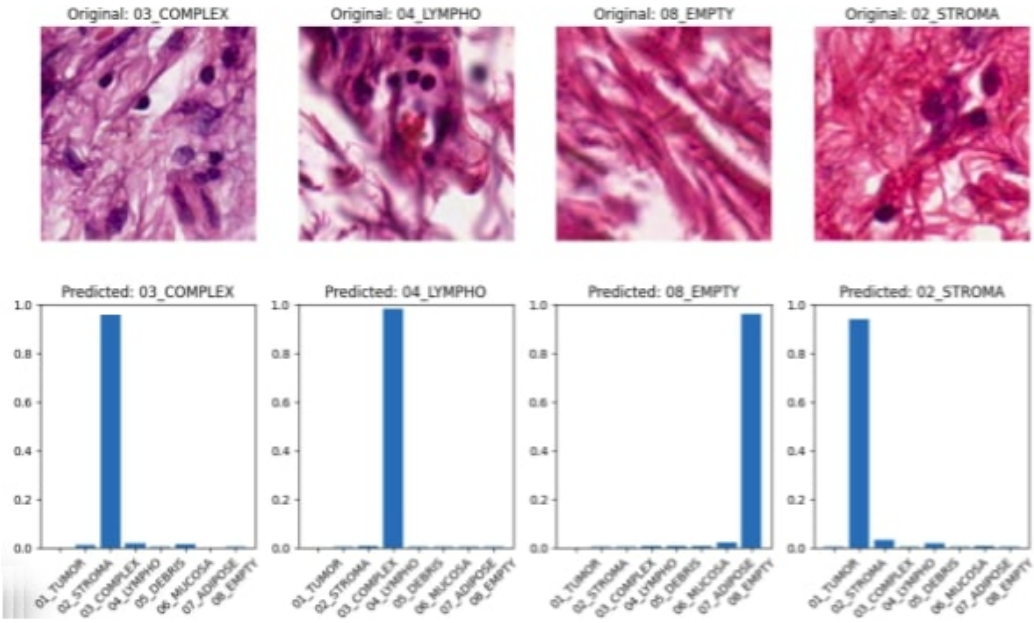
- Test AUC (0.9923): - The Area Under the ROC Curve is close to 1, indicating excellent discrimination between classes. This is a positive sign of the model's ability to distinguish between different categories.
- Test F1 Score (0.9199): - The F1 score is the harmonic mean of precision and recall. With an average F1 score of approximately 92%, the model demonstrates a good balance between precision and recall for each class.

These results suggest that the proposed algorithm is robust and effective in classifying histological images of human colorectal adenocarcinomas into multiple categories. The high values across accuracy, precision, recall, AUC, and F1 score indicate a strong performance in multi-class classification.

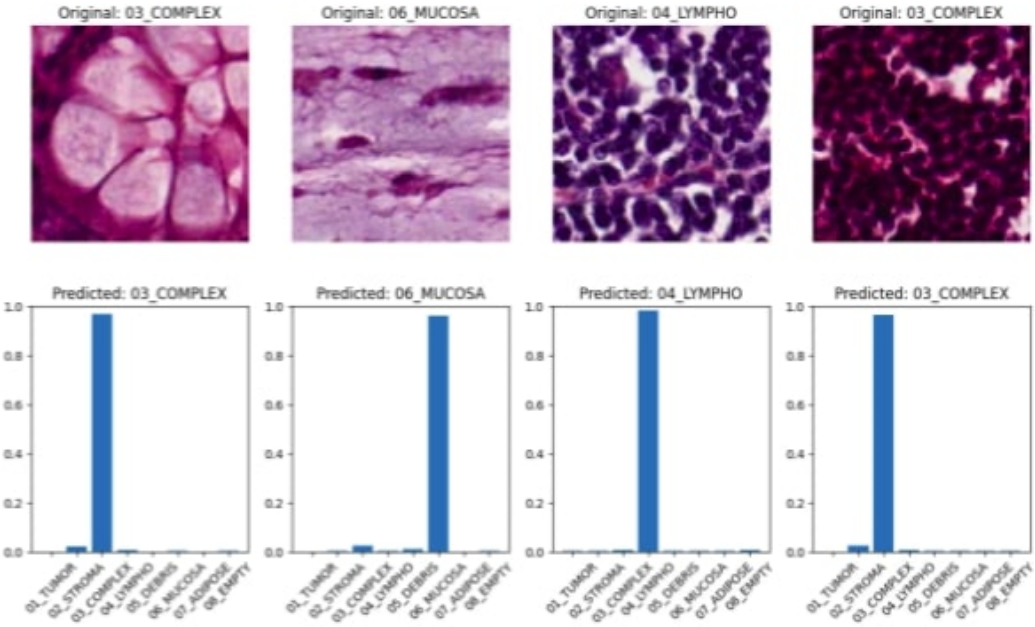
Figure 4.4 illustrates the outcome of the classification test, comparing the original and predicted tissue types in histological images. Upon comparing with the original images, it is evident that the classification test results closely align with the ground truth, resulting in enhanced accuracy in categorizing both the type and stage of colorectal cancer histology.

4.5 Conclusion

In conclusion, we presented in this paper a new deep-learning model for efficiently categorizing colorectal carcinoma images and accurately diagnosing the disease. Our novel approach using a fully convolutional network, augmented with a pre-trained convolutional neural network and a spatial attention module, represents a significant advancement in computer-aided detection and diagnosis within the medical research landscape. The carefully curated architectural design has proven instrumental in achieving accurate image categorization. The promising results



(a) Examples of the predicted 4 classes.



(b) Examples of the remaining 4 predicted classes.

Figure 4.4: The result of the evaluation of the proposed model

obtained from our model underscore its efficacy in addressing the multi-class challenge associated with colorectal cancer detection in digital pathology. As we navigate the evolving landscape of medical research and diagnostic methodologies, the success of our approach highlights the potential of deep learning and attention mechanisms in significantly enhancing classification accuracy. This research not only contributes to the specific domain of colorectal cancer detection but also serves as a blueprint for the broader application of advanced technologies in medical imaging and pathology. Continued collaboration between computer science and medical research communities will be crucial for further refinement, validation, and integration of such innovative models into clinical practice, ultimately shaping the future of computer-aided medical diagnostics.

Conclusion

Object detection plays a crucial role in computer vision, serving as a cornerstone for various applications such as autonomous driving, surveillance, and image analysis. However, despite significant advancements in recent years, it remains a challenging task due to factors such as varying object scales, occlusions, complex backgrounds, and limited training data. Efforts in research and development continually strive to enhance the accuracy, efficiency, and robustness of object detection algorithms to address these challenges and unlock the full potential of computer vision technology.

This work addresses the challenge of object detection and identification by introducing an innovative technique that integrates standard deviation values into the learning process (σNet). By doing so, it significantly improves object detection accuracy. Applications such as text detection in document analysis and scene text recognition can benefit from this approach. Notably, the method enhances text detection algorithms' robustness and effectively handles variations in image quality. This adaptability is particularly valuable in scenarios where image quality varies significantly.

On the other hand, we introduced a novel deep learning model that effectively classifies colorectal cancer images and provides precise diagnoses. Our innovative method, which incorporates the σNet principle, represents a substantial advancement in computer-aided detection and diagnosis within medical research. The carefully chosen architectural design has been instrumental in achieving accurate image categorization. The encouraging results of our approach demonstrate its efficacy in addressing the multi-class challenge of colorectal cancer diagnosis in digital pathology.

We concentrate on the thesis's context in the first chapter and the general introduction. The study challenge, the thesis's motivations and goals, and the key contributions are all outlined in the general introduction.

In the first chapter, the Deep Learning methodology used in computer vision is presented, with particular emphasis on Convolutional Neural Networks (CNNs) for tasks like object detection and semantic segmentation. Since CNNs can automatically learn hierarchical features from unprocessed input data, they are especially well suited for these kinds of applications.

Chapters two and three discuss the thesis' primary contribution. The color divergence method used for object recognition and image segmentation is presented in the second Chapter. This technique uses color information to identify objects in images and segment them. The third Chapter focuses on the use of the σNet strategy for text detection in natural scene images. It builds upon the σNet approach by incorporating the color divergence technique. This chapter explores the intricacies of the σNet methodology, including its architecture, training regimen, and performance assessment.

In chapter four, a σNet with a pre-trained convolutional neural network and a spatial attention module is used to address the multi-class challenge in colorectal cancer diagnosis.

This study not only advances the field of colorectal cancer diagnosis specifically but also provides a model for more widely utilizing cutting-edge technology in pathology and medical imaging. The future of computer-aided medical diagnostics will ultimately be shaped by the ongoing cooperation between the computer science and medical research communities, which will be essential for the further development, verification, and application of such innovative models in clinical practice.

The suggested technique presents exciting opportunities for use in low-end devices and the Internet of Things (IoT). By leveraging standard deviation values, it offers a lightweight and effective solution for text identification on devices with limited resources. This is especially crucial in situations where processing power is scarce. For IoT applications, where low-processing-power and memory devices are prevalent, this approach simplifies object detection. Its increased precision and efficiency can significantly benefit these resource-constrained devices. Looking ahead, future research should explore its potential in other computer vision tasks and its applicability across a wide range of real-world scenarios.

Bibliography

Bibliography

[1] Sibanjan Das and Umit Mert Cakmak. *Hands-On Automated Machine Learning: A beginner's guide to building automated machine learning systems using AutoML and Python*. Packt Publishing Ltd, 2018.

[2] Sana Sahar Guia, Abdelkader Laouid, Mohammad Hammoudeh, Ahcène Bounceur, Mai Alfawair, and Amna Eleyan. Co-simulation of multiple vehicle routing problem models. *Future Internet*, 14(5):137, 2022.

[3] Sana Sahar Guia, Abdelkader Laouid, Mohammad Hammoudeh, and Mostafa Kara. Intelligent image text detection via pixel standard deviation representation. *Computer Systems Science & Engineering*, 48(4), 2024.

[4] Sana Sahar Guia, Abdelkader Laouid, Reinhardt Euler, Mohammed Amine Yagoub, Ahcene Bounceur, and Mohammad Hammoudeh. A salient object detection technique based on color divergence. In *Proceedings of the 4th International Conference on Future Networks and Distributed Systems*, pages 1–6, 2020.

[5] Sana Sahar Guia, Abdelkader Laouid, and Khaled Chait. Categorization of digital pathology image using deep learning model. In *Proceedings of the 7th International Conference on Future Networks and Distributed Systems*, pages 397–402, 2023.

[6] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[7] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[9] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. Recent advances in convolutional neural networks. *Pattern recognition*, 77:354–377, 2018.

[10] Kazuyuki Hara, Daisuke Saito, and Hayaru Shouno. Analysis of function of rectified linear unit used in deep learning. In *2015 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2015.

[11] Leiyu Chen, Shaobo Li, Qiang Bai, Jing Yang, Sanlong Jiang, and Yanming Miao. Review of image classification algorithms based on convolutional neural networks. *Remote Sensing*, 13(22):4712, 2021.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[14] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer, 2014.

[15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[16] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520, 2016.

[17] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[18] Amit Kumar, Azadeh Alavi, and Rama Chellappa. Kepler: Keypoint and pose estimation of unconstrained faces by learning efficient h-cnn regressors. In *2017 12th ieee international conference on automatic face & gesture recognition (fg 2017)*, pages 258–265. IEEE, 2017.

[19] Li Yi, Hao Su, Xingwen Guo, and Leonidas J Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2282–2290, 2017.

[20] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? *Advances in neural information processing systems*, 27, 2014.

[21] Xintao Ding, Qingde Li, Yongqiang Cheng, Jinbao Wang, Weixin Bian, and Biao Jie. Local keypoint-based faster r-cnn. *Applied Intelligence*, 50:3007–3022, 2020.

[22] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Ncnet: Neighbourhood consensus networks for estimating image correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):1020–1034, 2022.

[23] Qinsong Li, Shengjun Liu, Ling Hu, and Xinru Liu. Shape correspondence using anisotropic chebyshev spectral cnns. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 14658–14667, 2020.

[24] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. Sfnet: Learning object-aware semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2278–2287, 2019.

[25] Andrej Krenker, Janez Bešter, and Andrej Kos. Introduction to the artificial neural networks. *Artificial Neural Networks: Methodological Advances and Biomedical Applications. InTech*, pages 1–18, 2011.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[27] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[28] Cheng Tai, Tong Xiao, Yi Zhang, Xiaogang Wang, et al. Convolutional neural networks with low-rank regularization. *arXiv preprint arXiv:1511.06067*, 2015.

[29] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[30] Jia Song, Shaohua Gao, Yunqiang Zhu, and Chenyan Ma. A survey of remote sensing image classification based on cnns. *Big earth data*, 3(3):232–254, 2019.

[31] Tao Lei, Yu Zhang, and Yoav Artzi. Training rnns as fast as cnns. 2018.

[32] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021.

[33] Kyle Aitken, Vinay Ramasesh, Yuan Cao, and Niru Maheswaranathan. Understanding how encoder-decoder architectures attend. *Advances in Neural Information Processing Systems*, 34:22184–22195, 2021.

[34] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[35] Baozhou Zhu, Peter Hofstee, Jinho Lee, and Zaid Al-Ars. An attention module for convolutional neural networks. In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part I 30*, pages 167–178. Springer, 2021.

[36] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

[37] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[38] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1605.06211, 2016.

[39] Critically Sampled Perfect Reconstruction Filter Banks. Mus421/ee367b lecture 9 multirate, polyphase, and wavelet filter banks. 2020.

[40] Dragoș Dumitrescu and Costin-Anton Boiangiu. A study of image upsampling and downsampling filters. *Computers*, 8(2):30, 2019.

[41] Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal. The importance of skip connections in biomedical image segmentation. In *International workshop on deep learning in medical image analysis, international workshop on large-scale annotation of biomedical data and expert label synthesis*, pages 179–187. Springer, 2016.

[42] Yongfeng Xing, Luo Zhong, Xian Zhong, et al. An encoder-decoder network based fcn architecture for semantic segmentation. *Wireless Communications and Mobile Computing*, 2020, 2020.

[43] Sen Wang, Yunlong Pan, Mingfang Chen, Yinhui Zhang, and Xing Wu. Fcn-sfw: steel structure crack segmentation using a fully convolutional network and structured forests. *IEEE Access*, 8:214358–214373, 2020.

[44] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3523–3542, 2021.

[45] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[46] Linda G Shapiro and George C STockMan. Icomputer vision", pp 279y325. *New Jersey*, 2001.

[47] Ta Yang Goh, Shafriza Nisha Basah, Haniza Yazid, Muhammad Juhairi Aziz Safar, and Fathinul Syahir Ahmad Saad. Performance analysis of image thresholding: Otsu technique. *Measurement*, 114:298–307, 2018.

[48] Dazhou Guo, Yanting Pei, Kang Zheng, Hongkai Yu, Yuhang Lu, and Song Wang. Degraded image semantic segmentation with dense-gram networks. *IEEE Transactions on Image Processing*, 29:782–795, 2019.

[49] Shijie Hao, Yuan Zhou, and Yanrong Guo. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406:302–321, 2020.

[50] Abdul Mueed Hafiz and Ghulam Mohiuddin Bhat. A survey on instance segmentation: state of the art. *International journal of multimedia information retrieval*, 9(3):171–189, 2020.

[51] Alexander Kirillov, Kaiming He, Ross B. Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. *CoRR*, abs/1801.00868, 2018.

[52] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[53] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[54] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[55] Salman H. Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *CoRR*, abs/2101.01169, 2021.

[56] Xiaofeng Zhu, Heung-Il Suk, Seong-Whan Lee, and Dinggang Shen. Subspace regularized sparse multitask learning for multiclass neurodegenerative disease identification. *IEEE Transactions on Biomedical Engineering*, 63(3):607–618, 2015.

[57] Xiaofeng Zhu, Heung-Il Suk, and Dinggang Shen. A novel matrix-similarity based loss function for joint regression and classification in ad diagnosis. *NeuroImage*, 100:91–105, 2014.

[58] Fabian Flohr, Dariu Gavrila, et al. Pedcut: an iterative framework for pedestrian segmentation combining shape models and multiple data cues. In *BMVC*, 2013.

[59] Garrick Brazil, Xi Yin, and Xiaoming Liu. Illuminating pedestrians via simultaneous detection & segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 4950–4959, 2017.

[60] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 111(3):257–276, 2023.

[61] Machine Mind Matters. Understanding object detection in ai. `https://machinemindmatters.com/posts/understanding-object-detection-in-ai/`, 2024.

[62] Augmented Startups. How to implement object detection using deep learning: A step-by-step guide. `https://www.augmentedstartups.com/blog/how-to-implement-object-detection-using-deep-learning-a-step-by-step-guide` 2024.

[63] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.

[64] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015.

[65] Zhipeng Deng, Hao Sun, Shilin Zhou, Juanping Zhao, and Huanxin Zou. Toward fast and accurate vehicle detection in aerial images using coupled region-based convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(8):3652–3664, 2017.

[66] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[67] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia computer science*, 199:1066–1073, 2022.

[68] Tausif Diwan, G Anirudh, and Jitendra V Tembhurne. Object detection using yolo: Challenges, architectural successors, datasets and applications. *multimedia Tools and Applications*, 82(6):9243–9275, 2023.

[69] Junwei Han, Dingwen Zhang, Gong Cheng, Nian Liu, and Dong Xu. Advanced deep-learning techniques for salient and category-specific object detection: a survey. *IEEE Signal Processing Magazine*, 35(1):84–100, 2018.

[70] Ali Borji and Laurent Itti. State-of-the-art in visual attention modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):185–207, 2012.

[71] Ashish Kumar Gupta, Ayan Seal, Mukesh Prasad, and Pritee Khanna. Salient object detection techniques in computer vision—a survey. *Entropy*, 22(10):1174, 2020.

[72] Cong Ma, Zhenjiang Miao, Xiao-Ping Zhang, and Min Li. A saliency prior context model for real-time object tracking. *IEEE Transactions on Multimedia*, 19(11):2415–2424, 2017.

[73] Hyemin Lee and Daijin Kim. Salient region-based online object tracking. In *2018 IEEE Winter conference on applications of computer vision (WACV)*, pages 1170–1177. IEEE, 2018.

[74] Huazhu Fu, Dong Xu, and Stephen Lin. Object-based multiple foreground segmentation in rgbd video. *IEEE Transactions on Image Processing*, 26(3):1418–1427, 2017.

[75] Yanwei Pang, Li Ye, Xuelong Li, and Jing Pan. Incremental learning with saliency map for moving object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(3):640–651, 2016.

[76] Xiang Wang, Shaodi You, Xi Li, and Huimin Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[77] Muath AlShaikh, Lamri Laouamer, Abdelkader Laouid, Ahcene Bounceur, and Mohammad Hammoudeh. Robust and imperceptible medical image watermarking based on dijkstra algorithm. In *Proceedings of the 3rd International Conference on Future Networks and Distributed Systems*, pages 1–6, 2019.

[78] Ahcène Bounceur, Madani Bezoui, Loïc Lagadec, Reinhardt Euler, Abdelkader Laouid, Mahamadou Traore, and Mounir Lallali. Detecting gaps and voids in wsns and iot networks: the minimum x-coordinate based method. In *Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, pages 1–7, 2018.

[79] P Ganesan and G Sajiv. A comprehensive study of edge detection for image processing applications. In *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pages 1–6. IEEE, 2017.

[80] Mohammad Hammoudeh, Robert Newman, Christopher Dennett, Sarah Mount, and Omar Aldabbas. Map as a service: A framework for visualising and maximising information return from multi-modalwireless sensor networks. *Sensors*, 15(9):22970–23003, 2015.

[81] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip HS Torr. Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3203–3212, 2017.

[82] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2083–2090, 2013.

[83] Junwei Han, Gong Cheng, Zhenpeng Li, and Dingwen Zhang. A unified metric learning-based framework for co-saliency detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2473–2483, 2017.

[84] Wangjiang Zhu, Shuang Liang, Yichen Wei, and Jian Sun. Saliency optimization from robust background detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[85] Wenguan Wang, Qiuxia Lai, Huazhu Fu, Jianbing Shen, and Haibin Ling. Salient object detection in the deep learning era: An in-depth survey. *arXiv preprint arXiv:1904.09146*, 2019.

[86] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *Computational Visual Media*, pages 1–34, 2014.

[87] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 478–487, 2016.

[88] Jason Kuen, Zhenhua Wang, and Gang Wang. Recurrent attentional networks for saliency detection. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition (CVPR)*, pages 3668–3677, 2016.

[89] Srinivas SS Kruthiventi, Vennela Gudisa, Jaley H Dholakiya, and R Venkatesh Babu. Saliency unified: A deep architecture for simultaneous eye fixation prediction and salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5781–5790, 2016.

[90] Jing Zhang, Yuchao Dai, and Fatih Porikli. Deep salient object detection by integrating multi-level cues. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–10. IEEE, 2017.

[91] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1395–1403, 2015.

[92] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 202–211, 2017.

[93] Xiaowei Hu, Lei Zhu, Jing Qin, Chi-Wing Fu, and Pheng-Ann Heng. Recurrently aggregating deep features for salient object detection. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[94] Xin Li, Fan Yang, Hong Cheng, Wei Liu, and Dinggang Shen. Contour knowledge transfer for salient object detection. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[95] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3089–3098, 2018.

[96] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. *arXiv preprint arXiv:1904.09569*, 2019.

[97] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *The*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[98] Yuzhu Ji, Haijun Zhang, Kuo-Kun Tseng, Tommy WS Chow, and QM Jonathan Wu. Graph model-based salient object detection using objectness and multiple saliency cues. *Neurocomputing*, 323:188–202, 2019.

[99] Xuanyang Xi, Yongkang Luo, Peng Wang, and Hong Qiao. Salient object detection based on an efficient end-to-end saliency regression network. *Neurocomputing*, 323:265–276, 2019.

[100] Mohammad Hammoudeh and Robert Newman. Information extraction from sensor networks using the watershed transform algorithm. *Information Fusion*, 22:39–49, 2015.

[101] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence*, 33(2):353–367, 2010.

[102] Mohd Anul Haq. Planetscope nanosatellites image classification using machine learning. *Computer Systems Science & Engineering*, 42(3), 2022.

[103] Mohd Anul Haq. Cnn based automated weed detection system using uav imagery. *Computer Systems Science & Engineering*, 42(2), 2022.

[104] Seyedeh Newsha Estiri, Amir Hossein Jalilvand, Samaneh Naderi, M Hassan Najafi, and Mahdi Fazeli. A low-cost stochastic computing-based fuzzy filtering for image noise reduction. In *2022 IEEE 13th International Green and Sustainable Computing Conference (IGSC)*, pages 1–6. IEEE, 2022.

[105] Xu-Cheng Yin, Xuwang Yin, Kaizhu Huang, and Hong-Wei Hao. Robust text detection in natural scene images. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):970–983, 2013.

[106] Jerod J Weinman, Erik Learned-Miller, and Allen R Hanson. Scene text recognition using similarity and a lexicon with sparse belief propagation. *IEEE Transactions on pattern analysis and machine intelligence*, 31(10):1733–1746, 2009.

[107] Sezer Karaoglu, Ran Tao, Theo Gevers, and Arnold WM Smeulders. Words matter: Scene text for image classification and retrieval. *IEEE transactions on multimedia*, 19(5):1063–1076, 2016.

[108] Mostefa Kara, Abdelkader Laouid, and Mohammad Hammoudeh. An efficient multi-signature scheme for blockchain. *Cryptology ePrint Archive*, 2023.

[109] Mostefa Kara, Abdelkader Laouid, Mohammad Hammoudeh, Ahcène Bounceur, et al. One digit checksum for data integrity verification of cloud-executed homomorphic encryption operations. *Cryptology ePrint Archive*, 2023.

[110] Alexander Theus, Luca Rossetto, and Abraham Bernstein. Hytext–a scene-text extraction method for video retrieval. In *International Conference on Multimedia Modeling*, pages 182–193. Springer, 2022.

[111] Fatemeh Naiemi, Vahid Ghods, and Hassan Khalesi. Scene text detection and recognition: a survey. *Multimedia Tools and Applications*, 81(14):20255–20290, 2022.

[112] Lingqian Yang, Daji Ergu, Ying Cai, Fangyao Liu, and Bo Ma. A review of natural scene text detection methods. *Procedia Computer Science*, 199:1458–1465, 2022.

[113] Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. Towards end-to-end unified scene text detection and layout analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1049–1059, 2022.

[114] Qixiang Ye and David Doermann. Text detection and recognition in imagery: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 37(7):1480–1500, 2014.

[115] Salem Sayahi and Mohamed Ben Halima. An intelligent and robust multi-oriented image scene text detection. In *2014 6th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pages 418–422. IEEE, 2014.

[116] Seiichi Uchida. 25–text localization and recognition in images and video. *Handbook of document image processing and recognition*, pages 843–883, 2014.

[117] Hongyuan Yu, Yan Huang, Lihong Pi, Chengquan Zhang, Xuan Li, and Liang Wang. End-to-end video text detection with online tracking. *Pattern Recognition*, 113:107791, 2021.

[118] Shivani Surana, Komal Pathak, Mehul Gagnani, Vidhan Shrivastava, TR Mahesh, et al. Text extraction and detection from images using machine learning techniques: A research review. In *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, pages 1201–1207. IEEE, 2022.

[119] Sandeep Dwarkanath Pande, Pramod Pandurang Jadhav, Rahul Joshi, Amol Dattatray Sawant, Vaibhav Muddebihalkar, Suresh Rathod, Madhuri Navnath Gurav, and Soumitra Das. Digitization of handwritten devanagari text using cnn transfer learning–a better customer service support. *Neuroscience Informatics*, 2(3):100016, 2022.

[120] Khalil Boukthir, Abdulrahman M Qahtani, Omar Almutiry, Habib Dhahri, and Adel M Alimi. Reduced annotation based on deep active learning for arabic text detection in natural scene images. *Pattern Recognition Letters*, 157:42–48, 2022.

[121] Pratik Madhukar Manwatkar and Shashank H Yadav. Text recognition from images. In *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pages 1–6. IEEE, 2015.

[122] Peyman Alipour and Sina E Charandabi. Analyzing the interaction between tweet sentiments and price volatility of cryptocurrencies. *European Journal of Business and Management Research*, 8(2):211–215, 2023.

[123] Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. *International Journal of Computer Vision*, 129(1):161–184, 2021.

[124] Weilin Huang, Yu Qiao, and Xiaoou Tang. Robust scene text detection with convolution neural network induced mser trees. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 497–511. Springer, 2014.

[125] Shangxuan Tian, Yifeng Pan, Chang Huang, Shijian Lu, Kai Yu, and Chew Lim Tan. Text flow: A unified text detection system in natural scene images. In *Proceedings of the IEEE international conference on computer vision*, pages 4651–4659, 2015.

[126] Cong Yao, Xiang Bai, Nong Sang, Xinyu Zhou, Shuchang Zhou, and Zhimin Cao. Scene text detection via holistic, multi-channel prediction. *arXiv preprint arXiv:1606.09002*, 2016.

[127] Lei Sun, Qiang Huo, Wei Jia, and Kai Chen. A robust approach for text detection from natural scene images. *Pattern Recognition*, 48(9):2906–2920, 2015.

[128] Ebin Zacharias, Martin Teuchler, and Bénédicte Bernier. Image processing based scene-text detection and recognition with tesseract. *arXiv preprint arXiv:2004.08079*, 2020.

[129] Chenggang Yan, Hongtao Xie, Shun Liu, Jian Yin, Yongdong Zhang, and Qionghai Dai. Effective uyghur language text detection in complex background images for traffic prompt identification. *IEEE transactions on intelligent transportation systems*, 19(1):220–229, 2017.

[130] Shehzad Muhammad Hanif and Lionel Prevost. Text detection and localization in complex scene images using constrained adaboost algorithm. In *2009 10th international conference on document analysis and recognition*, pages 1–5. IEEE, 2009.

[131] Alvaro Gonzalez, Luis M Bergasa, and J Javier Yebes. Text detection and recognition on traffic panels from street-level imagery using visual appearance. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):228–238, 2013.

[132] Syed Yasser Arafat and Muhammad Javed Iqbal. Urdu-text detection and recognition in natural scene images using deep learning. *IEEE Access*, 8:96787–96803, 2020.

[133] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017.

[134] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11474–11481, 2020.

[135] Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao, and Xiang Bai. Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):919–931, 2022.

[136] Shi-Xue Zhang, Xiaobin Zhu, Jie-Bo Hou, Chang Liu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. Deep relational reasoning graph network for arbitrary shape text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9699–9708, 2020.

[137] Yiqin Zhu, Jianyong Chen, Lingyu Liang, Zhanghui Kuang, Lianwen Jin, and Wayne Zhang. Fourier contour embedding for arbitrary-shaped text detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3123–3131, 2021.

[138] Zhida Huang, Zhuoyao Zhong, Lei Sun, and Qiang Huo. Mask r-cnn with pyramid attention network for scene text detection. In *2019 IEEE Winter*

Conference on Applications of Computer Vision (WACV), pages 764–772. IEEE, 2019.

[139] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, and Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8440–8449, 2019.

[140] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9336–9345, 2019.

[141] Xiaoxue Chen, Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, and Canjie Luo. Adaptive embedding gate for attention-based scene text recognition. *Neurocomputing*, 381:261–271, 2020.

[142] Jianming Zhang, Xin Zou, Li-Dan Kuang, Jin Wang, R Simon Sherratt, and Xioafeng Yu. Cctsdb 2021: a more comprehensive traffic sign detection benchmark. *Human-centric Computing and Information Sciences*, 12, 2022.

[143] Weiyu Wang and Keng Siau. Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: A review and research agenda. *Journal of Database Management (JDM)*, 30(1):61–79, 2019.

[144] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[145] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

[146] Kristina P Sinaga and Miin-Shen Yang. Unsupervised k-means clustering algorithm. *IEEE access*, 8:80716–80727, 2020.

[147] Artúr István Károly, Róbert Fullér, and Péter Galambos. Unsupervised clustering for deep learning: A tutorial survey. *Acta Polytechnica Hungarica*, 15(8):29–53, 2018.

[148] Fatemeh Naiemi, Vahid Ghods, and Hassan Khalesi. A novel pipeline framework for multi oriented scene text image detection and recognition. *Expert Systems with Applications*, 170:114549, 2021.

[149] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE, 2015.

[150] Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 935–942. IEEE, 2017.

[151] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1083–1090. IEEE, 2012.

[152] Cong Yao, Xiang Bai, and Wenyu Liu. A unified framework for multioriented text detection and recognition. *IEEE Transactions on Image Processing*, 23(11):4737–4749, 2014.

[153] Matthew Fleming, Sreelakshmi Ravula, Sergei F Tatishchev, and Hanlin L Wang. Colorectal carcinoma: Pathologic aspects. *Journal of gastrointestinal oncology*, 3(3):153, 2012.

[154] Mikala Egeblad, Elizabeth S Nakasone, and Zena Werb. Tumors as organs: complex tissues that interface with the entire organism. *Developmental cell*, 18(6):884–901, 2010.

[155] Mingchao Du, Min Tao, Jian Hong, Dian Zhou, and Shuihua Wang. Application of deep learning algorithm in feature mining and rapid identification of colorectal image. *IEEE Access*, 8:128830–128844, 2020.

[156] Mostefa Kara, Konstantinos Karampidis, Zaoui Sayah, Abdelkader Laouid, Giorgos Papadourakis, and Mohammad Nadir Abid. A password-based mutual authentication protocol via zero-knowledge proof solution. In *International Conference on Applied CyberSecurity*, pages 31–40. Springer, 2023.

[157] Farid Lalem, Abdelkader Laouid, Mostefa Kara, Mohammed Al-Khalidi, and Amna Eleyan. A novel digital signature scheme for advanced asymmetric encryption techniques. *Applied Sciences*, 13(8):5172, 2023.

[158] Saci Medileh, Abdelkader Laouid, Mohammad Hammoudeh, Mostefa Kara, Tarek Bejaoui, Amna Eleyan, and Mohammed Al-Khalidi. A multi-key with partially homomorphic encryption scheme for low-end devices ensuring data integrity. *Information*, 14(5):263, 2023.

[159] Mostefa Kara, Konstantinos Karampidis, Giorgos Papadourakis, Abdelkader Laouid, and Muath AlShaikh. A probabilistic public-key encryption with ensuring data integrity in cloud computing. In *2023 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO)*, pages 59–66. IEEE, 2023.

[160] Jakob Nikolas Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne M Melchers, Lothar R Schad, Timo Gaiser, Alexander Marx, and Frank Gerrit Zöllner. Multi-class texture analysis in colorectal cancer histology. *Scientific reports*, 6(1):27988, 2016.

[161] Srinath Jayachandran and Ashlin Ghosh. Deep transfer learning for texture classification in colorectal cancer histology. In *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, pages 173–186. Springer, 2020.

[162] World Health Organization. Colorectal cancer fact sheet. Accessed: 18/10/2023.

[163] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017.

[164] Geert Litjens, Clara I Sánchez, Nadya Timofeeva, Meyke Hermsen, Iris Nagtegaal, Iringo Kovacs, Christina Hulsbergen-Van De Kaa, Peter Bult, Bram Van Ginneken, and Jeroen Van Der Laak. Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. *Scientific reports*, 6(1):26286, 2016.

[165] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[166] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[167] Jakob Nikolas Kather, Frank Gerrit Zöllner, Francesco Bianconi, Susanne M Melchers, Lothar R Schad, Timo Gaiser, Alexander Marx, and Cleo-Aron Weis. Collection of textures in colorectal cancer histology, June 2016.