
Polytechnic of Leiria
School of Technology and Management
Department of Electrical and Electronics Engineering
Master in Electrical and Electronic Engineering - Electronics and Telecommunications

# ROI-BASED CODING OF BIOMEDICAL IMAGES FOR MACHINE ANALYSIS

DANIEL FILIPE DA SILVA NICOLAU

Leiria, March 2025


Polytechnic of Leiria
School of Technology and Management
Department of Electrical and Electronics Engineering
Master in Electrical and Electronic Engineering - Electronics and Telecommunications

# ROI-BASED CODING OF BIOMEDICAL IMAGES FOR MACHINE ANALYSIS

DANIEL FILIPE DA SILVA NICOLAU
Number: 2232623

Dissertation supervised by Professor Sérgio M. M. Faria (sergio.faria@ipleiria.pt), Professor Lucas A. Thomaz (lucas.thomaz@ipleiria.pt), and Professor Luís M. N. Távora (luis.tavora@ipleiria.pt).

Leiria, March 2025

# ACKNOWLEDGMENTS

I would like to start by thanking my supervisors, Prof. Lucas Arrabal Thomaz, Prof. Luís Miguel de Oliveira Pegado de Noronha e Távora, and Prof. Sérgio Manuel Maciel de Faria. Their unwavering guidance, availability, encouragement, and expertise have been fundamental throughout this journey. I am deeply thankful not only for their trust but also for their exemplary dedication and leadership, which continue to greatly inspire me.

I would like to extend my thanks to the host institution, Instituto de Telecomunicações, for providing excellent research conditions and computational resources, as well as to the School of Technology and Management of the Polytechnic of Leiria for their facilities which were crucial to the success of this research. Additionally, I acknowledge the financial support provided by the Fundação para a Ciência e a Tecnologia (FCT), Portugal under projects CoMBINNe 2022.09914.PTDC (DOI:10.54499/2022.09914.PTDC), Programa Operacional Regional do Centro, and by FCT/MCTES through national funds and when applicable co-funded by EU funds under the project UIDB/EEA/50008/2020 (DOI: 10.54499/UIDB/50008/2020) and LA/P/0109/2020 (DOI: 10.54499/LA/P/0109/2020).

Beyond infrastructure and resources, a research journey is shaped by the people who share it. Over my final year of my bachelor's and throughout my master's, I had the privilege of meeting remarkable colleagues and friends at the Multimedia Signal Processing group from the Leiria Delegation of Instituto de Telecomunicações: José Filipe, Rui Lourenço, João Parracho, José Rosa, Edgar Paulo, Rúben Susano, Nicolas Vasconcellos, Rúben Francisco, and Nuno Fernandes. Their companionship, knowledge exchange, work debating, and readiness to help, whether with technical challenges or moments of doubt, made this journey significantly more enriching and enjoyable.

My deepest gratitude goes to my family - my parents, Rita Silva and João Nicolau, my brother, Leonardo Nicolau, and my grandparents, Helena, Amadeu, José, and Fátima. Their constant love, support, and encouragement have been an essential pillar, not only during this work, but also in my life. I only hope to reciprocate the affection and patience and to continue making them proud. To my girlfriend, Rahma Azzalia, whose support, patience, and companionship during long working hours have been invaluable and I am profoundly grateful. Her kindness and encouragement have been a source of strength throughout this process. Terima kasih banyak, sayangku.

Finally, I extend my heartfelt thanks to all those who, in one way or another, have contributed to this journey. A special mention to my friends Francisco Ferreira, Pedro Gaspar, Ana Neto, and Sofia Balau, whose friendship, generosity, and support have had a lasting impact. Their presence has undoubtedly shaped the path that led me here.

With deep gratitude,
**Thank You!**

# RESUMO

O aumento do volume de dados adquiridos e gerados diariamente na área da saúde, impulsionado pelos avanços tecnológicos, traz benefícios significativos para o diagnóstico de pacientes e para a investigação. No entanto, esta evolução também representa desafios consideráveis na análise e processamento desses dados. Para enfrentar estas dificuldades, os algoritmos de visão computacional surgiram como ferramentas poderosas, capazes de automatizar tarefas repetitivas e demoradas, permitindo um processamento mais rápido e preciso.

Paralelamente, o crescimento do volume de dados coloca pressão sobre as capacidades de armazenamento e transmissão, exigindo métodos de compressão eficientes para minimizar o seu tamanho. Na literatura, encontram-se diversas abordagens, maioritariamente divididas em duas categorias: compressão com perdas e sem perdas. Embora os métodos sem perdas garantam a integridade dos dados, não atingem taxas de compressão tão elevadas como os algoritmos com perdas. Estes últimos, apesar de proporcionarem uma redução significativa do tamanho dos ficheiros, introduzem distorções que podem comprometer a qualidade das imagens, afetando a precisão de sistemas automatizados.

Esta dissertação foca-se em dois desafios principais: primeiro, avaliar o impacto da compressão de imagens no desempenho de sistemas de visão computacional biomédicos e, segundo, aumentar a eficiência da compressão sem prejudicar a precisão destes algoritmos. Para tal, foram utilizados modelos de deteção e segmentação, como o YOLOv8 e o SAM, para analisar o efeito da distorção causada pela codificação na localização e segmentação de mitocôndrias em dois conjuntos de imagens de microscopia eletrónica.

Com o objetivo de melhorar o desempenho dos modelos face a níveis de compressão mais elevados, duas metodologias foram implementadas. A primeira centra-se na adaptação ao domínio, ajustando os modelos para reconhecer e compensar as distorções introduzidas pela compressão, especificamente nos codificadores HEVC/H.265 e VVC/H.266. A segunda abordagem propõe uma adaptação do codificador ao conteúdo, permitindo a atribuição de diferentes níveis de qualidade a regiões de interesse selecionadas. Este método visa reduzir os requisitos de armazenamento e largura de banda sem comprometer significativamente o desempenho dos modelos baseados em *deep learning*.

Os resultados experimentais demonstram que as estratégias de codificação baseadas em regiões de interesse reduzem eficazmente a taxa de compressão, mantendo a

precisão dos modelos. Em particular, as metodologias propostas alcançam, em média, um aumento de desempenho de até 23,70% para as mesmas taxas e uma redução do tamanho dos dados de até 74,96%,. Adicionalmente, foi proposto um algoritmo de otimização baseado na solução de Pareto, que determina as configurações de codificação mais adequadas para diferentes normas e modelos, garantindo um equilíbrio entre eficiência de compressão e desempenho na deteção de objetos.

***Palavras-chave:*** Imagiologia Biomédica, Microscopia Eletrónica, Compressão com Perdas, YOLO, SAM, HEVC, VVC, Codificação de Regiões de Interesse, Preservação de Conteúdo

# ABSTRACT

The increasing volume of data acquired and generated daily in the healthcare sector, driven by technological advancements, brings significant benefits to patient diagnosis and research. However, this growth also presents considerable challenges in the analysis and processing of such data. To address these difficulties, computer vision algorithms have emerged as powerful tools, capable of automating repetitive and time-consuming tasks, enabling faster and more accurate processing.

At the same time, the growing volume of data places pressure on storage and transmission capabilities, demanding efficient compression methods to minimise its size. In the literature, various approaches are found, primarily divided into two categories: lossy and lossless compression. While lossless methods ensure data integrity, they do not achieve compression rates as high as lossy algorithms. The latter, despite significantly reducing file sizes, introduces distortions that may compromise image quality, affecting the accuracy of automated systems.

This dissertation focuses on two main challenges: first, evaluating the impact of image compression on the performance of biomedical computer vision systems, and second, improving compression efficiency without compromising the accuracy of these algorithms. To this end, detection and segmentation models, such as YOLOv8 and SAM, were used to analyse the effect of distortion caused by encoding on the localisation and segmentation of mitochondria in two datasets of electron microscopy images.

To enhance model performance at higher compression levels, two methodologies were implemented. The first focuses on domain adaptation, fine-tuning the models to recognise and compensate for distortions introduced by compression, specifically in HEVC/H.265 and VVC/H.266 encoders. The second approach proposes content-aware encoder adaptation, allowing the assignment of different quality levels to selected regions of interest. This method aims to reduce storage and bandwidth requirements without significantly compromising the performance of deep learning-based models.

Experimental results demonstrate that region-of-interest-based encoding strategies effectively reduce compression rates while maintaining model accuracy. In particular, the proposed methodologies allowed to achieve an average performance improvement of up to 23.70% for the same bpp range and a data size reduction of up to 74.96%. Additionally, a Pareto-based optimisation algorithm was proposed to determine the

most suitable encoding configurations for different standards and models, ensuring a balance between compression efficiency and object detection performance.

***Keywords:*** Biomedical Imaging, Electron Microscopy, Lossy Compression, YOLO, SAM, HEVC, VVC, Region-of-Interest Coding, Content Preservation

CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ACRONYMS

| | |
|---|---|
| AVC | Advanced Video Coding. |
| BD | Bjøntegaard Delta. |
| bpp | bits per pixel. |
| CIoU | Complete Intersection over Union. |
| CNN | Convolutional Neural Network. |
| CT | Computed Tomography. |
| CTB | Coding Tree Block. |
| CTU | Coding Tree Unit. |
| CU | Coding Unit. |
| DCT | Discrete Cosine Transform. |
| DICOM | Digital Imaging and Communications in Medicine. |
| DNA | Deoxyribonucleic Acid. |
| DSC | Dice-Sørensen coefficient. |
| EM | Electron Microscopy. |
| FIB-SEM | Focused Ion Beam Scanning Electron Microscopy. |
| FN | False Negative. |
| FP | False Positive. |
| FPN | Feature Pyramid Network. |
| GAN | Generative Adversarial Network. |
| GPU | Graphics Processing Unit. |
| HEVC | High Efficiency Video Coding. |

| | |
|---|---|
| HM | HEVC Test Model. |
| ID | Identification. |
| IEC | International Electrotechnical Commission. |
| IoU | Intersection over Union. |
| ISO | International Organization for Standardization. |
| ITU-T | International Telecommunication Union Telecommunication Standardization Sector. |
| JCT-VC | Joint Collaborative Team on Video Coding. |
| JPEG | Joint Photographic Experts Group. |
| JVET | Joint Video Experts Team. |
| MedSAM | Medical Segment Anything Model. |
| MPEG | Moving Picture Experts Group. |
| MRI | Magnetic Resonance Imaging. |
| MSE | Mean Squared Error. |
| MTT | Multi-Type Tree. |
| PCHIP | Piecewise Cubic Hermite Interpolation. |
| PET | Positron Emission Tomography. |
| pp | percentage point. |
| PSNR | Peak Signal-to-Noise Ratio. |
| PU | Prediction Unit. |
| QG | Quantisation Group. |
| QP | Quantisation Parameter. |
| R-CNN | Region Based Convolutional Neural Networks. |
| RD | Rate Distortion. |
| ROI | Region of Interest. |
| SAM | Segment Anything Model. |

SEM Scanning Electron Microscopy.

SSE Sum of Squared Errors.

ssEM serial section Electron Microscopy.

STEM Scanning Transmission Electron Microscopy.

TEM Transmission Electron Microscopy.

TN True Negative.

TP True Positive.

TU Transform Unit.

UHD Ultra-High-Definition.

VCEG Video Coding Experts Group.

ViT Vision Transformer.

VTM VVC Test Model.

VVC Versatile Video Coding.

YOLO You Only Look Once.

# 1

# INTRODUCTION

## 1.1 CONTEXT AND MOTIVATION

In clinical and research environments, machine learning models play a crucial role in automating repetitive and time-consuming tasks, enabling faster and more accurate analysis of biomedical data [1], [2]. Advances in biomedical imaging technology, such as higher spatial resolutions and the acquisition of volumetric data, have significantly enhanced the quality and detail of information available for medical diagnosis and research [3]. However, the growing volume and complexity of this type of data create significant challenges in storage, management, and transmission [4], placing a considerable burden on medical infrastructures [5].

Nowadays, it is expected that the healthcare industry produces approximately 30% of the global data [6]. Additionally, compliance with legal terms often requires the retention of data for long periods of time, increasing the scarcity of computational and storage resources. For instance, in Spain, medical data must be retained for at least five years after a patient's discharge, although this period can extend depending on the legislation of autonomous regions [7], further intensifying the demand for efficient storage solutions. Furthermore, the transmission of large medical datasets over networks, particularly in telemedicine and collaborative research, requires strategies to balance bandwidth constraints and image quality [8], [9].

To address these challenges, image compression techniques have been widely adopted to reduce storage requirements and improve data transmission [10]. Compression methods can be broadly classified into two categories: lossless and lossy. Lossless compression ensures the exact reconstruction of the original image, thus preserving all image details. However, it typically achieves limited compression ratios, which may not be sufficient to alleviate the constraints mentioned previously. In contrast, lossy compression achieves significantly higher compression ratios by discarding less perceptible image information [11]. While this approach effectively reduces file sizes, it introduces distortions that can degrade the accuracy of clinicians and automated analysis performed by machine learning models [12].

In biomedical imaging, selecting an appropriate compression method requires a careful balance between reducing storage costs and maintaining the integrity of diagnostically relevant features [13]. Machine learning models, particularly those

used for segmentation and detection tasks, rely on fine image details to achieve accurate predictions [14]. The introduction of compression artefacts may lead to misinterpretations, reducing the effectiveness of these models [15]. Therefore, understanding the trade-offs between compression efficiency and model performance is essential for developing optimised compression strategies tailored to biomedical applications.

## 1.2 OBJECTIVES

This dissertation aims to evaluate the impact of lossy coding on biomedical images within computer vision systems and to develop approaches that enhance their performance. To achieve these objective, the following tasks were carried out:

- Investigation of the performance of detection and segmentation models (both trained from scratch and fine-tuned) on compressed biomedical images, on the identification of mitochondria within Electron Microscopy (EM) images.
- Adaptation of a state-of-the-art Region of Interest (ROI)-based coding algorithm for High Efficiency Video Coding (HEVC)/H.265 to support non-rectangular ROI coding.
- Development of an ROI-based coding solution for the Versatile Video Coding (VVC)/H.266 standard.
- Comprehensive evaluation of the performance of the proposed methods.

## 1.3 DISSERTATION OUTLINE

The remainder of this document is organised as follows: Chapter 2 provides an overview of biomedical imaging, with a particular focus on EM imaging. It also discusses state-of-the-art standard encoders, ROI-based coding techniques, and computer vision tasks in biomedical imaging, emphasising detection and segmentation. Chapter 3 details the methodologies and proposed frameworks for ROI-based coding using standard encoders. Chapter 4 describes the experimental setup and presents an analysis of the results. Finally, in Chapter 5 concluding remarks and insights are presented.

# 2

# BACKGROUND

This chapter presents the background and state-of-the-art relevant to this work. The first section provides a brief overview of biomedical imaging, with a particular emphasis on EM imaging and the datasets used. Subsequently, the chapter explores key computer vision tasks in biomedical imaging, with a focus on detection and segmentation. Finally, the codecs employed in the research, as well as state-of-the-art techniques for ROI-based coding, are described and discussed.

## 2.1 BIOMEDICAL IMAGING

Biomedical imaging is fundamental in modern medicine and biological research [16], [17], providing detailed visualizations of anatomical structures [18], physiological processes [19], and pathological conditions across multiple scales [20]. By capturing images of organs, tissues, cells, and even molecules, as well as metabolic activity, these techniques offer critical insights into the complexity of biological systems [20], [21]. From disease diagnosis to surgical guidance and the study of cellular mechanisms, biomedical imaging has significantly advanced healthcare, therapy, and scientific discovery [22].

This field encompasses a wide range of imaging modalities, each optimized for specific applications and levels of detail. Techniques such as Magnetic Resonance Imaging (MRI) [23] and Computed Tomography (CT) [24] are widely used in clinical practice for (nearly) non-invasive visualization of structural information, such as internal organs and tissues. These modalities provide macroscopic views at the centimetre and millimetre scale, making them essential for diagnosing and monitoring conditions such as tumours, cardiovascular diseases, and neurological disorders. Alternatively, imaging technologies like Positron Emission Tomography (PET) [25] enable the analysis of functional information, such as metabolic activity and biochemical processes, offering complementary insights to structural imaging [26]. At a finer scale, optical microscopy techniques [27] are indispensable for studying cellular and subcellular structures. These methods produce high-resolution images of living cells and tissues, achieving sub-millimetre resolution.

Recent advances in hardware, software, and computational methods continue to drive innovation in biomedical imaging. Techniques such as super-resolution

microscopy [28] push the boundaries of resolution and contrast, enabling researchers to explore biological processes with unprecedented clarity. Additionally, the integration of artificial intelligence and machine learning is transforming image analysis, enabling automated detection and segmentation of complex structures [29], thus supporting and pushing forward both research and clinical applications.

### 2.1.1 *Electron Microscopy Images*

Several technological fields require the ability to visualise structures at extremely small scales. For instance, advancements in semiconductor research and nanotechnology depend on analysing integrated circuits to optimise fabrication processes [30]. In biological studies, nanoscale visualisation allows the advancement of the understanding of cellular processes and the inner workings of biological systems [31], [32]. To tackle the demands of these fields, researchers leverage, among others, the power of EM.

EM is an imaging technique that uses beams of electrons instead of light to create images of specimens at resolutions far beyond what conventional light microscopy can achieve. Modern electron microscopes can achieve resolutions down to 0.1nm [33]. This exceptional imaging capability allows researchers to examine structures at near-atomic resolution, providing unprecedented insights into the cellular architecture and material properties [34], [35].

The technology encompasses three primary types of electron microscopes, each distinguished by its method of electron collection, as illustrated in Figure 1. Scanning Electron Microscopy (SEM) probes the surface of specimens with a focused beam of electrons, detecting the reflected ones. In contrast, in Transmission Electron Microscopy (TEM), a beam of electrons is transmitted through an ultra-thin specimen, creating a detailed image of the internal structure as electrons interact with and pass through the sample. Finally, Scanning Transmission Electron Microscopy (STEM) combines features of both SEM and TEM, providing high-resolution imaging of thin samples, similar to TEM, but with data acquired sequentially, enabling the creation of 3D reconstructions.

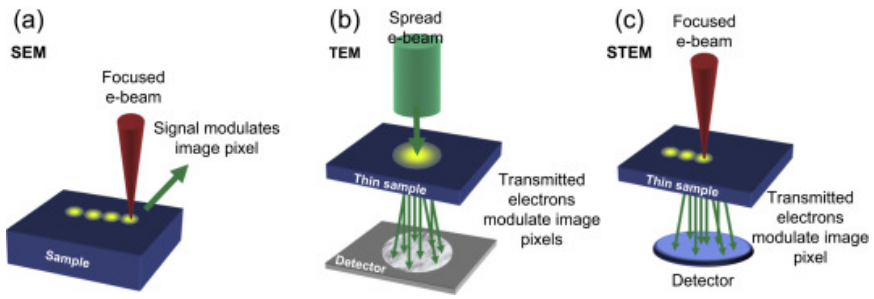



Figure 1: Diagram of SEM, TEM, and STEM acquisition techniques [36].

Automated detection and analysis of cellular structures in EM has become increasingly crucial for modern biomedical and neurobiological research. Of particular importance is the identification of mitochondria, cellular organelles that play a vital role in energy production and cellular homeostasis. Inherited mitochondrial dysfunction, resulting from deficiencies in mitochondrial Deoxyribonucleic Acid (DNA) maintenance and translation, is a key determinant in several neurological disorders, such as autism, Leigh syndrome, and stroke-like episodes, and it is also associated with a variety of other systemic diseases such as cancer, heart failure, and diabetes [37], [38]. Additionally, studies also suggest that mitochondria may occupy twice as much volume in inhibitory dendrites and axons compared to excitatory ones [39]. To maintain cellular balance, surveillance mechanisms closely monitor mitochondrial functions, allowing to anticipate potential disruptions within these multifunctional organelles [40].

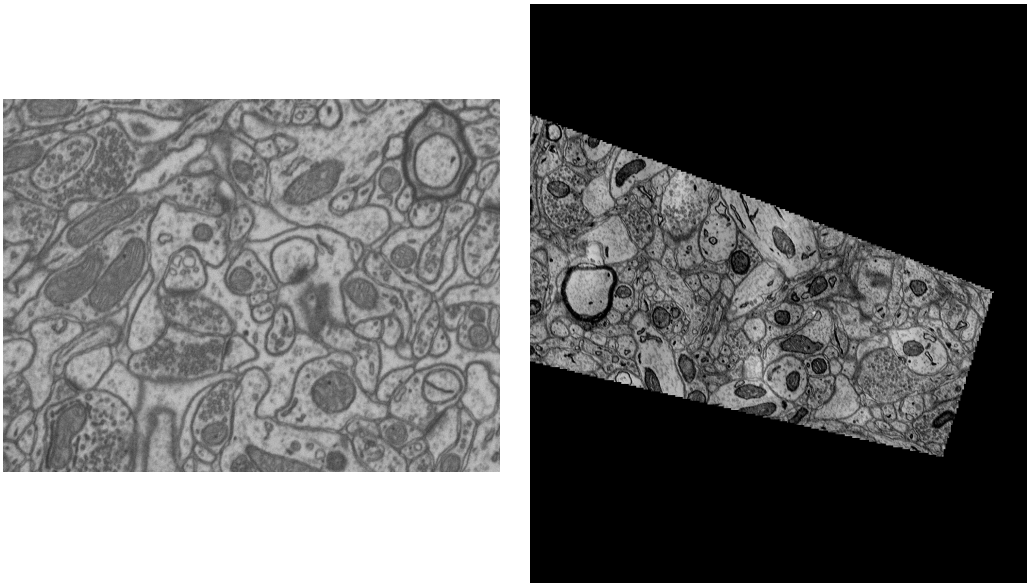
### 2.1.2 *Datasets*

In this work, two publicly available EM datasets are used, denoted as Lucchi++ and Kasthuri++. These datasets were initially introduced in [41] and [39], respectively. To address boundary inconsistencies and misclassifications in the original annotations, the datasets were re-annotated by experts, resulting in the refined versions described in [42]. The re-annotation process involved a senior biologist performing the initial corrections, followed by validation from two neuroscientists to ensure consensus-driven labelling.

The Lucchi++ dataset is based on Focused Ion Beam Scanning Electron Microscopy (FIB-SEM) images from a $5 \times 5 \times 5$ µm section of the mouse hippocampus, with a voxel size of $5 \times 5 \times 5$ nm. The dataset consists of two neighbouring stacks, each with dimensions of $1024 \times 768 \times 165$ voxels at 8-bit greyscale depth. The re-annotation process reduced the number of annotated mitochondria from 99 to 80, with the average 2D area of the labels increasing approximately 20.2%.

The Kasthuri++ dataset contains mitochondria annotations of a mouse cortex volume acquired using serial section Electron Microscopy (ssEM). The dataset comprises two neighbouring sub-volumes with dimensions $1463 \times 1613 \times 85$ voxels and $1334 \times 1553 \times 75$ voxels, also at 8-bit greyscale depth, with a voxel size of $3 \times 3 \times 3$ nm. The expert re-annotation process resulted in a reduction from 242 to 208 mitochondria, with the average 2D area slightly increasing approximately 2.7%. Figure 2 presents a sample image of each dataset.

As both datasets consist of two volumes each, they are commonly used as separate training and testing sets [12], [42], [43]. However, to also include a validation set, each



(a) Lucchi++. (b) Kasthuri++.

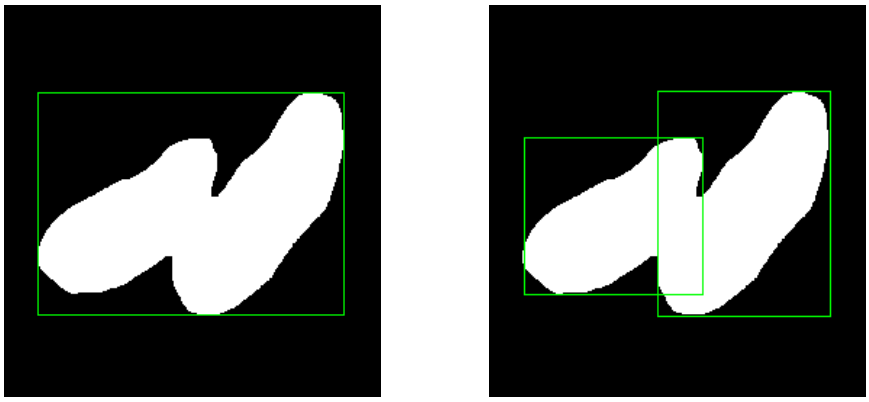
Figure 2: Sample images from the used datasets.

training dataset was subdivided into training and validation data. In Appendix A, the image Identification (ID)s for each validation and training set are defined to ensure reproducibility of the study. The resulting datasets consist of 125, 40, and 165 images for the training, validation, and testing sets, respectively, for the Lucchi++ dataset, and 61, 24, and 75 images for the respective sets of the Kasthuri++ dataset.

The refined re-annotation of mitochondria in both datasets are presented as binary masks. However, for detection tasks, bounding boxes are required for the correct training and evaluation of these models. While mitochondria location can be extracted directly from the ground truth masks, it was observed that some adjacent structures were connected by a few pixels, leading to the extraction of a single bounding box for multiple instances, as illustrated in Figure 3a. To address this issue, bounding box annotations for both datasets were refined in [44]. Initially, these were extracted from the segmentation masks, followed by manual corrections to separate connected instances. This process resulted in a more accurate delineation of individual mitochondria, as illustrated in Figure 3b. The re-annotation increased the number of annotated mitochondria by 2.9% for the Lucchi++ dataset, and 2.1% for the Kasthuri++ dataset.

## 2.2 COMPUTER VISION IN BIOMEDICAL IMAGING

As discussed in Section 2.1.1, identifying mitochondria is a critical task for neurobiological research, requiring automated detection methods capable of matching the acquisition speed of modern electron microscopes [39]. While detection provides



(a) Bounding box surrounding all connected components.

(b) Individual bounding boxes assigned to each mitochondrion.

Figure 3: Comparison of bounding box annotations derived from segmentation masks.

a fast location definition through bounding boxes, a more precise analysis of its morphology and volume demands pixel-level segmentation. In EM images, mitochondria appear as dark, predominantly elliptical structures with high membrane contrast. However, their appearance and shape can vary significantly depending on the sectioning angle and cellular context [42]. This variability, combined with the massive scale of modern EM datasets, needs robust automated approaches for both detection and segmentation tasks. Such systems offer the additional advantage of being time-invariant and immune to intra- and inter-operator variability [45].

### 2.2.1 *Detection*

Object detection in biomedical imaging represents a fundamental computer vision task that aims to locate objects of interest within medical images [46]. In the context of EM, detection typically involves identifying the presence and location of cellular structures through bounding boxes, providing a faster but less detailed alternative to full segmentation approaches [47]. Traditional detection approaches relied heavily on hand-crafted features, such as texture descriptors and intensity-based characteristics, combined with classical machine learning algorithms [48]–[50].

The advent of deep learning, particularly Convolutional Neural Network (CNN)s, has significantly advanced the field of object detection by enabling the automatic learning of hierarchical features directly from image data [51], [52]. One of the pioneering techniques in this domain is Region Based Convolutional Neural Networks (R-CNN) [53], which introduced a two-stage approach to object detection. In the first stage, the algorithm generates region proposals, which are potential areas of interest in the image, using a selective search algorithm [54]. These region proposals are then inputted to a CNN, which extracts features and classifies the objects within each region. Finally, a bounding box regressor refines the coordinates of the detected

objects. While R-CNN was a breakthrough, its computational inefficiency led to the development of faster and more efficient variants, such as Fast R-CNN [55] and Faster R-CNN [56], which streamlined the process by sharing computations across region proposals.

Among several detection models, You Only Look Once (YOLO)'s architectures [57] have exhibited strong performance in various tasks and objects of interest within medical imaging [58], including anomaly [59], [60] and lesion detection [61], [62]. Unlike two-stage detectors such as R-CNN and its variants, the YOLO framework revolutionized object detection by introducing a single-stage approach that treats detection as a direct regression problem [63]. Instead of generating region proposals, YOLO divides the image into a grid of cells, as illustrated in Figure 4. Each cell is responsible for predicting bounding boxes and class probabilities for objects within its boundaries. This design significantly improved inference speed, making YOLO well-suited for real-time applications. However, early versions of YOLO had limitations in handling small objects and overlapping instances due to their coarse grid-based predictions.

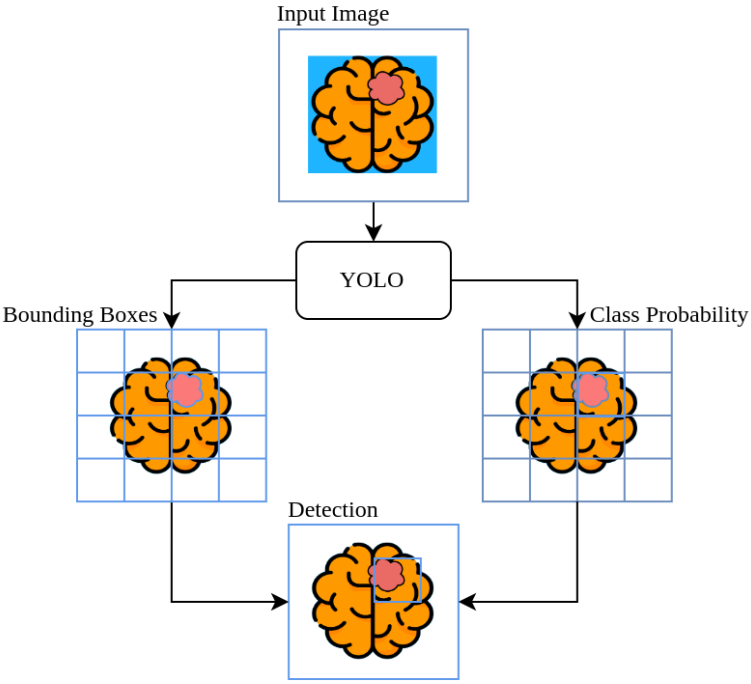



Figure 4: YOLO's detection core idea [58].

To address these limitations, the next versions of YOLO, starting with YOLOv2 [64], introduced an anchor-based approach. Anchors are pre-defined bounding boxes of various sizes and aspect ratios, which serve as templates for object detection. Instead of predicting bounding boxes directly, the model predicts offsets relative to these anchors, improving its ability to handle objects of different shapes and sizes, improving the detection of small and overlapping objects.

Compared to previous versions, YOLOv4 [65] proposed several tools and techniques to improve its performance. It introduced a variety of data augmentation methods, including photometric distortions (e.g., brightness, contrast, and noise adjustments) and geometric deformations (e.g., random scaling, cropping, and rotation), to improve the model's robustness and generalisation. Additionally, YOLOv4 refined the bounding box regression process by replacing the traditional Mean Squared Error (MSE) loss with the Complete Intersection over Union (CIoU) loss [66], which considers the overlap area, centre point distance, and aspect ratio between predicted and ground truth bounding boxes. This change improved the accuracy of bounding box predictions, particularly for objects with complex shapes or orientations.

YOLOv8 [67] also represents a significant evolution over its predecessors, particularly due to its backbone design, based on the Feature Pyramid Network (FPN) [68], and its detection head structures. The FPN enhances multi-scale feature representation by progressively reducing the spatial resolution of the input image while preserving essential details, enabling effective object detection at different scales. Additionally, YOLOv8 adopts an anchor-free detection paradigm, where the detection head directly predicts object centres and dimensions. This approach eliminates the need for predefined anchor boxes, reducing the number of hyperparameters that require manual tuning and decreasing memory requirements during training and inference. Due to its optimized performance, this model has been successfully applied to medical imaging detection tasks [69], [70]. Building on these advancements, subsequent versions of YOLO, such as YOLOv9 [71], YOLOv10 [72], and the latest, YOLOv11 [73], have introduced further architectural improvements. These version focus not only on enhancing detection accuracy, but also on optimising inference speed to achieve real-time performance in computationally demanding environments.

### 2.2.2 *Segmentation*

Object segmentation in biomedical imaging plays a crucial role in precisely delineating structures of interest at the pixel level, providing a more detailed alternative to object detection [74]. Unlike detection, which relies on bounding boxes that may include background regions, segmentation ensures that only the relevant anatomical or pathological structures are highlighted pixel-wise, improving diagnostic accuracy and enabling more precise downstream analysis.

Early segmentation approaches relied on classical image processing techniques, such as thresholding and edge detection methods [75], [76]. While effective in controlled settings, these techniques often struggled with variations in contrast,

illumination, and noise, which are common in biomedical imaging [77]. To address these limitations, machine learning-based methods, including clustering [78], [79] and graph-based approaches [54], were introduced to enhance robustness and adaptability to complex image characteristics.

The introduction of deep learning, particularly CNNs, revolutionized biomedical image segmentation. One of the most influential architectures is U-Net [80], which employs an encoder-decoder structure with skip connections to capture both low-level and high-level features. This design allows for precise segmentation, even with limited training data, making U-Net highly effective for medical imaging tasks. U-Net has been widely adopted in various applications, including cell segmentation [81]–[83], tumour delineation [84], [85], and organ segmentation [86]–[88].

YOLO architectures have also been adapted for segmentation tasks, expanding their applicability beyond object detection. One notable example is YOLOv8's segmentation variant [89], which extends the traditional YOLOv8 framework to provide pixel-wise object delineation while maintaining the model's real-time performance. This variant introduces segmentation-specific heads to the YOLOv8 architecture, enabling it to output binary masks alongside bounding boxes and class probabilities [90].

The Segment Anything Model (SAM) [91] represents a breakthrough in segmentation, providing zero-shot segmentation on a wide range of objects and domains, without the need for extensive training. SAM is built around three core components, as illustrated in Figure 5: an image encoder, a prompt encoder, and a mask decoder. The first module, the image encoder, uses a Vision Transformer (ViT)-based architecture [92] to process and transform the input images into feature embeddings. The second module, the prompt encoder, is the module that distinguishes SAM from any traditional segmentation models. This component allows SAM to take inputs from the user in the form of prompts, such as points, bounding boxes, text, or binary masks, to guide the segmentation process. This module embeds these prompts as multi-dimensional vectors into a feature space that aligns with the image data. This becomes extremely useful in medical imaging, as the user can quickly specify the position of structures, and SAM adapts the segmentation accordingly. Finally, the mask decoder uses information from both the image decoder and the embedded prompts to produce pixel-wise segmentation results. To deal with ambiguity during training, rather than presenting only one output that would average multiple valid masks, the model learns to predict multiple outputs for a single prompt. The model then learns to rank these masks, by estimating the Intersection over Union (IoU) between each prediction and the object it covers.

While SAM has demonstrated exceptional performance in general object segmentation tasks [93], its application to medical imaging presents unique challenges.

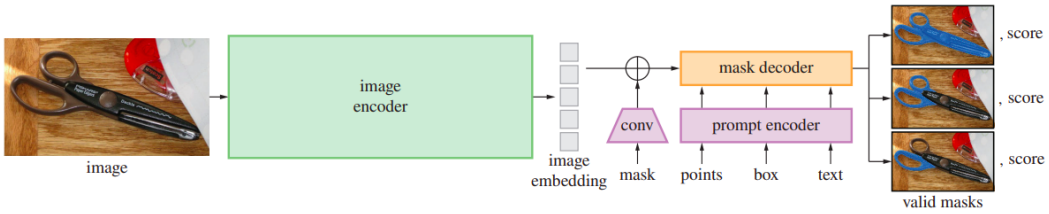



Figure 5: SAM architecture diagram [91].

Medical images often contain fine-grained, subtle distinctions that vary significantly across imaging modalities, such as CT, MRI, and ultrasound. These variations, combined with noise, low contrast, and complex anatomical structures, make medical image segmentation a particularly challenging domain for zero-shot models like SAM [14].

To address these challenges, Medical Segment Anything Model (MedSAM) [14] was developed as an extension of SAM specifically tailored for medical image segmentation. MedSAM fine-tunes the mask decoder of SAM on a diverse set of medical datasets, enhancing its ability to handle domain-specific noise and variations. This adaptation improves the model's robustness and accuracy, enabling it to achieve more reliable segmentation results for medical use cases. MedSAM has been successfully applied to tasks such as organ segmentation, lesion detection, and tissue analysis, demonstrating its potential to support clinical analysis and research [94]–[96].

Research in biomedical imaging has shown that the original SAM model performs more reliably and accurately when prompted with bounding boxes compared to other input types, such as points [97]. This observation has led to the development of hybrid systems that combine automated detection algorithms with SAM for precise object segmentation. For instance, in [98], YOLOv8 was used to generate bounding box prompts for SAM in ultrasound, CT, and X-ray images, achieving higher segmentation accuracy compared to other tested models.

## 2.3 IMAGE COMPRESSION

In its simplest form, data compression schemes can be seen as two mappings presented in Figure 6: an encoder (red dashed box), which maps the data $x \in \mathbb{R}^N$ into a compressed bitstream $b$; and a decoder (blue dashed box), which maps the bitstream back to an approximation of the original $x$, $\hat{x} \in \mathbb{R}^N$ (if the coding process is done losslessly, $\hat{x} = x$). For images, $x$ represents the pixels' vector of length $N$, where $N = W \times H \times C$. Here, $W$ and $H$ represent both the width and height of the image, respectively, and $C$ the number of channels.

The encoder's mapping is composed as $\gamma \circ \alpha$, where $\alpha$ represents the mapping from the data domain to a discrete set $s$ (typically this process is a quantiser) and $\gamma$ an invertible entropy code allowing for the mapping of $s$ into the bitstream. In order to reconstruct the approximation of the original $x$, the decoder reverses the steps performed by the encoder by inverting the $\gamma$ and posteriorly mapping the discrete values to the data domain (often called a dequantiser).

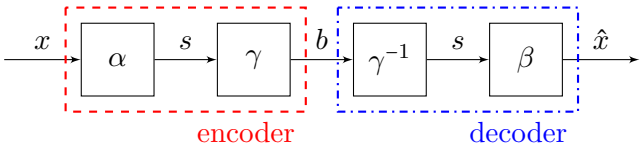



Figure 6: Generic encoder-decoder scheme.

Using this approach, the bitstream representation may be significantly reduced in comparison to the raw image, but it will produce low-quality images. Regarding this limitation, an additional block, transform coding ($T$), is introduced in the encoder, and its inverse on the decoder, generating the framework presented in Figure 7. By mapping the image into another domain, such as the frequency domain, that is able to compact the energy of the image in a smaller number of coefficients $y$. This set compacted representation of the image can be further quantised to discard less important information. The inverse transformation is also applied at the decoder, allowing to reconstruct the approximation of the original image from the decoded coefficients $\hat{y}$.

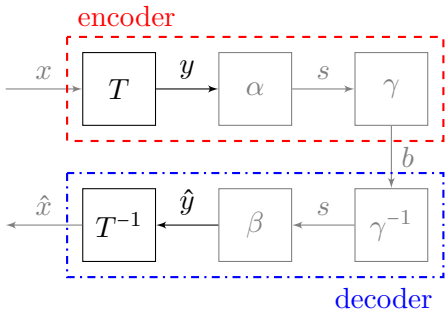



Figure 7: Transform coding for a generic compression scheme.

While transform coding and other compression techniques offer benefits in terms of storage and transmission efficiency, their application to biomedical imaging requires careful consideration. Biomedical data, due to its unique nature and potential loss of critical diagnostic information, is subject to strict legal and regulatory standards, such as compliance with the Digital Imaging and Communications in Medicine (DICOM) standard [99]. These regulations ensure trust, ethical practices, and interoperability across biomedical systems. To preserve data integrity, conventional lossless image coding is often employed [11]. However, lossless compression ratios are significantly lower compared to lossy or near-lossless methods, prompting the need for more advanced approaches that can retain essential biomedical information

while achieving higher compression efficiency [100]. The choice of the compression method ultimately depends on the acceptable balance between file size reduction and the fidelity of the compressed data.

Currently, DICOM supports various standards for image and volume coding, providing guidelines to integrate these algorithms into DICOM-compliant systems. For image compression, DICOM includes standards such as Joint Photographic Experts Group (JPEG), JPEG-LS, JPEG 2000, and JPEG-XL, while for slice-stack coding, it supports standards like MPEG-2, AVC/H.264, and HEVC/H.265.

### 2.3.1 *High Efficiency Video Coding*

HEVC/H.265 [101] is a hybrid video coding standard approved by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) in 2013, developed by the Joint Collaborative Team on Video Coding (JCT-VC), which is composed by the ITU-T Video Coding Experts Group (VCEG) and the International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) Moving Picture Experts Group (MPEG) standardisation organisations, as the successor to Advanced Video Coding (AVC)/H.264. Designed to address the growing demand for high-resolution video, HEVC supports resolutions up to $8192 \times 4320$, including 8K Ultra-High-Definition (UHD) ($7680 \times 4320$). It achieves significant improvements in data compression, offering 25% to 50% better coding efficiency compared to H.264 for the same level of video quality [102]. The generic HEVC encoder block diagram is illustrated in Figure 8.

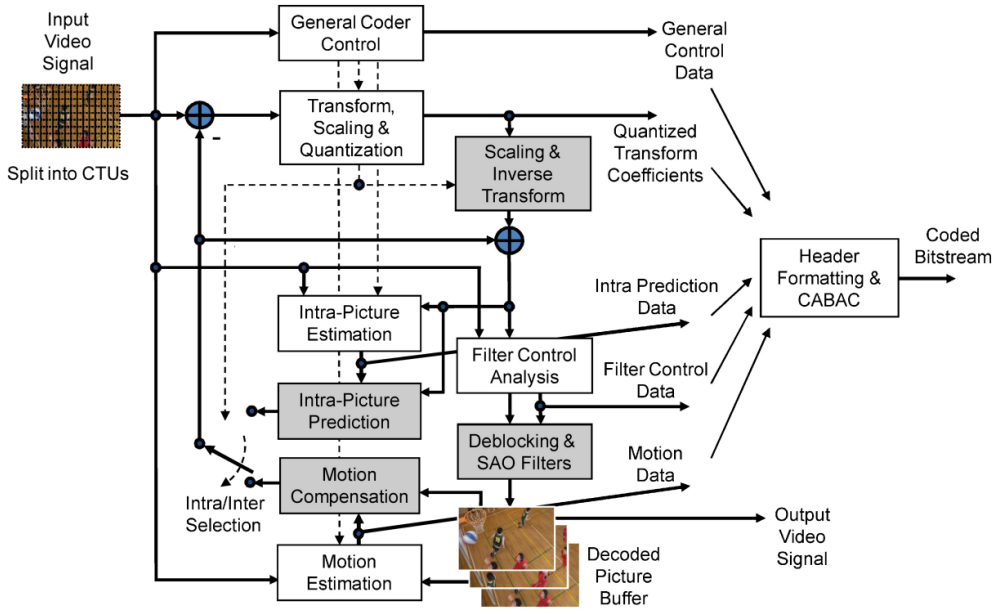



Figure 8: Typical HEVC encoder block diagram [102].

Similar to its predecessor, HEVC employs a hybrid video coding approach that combines block-based motion-compensated prediction and transform coding. However, HEVC introduces several advancements to enhance compression efficiency and adaptability. One of the key differences lies in the block-based structures used for encoding. While AVC relies on a partition of the image into macroblocks of $16 \times 16$ pixels, HEVC adopts a more flexible structure called the Coding Tree Unit (CTU), which can range in size from $16 \times 16$ to $64 \times 64$ pixels. The increase in the block's size allows the exploitation of more spatial redundancy, which is useful for higher-resolution content. Each CTU can be recursively divided into smaller blocks using a quadtree structure, enabling efficient representation of both large uniform areas, and fine details within a single frame.

Within each CTU, the frame is further partitioned into Coding Unit (CU)s. These blocks are the basic units for prediction and can vary in size from $8 \times 8$ to $64 \times 64$. Each CU contains information about the prediction mode (intra or inter) and serves as the root for further partitioning into Prediction Unit (PU)s and Transform Unit (TU)s. PUs represent the regions for which motion compensation (inter-prediction) or intra-prediction is performed. For intra-prediction, PUs use neighbouring spatial information from already encoded pixels to predict the current block, reducing spatial redundancy.

TUs define the regions where transform coding is applied. They are used to convert the residual data into frequency-domain coefficients using transforms, such as the Discrete Cosine Transform (DCT). TUs can vary in size, ranging from $4 \times 4$ to $32 \times 32$ pixels, and are determined by the quadtree partitioning within the CU. The transform coefficients are quantised using a Quantisation Parameter (QP) defined by the user, which controls the trade-off between bitrate and quality, and ranges between 0 to 51. For instance, a QP of 51 achieves maximum compression at the cost of severe image degradation, while lower QP values preserve higher image quality but result in larger bitrates. Following both prediction and transformation, the resulting residual information is entropy-encoded. This hierarchical and flexible approach to partitioning and encoding, combined with other introduced advanced prediction and transform techniques, allows HEVC to achieve significant improvements in compression efficiency over previous standards. The standard supports the encoding of image and video data with 8 and 10 bits per sample; however, a range extension was introduced [103], allowing for the encoding of images with higher bit depths, up to a maximum of 16 bits per sample. This extension targets applications such as medical imaging, screen content coding, and other scenarios where higher bit-depth is required.

### 2.3.2 *Versatile Video Coding*

VVC/H.266 [104], is the latest video coding standard approved by the ITU-T in 2020, developed by the Joint Video Experts Team (JVET), a collaboration between the ITU-T VCEG and the ISO/IEC MPEG. As the successor to HEVC, VVC was designed to address the increasing demand for efficient compression of ultra-high-resolution video, including 8K and beyond, as well as emerging applications such as 360-degree video, screen content, and machine-to-machine communication. VVC achieves 30–50% better compression efficiency compared to HEVC for the same visual quality [105]. The typical VVC encoder block diagram is illustrated in Figure 9.

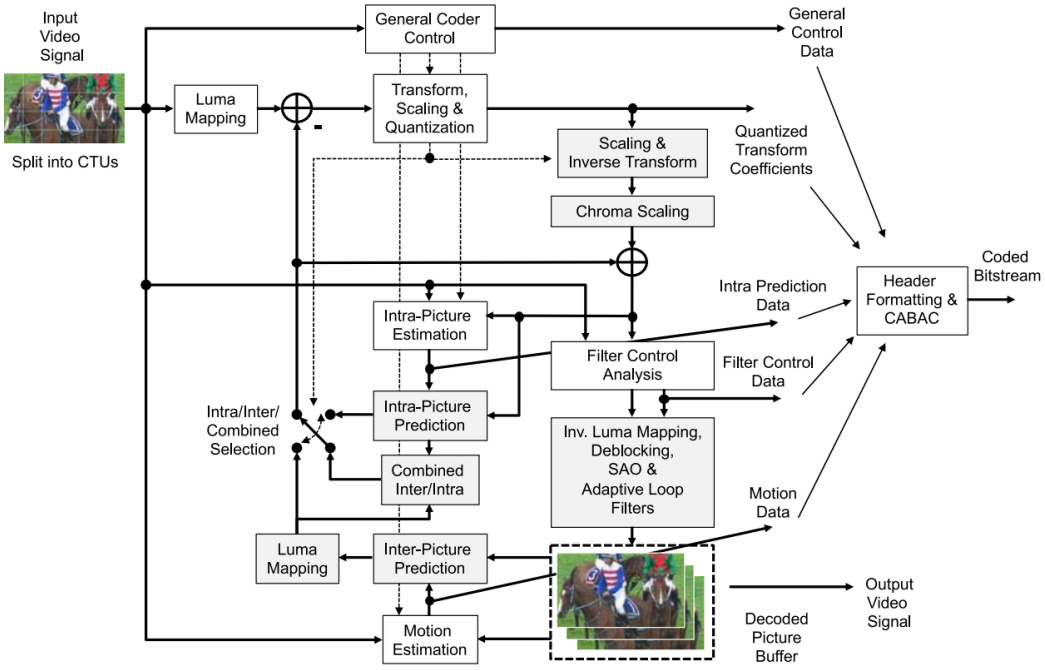
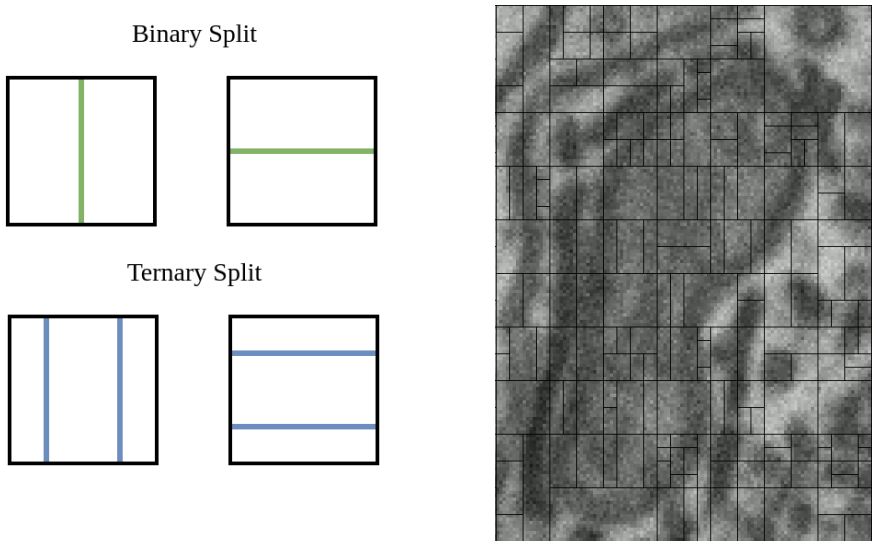



Figure 9: VVC encoder block diagram [105].

Built on the hybrid video coding framework of its predecessors, VVC introduces several advanced tools and techniques to further improve compression performance. One of the most notable enhancements is the flexible partitioning structure. While HEVC uses a quadtree-based partitioning scheme for CTUs, VVC extends it with a Multi-Type Tree (MTT) structure. This allows for more granular partitioning, combining quadtree, binary, and ternary splits, illustrated at Figure 10a, enabling better adaptation to complex textures and motion patterns, as shown in Figure 10b. The CTU size in VVC remains similar to HEVC, ranging from $16 \times 16$ to $128 \times 128$ pixels, but the additional partitioning options provide greater flexibility.

Similar to HEVC, within each CTU, VVC partitions the frame into CUs, but with size ranging from $4 \times 4$ to $128 \times 128$, offering a wider range of block sizes when compared to its predecessor. Within intra-prediction, VVC supports 67 directional




(a) MTT structures. (b) Partition example using VVC.

Figure 10: MTT partitioning in VVC.

modes, compared to 35 in HEVC, along with wide-angle intra-prediction, improving the accuracy of spatial prediction. The quantisation process is also enhanced with dependent quantisation, which reduces quantisation artefacts by considering the context of neighbouring coefficients. Finally, the QP ranges from 0 to 61, extending the range of HEVC.

### 2.3.3 *ROI-Based Coding*

Classical lossy and lossless compression methods, such as the previously presented HEVC and VVC, treat the entire image uniformly, applying the same encoding parameters across all regions. While this approach simplifies the compression process, it often fails to account for the varying importance of different regions within an image. In many applications, particularly in biomedical imaging, certain regions contain critical information that must be preserved with high fidelity, while other regions may tolerate higher levels of degradation due to compression, without significant loss of diagnostic value. This limitation in standard coding has led to the development of ROI-based coding techniques, which prioritise the preservation of important regions while allowing more aggressive compression in less critical areas [106]–[108]. By focusing computational resources on ROIs, ROI-based coding can achieve significant reductions in file size, while maintaining high visual quality in critical areas [109].

One of the earliest implementations of ROI-based coding can be found in the JPEG2000 standard [110], which introduced a mechanism to define and encode regions of interest with higher quality, ensuring that important details are preserved.

This approach has been widely adopted in medical imaging, where preserving diagnostic information is paramount [111]–[113]. However, many of these methods end up completely removing the background region, resulting in a loss of spatial coherence and contextual information, which can be critical for certain applications.

In more recent coding standards, such as HEVC and VVC, some works have employed tiling rate-control algorithms to achieve ROI-based coding at the CTU level, significantly reducing the bitrate while preserving critical content [114]–[116]. However, these approaches often rely on coarse-grained partitioning, limiting their ability to adapt to complex shapes and fine details. To address this limitation, the authors in [44] proposed a novel methodology for defining ROIs in HEVC coding at the CU level, achieving a resolution down to $8 \times 8$ pixels. This approach allows accurate control over the quality of ROI and non-ROI regions through the QP parameter, enabling finer adaptation to the image content.

Recent advancements in ROI-based compression have explored the use of learning-based and transformer-based architectures to further optimise rate control and quality trade-off. In [117], a learnable compression system capable of achieving variable-rate compression and ROI bit allocation is employed. This framework uses a prompt generation network to condition a transformer-based autoencoder using an ROI mask, enabling spatially adaptive bit allocation. While this approach demonstrates promising results, it primarily focuses on preserving ROI information, often leading to the complete degradation or destruction of background information. This results in a loss of spatial coherence, which can be detrimental for applications requiring a holistic understanding of the scene. To tackle this loss of spatial coherence, [118] applies different loss functions for ROI and background regions, ensuring that critical areas are preserved with high fidelity while background regions undergo more aggressive compression. Additionally, a Generative Adversarial Network (GAN) is further used to maintain perceptual quality in non-ROI regions, providing a balance between compression efficiency and visual realism.

## 2.4 SUMMARY

This chapter provides the necessary background and state-of-the-art for this dissertation, beginning with an overview of biomedical imaging, emphasising its role in biological research. Particular focus is given to EM imaging due to its relevance to this study, along with a description of the datasets used. Subsequently, the role of computer vision in biomedical imaging, particularly for detection and segmentation tasks, and their importance for automation are explored. Finally, the chapter introduces some concepts on image compression, discussing standard coding techniques relevant to the study, and a review of the state-of-the-art in ROI-based coding.

# 3

# IMAGE COMPRESSION FOR MACHINE-BASED ANALYSIS

This chapter outlines the contributions made to improve learning-based computer vision models' performance when dealing with compressed images. First, the proposed methodologies are detailed, focusing on approaches to mitigate the impact of compression artefacts on these models. Subsequently, modifications for the HEVC codec are described, enabling non-rectangular ROI-based coding for better ROI shape matching. Finally, the modifications implemented in the VVC codec are discussed, specifically addressing the adaptations for ROI-based coding at the CU level to support generic ROI shapes.

## 3.1 PROPOSED METHODOLOGIES

To establish a baseline understanding of how compression artefacts influence the model's performance, especially for biomedical images, an initial experiment is conducted to evaluate the relationship between image degradation and machine vision task performance. As illustrated in Figure 11, the pipeline begins with an input biomedical image subjected to lossy compression using a standard codec with various QP levels. The image is then reconstructed and fed into a machine vision model, pre-trained with uncompressed images, to perform an automatic task, such as detection or segmentation of mitochondria in EM images.

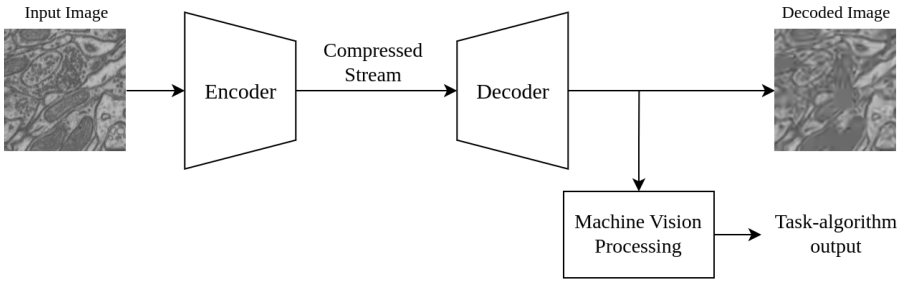
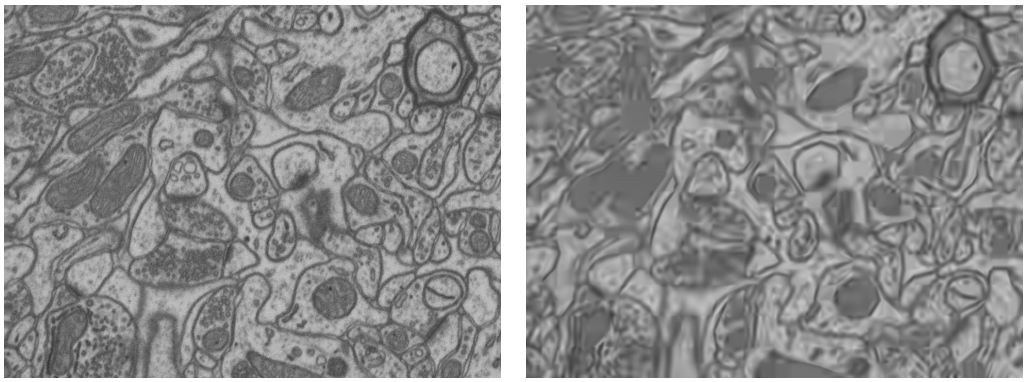



Figure 11: Basic pipeline for evaluating the impact of image compression on machine vision tasks.

The experiment intends to evaluate how increasing the levels of image distortion, resulted from the coding process and controlled through QP, degrades the structural and textural features that are critical for a computer vision model's interpretation. For instance, in segmentation models, which rely on precise boundary delineation

of mitochondria, coding artefacts such as blurred edges and quantised textures, as illustrated in Figure 12, may erode the spatial coherence required for accurate pixel-wise classification [14]. For machine analysis, degradations due to compression become an even greater challenge due to a key limitation of conventional codecs: their optimisation for objective metrics, such as Peak Signal-to-Noise Ratio (PSNR), often fails to prioritise features that machine vision systems inherently rely on [119]. This discrepancy can be addressed through two potential solutions: (1) either the model is adapted for the analysis of degraded images, as demonstrated in studies like [12], or (2) the codecs are adapted for specific area preservation, specifically for mitochondrial structures [44].



(a) Uncompressed sample. (b) Sample compressed with QP 51.

Figure 12: Comparison between uncompressed and compressed samples from the Lucchi++ dataset.

For the first approach, the computer vision model is fine-tuned using both uncompressed and compressed images. The process begins by importing the weights from the previously used model trained with uncompressed images. Subsequently, the model undergoes further training, using an expanded dataset that includes both original images and respective compressed versions, with different levels of distortion. This dual-exposure strategy ensures that the model becomes robust to artefacts and distortions introduced by the codec's compression, such as quantised textures. Once the model is retrained, the evaluation process follows the same methodology as the baseline approach previously outlined in Figure 11. The only distinction lies in the updated computer vision model, which is now optimised to handle compressed inputs.

For the second approach, the model's performance is preserved through selective content preservation by the codec, defined by ROIs. To implement a fully automated framework, a trained model is used to identify and select the ROIs that require higher quality during compression. Since the input consists of original and uncompressed images, a model trained exclusively on uncompressed data is preferred. This is because such a model is better suited to accurately identify ROIs without

being influenced by compression artefacts, ensuring more reliable content selection compared to a fine-tuned model that has been exposed to degraded inputs [44].

Once the ROIs are identified, a modified encoder is used to compress these images, allocating higher quality, and consequently more bits, to the selected regions. This quality-selective encoding ensures that relevant features, such as mitochondria for instance, are preserved with minimal distortion. Importantly, the adapted encoder needs to maintain standard compliance, allowing the compressed images to be decoded using standard decoders, without requiring modifications.

The resulting compressed image, which exhibits higher quality in the ROIs, is then evaluated using a computer vision model. This model can be either the one used for ROI identification or the retrained model, depending on the experimental setup. The framework for this method is illustrated in Figure 13.

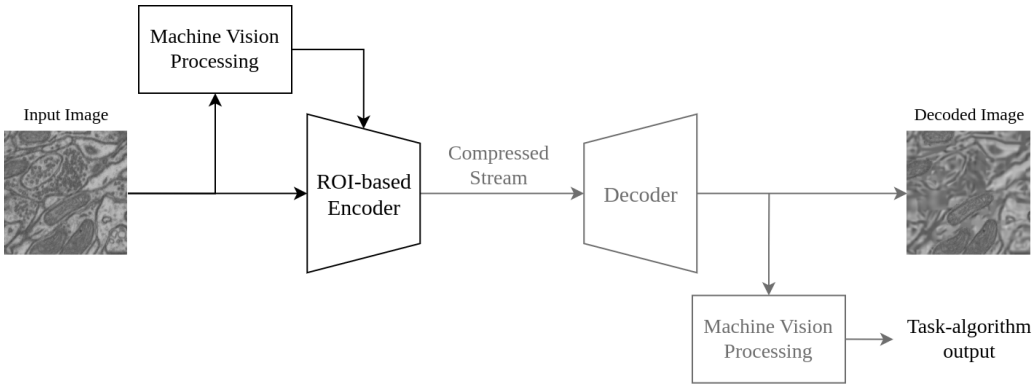



Figure 13: Pipeline for evaluating the impact of content-selective image compression on machine vision tasks.

## 3.2 ROI-BASED HEVC

Previous work [44], as explored in Section 2.3.3, introduced a modification of the HEVC standard, enabling selective compression between ROIs and background areas while maintaining standard compliance. This approach relied on rectangular ROIs, where each region was defined by its top-left and bottom-right coordinates. The compression quality was controlled at the CU level through a QP assignment function, according to

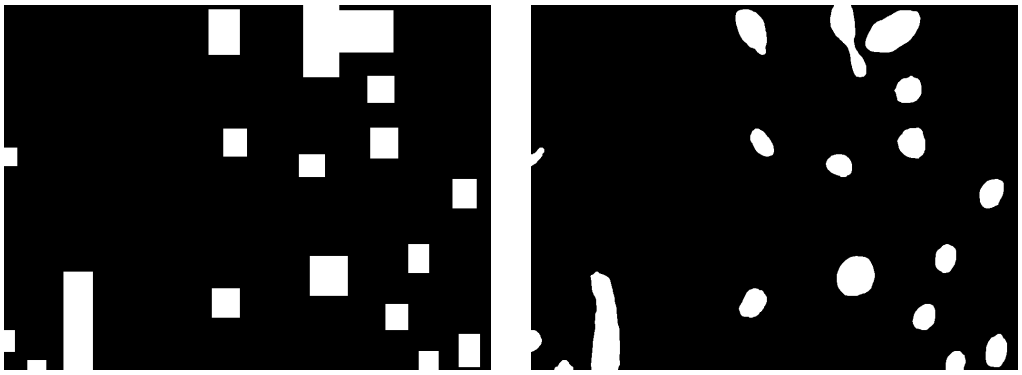
$$\mathrm{QP}_{\mathrm{CU}} = \begin{cases} \mathrm{QP}_{\mathrm{F}}, & \text{if } \exists \text{ ROI s.t. ROI} \cap \mathrm{CU} \neq \varnothing, \\ \mathrm{QP}_{\mathrm{B}}, & \text{otherwise} \end{cases}. \tag{1}$$

This function assigns different QP values to CUs based on their intersection with ROIs, where $\mathrm{QP}_{\mathrm{F}}$ represents the QP for foreground regions (ROI) and $\mathrm{QP}_{\mathrm{B}}$ for background areas.

Table 1: Average percentage of area covered by ground truth ROIs for each dataset, distinguishing between rectangular (bounding boxes) and non-rectangular (masks) ROIs.

| **ROI type** | **Lucchi++** | **Kasthuri++** |
|---|---|---|
| rectangular | 9.87% | 5.45% |
| non-rectangular | 6.84% | 3.17% |

However, an analysis of the Luchi++ and Kasthuri++ datasets revealed a significant limitation of the rectangular ROI approach. As shown in Table 1, using non-rectangular ROIs instead of regions defined by bounding boxes results in reducing the total area occupied by ROIs by 3.03 percentage point (pp) (9.87% to 6.84%) and 2.28 pp (5.45% to 3.17%) for the Lucchi++ and Kasthuri++ datasets, respectively. This reduction in ROI area suggests that non-rectangular regions could potentially improve coding efficiency while still preserving the critical mitochondria regions. Figure 14 presents the difference between rectangular and non-rectangular ROI representations in binary masks.



(a) Rectangular ROIs. (b) Non-rectangular ROIs.

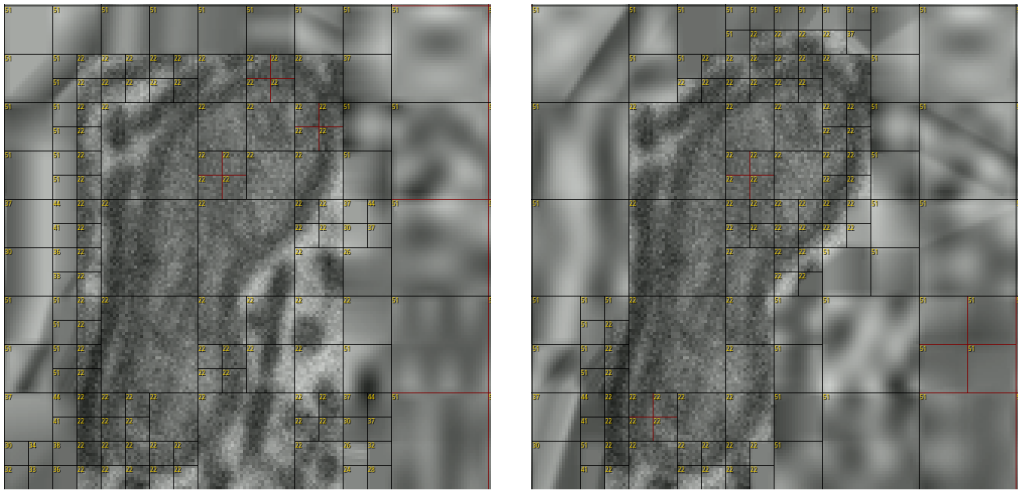
Figure 14: Binary masks with ROIs identifying mitochondria.

In the original work, the intersection between ROIs and CUs was determined using only the boundaries of these blocks. However, this boundary-based approach becomes challenging when dealing with non-rectangular ROIs. To simplify the process and enable non-rectangular ROI-based compression, the QP assignment function was reformulated to use binary masks instead of bounding boxes' coordinates, resulting in

$$\mathrm{QP} = \begin{cases} \mathrm{QP_F}, & \text{if } \exists (x,y) \in \mathrm{CU} \text{ s.t. } M(x,y) = 1, \\ \mathrm{QP_B}, & \text{otherwise} \end{cases}, \tag{2}$$

where $M(x,y)$ represents a binary mask value at position $(x,y)$ within the CU. This modification allows for arbitrary ROI shapes, generalising the region-adaptive compression for any object.

The impact of this enhancement is demonstrated in Figure 15, which compares the compression results using rectangular and non-rectangular ROI-based coding. As shown in this figure, when applying QP 22 for mitochondria regions and QP 51 for background areas, the non-rectangular approach achieves a more precise region targeting, allowing for a more efficient compression of background regions while maintaining higher quality in mitochondria information.



(a) Rectangular coding. (b) Non-rectangular coding.

Figure 15: Comparison between rectangular and non-rectangular ROI-based coding with HEVC for a QP 22 in the mitochondria region ($QP_F$) and QP 51 for the background ($QP_B$). The smallest partitioning is $8 \times 8$ pixels.

## 3.3 ROI-BASED VVC

As discussed in Section 2.3.2, VVC inherently provides improved coding efficiency for the same quality. Additionally, two specific features of this codec enhance its ability to adapt to different ROI shapes, particularly when they are non-rectangular. First, the minimum size of CUs is reduced to $4 \times 4$, compared to $8 \times 8$ in HEVC. Second, the introduction of MTT partitioning allows for better shape matching of CU partitions to the segmented masks. Both these features are particularly suitable for detailed and irregularly shaped regions, offering advantages over its predecessor.

The same approach used in HEVC was attempted with VVC; however, a direct implementation proved unfeasible due to the increased complexity of VVC architecture. Consequently, two key aspects had to be considered: the use of Quantisation Group (QG) and the modification of the Lagrangian bit-allocation.

### 3.3.1 *Quantisation Groups*

In VVC, rate control algorithms and perceptually optimized encoding approaches can assign different QPs for different regions within a frame [120]. These adjustable regions are denoted as QGs, allowing larger coding units, CTUs, to be divided into smaller groups. The size of each QG is specified on the picture header. For each QG, the QP information is stored on the first TU within the QG. Throughout this subsection, variables defined in bold correspond to values transmitted in the bitstream, whereas variables in italics represent those specified in the ITU standard, which can be derived from input arguments and/or variables transmitted in the bitstream. The QPs for the luma component are encoded differentially. For each QG, that contains at least one TU to transport the QP information, the difference (*CuQpDeltaVal*) between the current group's QP, and a prediction derived from spatially neighbouring QGs is transmitted. To represent the *CuQpDeltaVal* in the bitstream, two values are transmitted for each QG:

- **cu_qp_delta_abs:** specifies the absolute value of *CuQpDeltaVal*, allocating 6 bits in the bitstream;
- **cu_qp_delta_sign_flag:** indicates the sign of *CuQpDeltaVal*, where 0 denotes a positive value and 1 a negative value. If this flag is absent, it is inferred as 0.

The range of *CuQpDeltaVal* is defined as

$$\left[\frac{QpBdOffset}{2} - 32, \frac{QpBdOffset}{2} + 31\right], \tag{3}$$

where *QpBdOffset* is the luma QP range offset, given by

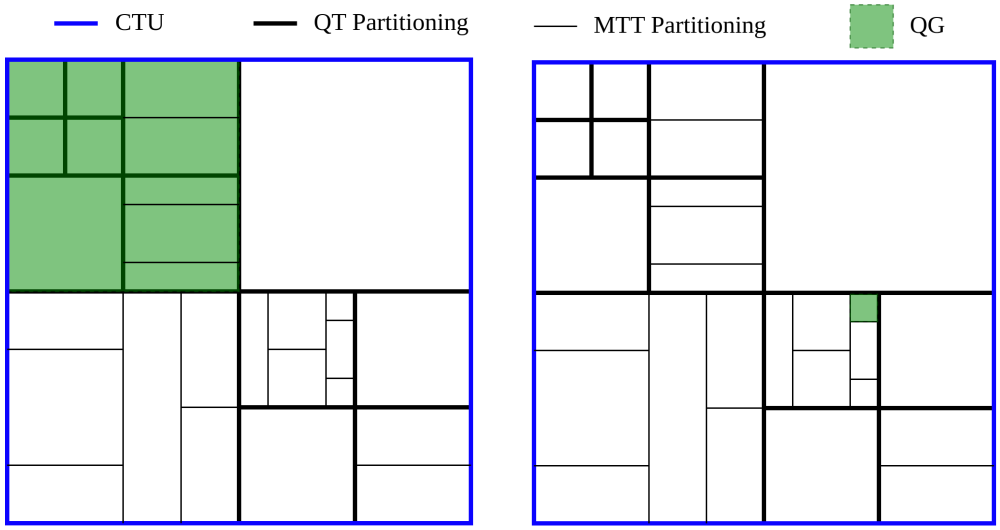
$$QpBdOffset = 6 \times \textbf{sps\_bitdepth\_minus8}, \tag{4}$$

and **sps_bitdepth_minus8** specifies the bit depth of luma samples in the bitstream, minus 8. Since VVC supports bit depths in the range [8,16], **sps_bitdepth_minus8** can range from [0,8]. In this work, the datasets used, described in Section 2.1.2, have a bit depth of 8, resulting in a *CuQpDeltaVal* range of [−32,31]. Additionally, the encoding argument *MaxDeltaQP* can further constrain and reduce this range.

The QG is a rectangular region within a Coding Tree Block (CTB) that shares a uniform QP. The size of QGs is determined by the parameter *MaxCuDQPDepth*, which specifies the maximum CTU subdivision level used to define luma QGs. Subdivision levels increase based on the type of partitioning, as follows:

- Binary and ternary splits (MTT splits) increase the level by 1;
- Quaternary splits increase the level by 2.

For instance, to define regions within a CTU that match the resolution of the region highlighted in green in Figure 16a, where all CUs inside each region share the same QP but differ from the QPs of neighbouring regions, a QG of this size must be created. To achieve this, the *MaxCuDQPDepth* parameter must be set to at least 2, as the region is generated through a quaternary split of the CTU. In contrast, if a smaller region resolution is required, such as the one shown in green in Figure 16b, the *MaxCuDQPDepth* must be at least 6, as the subdivision involves two quaternary splits and two ternary splits. In summary, the proper configuration of QGs allows for finer control of QPs and efficient bitstream representation within CTUs, aligning with the goals of ROI-based coding.




(a) MaxCuDQPDepth = 2. (b) MaxCuDQPDepth = 6.

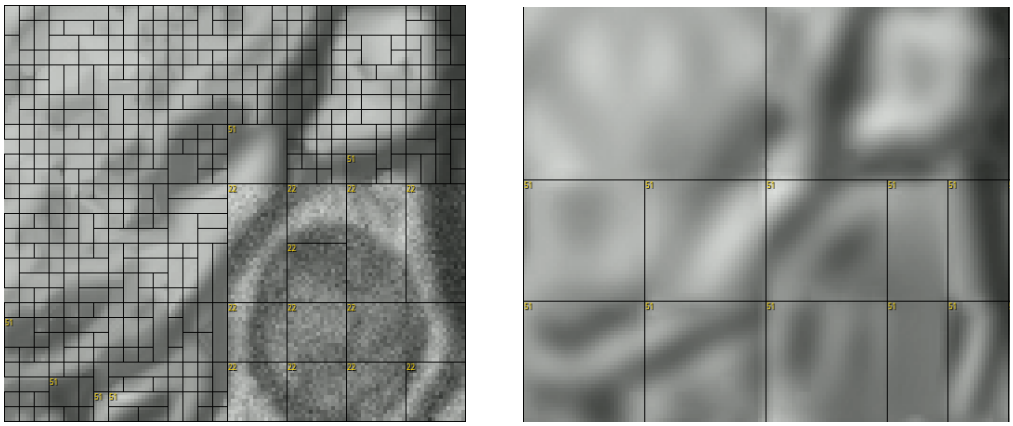
Figure 16: QG size representation based on the "MaxCuDQPDepth" argument.

### 3.3.2 *Lagrangian bit-allocation*

Applying the correct QG, as previously discussed, enables the differentiation of the QP used for each CU. However, even with this tool and the strategy employed in HEVC, presented in Equation (2), the desired outcome is not fully achieved. For instance, Figure 17a illustrates the result of applying this strategy to a sample from the Lucchi++ dataset.

In this case, a QP of 51 was defined for the background, while the ROI, defined in Figure 18, was assigned a QP of 22. As observed, two inconsistencies emerge with the desired outcome. First, although the defined ROI has a non-rectangular shape, the CU definition of the foreground does not match the correct shape, contrary to the desired behaviour of HEVC shown in Figure 15b. Additionally, the block partitioning of the background region produces very small blocks, mostly of size

$4 \times 4$, which is unexpected when compared to the block partitioning produced when coding the same sample with standard VVC at QP $= 51$.



(a) Partitioning of a sample coded using VVC with ROI-based encoding (incomplete), where $\mathrm{QP_F} = 22$ and $\mathrm{QP_B} = 51$. The ROI is defined in Figure 18.

(b) Partitioning of a sample coded using standard VVC with a fixed QP of 51.

Figure 17: Comparison of CU partitioning between VVC with ROI-based encoding (incomplete) and standard VVC.

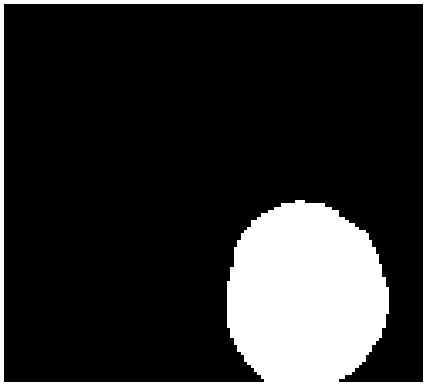


Figure 18: ROI mask corresponding to the sample shown in Figure 17a.

Furthermore, in the background region, nearly all CUs contain no TUs, and only PUs are used. This can be observed in Figure 17b, as only TUs use the QP parameter, as previously discussed in Section 2.3.1. Moreover, this background region exhibits higher quality than expected for QP 51, accompanied by an excessively high data size per block. This phenomenon occurs because, during encoding, a QP of 51 fails to provide the CUs with sufficient quality, causing the encoder to rely solely on PUs and residual information to meet the desired quality level.

This behaviour stems from the cost function optimized by the VVC encoder, which is based on the Lagrangian formulation:

$$\mathcal{L} = D + \lambda R, \tag{5}$$

where $\lambda$ represents the Lagrange multiplier that balances the trade-off between distortion, $D$, and rate, $R$. In VVC, by default, $\lambda$ is derived from the QP of the CTU, ensuring a constant rate-distortion trade-off within it. However, this approach is not suitable when different rate-distortion trade-offs are desired at the CU level.

Returning to the analysis of Figure 17a, the constant $\lambda$ derived from the foreground QP results in a uniform rate-distortion optimisation across the entire image, preventing the encoder from achieving the intended quality differentiation between foreground and background regions. In contrast, this issue did not arise in HEVC. In there, when $\mathrm{QP}_{\mathrm{CU}}$ was modified, the corresponding $\lambda$ was automatically adjusted to reflect the new QP value, allowing the encoder to achieve the desired quality differentiation at the CU level. However, in VVC, a different mechanism for the adjustment of $\lambda$ requires additional modifications on the Lagrangian formulation to achieve the intended results. To formalise these modifications, it is necessary to first define the distortion metric used by the encoder.

Let $D_k$ denote the Sum of Squared Errors (SSE) for a given CTU, $B_k$, at index $k$, defined as

$$D_k = \sum_{(x,y)\in B_k} \left(s_{(x,y)} - \hat{s}_{(x,y)}\right)^2, \tag{6}$$

where $s$ and $\hat{s}$ represent the original and reconstructed pixel values, respectively, at position $(x,y)$ within $B_k$. This distortion metric is used by default in both HEVC and VVC standards, in order to evaluate the reconstruction quality of encoded CTUs.

Furthermore, let $\mathbf{p}_k$ represent the vector of encoding decisions for $B_k$, incorporating parameters such as prediction mode, QP, and block partitioning. The encoder's primary goal is to determine the set of coding parameters $\{\mathbf{p}_k\}$ that minimise the overall distortion $D_{\mathrm{pic}}$ and the corresponding rate $R_{\mathrm{pic}}$ [121]. Using the concept of the Lagrangian multipliers, this optimisation problem is expressed as

$$\min_{\{\mathbf{p}_k\}} D_{\mathrm{pic}}(\{\mathbf{p}_k\}) + \lambda_k \cdot R_{\mathrm{pic}}(\{\mathbf{p}_k\}), \tag{7}$$

where $\lambda_k$ is the Lagrange multiplier for $B_k$, that balances the trade-off between distortion and rate. Given that the SSE is a block-additive distortion measure, Equation (7) can be reformulated as

$$\min_{\{\mathbf{p}_k\}} \sum_k D_k(\mathbf{p}_k) + \lambda \cdot R_k(\mathbf{p}_k), \tag{8}$$

where $D_k(\mathbf{p}_k)$ is the distortion for the CTU, and $R_k(\mathbf{p}_k)$ the rate required to transmit all encoding decisions $\mathbf{p}_k$. To simplify the encoding process, dependencies

between CTUs are typically ignored [121], allowing the optimisation problem to be solved approximately by optimising each CTU independently:

$$\min_{\mathbf{p}_k} D_k(\mathbf{p}_k) + \lambda \cdot R_k(\mathbf{p}_k),\ \forall k. \tag{9}$$

To extend the optimisation problem to include CU-level lambda, $\lambda_{k,j}$ for each CU within the CTU, the proposed implementation considers each CU independently. Denoting the set of coding decisions for all CUs in a CTU as $\mathbf{p}_k = \{\mathbf{p}_{k,j}\}$, where $\mathbf{p}_{k,j}$ represents the encoding decision for the $j$-th CU within the $k$-th CTU and ignoring dependencies between CUs within a CTU, the optimisation problem can be expressed as

$$\min_{\mathbf{p}_{k,j}} D_{k,j}\left(\mathbf{p}_{k,j}\right) + \lambda_{k,j} \cdot R_{k,j}\left(\mathbf{p}_{k,j}\right),\ \forall k, \forall j. \tag{10}$$

Similarly to the HEVC adaptation, for each $j$-th CU within the $k$-th CTU, the $\lambda_{k,j}$ is selected based on whether the CU overlaps with the foreground mask:

$$\lambda_{k,j} = \begin{cases} \lambda_{\mathrm{F}}, & \text{if } \exists (x,y) \in B_{k,j} \text{ s.t. } M(x,y) = 1, \\ \lambda_{\mathrm{B}}, & \text{otherwise} \end{cases}, \tag{11}$$

where $\lambda_{\mathrm{F}}$ and $\lambda_{\mathrm{B}}$ represent the foreground and background Lagrangian multiplier, respectively. Note that, since the foreground will have a higher quality than the background, $\lambda_{\mathrm{F}} < \lambda_{\mathrm{B}}$. The $\lambda_{\mathrm{B}}$ is derived from the respective $\mathrm{QP}_{\mathrm{B}}$ as

$$\lambda_{\mathrm{B}} = c \cdot 2^{\frac{\mathrm{QP}_{\mathrm{B}}-12}{3}}, \tag{12}$$

where $c$ is a constant determined by the VVC [122][1]. By calculating the relation between $\lambda_{\mathrm{B}}$ and $\lambda_{\mathrm{F}}$, $\lambda_{\mathrm{F}}$ can be mathematically deduced as

$$\lambda_{\mathrm{F}} = \lambda_{\mathrm{B}} \cdot 2^{\frac{-\Delta\mathrm{QP}}{3}}, \tag{13}$$

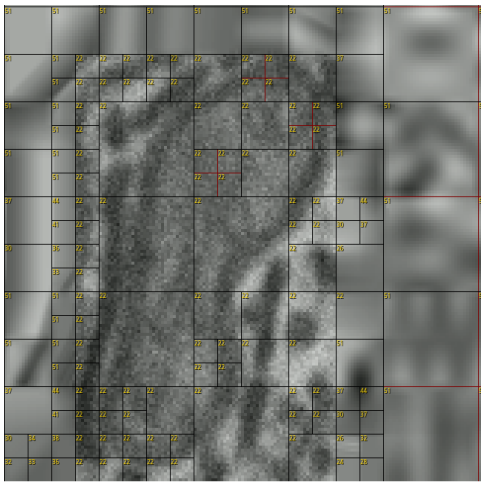
where $\Delta\mathrm{QP} = \mathrm{QP}_{\mathrm{B}} - \mathrm{QP}_{\mathrm{F}}$.

Given the challenges in tracking the rate contributions across multiple parts of the codec implementation, a practical approach involves normalising the optimisation objective for each CU by $\lambda_{k,j}$. This adjustment transfers the influence of $\lambda_{k,j}$ from the rate term to the distortion term, consolidating its effect into a single point where distortion is calculated. By applying this normalisation to all CUs across all CTUs in the entire picture, the following Lagrangian optimisation function is obtained:

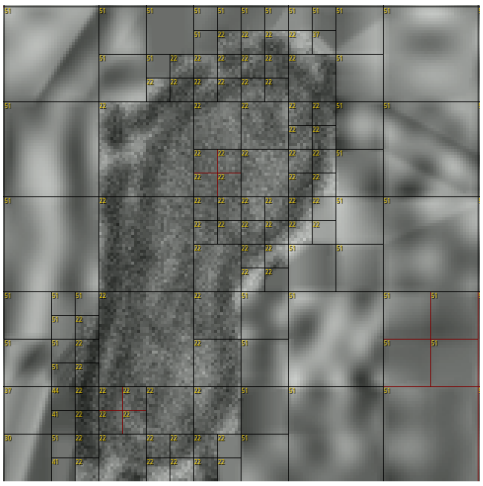
1 The software manual contains some inconsistencies in defining $\lambda$ based on QP. The equation presented here has been corrected using the implemented software as baseline.

$$\mathcal{L} = \sum_{k} \left[ \sum_{j} \left( \frac{D_{k,j}\left(\mathbf{p}_{k,j}\right)}{\lambda_{k,j}} \right) + R_k\left(\mathbf{p}_k\right) \right], \tag{14}$$

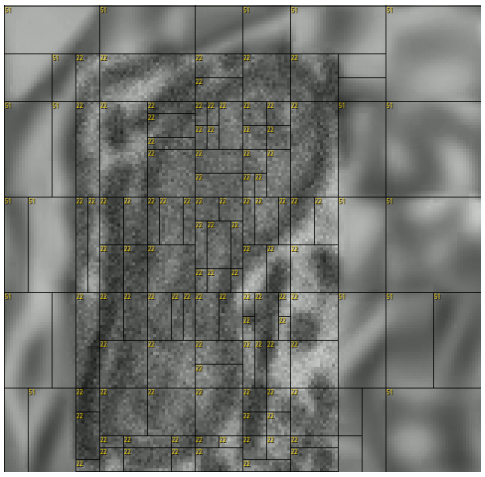
where the desired balance between distortion and rate is achieved for each region. For instance, comparing to the compression achievable in HEVC in Figures 19a and 19b, Figures 19c and 19d illustrate the partitioning of the implemented ROI-based encoding in VVC, using $\mathrm{QP_F} = 22$ and $\mathrm{QP_B} = 51$, for both rectangular and non-rectangular regions, respectively. Compared to its predecessor, VVC achieves a more fine-grained adaptation to the shape of the mitochondria, enabling a larger portion of the background to be encoded at lower quality while preserving the quality of the mitochondria content. This behaviour is particularly noticeable when analysing the coding partitioning using non-rectangular ROIs. The improved adaptability of VVC is enabled by its smaller minimum block size and the use of non-square MTT splits, providing greater flexibility in partitioning.



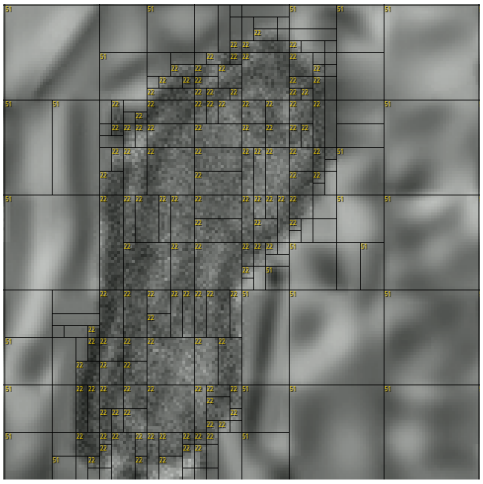
(a) Rectangular HEVC coding.



(b) Non-rectangular HEVC coding.



(c) Rectangular VVC coding.



(d) Non-rectangular VVC coding.

Figure 19: Comparison between ROI-based coding in HEVC and VVC for a QP 22 in the mitochondria region and QP 51 for the background. The smallest partitioning is $8 \times 8$ and $4 \times 4$ pixels for HEVC and VVC, respectively.

## 3.4 SUMMARY

This chapter describes methods to enhance learning-based computer vision models to perform detection and segmentation tasks by addressing compression artefacts and improving coding efficiency through ROI-based compression. Two approaches were explored: improving models to better handle compressed images, by incorporating compressed images in the training process, and modifying codecs to preserve the quality of critical ROIs. For the second approach, non-rectangular ROI coding using binary masks is proposed for HEVC to efficiently handle a more selective quality preservation. Finally, a similar method was developed for VVC, where a ROI-based approach is proposed to further improve the coding efficiency in relation to its predecessor. The results of these methods will be discussed in Chapter 4.

# 4

# EXPERIMENTAL ASSESSMENT

In this chapter, the results obtained by applying the methodologies outlined in Chapter 3 are presented and discussed. First, it begins with a detailed description of the experimental setup for both the learning-based models and video codecs, to ensure the reproducibility of the study. Following this, the metrics used to evaluate the proposed methods are introduced and explained. Subsequently, an optimisation approach is proposed to select the most representative points for each ROI-based coding approach. The results of the study are then presented and analysed in detail, highlighting key findings and insights. Finally, an additional discussion is provided, introducing supplementary studies to enrich the main findings.

## 4.1 EXPERIMENTAL SETUP

### 4.1.1 *Learning-based Models*

From the pipelines defined in Section 3.1, four models were considered in this work for EM image analysis of mitochondria: two detection models, YOLOv4 and YOLOv8, and two segmentation models, YOLOv8 (segmentation) and SAM. For simplicity, YOLOv8 (segmentation) will be denoted as YOLOv8-Seg throughout the remainder of this dissertation.

The experimental methodology involves two distinct approaches to model preparation: training and fine-tuning. In the first approach, models were trained exclusively using the original, uncompressed training images of the respective dataset. These will be denoted as trained models. In the second approach, (pre-)trained models underwent additional fine-tuning using both uncompressed and compressed versions of the training datasets, with image coding applied at various QP levels, as detailed in Section 4.1.2. These will be referred to as fine-tuned models throughout the results analysis. It is important to note that when fine-tuning models for evaluation with a specific codec, such as HEVC or ROI-based HEVC, only images compressed with the standard version of the respective codec were used in the fine-tuning process. ROI-based coded images were not included in this stage.

To aim for optimal performance, the Optuna framework [123] was employed for hyperparameter optimisation, where 300 trials per model were conducted for both

the training and fine-tuning phases. All experiments were conducted on a NVIDIA GeForce RTX 3090 Graphics Processing Unit (GPU) with 24 GB of VRAM, which provided the computational capacity to execute the extensive hyperparameter trials. Throughout the following section, any hyperparameters presented as ranges indicate optimisation parameters defined within those bounds by Optuna.

For the YOLOv4 training and fine-tuning setup, the learning rate and batch size were optimised by Optuna. The learning rate range was determined based on an initial evaluation of the models to identify values that yielded effective training performance, resulting in a range of $[5 \times 10^{-5}, 1 \times 10^{-2}]$. The batch size range was set to [4,16], with the upper limit constrained by computational resources. The number of batches was set to 6000 for each training, with a burn-in phase of 1000 iterations to gradually increase the learning rate to the defined value. A learning rate decay policy was applied starting at the $4800^{\text{th}}$ batch, progressively reducing the learning rate until reaching zero. Additionally, anchor box sizes were pre-calculated for each dataset using K-Means clustering on the training ground truth bounding boxes to optimise model initialisation [124].

For the YOLOv8 detection and YOLOv8-Seg models, the pre-trained weights from the *yolo8x* and *yolo8x-seg* models were used, respectively. For both models, the learning rate was optimised within the range $\left[1 \times 10^{-5}, 1 \times 10^{-2}\right]$, gradually reducing it to zero after 300 epochs, with the batch size varying between [4,16]. Each training and fine-tuning process ran for a maximum of 1000 epochs, with an early stopping criterion based on a patience of 15 epochs.

The training approach for the SAM model follows a parameter-freezing methodology, focusing exclusively on optimising the mask decoder while maintaining the pre-trained weights of both the image encoder and prompt encoder. This strategy is primarily motivated by computational efficiency considerations, as training the complete architecture is highly resource-intensive. For reference, in the original work [91], the SAM architecture used 256 A100 GPUs for 68 hours during training. The model was initialised with the pre-trained SAM model using the ViT-Base architecture. The image encoder has already developed robust feature extraction capabilities through extensive pre-training, enabling it to effectively process visual information across diverse image domains. Similarly, the prompt encoder has been optimised to process various types of user inputs, such as bounding boxes and point prompts, transforming them into meaningful embedding representations. While fine-tuning these components could yield marginal improvements, the computational cost typically outweighs the benefits [14], [125].

Hence, only the mask decoder was trained, as it is responsible for synthesising information from both encoders to generate accurate segmentation masks tailored to the domain content, i.e., mitochondria segmentation. Similar to the previous models,

the learning rate was optimised within the range $\left[1 \times 10^{-8}, 1 \times 10^{-2}\right]$, with the final learning rate set to 0. Despite the reduction in the number of trainable parameters, a batch size of 1 was used during training due to computational constraints. The training process ran for a maximum of 10,000 epochs, with an early stopping mechanism employing a patience of 15 epochs to prevent overfitting. Additionally, a warm-up period of 250 steps was implemented to gradually increase the learning rate and stabilise the initial training phase.

### 4.1.2 *Image Compression*

The HEVC Test Model (HM) 18.0 and VVC Test Model (VTM) 23.4 were used for both the standard experiments and the adaptation for ROI-based encoding. To ensure reproducibility and consistency across experiments, the input images were encoded using the configuration files specified in the Common Test Conditions (CTC) for each codec [126], [127]. Specifically, *encoder_intra_main.cfg* and *encoder_intra_vtm.cfg* files were employed for HEVC and VVC coding, respectively. For all experiments, a wide range of QPs was used to test the learning-based models across different levels of coding degradation. For both encoders, the following set was used: $\mathrm{QP} = \{22, 27, 30, 32, 35, 37, 41, 43, 45, 47, 49, 51\}$. Additionally, for simplicity, compression using the non-ROI-based coding will be referred to as standard coding throughout the remainder of the document.

For the ROI-based coding experiments, the same configuration files were used, with the addition of the argument *MaxCuDQPDepth* set to 12 in VVC, previously described in Section 3.3.1. Since the maximum possible CTU size is $128 \times 128$ and the smallest CU size is $4 \times 4$, this results in 6 quaternary divisions, leading to the maximum value of 12. Although the configuration files used in this study defined the maximum CU size as $64 \times 64$, setting *MaxCuDQPDepth* to 12 remains valid and ensures the expected behaviour.

For ROI-based coding, two QP values must be defined for each experiment: one for the background ($\mathrm{QP_B}$) and one for the froeground ($\mathrm{QP_F}$). All possible combinations of the previously defined QP set were used, with the constraint that $\mathrm{QP_B}$ must always be greater than $\mathrm{QP_F}$, resulting in 66 combinations. The possible combinations are presented in Table 2. As discussed in Section 3.3.1, the maximum range for *CuQpDeltaVal* (the difference between $\mathrm{QP_F}$ and $\mathrm{QP_B}$) in VVC is $[-32, 31]$. Since the highest difference in the experiments is 29, i.e. $51 - 22$, no further modifications were required for the experiments to be executed.

Finally, for each combination of $\mathrm{QP_B}$ and $\mathrm{QP_F}$, two additional parameters were tested for both codecs:

Table 2: Possible combinations for $QP_F$ and $QP_B$.

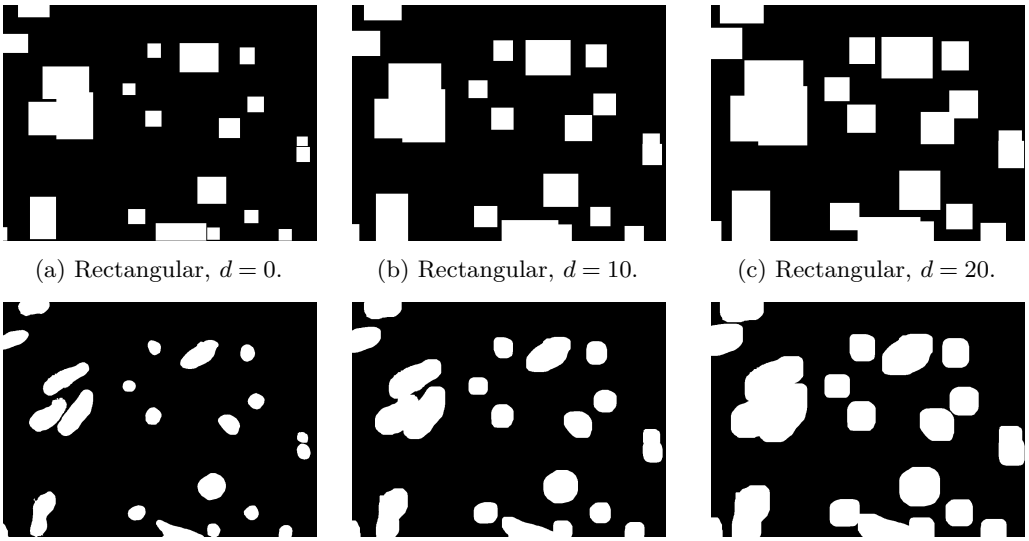
| Foreground QP \ Background QP | 22 | 27 | 30 | 32 | 35 | 37 | 41 | 43 | 45 | 47 | 49 | 51 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **22** | - | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **27** | - | - | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **30** | - | - | - | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **32** | - | - | - | - | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **35** | - | - | - | - | - | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **37** | - | - | - | - | - | - | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| **41** | - | - | - | - | - | - | - | ✓ | ✓ | ✓ | ✓ | ✓ |
| **43** | - | - | - | - | - | - | - | - | ✓ | ✓ | ✓ | ✓ |
| **45** | - | - | - | - | - | - | - | - | - | ✓ | ✓ | ✓ |
| **47** | - | - | - | - | - | - | - | - | - | - | ✓ | ✓ |
| **49** | - | - | - | - | - | - | - | - | - | - | - | ✓ |
| **51** | - | - | - | - | - | - | - | - | - | - | - | - |

1. **Mask Type:** The mask is either rectangular, if obtained from the trained detection model, or non-rectangular, if obtained from the trained segmentation model;

2. **Mask Dilation:** The morphological operation of dilation was applied to the masks using a $5 \times 5$ kernel, with 0, 5, 10, 15, and 20 iterations (0 accounts for no dilation, i.e. using the model's original mask). Each iteration expands the mask by one pixel on each side, resulting in five levels of mask expansion: 0, 5, 10, 15, and 20 pixels. This dilation process gradually increases the size of the regions of interest, allowing for an analysis of how mask expansion impacts coding efficiency and quality.

Figure 20 illustrates the influence of these parameters on the masks used to define the regions of interest.

Additionally, Figure 21 illustrates the process of ROI-based coding. Specifically, Figure 21a shows in green the predicted segmentation masks generated by the SAM-trained model on the Lucchi++ dataset. Both the image and the mask are input to the VVC ROI-based encoder, which assigns a QP of 22 to the predicted mask and a QP of 51 to the remaining regions; the resulting QP mapping is depicted in Figure 21b. The corresponding partitioning structure produced by VVC is represented in Figure 21c. Finally, the encoded image with ROI-based coding can be observed in Figure 21d, where the mitochondrial region is assigned significantly higher quality than the background. Appendix B showcases encoded images for the same setup, varying the mask type and dilation.



(a) Rectangular, $d = 0$. (b) Rectangular, $d = 10$. (c) Rectangular, $d = 20$.

(d) Non-rectangular, $d = 0$. (e) Non-rectangular, $d = 10$. (f) Non-rectangular, $d = 20$.

Figure 20: Comparison of rectangular and non-rectangular ROIs with different dilation levels. The top row and bottom rows represent rectangular and non-rectangular ROIs, respectively, with dilation values, $d$, increasing from left to right.

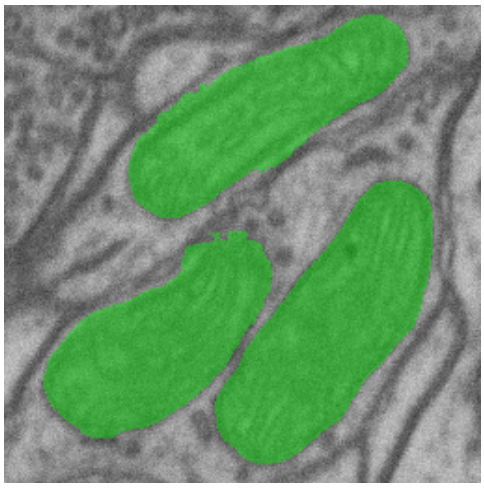
## 4.2 ASSESSMENT METRICS

### 4.2.1 *Detection and Segmentation Evaluation*

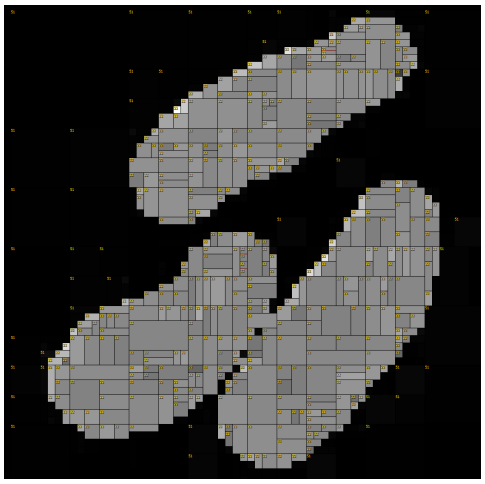
In detection tasks, the performance evaluation of a model focuses on its ability to accurately detect objects within a given scene. For the current work, this task involves only one class, as it is expected only to detect the location of mitochondria. To this extent, the model's performance can be analysed using metrics that categorise outcomes into True Positive (TP), False Positive (FP), and False Negative (FN). A TP occurs when the model correctly detects an object that exists, whereas a FP represents an incorrect detection, where the model predicts a bounding box with no corresponding ground truth region. An FN happens when the model fails to detect an object that exists accurately.

In the context of single-class detection tasks, the concept of True Negative (TN), which represents cases where the model correctly identifies the absence of an object, does not have a clear physical meaning. This is because the background or non-object regions are typically vast and undefined, making it impractical to quantify or evaluate TN. Therefore, the focus remains on TP, FP, and FN for detection tasks.

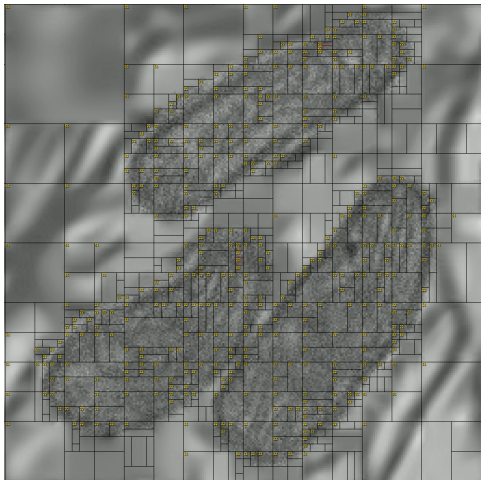
For detection tasks, a prediction is classified as a TP when its bounding box exhibits overlap with the corresponding ground truth annotation above a given threshold, quantified by the IoU metric [128]. The IoU evaluates localization accuracy by measuring the spatial agreement between a predicted region, $X$, and the ground truth counterpart, $Y$:



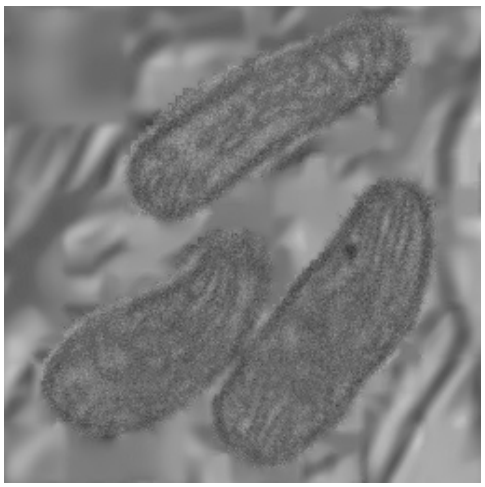
(a) ROI definition (non-rectangular) by a trained model (SAM).



(b) QP assignment: grey blocks represent regions coded with QP 22 (higher quality), while black blocks correspond to QP 51 (lower quality).



(c) Partition structure.



(d) Encoded image with ROI-based quality allocation.

Figure 21: Example of ROI-based coding with $\mathrm{QP_B} = 51$ and $\mathrm{QP_F} = 22$ for VVC.

$$\mathrm{IoU} = \frac{|X \cap Y|}{|X \cup Y|}, \tag{15}$$

where $|\cdot|$ represents the area in pixels of a given region. The metric's value can range from 0 to 1, with 0 indicating there is no overlap between bounding boxes, and 1 the perfect overlap. A common IoU threshold for considering a prediction as TP in the state-of-the-art is $\mathrm{IoU} \geq 0.5$ [129]. Mathematically, this value ensures that the intersection area is at least half of the union area, generally acknowledged as sufficient to ensure that the detection is both accurate and practically useful; hence this was the threshold adopted for the current work.

To improve the evaluation of a detection model's performance, several metrics derived from the TP, FP, and FN values are usually used. For instance, Recall, also

known as *sensitivity*, measures the model's ability to detect all objects in the image, defined as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{16}$$

A high recall indicates that the model misses few objects, making it particularly important in the current work, as it is expected to preserve the quality of as many mitochondria as possible. Precision measures the model's ability to avoid false detections and it is calculated as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{17}$$

where a high precision denotes most detected objects are correct, which is extremely useful for the current work to avoid coding false image regions with a quality higher than intended.

To fully evaluate the detection model's performance, the $\text{F}_1$ score is employed, defined as

$$F_1 = 2 \cdot \frac{\mathit{Precision} \cdot \mathit{Recall}}{\mathit{Precision} + \mathit{Recall}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \tag{18}$$

This metric combines both precision and recall into a single value using their harmonic mean, which inherently penalises extreme imbalances between the two metrics [130]. This is particularly important in biomedical imaging tasks, where false negatives (missing mitochondria, compromising diagnostic utility) and false positives (incorrect detections, wasting resources on irrelevant regions) carry significant consequences. To this end, the $\text{F}_1$ score ensures neither precision nor recall disproportionately dominates the evaluation, reflecting a balanced measure of the model's effectiveness.

For the segmentation task, however, the Dice-Sørensen coefficient (DSC) is used to evaluate its performance. The DSC measures the similarity between a predicted segmentation mask, $Y$, and the corresponding ground truth mask, $X$. It is mathematically represented as

$$\text{DSC} = 2 \cdot \frac{|X \cap Y|}{|X| + |Y|}. \tag{19}$$

DSC also ranges from 0 to 1, where 0 indicates there is no overlap between the predicted and ground truth segmentation masks, and 1 indicates perfect overlap.

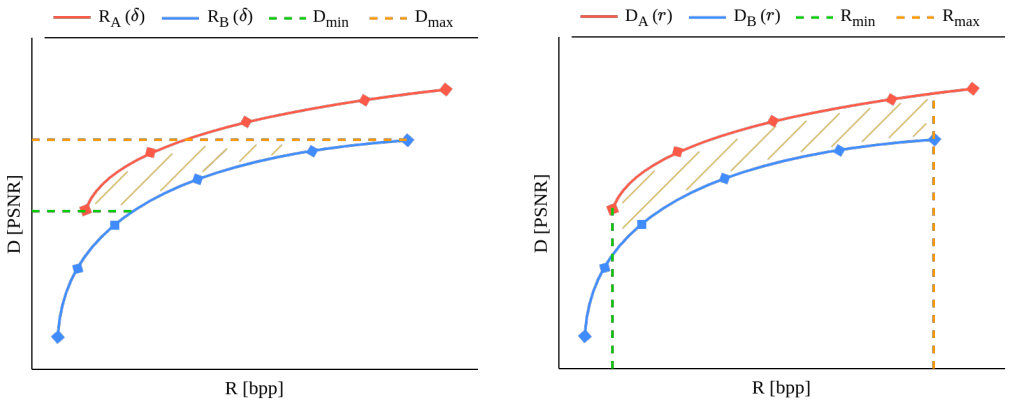
### 4.2.2 *Rate-Performance Evaluation*

When evaluating different compression algorithms across various bits per pixel (bpp) ranges, the use of Rate Distortion (RD) curves is extremely useful. It provides the trade-off between the rate and distortion characteristics obtained using different encoding configurations, e.g., QP.

Two metrics are widely used to compare the efficiency of two algorithms using the RD curves [131]. These metrics summarise the rate-distortion performance into a single, interpretable result: (1) the Bjøntegaard Delta (BD)-Rate and (2) the BD-PSNR. The BD-Rate, as illustrated in Figure 22a, measures the average difference in bitrate between the compression performance of two encoders for the same level of distortion. Mathematically, it can be defined as

$$\text{BD-Rate} = \frac{1}{D_{\max} - D_{\min}} \int_{D_{\min}}^{D_{\max}} [R_A(\delta) - R_B(\delta)] d\delta, \tag{20}$$

where $R_A(\delta)$ and $R_B(\delta)$ represent the bitrates of encoders $A$ and $B$ at a distortion $\delta$, respectively. The overlap between the two RD curves is illustrated using the dashed lines in Figure 22a, with $D_{\max}$ and $D_{\min}$ denoting the upper and lower distortion bounds, respectively. Note $D_{\min} = \max(\min D_A, \min D_B)$ and $D_{\max} = \min(\max D_A, \max D_B)$. $\delta$ represents the integration variable. A negative BD-Rate value indicates that encoder $B$ is more efficient than encoder $A$, as it requires a lower bitrate to achieve the same quality.




(a) BD-Rate. (b) BD-PSNR.

Figure 22: Rate-distortion performance comparison of two codecs using (a) BD-Rate and (b) BD-PSNR. The curves represent the distortion, $D$, in PSNR as a function of bitrate, $R$, with the anchor codec in blue and the compared codec in red.

Similarly, BD-PSNR, illustrated in Figure 22b, measures the average difference in quality in PSNR between two RD curves for the same bitrate and is defined as

$$\text{BD-PSNR} = \frac{1}{R_{\max} - R_{\min}} \int_{R_{\min}}^{R_{\max}} [D_A(r) - D_B(r)] dr, \tag{21}$$

where $D_A(r)$ and $D_B(r)$ represent the distortions of encoders $A$ and $B$ at a bitrate value $r$, respectively. A positive BD-PSNR value indicates that encoder $B$ provides better quality than encoder $A$ for the same bitrate, while a negative value suggests the opposite.

In the VCEG-M33 contribution [131], which first proposed the use of BD-Rate and BD-PSNR metrics to calculate rate and PSNR savings, a third-order (cubic) polynomial fit was employed to interpolate the RD curves [132]. This cubic fit was used to derive the expression for the integral of the curve, allowing the calculation of the BD-Rate and BD-PSNR metrics. However, subsequent work highlighted issues with this approach, reporting instability and unexpected results in the BD-Rate calculations when using this interpolation. It was later proposed the use of Piecewise Cubic Hermite Interpolation (PCHIP) [133] in replacement. While the cubic polynomial interpolation fits a single curve across all data points, leading to oscillations and overshooting, particularly in cases where RD curves exhibited non-smooth behaviour or sparse data points, the PCHIP calculates a piecewise cubic polynomial for each interval between data points. This approach ensures that the interpolation is locally monotonic, meaning it avoids oscillations while preserving the shape of the data, ensuring stability in the interpolation.

In addition to these metrics, BD-Rate and BD-PSNR, the authors in [134] incorporated perceptual quality metrics for 3D point clouds, such as 1-PCQM, to explore the performance of deep learning models on images degraded by compression. Similarly, in [135], introduced the BD-Top-1Accu metric to evaluate model efficiency using Rate-Performance curves. Building upon these methodologies, new BD-based metrics are proposed in this work, specifically designed for evaluating detection and segmentation algorithms under different compression conditions.

For detection models, the BD-$F_1$ metric is introduced to measure the average difference in $F_1$ score between two compression algorithms or two detection models across a range of bitrates:

$$\text{BD-F}_1 = \frac{1}{R_{\max} - R_{\min}} \int_{R_{\min}}^{R_{\max}} [F1_A(r) - F1_B(r)]dr, \tag{22}$$

where $F1_A$ and $F1_B$ represent the scores achieved by detection models when processing images compressed by algorithms $A$ and $B$, respectively. Similarly, for the segmentation task, where the DSC metric is used for evaluation, the BD-DSC metric is proposed and defined as

$$\text{BD-DSC} = \frac{1}{R_{\max} - R_{\min}} \int_{R_{\min}}^{R_{\max}} [\text{DSC}_A(r) - \text{DSC}_B(r)]dr, \tag{23}$$

where $\text{DSC}_A$ and $\text{DSC}_B$ represent the segmentation performance obtained when using compression algorithms $A$ and $B$. For each task, the BD-Rate can also be

calculated using the Equation (20), where the distortion parameter corresponds to either the $F_1$ score for detection or DSC for segmentation.

## 4.3 OPTIMAL POINT SELECTION FOR ROI-BASED CODING

As mentioned in Section 4.1.2, several parameters were introduced during the coding of the ROI-based approaches. These include foreground and background QPs, mask types, and dilation levels, which significantly influence the rate-performance trade-off in ROI-based coding. When coding all possible combinations of these parameters, a wide range of rate-performance results is obtained, as illustrated in Figure 23. A key challenge arises when analysing these results: to determine which parameters are optimal and which points best represent the curve for the ROI-based coding in the context of a specific codec and computer vision model.

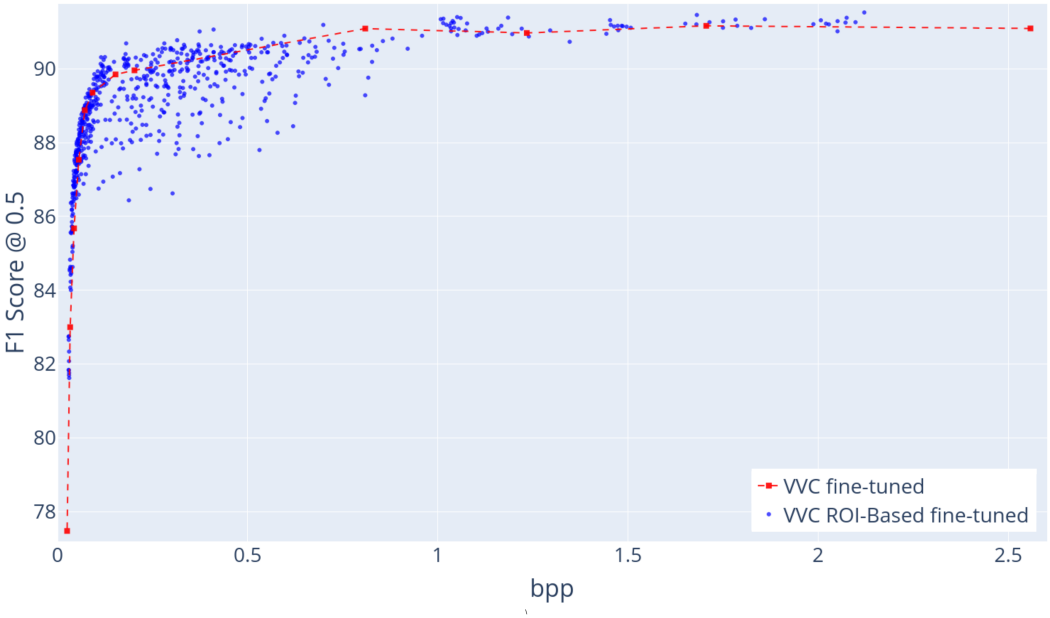



Figure 23: Performance comparison of the YOLOv8 fine-tuned model for images coded with two different approaches: standard VVC encoding (red dashed line) and ROI-based VVC encoding (blue scatter points). The ROI-based approach showcases varied performance across different combinations of foreground/background QP, mask types, and dilation levels.

To address this challenge, an optimisation mechanism based on the Pareto frontier algorithm [136] was proposed to select the best points for each curve. The Pareto frontier, also known as the Pareto optimal set, consists of points where no single parameter combination can improve one objective (e.g., compression rate) without worsening another (e.g., model performance). The proposed method, detailed in Algorithm 1, systematically evaluates the rate-performance trade-offs for each parameter combination and identifies the Pareto-optimal points. These points are

then used to represent the best possible performance of the ROI-based coding approach for a given codec and computer vision model.

**Algorithm 1** Pareto Frontier Optimisation

**Input:** Set of points $P = \{(x_i, y_i) \mid i \in 1, \ldots, n \wedge x_i, y_i \in \mathbb{R}^+\}$
**Output:** Sequence of Pareto-optimal points $S$

$i_{\min} \leftarrow \arg\min_i x_i$
$S \leftarrow (x_{i_{\min}}, y_{i_{\min}})$
$P \leftarrow P \backslash S$ ▷ Remove selected point from $P$
**while** $P \neq \varnothing$ **do**
    $(x_j, y_j) \leftarrow$ last point in $S$
    $V \leftarrow (x, y) \in P \mid x > x_j \wedge y > y_j$ ▷ Get Valid candidates
    **if** V$= \varnothing$ **then**
        **break**
    **end if**
    $\theta(x, y) \leftarrow \arctan\left(\frac{y - y_j}{x - x_j}\right)$
    $(x', y') \leftarrow \arg\max_{(x,y)} \theta(x, y)$ ▷ Candidate with the largest angle
    $S \leftarrow S \cup (x', y')$
    $P \leftarrow V \setminus (x', y')$ ▷ Set of Valid candidates excluding the chosen one
**end while**
**return** $S$

The algorithm begins by selecting the point with the smallest bpp value, initialising the sequence of Pareto-optimal points. This ensures that the curve starts with the most efficient point in terms of compression. The algorithm then discards all points that do not contain higher values in both objectives compared to the last point in the sequence, i.e. the first selected point. From the remaining valid candidates, the algorithm selects the point that, alongside with the last selected optimal point, forms the line with the highest slope. This prioritises points that offer the most significant performance gain for the smallest increase in compression rate, ensuring that the algorithm captures the most efficient trade-offs.

The selected point is added to the sequence, and the process repeats until no more valid candidates are available. This ensures that the curve ends with the point that corresponds to the highest achievable performance for the current model. From the set of points illustrated in Figure 23, the optimised curve after the proposed Pareto Frontier optimisation is illustrated in Figure 24.

## 4.4 RESULTS

### 4.4.1 *Performance evaluation*

Before evaluating the impact of compression on the detection and segmentation models' performance, it is essential to establish a baseline, by assessing the perfor-

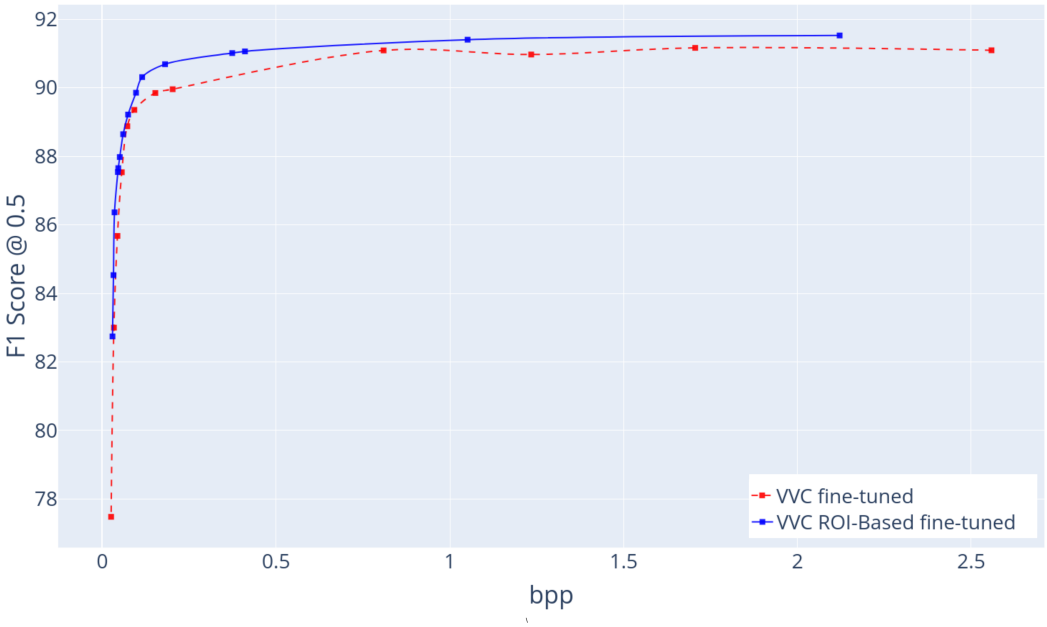



Figure 24: Performance comparison of the YOLOv8 fine-tuned model for images coded with two different approaches: standard VVC encoding (red dashed line) and ROI-based VVC encoding (blue solid line). Data points for the ROI-based approach were selected using the Pareto Frontier Optimisation criteria.

mance of the detection and segmentation models using the uncompressed images. Table 3 compares the detection performance of YOLOv4 and YOLOv8 using the $F_1$ score on the Lucchi++ and Kasthuri++ test datasets, previously detailed in Section 2.1.2. These results demonstrate that both models achieve strong detection and segmentation performance on uncompressed images, with minor variations depending on the dataset. For detection, YOLOv8 achieves a slightly higher $F_1$ score of 91.79% on the Lucchi++ dataset, outperforming YOLOv4 (91.67%). However, on the Kasthuri++ dataset, YOLOv4 performs marginally better, with a $F_1$ score of 91.14% compared to YOLOv8's 90.47%.

Table 3: Comparison of detection performance using $F_1$ score (%) between YOLOv4 and YOLOv8 on the original Lucchi++ and Kasthuri++ datasets.

| | **$F_1$ Score (%)** | |
|---|---|---|
| **Model** | **Lucchi++** | **Kasthuri++** |
| **YOLOv4** | 91.67 | **91.14** |
| **YOLOv8** | **91.79** | 90.47 |

Table 4 shows the comparison, in terms of segmentation, between the performance of YOLOv8-Seg and SAM (using the trained YOLOv8 model's detection outputs as prompts) using the DSC metric. SAM outperforms YOLOv8-Seg on both datasets, achieving a DSC of 90.85% on Lucchi++ and 93.24% on Kasthuri++, compared to YOLOv8-Seg's 89.07% and 90.43%, respectively. It is worth noticing that, while YOLOv8-Seg must perform both detection and pixel-to-pixel segmentation of

mitochondria, SAM benefits from the localisation provided by the YOLOv8 detection model. This separation of tasks allows SAM to focus solely on improving pixel-to-pixel segmentation performance, leading to higher accuracy.

Table 4: Comparison of segmentation performance using the DSC metric between YOLOv8-Seg and SAM (with YOLOv8 detection outputs as prompts) on the original Lucchi++ and Kasthuri++ datasets.

| | **DSC (%)** | |
|---|---|---|
| **Model** | **Lucchi++** | **Kasthuri++** |
| **YOLOv8-Seg** | 89.07 | 90.43 |
| **SAM** | **90.85** | **93.24** |

With the baseline performance established, the study now turns to evaluating the impact of lossy compression using standard codecs, namely HEVC and VVC, on the previous models' performance. The impact of compression on the model performance is further evaluated by comparing the detection and segmentation metrics for models trained on uncompressed images and fine-tuned with compressed images. Table 5 presents the detection performance of YOLOv4 and YOLOv8 using the BD-Rate and BD-$F_1$, while Table 6 shows the segmentation performance of YOLOv8-Seg and SAM using the BD-Rate and BD-DSC metric. Both tables include results for models fine-tuned on HEVC- and VVC-compressed images, evaluated across the tested set of QP values.

Table 5: BD-Rate ↓ and BD-$F_1$ ↑ performance metrics for trained and fine-tuned detection models on images coded with standard HEVC and VVC on the Lucchi++ and Kasthuri++ datasets. Metrics use as baseline the trained YOLOv8 model evaluated for HEVC-coded images.

| | | | **Lucchi++** | | **Kasthuri++** | |
|---|---|---|---|---|---|---|
| **Model** | **Fine-tuned** | **Encoder** | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** |
| **YOLOv4** | - | **HEVC** | 7.46 | -2.34 | 20.17 | -0.35 |
| | - | **VVC** | -3.32 | 1.64 | 25.62 | -0.39 |
| | ✓ | **HEVC** | -40.19 | 10.40 | 0.63 | -0.07 |
| | ✓ | **VVC** | -54.97 | 13.51 | 15.34 | -0.24 |
| **YOLOv8** | - | **HEVC** | - | - | - | - |
| | - | **VVC** | -4.40 | 1.00 | -40.74 | 0.48 |
| | ✓ | **HEVC** | -47.82 | 11.34 | -38.16 | 0.85 |
| | ✓ | **VVC** | **-66.77** | **14.46** | **-84.89** | **0.95** |

From the obtained results, two key conclusions can be drawn: the use of the VVC codec generally improves the performance of learning-based models when compared to HEVC; fine-tuning models with compressed images leads to a significant increase in performance. While VVC-compressed images lead to higher performance, this improvement is less pronounced when comparing the trained models using standard codecs. For instance, in the Lucchi++ dataset, using the YOLOv8 model,

VVC achieves an average performance increase of 1.00% for the same bpp and an average bpp reduction of 4.40% for the same performance in the detection task compared to HEVC. This limited improvement may stem from the fact that both codecs are optimised for objective quality metrics, such as SSE and PSNR, and the models are not specifically trained to handle the degradation introduced by these codecs. Nevertheless, the results indicate that VVC introduces less severe quality degradation than HEVC at equivalent bitrates, which translates to better preservation of features relevant to detection and segmentation tasks, evidenced by the consistent improvements across datasets.

Subsequently, by adapting the model to the degradation caused by the codecs (fine-tuning), the models become more robust and achieve better results on compressed data. By fine-tuning the YOLOv8 detection model on HEVC-compressed images results in an 11.34% increase in $F_1$ score and a 47.82% reduction in bpp. Similarly, YOLOv4 achieves an average $F_1$ score increase of 13.51% when fine-tuned on VVC-compressed images, compared to 1.64% without fine-tuning. These improvements highlight the importance of domain-specific fine-tuning to mitigate the impact of compression artefacts.

Conversely, by using the Kasthuri++ dataset, the improvements are less pronounced but still significant. YOLOv8 fine-tuned on VVC-compressed images achieves an average $F_1$ score increase of 2.52%, compared to 0.48% without fine-tuning. However, YOLOv4 performs poorly on this dataset. For instance, using the fine-tuned YOLOv4 model with VVC-compressed images, reduces the model's performance by 0.24%, while requiring, on average, a 15.34% increase in file size to achieve the same performance as YOLOv8's trained model evaluated on HEVC-compressed images.

Table 6: BD-Rate ↓ and BD-DSC ↑ performance metrics for trained and fine-tuned segmentation models on images coded with standard HEVC and VVC on the Lucchi++ and Kasthuri++ datasets. Metrics use as baseline the trained SAM model evaluated for HEVC-coded images.

| | | | **Lucchi++** | | **Kasthuri++** | |
|---|---|---|---|---|---|---|
| **Model** | **Fine-tuned** | **Encoder** | **↓BD-Rate (%)** | **↑BD-DSC (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **YOLOv8-Seg** | - | **HEVC** | -3.57 | 0.23 | 70.61 | -7.86 |
| | - | **VVC** | -4.13 | 0.74 | 64.13 | -7.49 |
| | ✓ | **HEVC** | -27.92 | 5.21 | 32.88 | -2.64 |
| | ✓ | **VVC** | -26.74 | 4.98 | 14.62 | -1.14 |
| **SAM** | - | **HEVC** | - | - | - | - |
| | - | **VVC** | -6.48 | 1.67 | -29.37 | 0.59 |
| | ✓ | **HEVC** | -40.37 | 8.20 | -38.97 | 0.93 |
| | ✓ | **VVC** | **-73.96** | **14.76** | **-47.90** | **0.97** |

For segmentation tasks, fine-tuning on VVC-coded images also leads to substantial performance gains. By using the Lucchi++ dataset, SAM fine-tuned on VVC-compressed images achieves an average DSC increase of 14.76% and a 73.96%

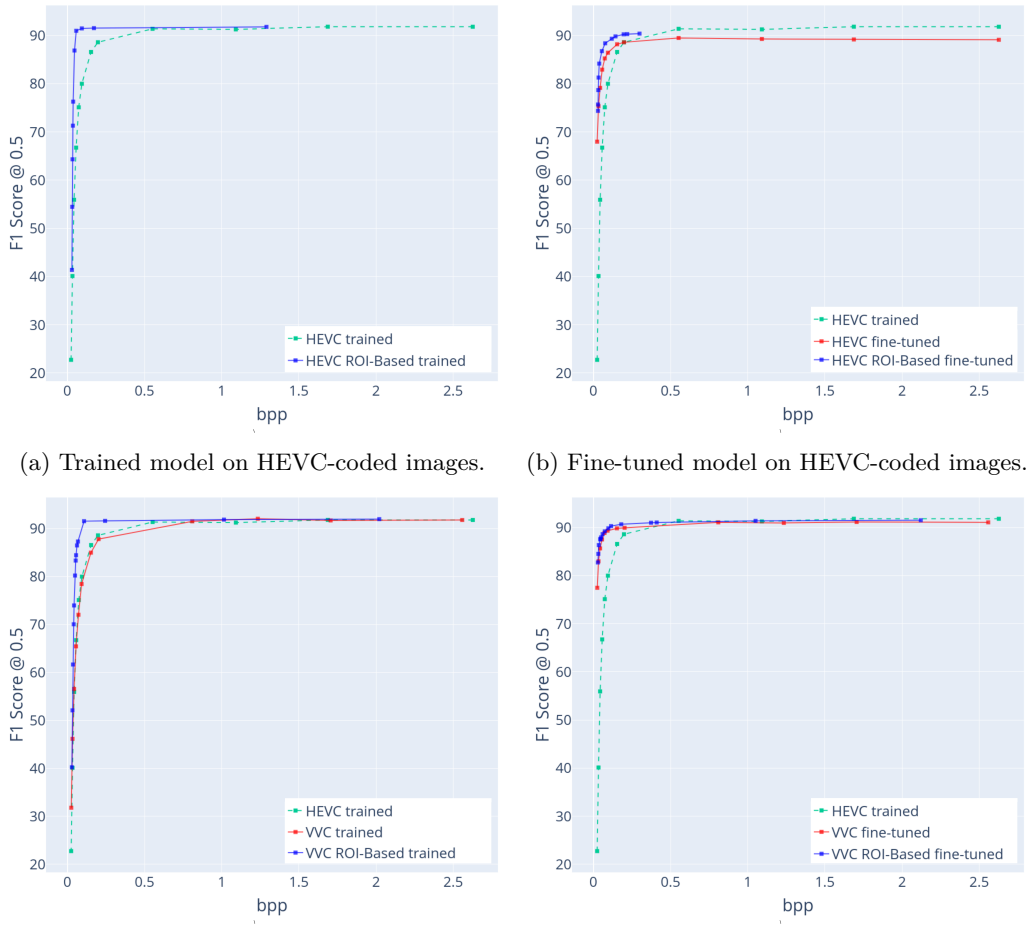
reduction in file size for the same performance, compared to a 1.67% increase and 6.48% reduction, respectively, without fine-tuning. Similarly, YOLOv8-Seg fine-tuned on HEVC-compressed images improves its DSC from 0.23% to 5.21%. These results underscore the effectiveness of fine-tuning in enhancing segmentation performance on compressed images.

When the Kasthuri++ dataset is used, SAM fine-tuned on VVC-compressed images achieves the highest average DSC increase of 0.97%, slightly outperforming the model fine-tuned on HEVC-compressed images (0.94%). In contrast, YOLOv8-Seg performs poorly, with the highest result being an average DSC loss of 1.14%. Additionally, to achieve the same performance as SAM on HEVC-encoded images, YOLOv8-Seg requires, on average, a 14.62% increase in file size. These findings highlight the variability in model performance across datasets and emphasize the importance of selecting appropriate models and fine-tuning strategies for specific tasks.

### 4.4.2 *Performance evaluation of ROI-based coding*

Given the extensive number of combinations involved in the evaluation of ROI-based coding, the study focused on two key models: YOLOv8 for detection and SAM for segmentation, using the bounding boxes detected by YOLOv8 as prompts. As discussed in Section 4.3, the proposed Pareto Frontier optimisation algorithm was applied to each combination of computer vision model and ROI-based encoder to reduce the number of points per curve, ensuring the resulting points achieve the best rate-performance trade-off. The curves generated by this optimisation are illustrated in Figure 25 for the detection task on the Lucchi++ dataset, while the curves for the remaining tasks and datasets are presented in Appendix C. For simplicity, each chart presents the curves for the same model, whether trained or fine-tuned, and compares the performance between standard coding and the respective ROI-based approach. In Appendix D, the selected points' parameters, alongside the respective model's performance and compression rate, are defined.

Initial tests employed a third-order polynomial interpolation method, described in Section 4.2, which led to inconsistencies in the results. This behaviour is detailed in Appendix E. However, by using the PCHIP interpolation, the results are refined, as shown in Tables 7 and 8. These tables compare detection and segmentation models, both trained and fine-tuned, on compressed images through standard coding, as well as their corresponding ROI-based versions, for both datasets. All results are obtained using as reference the baseline curves from standard HEVC coding applied to models trained on uncompressed images.




(a) Trained model on HEVC-coded images.

(b) Fine-tuned model on HEVC-coded images.

(c) Trained model on VVC-coded images.

(d) Fine-tuned model on VVC-coded images.

Figure 25: Detection performance comparison between Lucchi++ images coded using standard coding (red line) and ROI-based coding (blue line) against the baseline (green line; trained model on standard HEVC-coded images). The top row shows results for the HEVC codec, while the bottom row corresponds to the VVC codec. The left column represents trained models, and the right column represents fine-tuned models.

From the obtained results, it can be seen that ROI-based coding consistently delivers superior results compared to standard coding. For instance, on the Lucchi++ dataset, the fine-tuned on VVC-compressed images segmentation model applied to ROI-based coded images using VVC outperforms the same model applied to standard VVC-coded images. Specifically, the average DSC increases from 14.76% to 16.26%, while the BD-Rate reduction improves from 73.96% to 84.34%. This demonstrates that ROI-based coding not only preserves critical information in regions of interest but also enhances the overall efficiency of the coding process. Similar trends are observed on the Kasthuri++ dataset, where ROI-based VVC achieves a 2.52% increase in $F_1$ score and a 93.87% reduction in bpp for the fine-tuned detection model, further validating the effectiveness of this approach.

Despite the optimisation algorithm selecting the most representative parameters for each curve, it does not guarantee that the chosen parameters encompass the entire bpp range. As previously discussed, the algorithm ensures that the curve

Table 7: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for trained and fine-tuned models using standard and ROI-based HEVC and VVC codecs with the Lucchi++ dataset. All metrics use as baseline the trained model with standard HEVC encoding.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| | | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Trained Model** | **HEVC** | - | - | - | - |
| | **VVC** | -4.40 | 1.00 | -6.48 | 1.67 |
| | **HEVC-ROI** | -42.14 | 12.17 | -41.84 | 11.03 |
| | **VVC-ROI** | -34.34 | 8.49 | -33.70 | 7.06 |
| **Fine-tuned Model** | **HEVC** | -47.82 | 11.34 | -40.37 | 8.20 |
| | **VVC** | -66.77 | 14.46 | -73.96 | 14.76 |
| | **HEVC-ROI** | -64.57 | **21.72** | -54.25 | 9.97 |
| | **VVC-ROI** | **-73.44** | 12.63 | **-84.34** | **16.26** |

Table 8: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for trained and fine-tuned models using standard and ROI-based HEVC and VVC codecs with the Kasthuri++ dataset. All metrics use as baseline the trained model with standard HEVC encoding.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| | | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Trained Model** | **HEVC** | - | - | - | - |
| | **VVC** | -40.74 | 0.48 | -29.37 | 0.59 |
| | **HEVC-ROI** | -51.94 | 1.23 | -32.22 | 0.97 |
| | **VVC-ROI** | -73.02 | 1.68 | -49.85 | **1.31** |
| **Fine-tuned Model** | **HEVC** | -38.16 | 0.85 | -38.97 | 0.93 |
| | **VVC** | -84.89 | 0.95 | -47.90 | 0.97 |
| | **HEVC-ROI** | -87.02 | 1.60 | -53.41 | 1.02 |
| | **VVC-ROI** | **-93.87** | **2.52** | **-59.42** | 1.21 |

ends at the highest performance value, but this can lead to inconsistencies in the results. For instance, as shown in Figure 26, the fine-tuned model achieves the highest performance and compression efficiency when using ROI-based coding with the VVC codec compared to ROI-based HEVC. However, the selected points for the HEVC version cover a much narrower bpp range than those for VVC. Since the BD-$F_1$ metric evaluates the average increase in performance across the bpp range, the narrower range for HEVC results in a higher average improvement of 21.72% compared to 12.63% for VVC, as previously seen in Table 7, even if VVC performs better.

To address these discrepancies, an additional study was conducted by restricting the integration limits to the smallest bpp range shared by all curves. This ensures a more fair comparison across all codecs and models. The results of this analysis are presented in Tables 9 and 10.

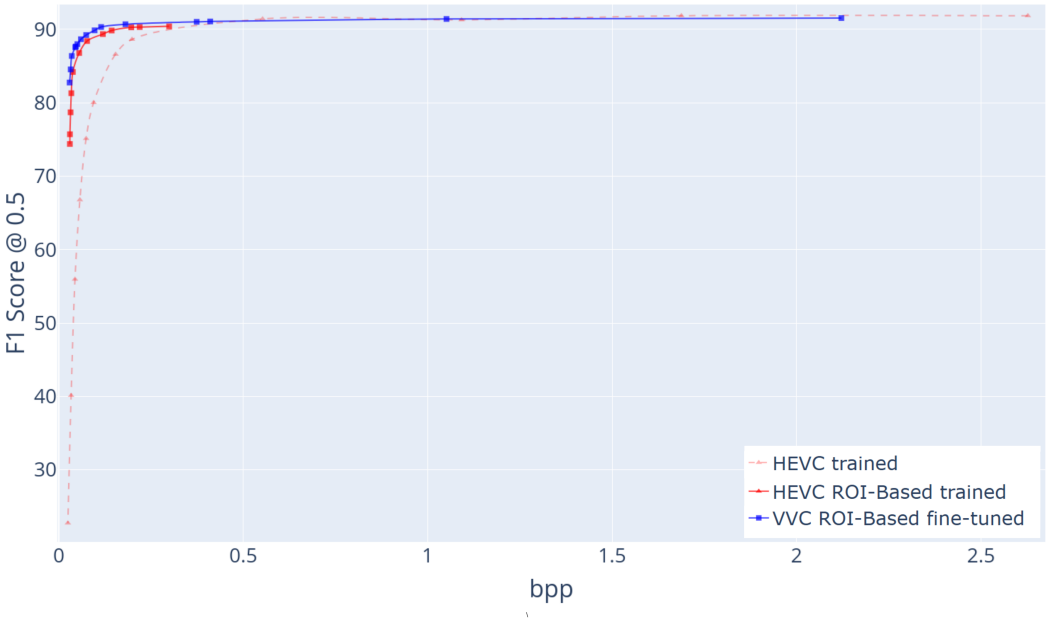



Figure 26: Performance comparison of the YOLOv8 fine-tuned model for images encoded using ROI-based HEVC (red solid line) and ROI-based VVC (blue solid line). The baseline performance of the detection model trained on uncompressed images and evaluated on HEVC-coded images is shown as a red dashed line.

For the Lucchi++ dataset, as shown in Table 9, the fine-tuned model using ROI-based VVC achieves the highest performance improvements, with a 23.70% increase in $F_1$ score and a 74.96% reduction in bpp for detection, and a 16.39% increase in DSC with a 78.38% reduction in bpp for segmentation. These results confirm that ROI-based VVC outperforms ROI-based HEVC when the integration range is constrained, aligning with the observations in Figure 26. Similarly, for the Kasthuri++ dataset, as shown in Table 10, ROI-based VVC achieves the best results, with a 2.23% increase in $F_1$ score and a 93.87% reduction in bpp for detection, and a 1.21% increase in DSC with a 59.42% reduction in bpp for segmentation.

These findings highlight the importance of carefully selecting the integration range when comparing codecs and models. By limiting the range to the smallest shared bpp interval, the results provide a more accurate representation of the relative performance of each approach. This adjustment ensures that the metrics reflect the true trade-offs between compression efficiency and model performance, avoiding biases introduced by uneven bpp ranges.

## 4.5 QUALITY ASSESSMENT AND PARAMETER ANALYSIS

To further exploit the efficiency of ROI-based coding, two additional studies were conducted. On the first one, the preservation of mitochondrial information was evaluated using an objective quality metric, specifically PSNR. The second builds on

Table 9: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for trained and fine-tuned models using standard and ROI-based HEVC and VVC codecs on the Lucchi++ dataset. All metrics use as baseline the trained model with standard HEVC encoding.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| | | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Trained Model** | **HEVC** | - | - | - | - |
| | **VVC** | -4.89 | 0.45 | -2.61 | 1.01 |
| | **HEVC-ROI** | -73.50 | 20.24 | -67.55 | 12.01 |
| | **VVC-ROI** | -63.14 | 15.37 | -55.77 | 8.65 |
| **Fine-tuned Model** | **HEVC** | -10.62 | 17.32 | -22.34 | 7.40 |
| | **VVC** | -67.67 | 22.36 | -72.41 | 14.88 |
| | **HEVC-ROI** | -65.78 | 21.72 | -28.72 | 9.97 |
| | **VVC-ROI** | **-74.96** | **23.70** | **-78.38** | **16.39** |

Table 10: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for trained and fine-tuned models using standard and ROI-based HEVC and VVC codecs on the Kasthuri++ dataset. All metrics use as baseline the trained model with standard HEVC encoding.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| | | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Trained Model** | **HEVC** | - | - | - | - |
| | **VVC** | -47.80 | 1.36 | -25.67 | 0.65 |
| | **HEVC-ROI** | -91.02 | 0.65 | -45.44 | 0.97 |
| | **VVC-ROI** | -91.70 | 1.68 | -53.72 | 1.14 |
| **Fine-tuned Model** | **HEVC** | -79.91 | 1.82 | -34.50 | 0.87 |
| | **VVC** | -90.30 | 1.79 | -47.09 | 0.94 |
| | **HEVC-ROI** | -92.60 | 2.02 | -54.39 | 1.20 |
| | **VVC-ROI** | **-93.87** | **2.23** | **-59.42** | **1.21** |

the analysis presented in Section 4.4.2, addressing the influence of coding parameters in ROI-based coding, particularly the effects of mask type and mask dilation.

### 4.5.1 *ROI quality reduction on ROI-based coding*

As previously discussed, the ROI-based approach improves the performance of computer vision models compared to standard codecs, while significantly enhancing compression efficiency. However, despite assigning a smaller QP (higher quality) to the region of interest, it remains unclear whether the mitochondrial information is preserved to the same extent as in standard coding. To address this question, an evaluation of the quality of the coded regions of interest, in comparison to standard coding, was conducted.

The results are illustrated in Figure 27 for the Lucchi++ test dataset, comparing the standard coding, using a QP of 22, with ROI-based coding, where $QP_F = 22$

and $QP_B$ varies. To calculate the PSNR exclusively within the defined mitochondrial regions, the metric was initially computed using the ROIs provided by the segmentation model. An example of this kind of region is illustrated in Figure 28, represented by the green. The bounding box prompt fed to SAM to generate the segmented mask is defined by the blue, rectangular mask in Figure 28, with the cyan mask representing the overlap between the bounding box and the predicted segmented mask.

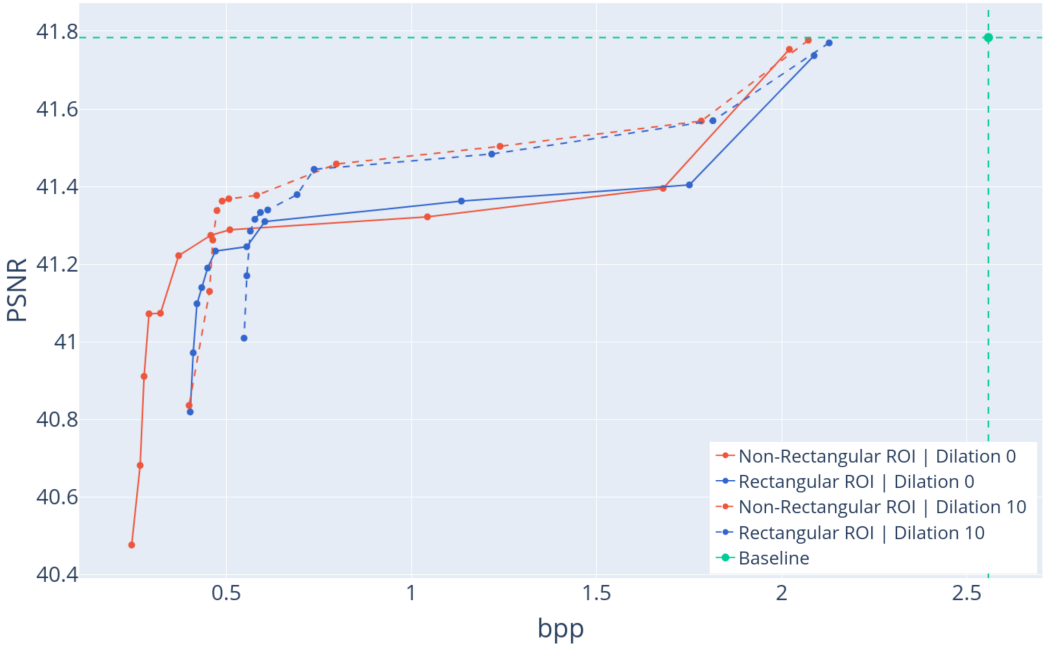



Figure 27: RD curves for the defined ROIs using VVC coding, with a fixed foreground QP of 22 and varying background QPs. The green curve represents standard VVC coding with a QP of 22. Blue curves correspond to ROI-based coding with rectangular regions, while red curves represent non-rectangular regions. Solid lines indicate ROI-based coding with no mask dilation, and dashed curves denote a dilation of 10 pixels.

As shown in Figure 28, the SAM model occasionally includes a few pixels outside the specified bounding box prompt. This means that, when calculating the PSNR for images coded using rectangular regions, some pixels coded with the background QP might inadvertently be included, potentially skewing the results. To address this issue, an intersection between the rectangular and non-rectangular masks (with no dilation), represented by the cyan region in Figure 28, was used for the PSNR calculation. This ensures that only the true regions of interest are considered, providing a fair and consistent comparison across all experiments.

The baseline standard coding, represented by the dashed green lines, achieves an average of 41.78 dB of PSNR and 2.56 bpp for the defined ROIs. From the curves, it can be analysed that increasing $\Delta$QP (difference between foreground and background QPs) leads to a slight decrease in mitochondrial quality. For instance, compared to the baseline, non-rectangular coding without dilation (red solid line)

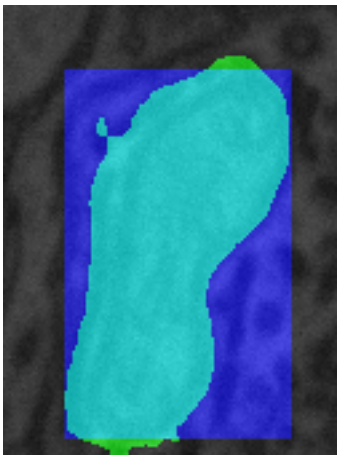


Figure 28: Comparison between predicted masks: the blue region represents the bounding box detection from YOLOv8, while the green mask shows the segmentation output from SAM. The cyan region highlights the overlap between both predictions.

shows that using a $QP_B = 27$ reduces the PSNR by 0.03 dB, while a $QP_B = 51$ results in a larger reduction of 1.31 dB.

This quality decrease can be explained by the encoder's optimisation process as a trade-off, as previously discussed in Section 3.3.2. In the baseline scenario, the ROI's neighbouring regions are coded with the same quality, i.e. QP 22. However, in ROI-based coding, where higher QPs are assigned to the neighbouring regions of the ROIs, the encoder faces poorer predictions and must consequently encode more residual information. This trade-off results in these regions requiring slightly higher bitrates compared to the same regions in the baseline coding, accompanied by a modest decrease in quality. Nevertheless, it's important to emphasise that while this quality decrease exists, it remains minimal relative to the substantial bitrate reductions achieved. The approach enables significant compression gains, achieving up to 90.44% reduction in bpp when comparing standard coding to coding with a $QP_B$ of 51, with a modest reduction of 1.31 dB in quality.

The use of rectangular masks in ROI-based coding, compared to non-rectangular regions, results in a subtle improvement in mitochondrial information preservation at the cost of reduced compression efficiency. In the former, regions close to the vicinities of the mitochondria are compressed with a lower quality, whereas in rectangular coding some non-mitochondrial regions are compressed with a lower QP, which creates higher-quality neighbouring regions of the mitochondria. These regions can then be used by the encoder for intra-prediction when coding the mitochondrial information, leading to an overall increase in the quality of the mitochondrial data compared to non-rectangular coding. However, this approach also increases the coding rate, as a larger area is coded at higher quality. For instance, when compared to non-rectangular coding with a $QP_B = 51$, rectangular coding for the same

$QP_B$achieves a 0.34 dB increase in quality but reduces the bpp by only 84.26% relative to the standard coding.

The use of dilation exhibits a similar behaviour. As the dilation increases, the quality of the mitochondrial information improves, approximating to the baseline. This is due to the larger area of neighbouring regions being coded with lower coding distortion, which improves intra-prediction for the mitochondrial regions. However, this also results in a reduction in bpp savings compared to using masks without dilation.

### 4.5.2 *Influence of ROI-based coding parameters*

The effectiveness of ROI-based coding depends heavily on the choice of mask type and the level of dilation applied to the binary mask. These parameters directly influence both compression efficiency and the preservation of critical features for machine vision tasks. To evaluate their impact, a study was conducted using models trained on uncompressed images from the Lucchi++ dataset, evaluating their coded versions with the ROI-based HEVC approach. The results of this study are summarised in Table 11. The evaluation compares two types of masks—rectangular and non-rectangular—as well as different dilation levels (ranging from 0 to 20 pixels), for both detection and segmentation tasks. The remainder of the results, including evaluations for other models, datasets, and codecs, are presented in Appendix F.

Table 11: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for trained detection and segmentation models using the ROI-based HEVC codec on the Lucchi++ dataset. Results are evaluated for varying mask types (rectangular and non-rectangular) and dilation levels (in pixels) of the binary mask. All metrics are compared against the baseline model trained with standard HEVC encoding.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| **Mask Type** | **Dilation** | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Rectangular** | **0** | -38.87 | 10.09 | -38.72 | 9.91 |
| | **5** | -38.27 | 9.74 | -37.62 | 10.55 |
| | **10** | -37.29 | 9.71 | -37.08 | 8.07 |
| | **15** | -36.51 | 9.42 | -36.83 | 9.46 |
| | **20** | -33.15 | 7.61 | -33.33 | 7.94 |
| **Non-rectangular** | **0** | **-42.09** | **12.31** | **-41.15** | **10.82** |
| | **5** | -39.34 | 10.26 | -36.29 | 8.69 |
| | **10** | -38.43 | 10.48 | -37.58 | 10.38 |
| | **15** | -35.70 | 8.59 | -36.09 | 8.99 |
| | **20** | -35.58 | 9.29 | -35.51 | 8.36 |

The choice between rectangular and non-rectangular masks significantly affects the performance of ROI-based coding. Non-rectangular masks consistently outperform rectangular regions in terms of both compression efficiency and model performance.

For instance, with a dilation of 0 pixels, i.e. no dilation, non-rectangular masks achieve a BD-Rate reduction of 42.09% and a BD-$F_1$ improvement of 12.31% for detection, compared to 38.87% and 10.09%, respectively, for rectangular masks. Similarly, for segmentation, non-rectangular masks yield a BD-Rate reduction of 41.15% and a BD-DSC improvement of 10.82%, outperforming rectangular masks (38.72% and 9.91%). This superior performance can be attributed to the precise delineation of mitochondrial regions by non-rectangular masks, which minimises the inclusion of irrelevant background areas with higher-quality while ensuring the preservation of critical features.

Dilation of the binary mask also plays a crucial role in ROI-based coding. Increasing the dilation level, for both mask types, generally leads to lower compression efficiency and model's performance. For example, with non-rectangular masks, increasing dilation from 0 to 20 pixels reduces the BD-Rate savings from 42.09% to 35.58% for detection and from 41.15% to 35.51% for segmentation. Similarly, the BD-$F_1$ improvement decreases from 12.31% to 9.29% for detection, and the BD-DSC improvement drops from 10.82% to 8.36% for segmentation. This trend occurs since dilation expands the region coded with a lower QP, increasing the overall bitrate while marginally improving the model's performance. To this end, the results suggest that non-rectangular masks without dilation (0 pixels) provide the best balance between compression efficiency and model performance.

## 4.6 SUMMARY

In this chapter, the experimental setup and results obtained from the methodologies described in Chapter 3 were presented and discussed. First, the configuration pipeline, including the detection and segmentation models as well as the codecs, was described. Then, the evaluation metrics were defined, including the proposed metrics for assessing rate-performance trade-offs. Additionally, an optimisation approach based on the Pareto frontier was introduced to iteratively select the most representative set of points for each ROI-based coding. The experimental results demonstrated that combining ROI-based coding with model fine-tuning effectively reduced file size while maintaining model performance. Finally, further analyses examined the impact of ROI-based coding parameters, particularly the effects of mask type and dilation on the preservation of mitochondrial information and the performance of computer vision models. This last evaluation led to the conclusion that non-rectangular masks with no dilation yield the best rate-performance results.

# 5

# CONCLUSION AND FUTURE WORK

## 5.1 CONCLUSIONS

The primary aim of this dissertation was to investigate the performance of learning-based detection and segmentation models on compression-degraded EM images. The study explores two main approaches: standard coding and ROI-based coding. The latter focuses on leveraging detection and segmentation results from these computer vision models to establish compression boundaries, ensuring that ROIs are preserved with higher quality while allowing more aggressive compression in less critical areas. This content-aware compression was implemented on HEVC/H.265 and VVC/H.266 standards, allowing the encoders to prioritise the preservation of ROIs' quality. Unlike traditional ROI-based coding approaches in the literature, the implemented solutions support non-rectangular regions with a CU level resolution, as fine as $4 \times 4$ pixels in VVC, without significantly compromising the background information.

The obtained results demonstrated a superior compression performance of the VVC codec over HEVC for similar detection and segmentation performance. This difference is particularly significant when the learning-based models used to analyse compressed images were trained using images with compression-induced distortions. Fine-tuning the detection and segmentation models on compressed images significantly enhanced their robustness and performance. Notably, fine-tuning on VVC-compressed images yielded substantial performance gains across different datasets, highlighting the importance of adapting the models to the degradations introduced by codecs.

The ROI-based coding strategy consistently outperformed standard coding approaches. By allocating higher quality to regions containing biomedical relevant information, it was shown that significant bitrate reductions were possible to attain while preserving essential details for model inference. The choice of mask type and dilation notably affects both compression efficiency and model performance. Studies revealed that non-rectangular masks without dilation are the most effective for ROI-based coding, significantly outperforming rectangular masks. While ROI-based coding introduces some quality degradation in mitochondrial regions, this trade-off is minimal compared to the substantial bitrate savings achieved. These findings

demonstrate that the proposed approach successfully balances compression efficiency with the preservation of critical features for computer vision tasks on these datasets.

Additionally, a framework for optimisation based on the Pareto frontier solution was presented to identify the optimal compression configurations. This approach ensured an optimal balance between bitrate reduction and model performance, enabling the selection of encoding parameters tailored to specific tasks and datasets. Overall, the findings of this research highlight the potential of ROI-based coding in biomedical image compression for machine analysis. The proposed methods effectively reduce storage and transmission requirements while maintaining high detection and segmentation accuracy, making them well-suited for applications in healthcare, research, and beyond.

## 5.2 FUTURE WORK

In this study, detection and computer vision models were used to define the ROIs. However, future work could explore the use of less complex networks to generate saliency or attention maps before compression. While this approach might result in larger mask areas, making it less efficient, it would significantly reduce the complexity on the encoder side. Although this complexity reduction may not be crucial in medical scenarios, it could be particularly beneficial in edge computing applications, such as autonomous driving and surveillance. In such contexts, a rough estimation of the regions of interest could be computed directly, for instance, on the vehicle and encoded accordingly, with the processing then performed on a remote server after transmission and decoding.

Additionally, having successfully implemented ROI-based coding in the latest video coding standards the focus of future work could shift to learning-based coding solutions. For instance, a ROI-based adaptation could be developed for the JPEG-AI standard, which is currently under development. JPEG-AI aims to use deep learning techniques to achieve state-of-the-art compression performance, and integrating ROI-based coding into this framework could further enhance its efficiency and applicability to domains requiring selective quality preservation.

# BIBLIOGRAPHY


[1] W. Fruehwirt and P. Duckworth, "Towards better healthcare: What could and should be automated?", *Technological Forecasting and Social Change*, vol. 172, p. 120 967, 2021, ISSN: 0040-1625. DOI: https://doi.org/10.1016/j.techfore.2021.120967. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0040162521003991.

[2] S. A. Alowais, S. S. Alghamdi, N. Alsuhebany, *et al.*, "Revolutionizing healthcare: The role of artificial intelligence in clinical practice", en, *BMC Med. Educ.*, vol. 23, no. 1, p. 689, Sep. 2023.

[3] R. Weissleder and M. Nahrendorf, "Advancing biomedical imaging", en, *Proc. Natl. Acad. Sci. U. S. A.*, vol. 112, no. 47, pp. 14 424–14 428, Nov. 2015.

[4] S. Dash, S. K. Shakyawar, M. Sharma, and S. Kaushik, "Big data in healthcare: Management, analysis and future prospects", en, *J. Big Data*, vol. 6, no. 1, Dec. 2019.

[5] A. A. T. Bui, J. D. Van Horn, and NIH BD2K Centers Consortium, "Envisioning the future of 'big data' biomedicine", en, *J. Biomed. Inform.*, vol. 69, pp. 115–117, May 2017.

[6] J. Thomason, "Data, digital worlds, and the avatarization of health care", *Global Health Journal*, vol. 8, no. 1, pp. 1–3, 2024, ISSN: 2414-6447. DOI: https://doi.org/10.1016/j.glohj.2024.02.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2414644724000034.

[7] C.-B. CHAFEA:Consumers, Health, Agriculture, and Food Executive Agency, *Assessment of the EU Member States' rules on health data in the light of GDPR*, en. Publications Office of the European Union, Feb. 2021.

[8] R. Monika and S. Dhanalakshmi, "An efficient medical image compression technique for telemedicine systems", en, *Biomed. Signal Process. Control*, vol. 80, no. 104404, p. 104 404, Feb. 2023.

[9] R. Yang, T. Xiao, Y. Cheng, *et al.*, "Sharing massive biomedical data at magnitudes lower bandwidth using implicit neural function", en, *Proc. Natl. Acad. Sci. U. S. A.*, vol. 121, no. 28, e2320870121, Jul. 2024.

[10] F. Liu, M. Hernandez-Cabronero, V. Sanchez, M. W. Marcellin, and A. Bilgin, "The current role of image compression standards in medical imaging", en, *Information (Basel)*, vol. 8, no. 4, p. 131, Dec. 2017.

[11] H. K. Huang, *PACS and Imaging Informatics*, en, 2nd ed. Hoboken, NJ: Wiley-Blackwell, Sep. 2011.

[12] E. S. Paulo, L. A. Thomaz, L. M. N. Távora, P. A. A. Assuncao, and S. M. M. Faria, "Extending the compression range of biomedical images for machine vision analysis", in *2022 30th European Signal Processing Conference (EUSIPCO)*, 2022, pp. 1273–1277. DOI: 10.23919/EUSIPCO55093.2022.9909663.

[13] A. C. Flint, "Determining optimal medical image compression: Psychometric and image distortion analysis", en, *BMC Med. Imaging*, vol. 12, no. 1, p. 24, Jul. 2012.

[14] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images", *Nature Communications*, vol. 15, no. 1, Jan. 2024, ISSN: 2041-1723. DOI: 10.1038/s41467-024-44824-z. [Online]. Available: http://dx.doi.org/10.1038/s41467-024-44824-z.

[15] T. Gandor and J. Nalepa, "First gradually, then suddenly: Understanding the impact of image compression on object detection using deep learning", en, *Sensors (Basel)*, vol. 22, no. 3, p. 1104, Feb. 2022.

[16] European Society of Radiology (ESR), "Medical imaging in personalised medicine: A white paper of the research committee of the european society of radiology (ESR)", en, *Insights Imaging*, vol. 6, no. 2, pp. 141–155, Apr. 2015.

[17] L. Papp, C. P. Spielvogel, and et al., "Personalizing medicine through hybrid imaging and medical big data analysis", *Frontiers in Physics*, vol. 6, 2018, ISSN: 2296-424X. DOI: 10.3389/fphy.2018.00051. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fphy.2018.00051.

[18] R. L. Wahl, L. E. Quint, and et al., ""anatometabolic" tumor imaging: Fusion of fdg pet with ct or mri to localize foci of increased activity", *Journal of nuclear medicine*, vol. 34, no. 7, pp. 1190–1197, 1993.

[19] S. Hussain, I. Mubeen, N. Ullah, *et al.*, "Modern diagnostic imaging technique applications and risk factors in the medical field: A review", en, *Biomed Res. Int.*, vol. 2022, no. 1, p. 5 164 970, Jun. 2022.

[20] X. Ying and T. M. Monticello, "Modern imaging technologies in toxicologic pathology: An overview", en, *Toxicol. Pathol.*, vol. 34, no. 7, pp. 815–826, 2006.

[21] P. M. A. Antony, C. Trefois, A. Stojanovic, A. S. Baumuratov, and K. Kozak, "Light microscopy applications in systems biology: Opportunities and challenges", en, *Cell Commun. Signal.*, vol. 11, no. 1, p. 24, Apr. 2013.

[22] C. M. Tempany and B. J. McNeil, "Advances in biomedical imaging", en, *JAMA*, vol. 285, no. 5, pp. 562–567, Feb. 2001.

[23] D. C. Preston, *Magnetic resonance imaging (mri) of the brain and spine: Basics*, 2006. [Online]. Available: https://case.edu/med/neurology/NR/MRI%20Basics.htm.

[24] W. Sun, S. B. Brown, and R. K. Leach, "An overview of industrial x-ray computed tomography", National Physical Laboratory, NPL Report ENG 32, 2012.

[25] K. Herholz and W.-D. Heiss, "Positron emission tomography in clinical neurology", en, *Mol. Imaging Biol.*, vol. 6, no. 4, pp. 239–269, Jul. 2004.

[26] J. S. Rabin, H. Klein, D. R. Kirn, *et al.*, "Associations of physical activity and b-amyloid with longitudinal cognition and neurodegeneration in clinically normal older adults", *JAMA Neurology*, vol. 76, no. 10, pp. 1203–1210, 2019. DOI: 10.1001/jamaneurol.2019.1879. [Online]. Available: https://jamanetwork.com/journals/jamaneurology/fullarticle/2738357.

[27] A. Stockhammer and F. Bottanelli, "Appreciating the small things in life: STED microscopy in living cells", *J. Phys. D Appl. Phys.*, vol. 54, no. 3, p. 033 001, Jan. 2021.

[28] L. Schermelleh, A. Ferrand, T. Huser, *et al.*, "Super-resolution microscopy demystified", en, *Nat. Cell Biol.*, vol. 21, no. 1, pp. 72–84, Jan. 2019.

[29] B. J. Erickson, P. Korfiatis, Z. Akkus, and T. L. Kline, "Machine learning for medical imaging", en, *Radiographics*, vol. 37, no. 2, pp. 505–515, Mar. 2017.

[30] K. Nakamae, "Electron microscopy in semiconductor inspection", *Meas. Sci. Technol.*, vol. 32, no. 5, p. 052 003, May 2021.

[31] T. Shirai, "Protein structure analysis and validation", in *Encyclopedia of Bioinformatics and Computational Biology*, S. Ranganathan, M. Gribskov, K. Nakai, and C. Schönbach, Eds., Oxford: Academic Press, 2019, pp. 512–519, ISBN: 978-0-12-811432-2. DOI: https://doi.org/10.1016/B978-0-12-809633-8.20282-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128096338202823.

[32] A. J. Rozo, M. H. Cox, A. Devitt, A. J. Rothnie, and A. D. Goddard, "Biophysical analysis of lipidic nanoparticles", en, *Methods*, vol. 180, pp. 45–55, Aug. 2020.

[33] D. Shindo and K. Hiraga, "Basis of high-resolution electron microscopy", in *High-Resolution Electron Microscopy for Materials Science*. Tokyo: Springer Japan, 1998, pp. 1–15, ISBN: 978-4-431-68422-0. DOI: 10.1007/978-4-431-68422-0_1. [Online]. Available: https://doi.org/10.1007/978-4-431-68422-0_1.

[34] N. Shibata, Y. Kohno, A. Nakamura, *et al.*, "Atomic resolution electron microscopy in a magnetic field free environment", en, *Nat. Commun.*, vol. 10, no. 1, p. 2308, May 2019.

[35] M. Agrawal, V. Prasad, G. Nijhawan, S. S. Jalal, B. Rajalakshmi, and S. P. Dwivedi, "A comprehensive review of electron microscopy in materials science: Technological advances and applications", *E3S Web Conf.*, vol. 505, p. 01 029, 2024.

[36] B. Inkson, "2 - scanning electron microscopy (sem) and transmission electron microscopy (tem) for materials characterization", in *Materials Characterization Using Nondestructive Evaluation (NDE) Methods*, G. Hübschen, I. Altpeter, R. Tschuncky, and H.-G. Herrmann, Eds., Woodhead Publishing, 2016, pp. 17–43, ISBN: 978-0-08-100040-3. DOI: https://doi.org/10.1016/B978-0-08-100040-3.00002-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978008100040300002X.

[37] M. Zeviani and S. Di Donato, "Mitochondrial disorders", en, *Brain*, vol. 127, no. Pt 10, pp. 2153–2172, Oct. 2004.

[38] Y. Zong, H. Li, P. Liao, *et al.*, "Mitochondrial dysfunction: Mechanisms and advances in therapy", en, *Signal Transduct. Target. Ther.*, vol. 9, no. 1, p. 124, May 2024.

[39] N. Kasthuri, K. J. Hayworth, D. R. Berger, *et al.*, "Saturated reconstruction of a volume of neocortex", *Cell*, vol. 162, no. 3, pp. 648–661, Jul. 2015, ISSN: 0092-8674. DOI: 10.1016/j.cell.2015.06.054. [Online]. Available: http://dx.doi.org/10.1016/j.cell.2015.06.054.

[40] E. Tjahjono, D. R. Kirienko, and N. V. Kirienko, "The emergent role of mitochondrial surveillance in cellular health", en, *Aging Cell*, vol. 21, no. 11, e13710, Nov. 2022.

[41] A. Lucchi, Y. Li, K. Smith, and P. Fua, "Structured image segmentation using kernelized features", in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 400–413, ISBN: 978-3-642-33709-3.

[42] V. Casser, K. Kang, H. Pfister, and D. Haehn, "Fast mitochondria detection for connectomics", in *Proceedings of the Third Conference on Medical Imaging with Deep Learning*, T. Arbel, I. Ben Ayed, M. de Bruijne, M. Descoteaux, H. Lombaert, and C. Pal, Eds., ser. Proceedings of Machine Learning Research, vol. 121, PMLR, Jul. 2020, pp. 111–120. [Online]. Available: https://proceedings.mlr.press/v121/casser20a.html.

[43] H.-C. Cheng and A. Varshney, "Volume segmentation using convolutional neural networks with limited training data", in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 590–594. DOI: 10.1109/ICIP.2017.8296349.

[44] E. S. Paulo, "Lossy compression of biomedical images for computer vision analysis", Available at http://hdl.handle.net/10400.8/10029, Master's thesis, School of Technology and Management, Leiria, Portugal, Jun. 2024.

[45] A. Storr, C. A. Venetis, S. Cooke, S. Kilani, and W. Ledger, "Inter-observer and intra-observer agreement between embryologists during selection of a single day 5 embryo for transfer: A multicenter study", *Human Reproduction*, vol. 32, no. 2, pp. 307–314, Jan. 2017, ISSN: 0268-1161. DOI: 10.1093/humrep/dew330. eprint: https://academic.oup.com/humrep/article-pdf/32/2/307/10237740/dew330.pdf. [Online]. Available: https://doi.org/10.1093/humrep/dew330.

[46] R. Kaur and S. Singh, "A comprehensive review of object detection with deep learning", *Digital Signal Processing*, vol. 132, p. 103 812, Jan. 2023, ISSN: 1051-2004. DOI: 10.1016/j.dsp.2022.103812. [Online]. Available: http://dx.doi.org/10.1016/j.dsp.2022.103812.

[47] A. Leitherer, B. C. Yeo, C. H. Liebscher, and L. M. Ghiringhelli, "Automatic identification of crystal structures and interfaces via artificial-intelligence-based electron microscopy", *npj Computational Materials*, vol. 9, no. 1, Oct. 2023, ISSN: 2057-3960. DOI: 10.1038/s41524-023-01133-1. [Online]. Available: http://dx.doi.org/10.1038/s41524-023-01133-1.

[48] D. J. C. Barbosa, J. Ramos, and C. S. Lima, "Detection of small bowel tumors in capsule endoscopy frames using texture analysis based on the discrete wavelet transform", in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, IEEE, Aug. 2008, pp. 3012–3015. DOI: 10.1109/iembs.2008.4649837. [Online]. Available: http://dx.doi.org/10.1109/IEMBS.2008.4649837.

[49] T. M. Mejía, M. G. Pérez, V. H. Andaluz, and A. Conci, "Automatic segmentation and analysis of thermograms using texture descriptors for breast cancer detection", in *2015 Asia-Pacific Conference on Computer Aided System Engineering*, 2015, pp. 24–29. DOI: 10.1109/APCASE.2015.12.

[50] L. Cai, X. Wang, Y. Wang, Y. Guo, J. Yu, and Y. Wang, "Robust phase-based texture descriptor for classification of breast ultrasound images", *BioMedical Engineering OnLine*, vol. 14, no. 1, Mar. 2015, ISSN: 1475-925X. DOI: 10.1186/s12938-015-0022-8. [Online]. Available: http://dx.doi.org/10.1186/s12938-015-0022-8.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition", in *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 346–361, ISBN: 9783319105789. DOI: 10.1007/978-3-319-10578-9_23. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10578-9_23.

[52] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, Feb. 2020, ISSN: 1939-3539. DOI: 10.1109/tpami.2018.2844175. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2018.2844175.

[53] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2014, pp. 580–587. DOI: 10.1109/cvpr.2014.81. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.81.

[54] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation", *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, Sep. 2004, ISSN: 0920-5691. DOI: 10.1023/b:visi.0000022288.19776.77. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000022288.19776.77.

[55] R. Girshick, *Fast r-cnn*, 2015. DOI: 10.48550/ARXIV.1504.08083. [Online]. Available: https://arxiv.org/abs/1504.08083.

[56] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, 2015. DOI: 10.48550/ARXIV.1506.01497. [Online]. Available: https://arxiv.org/abs/1506.01497.

[57] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas", *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, ISSN: 2504-4990. DOI: 10.3390/make5040083. [Online]. Available: http://dx.doi.org/10.3390/make5040083.

[58] M. G. Ragab, S. J. Abdulkadir, A. Muneer, *et al.*, "A comprehensive systematic review of yolo for medical object detection (2018 to 2023)", *IEEE Access*, vol. 12, pp. 57 815–57 836, 2024. DOI: 10.1109/ACCESS.2024.3386826.

[59] L. Guo, Y. Yang, H. Ding, *et al.*, "A deep learning-based hybrid artificial intelligence model for the detection and severity assessment of vitiligo lesions", *Annals of Translational Medicine*, vol. 10, no. 10, pp. 590–590, May 2022, ISSN: 2305-5847. DOI: 10.21037/atm-22-1738. [Online]. Available: http://dx.doi.org/10.21037/atm-22-1738.

[60] L. S. J, E. Joy, and S. R. M S, "Enhanced radiological anomaly detection using optimized yolo-nas model", in *2024 International Conference on Advances in Modern Age Technologies for Health and Engineering Science (AMATHE)*, 2024, pp. 1–6. DOI: 10.1109/AMATHE61652.2024.10582157.

[61] A. Baccouche, B. Garcia-Zapirain, C. Castillo Olea, and A. S. Elmaghraby, "Breast lesions detection and classification via yolo-based fusion models", *Computers, Materials amp; Continua*, vol. 69, no. 1, pp. 1407–1425, 2021, ISSN: 1546-2226. DOI: 10.32604/cmc.2021.018461. [Online]. Available: http://dx.doi.org/10.32604/cmc.2021.018461.

[62] M. A. Al-antari, M. A. Al-masni, and T.-S. Kim, "Deep learning computer-aided diagnosis for breast lesion in digital mammogram", in *Deep Learning in Medical Image Analysis*. Springer International Publishing, 2020, pp. 59–72, ISBN: 9783030331283. DOI: 10.1007/978-3-030-33128-3_4.

[63] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection*, 2015. DOI: 10.48550/ARXIV.1506.02640. [Online]. Available: https://arxiv.org/abs/1506.02640.

[64] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger", *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:786357.

[65] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, *Yolov4: Optimal speed and accuracy of object detection*, 2020. DOI: 10.48550/ARXIV.2004.10934. [Online]. Available: https://arxiv.org/abs/2004.10934.

[66] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, *Distance-iou loss: Faster and better learning for bounding box regression*, 2019. DOI: 10.48550/ARXIV.1911.08287. [Online]. Available: https://arxiv.org/abs/1911.08287.

[67] D. Reis, J. Kupec, J. Hong, and A. Daoudi, *Real-time flying object detection with yolov8*, 2023. DOI: 10.48550/ARXIV.2305.09972. [Online]. Available: https://arxiv.org/abs/2305.09972.

[68] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, *Feature pyramid networks for object detection*, 2016. DOI: 10.48550/ARXIV.1612.03144. [Online]. Available: https://arxiv.org/abs/1612.03144.

[69] M. Osama, R. Kumar, and M. Shahid, "Empowering cardiologists with deep learning yolov8 model for accurate coronary artery stenosis detection in angiography images", in *2023 International Conference on IoT, Communication and Automation Technology (ICICAT)*, 2023, pp. 1–6. DOI: 10.1109/ICICAT57735.2023.10263760.

[70] N. Palanivel, D. S, L. P. G, S. B, and S. M. M, "The art of yolov8 algorithm in cancer diagnosis using medical imaging", in *2023 International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2023, pp. 1–6. DOI: 10.1109/ICSCAN58655.2023.10395046.

[71] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, *Yolov9: Learning what you want to learn using programmable gradient information*, 2024. DOI: 10.48550/ARXIV.2402.13616. [Online]. Available: https://arxiv.org/abs/2402.13616.

[72] A. Wang, H. Chen, L. Liu, *et al.*, *Yolov10: Real-time end-to-end object detection*, 2024. DOI: 10.48550/ARXIV.2405.14458. [Online]. Available: https://arxiv.org/abs/2405.14458.

[73] R. Khanam and M. Hussain, *Yolov11: An overview of the key architectural enhancements*, 2024. DOI: 10.48550/ARXIV.2410.17725. [Online]. Available: https://arxiv.org/abs/2410.17725.

[74] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, "Nnu-net: A self-configuring method for deep learning-based biomedical image segmentation", *Nature Methods*, vol. 18, no. 2, pp. 203–211, Dec. 2020, ISSN: 1548-7105. DOI: 10.1038/s41592-020-01008-z. [Online]. Available: http://dx.doi.org/10.1038/s41592-020-01008-z.

[75] N. Otsu, "A threshold selection method from gray-level histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. DOI: 10.1109/TSMC.1979.4310076.

[76] J. Canny, "A computational approach to edge detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986. DOI: 10.1109/TPAMI.1986.4767851.

[77] Y. Xu, R. Quan, W. Xu, Y. Huang, X. Chen, and F. Liu, "Advances in medical image segmentation: A comprehensive review of traditional, deep learning and hybrid approaches", *Bioengineering*, vol. 11, no. 10, 2024, ISSN: 2306-5354. DOI: 10.3390/bioengineering11101034. [Online]. Available: https://www.mdpi.com/2306-5354/11/10/1034.

[78] A. Saxena, M. Prasad, A. Gupta, *et al.*, "A review of clustering techniques and developments", *Neurocomputing*, vol. 267, pp. 664–681, 2017, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.06.053. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231217311815.

[79] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis", *The Computer Journal*, vol. 41, no. 8, pp. 578–588, Jan. 1998, ISSN: 0010-4620. DOI: 10.1093/comjnl/41.8.578. eprint: https://academic.oup.com/comjnl/article-pdf/41/

8/578/1032918/410578.pdf. [Online]. Available: https://doi.org/10.1093/comjnl/41.8.578.

[80] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241, ISBN: 978-3-319-24574-4.

[81] J. C. Caicedo, J. Roth, A. Goodman, *et al.*, "Evaluation of deep learning strategies for nucleus segmentation in fluorescence images", *Cytometry Part A*, vol. 95, no. 9, pp. 952–965, Jul. 2019, ISSN: 1552-4930. DOI: 10.1002/cyto.a.23863. [Online]. Available: http://dx.doi.org/10.1002/cyto.a.23863.

[82] F. Long, "Microscopy cell nuclei segmentation with enhanced u-net", *BMC Bioinformatics*, vol. 21, no. 1, Jan. 2020, ISSN: 1471-2105. DOI: 10.1186/s12859-019-3332-1. [Online]. Available: http://dx.doi.org/10.1186/s12859-019-3332-1.

[83] S. Metlek, "Cellsegunet: An improved deep segmentation model for the cell segmentation based on unet++ and residual unet models", *Neural Computing and Applications*, vol. 36, no. 11, pp. 5799–5825, Jan. 2024, ISSN: 1433-3058. DOI: 10.1007/s00521-023-09374-3. [Online]. Available: http://dx.doi.org/10.1007/s00521-023-09374-3.

[84] P. Zheng, X. Zhu, and W. Guo, "Brain tumour segmentation based on an improved u-net", *BMC Medical Imaging*, vol. 22, no. 1, Nov. 2022, ISSN: 1471-2342. DOI: 10.1186/s12880-022-00931-1. [Online]. Available: http://dx.doi.org/10.1186/s12880-022-00931-1.

[85] J. Walsh, A. Othmani, M. Jain, and S. Dev, *Using u-net network for efficient brain tumor segmentation in mri images*, 2022. DOI: 10.48550/ARXIV.2211.01885. [Online]. Available: https://arxiv.org/abs/2211.01885.

[86] A. Hatamizadeh, Y. Tang, V. Nath, *et al.*, "Unetr: Transformers for 3d medical image segmentation", in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2022, pp. 574–584.

[87] Z. Liu, H. Wang, W. Lei, and G. Wang, "Csaf-cnn: Cross-layer spatial attention map fusion network for organ-at-risk segmentation in head and neck ct images", in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, 2020, pp. 1522–1525. DOI: 10.1109/ISBI45749.2020.9098711.

[88] S. Baek, D. H. Ye, and O. Lee, "Unet-based multi-organ segmentation in photon counting ct using virtual monoenergetic images", *Medical Physics*, vol. 52, no. 1, pp. 481–488, Oct. 2024, ISSN: 2473-4209. DOI: 10.1002/mp.17440. [Online]. Available: http://dx.doi.org/10.1002/mp.17440.

[89] A. Widayani, A. M. Putra, A. R. Maghriebi, D. Z. C. Adi, and M. H. F. Ridho, "Review of application yolov8 in medical imaging", *Indonesian Applied Physics Letters*, vol. 5, no. 1, pp. 23–33, May 2024. DOI: 10.20473/iapl.v5i1.57001. [Online]. Available: https://e-journal.unair.ac.id/IAPL/article/view/57001.

[90] D. M. Abed, S. Abdul-Rahman, and S. Mutalib, "Dental segmentation via enhanced yolov8 and image processing techniques", *Mesopotamian Journal of CyberSecurity*, vol. 4, no. 3, pp. 189–202, Dec. 2024, ISSN: 2958-6542. DOI: 10.58496/mjcs/2024/022. [Online]. Available: http://dx.doi.org/10.58496/MJCS/2024/022.

[91] A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything", 2023. eprint: 2304.02643 (cs.CV).

[92] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale", in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy.

[93] L. P. Osco, Q. Wu, E. L. de Lemos, *et al.*, "The segment anything model (sam) for remote sensing applications: From zero to one shot", *International Journal of Applied Earth Observation and Geoinformation*, vol. 124, p. 103 540, 2023, ISSN: 1569-8432. DOI: https://doi.org/10.1016/j.jag.2023.103540. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843223003643.

[94] W. Nazzal, K. Thurnhofer-Hemsi, and E. López-Rubio, "Improving medical image segmentation using test-time augmentation with medsam", *Mathematics*, vol. 12, no. 24, p. 4003, Dec. 2024, ISSN: 2227-7390. DOI: 10.3390/math12244003. [Online]. Available: http://dx.doi.org/10.3390/math12244003.

[95] X. Wei, J. Cao, Y. Jin, M. Lu, G. Wang, and S. Zhang, *I-medsam: Implicit medical image segmentation with segment anything*, 2023. DOI: 10.48550/ARXIV.2311.17081. [Online]. Available: https://arxiv.org/abs/2311.17081.

[96] K. Pang and S. Zeng, "Mediastinal image detection and segmentation based on MedSAM", in *Fourth International Conference on Image Processing and Intelligent Control (IPIC 2024)*, K. Du and A. bin Mohd Zain, Eds., International Society for Optics and Photonics, vol. 13250, SPIE, 2024,

132500O. DOI: 10.1117/12.3038517. [Online]. Available: https://doi.org/10.1117/12.3038517.

[97] M. A. Mazurowski, H. Dong, H. Gu, J. Yang, N. Konz, and Y. Zhang, "Segment anything model for medical image analysis: An experimental study", *Medical Image Analysis*, vol. 89, p. 102 918, 2023, ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2023.102918. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1361841523001780.

[98] S. Pandey, K.-F. Chen, and E. B. Dam, " Comprehensive Multimodal Segmentation in Medical Imaging: Combining YOLOv8 with SAM and HQ-SAM Models ", in *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2023, pp. 2584–2590. DOI: 10.1109/ICCVW60793.2023.00273. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCVW60793.2023.00273.

[99] M. Larobina, "Thirty years of the DICOM standard", en, *Tomography*, vol. 9, no. 5, pp. 1829–1838, Oct. 2023.

[100] F. F. Cunha, V. Blüml, L. M. Zopf, *et al.*, "Lossy image compression in a preclinical multimodal imaging study", en, *J. Digit. Imaging*, vol. 36, no. 4, pp. 1826–1850, Aug. 2023.

[101] "High Efficiency Video Coding", Joint Collaborative Team on Video Coding, Standard, 2013.

[102] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012. DOI: 10.1109/TCSVT.2012.2221191.

[103] D. Flynn, D. Marpe, M. Naccari, *et al.*, "Overview of the range extensions for the hevc standard: Tools, profiles, and performance", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 4–19, 2016. DOI: 10.1109/TCSVT.2015.2478707.

[104] "Versatile Video Coding", Joint Video Experts Team, Standard, 2020.

[105] B. Bross, Y.-K. Wang, Y. Ye, *et al.*, "Overview of the versatile video coding (vvc) standard and its applications", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021. DOI: 10.1109/TCSVT.2021.3101953.

[106] M. Xu, X. Deng, S. Li, and Z. Wang, "Region-of-interest based conversational hevc coding with hierarchical perception model of face", *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 3, pp. 475–489, 2014. DOI: 10.1109/JSTSP.2014.2314864.

[107] Y. Wu, P. Liu, Y. Gao, and K. Jia, "Medical ultrasound video coding with h.265/hevc based on roi extraction", *PLOS ONE*, vol. 11, no. 11, H. Zhang, Ed., e0165698, Nov. 2016, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0165698. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0165698.

[108] X. Sun, X. Yang, S. Wang, and M. Liu, "Content-aware rate control scheme for hevc based on static and dynamic saliency detection", *Neurocomputing*, vol. 411, pp. 393–405, Oct. 2020, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2020.06.003. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2020.06.003.

[109] V.-I. Ungureanu, P. Negirla, and A. Korodi, "Image-compression techniques: Classical and "region-of-interest-based" approaches presented in recent papers", en, *Sensors (Basel)*, vol. 24, no. 3, Jan. 2024.

[110] R. Grosbois, D. Santa-cruz, and T. Ebrahimi, "New approach to jpeg 2000 compliant region of interest coding", *Proc SPIE*, Aug. 2001. DOI: 10.1117/12.449760.

[111] G. Anastassopoulos and A. Skodras, "Jpeg2000 roi coding in medical imaging applications", *Proc. 2nd IASTED Int. Conf. on Visualisation, Imaging and Image Processing (VIIP2002)*, Jan. 2002.

[112] J. Bartrina-Rapesta, J. Serra-Sagristà, and F. Aulí-Llinàs, "JPEG2000 ROI coding through component priority for digital mammography", en, *Comput. Vis. Image Underst.*, vol. 115, no. 1, pp. 59–68, Jan. 2011.

[113] J. Bartrina-Rapesta, J. Serra-Sagristà, F. Aulí-Llinàs, and J. Muñoz Gómez, "Jpeg2000 roi coding method with perfect fine-grain accuracy and lossless recovery", in *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, 2009, pp. 558–562. DOI: 10.1109/ACSSC.2009.5469892.

[114] R. Antonio, J. Rosa, L. Ferreira, M. Figueiredo, P. Assuncao, and C. Ribeiro, "Enhanced object detection in highly compressed images using regions of interest", in *Proceedings of the 2023 6th International Conference on Sensors, Signal and Image Processing*, Nanjing China: ACM, Oct. 2023.

[115] M. Meddeb, M. Cagnazzo, and B. Pesquet-Popescu, "Roi-based rate control using tiles for an hevc encoded video stream over a lossy network", in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 1389–1393. DOI: 10.1109/ICIP.2015.7351028.

[116] G. Ren, Z. Liu, Z. Chen, and S. Liu, "Reinforcement learning based roi bit allocation for gaming video coding in vvc", in *2021 International Conference*

*on Visual Communications and Image Processing (VCIP)*, 2021, pp. 1–5. DOI: 10.1109/VCIP53242.2021.9675345.

[117] C.-H. Kao, Y.-C. Weng, Y.-H. Chen, W.-C. Chiu, and W.-H. Peng, "Transformer-based variable-rate image compression with region-of-interest control", in *2023 IEEE International Conference on Image Processing (ICIP)*, 2023, pp. 2960–2964. DOI: 10.1109/ICIP49359.2023.10222853.

[118] Y. Ma, Y. Zhai, C. Yang, *et al.*, "Variable rate roi image compression optimized for visual quality", in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 1936–1940. DOI: 10.1109/CVPRW53098.2021.00221.

[119] J. Ascenso, E. Alshina, and T. Ebrahimi, "The jpeg ai standard: Providing efficient human and machine visual data consumption", *IEEE MultiMedia*, vol. 30, no. 1, pp. 100–111, 2023. DOI: 10.1109/MMUL.2023.3245919.

[120] H. Schwarz, M. Coban, M. Karczewicz, *et al.*, "Quantization and entropy coding in the versatile video coding (vvc) standard", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3891–3906, 2021. DOI: 10.1109/TCSVT.2021.3072202.

[121] C. R. Helmrich, S. Bosse, M. Siekmann, H. Schwarz, D. Marpe, and T. Wiegand, "Perceptually optimized bit-allocation and associated distortion measure for block-based image or video coding", in *2019 Data Compression Conference (DCC)*, 2019, pp. 172–181. DOI: 10.1109/DCC.2019.00025.

[122] F. Bossen, D. Flynn, K. Sharman, and K. Sühring, *Hm software manual*, Version 18.0, Joint Video Experts Team (JVET), Apr. 2023. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jct-vc/HM.

[123] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework", in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[124] C. Peng, "Research on yolov4 object detection based on k-means algorithm and fusion attention mechanism", in *2023 International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, 2023, pp. 439–444. DOI: 10.1109/AEECA59734.2023.00084.

[125] W. Xie, N. Willems, S. Patil, Y. Li, and M. Kumar, "SAM fewshot finetuning for anatomical segmentation in medical images", in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA: IEEE, Jan. 2024, pp. 3241–3249.

[126] F. Bossen *et al.*, "Common test conditions and software reference configurations", *JCTVC-L1100*, vol. 12, no. 7, p. 1, 2013.

[127] J. Boyce, K. Suehring, X. Li, and V. Seregin, "Jvet-j1010: Jvet common test conditions and software reference configurations", Tech. Rep., Jul. 2018.

[128] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression", in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658–666. DOI: 10.1109/CVPR.2019.00075.

[129] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context", in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 740–755, ISBN: 978-3-319-10602-1.

[130] S. A. Hicks, I. Strümke, V. Thambawita, *et al.*, "On evaluation metrics for medical applications of artificial intelligence", en, *Sci. Rep.*, vol. 12, no. 1, p. 5979, Apr. 2022.

[131] G. Bjøntegaard, *Calculation of average psnr differences between rd-curves*, ITU-T SG16/Q6 Input Document VCEG-M33, Apr. 2001.

[132] N. Barman, M. G. Martini, and Y. Reznik, *Bjøntegaard delta (bd): A tutorial overview of the metric, evolution, challenges, and recommendations*, 2024. DOI: 10.48550/ARXIV.2401.04039. [Online]. Available: https://arxiv.org/abs/2401.04039.

[133] J. Zhao, Y. Su, and A. Segall, "On the calculation of psnr and bit-rate differences for the svt test data", ITU-T SG16, ITU Contribution COM16-C404-E, 2008.

[134] A. F. R. Guarda, N. M. M. Rodrigues, and F. Pereira, "The JPEG pleno learning-based point cloud coding standard: Serving man and machine", 2024. eprint: 2409.08130 (eess.IV).

[135] Z. Luo, W. Jia, and S. Perry, "Compressed point cloud classification with point-based edge sampling", en, *EURASIP J. Image Video Process.*, vol. 2024, no. 1, Aug. 2024.

[136] C. Yang, W. Ye, and Q. Li, "Review of the performance optimization of parallel manipulators", *Mechanism and Machine Theory*, vol. 170, p. 104 725, 2022, ISSN: 0094-114X. DOI: https://doi.org/10.1016/j.mechmachtheory.2022.104725. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094114X22000040.

# APPENDICES

# A

## APPENDIX A

In Table A.1, the image IDs used for the training and validation sets of the Lucchi++ and Kasthuri++ datasets are defined.

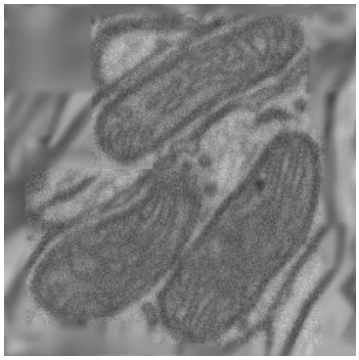
Table A.1: Number IDs used for the training and validation sets of the Lucchi++ and Kasthuri++ datasets.

| **Dataset** | **Instance number ID** | |
|---|---|---|
| | **Train set** | **Validation set** |
| Lucchi++ | 0,2,3,5,6,8,9,11,12,14,15,16, 17,20,21,22,25,26,28,29,30,32, 33,34,35,36,37,38,39,41,42,43, 45,46,48,49,50,51,53,54,55,56, 57,59,60,61,63,65,66,67,69,70, 72,73,74,75,77,79,80,81,82,84, 85,86,87,88,89,91,92,93,94,96, 97,99,100,102,103,104,105,107, 109,110,111,112,114,115,116,117, 119,120,121,122,124,125,126,127, 129,130,131,133,134,135,136,137, 138,140,142,143,144,146,147,148, 149,151,152,153,154,156,157,158, 159,161,162,163,164 | 1,4,7,10,13,18, 19,23,24,27,31, 40, 44,47,52,58, 62,64,68,71,76, 78,83, 90,95,98, 101,106,108,113, 118,123,128,132, 139,141,145,150, 155,160 |
| Kasthuri++ | 1032,1033,1034,1036,1037,1038,1039, 1041,1042,1044,1045,1046,1047,1049, 1050,1052,1054,1055,1057,1058,1060, 1061,1062,1063,1065,1067,1068,1069, 1070,1072,1073,1075,1076,1078,1079, 1080,1082,1083,1085,1086,1087,1088, 1089,1091,1092,1093,1095,1096,1098, 1099,1101,1102,1103,1105,1106,1108, 1109,1110,1111,1113,1114 | 1031,1035,1040, 1043,1048,1051, 1053,1056,1059, 1064,1066,1071, 1074,1077,1081, 1084,1090,1094, 1097,1100,1104, 1107,1112,1115 |

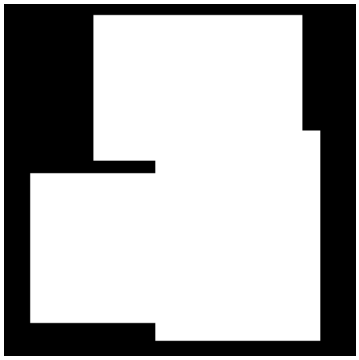
# B

## APPENDIX B

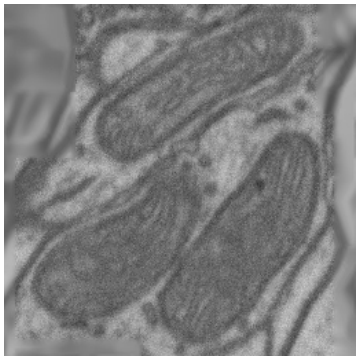
In this appendix, Figures B.1 and B.2 illustrate sample images encoded using ROI-based coding for different mask types (rectangular and non-rectangular) and varying dilation values.



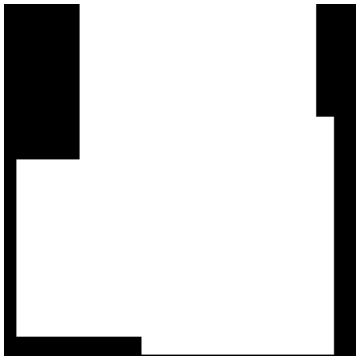
(a) Coded image with no ROI dilation.



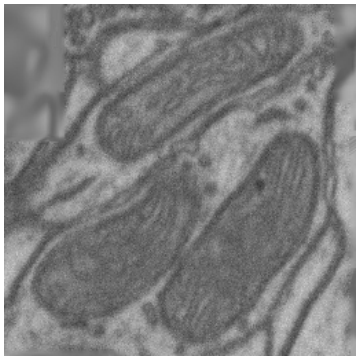
(b) ROI definition without dilation.



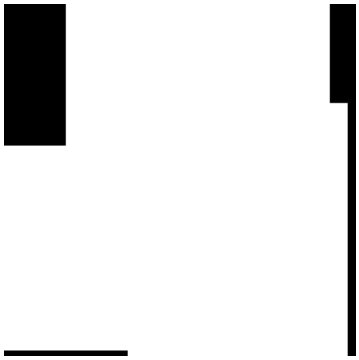
(c) Coded image with a 10-pixel ROI dilation.



(d) ROI definition with a 10-pixel dilation.

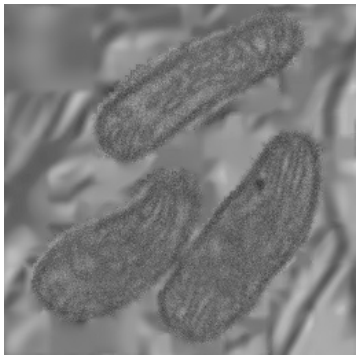


(e) Coded image with a 20-pixel ROI dilation.



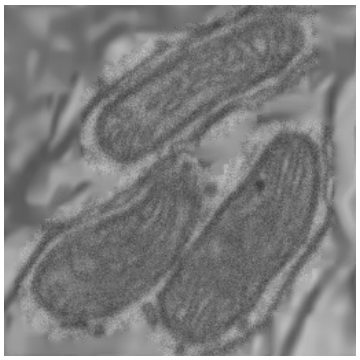
(f) ROI definition with a 20-pixel dilation.

Figure B.1: Examples of ROI-based coding with $QP_B = 51$ and $QP_F = 22$ for VVC, considering rectangular ROIs and varying dilation levels (no dilation at the top, 10-pixel dilation in the middle, 20-pixel dilation at the bottom). The resulting coded images are presented in the left column, with the respective ROI definitions shown on the right.



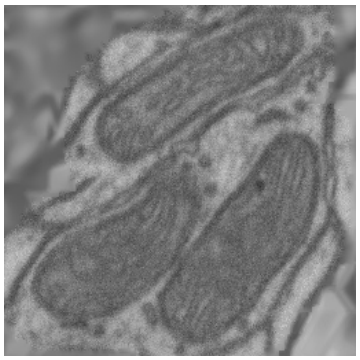
(a) Coded image with no ROI dilation.

(b) ROI definition without dilation.

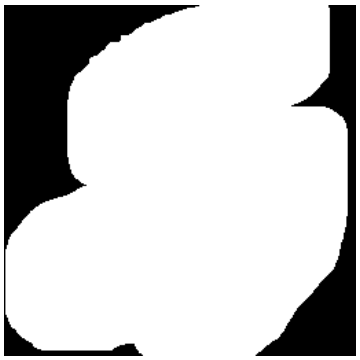


(c) Coded image with a 10-pixel ROI dilation.

(d) ROI definition with a 10-pixel dilation.



(e) Coded image with a 20-pixel ROI dilation.



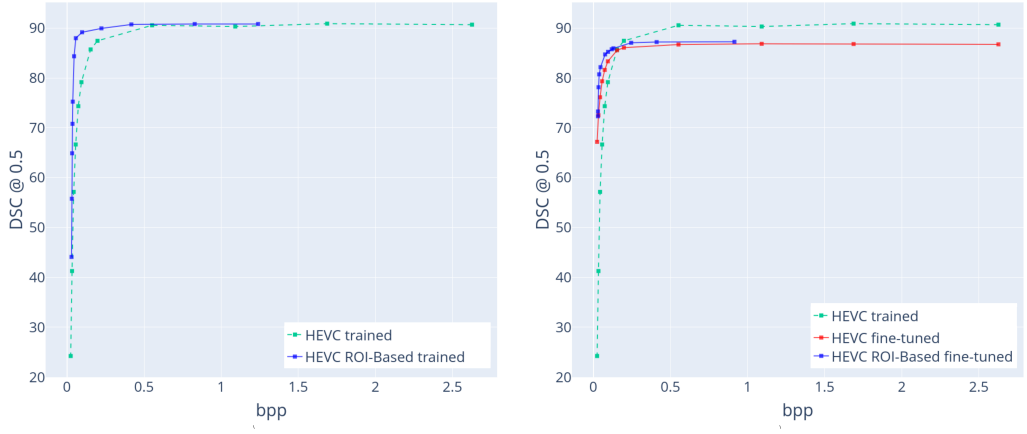
(f) ROI definition with a 20-pixel dilation.

Figure B.2: Examples of ROI-based coding with $QP_B = 51$ and $QP_F = 22$ for VVC, considering non-rectangular ROIs and varying dilation levels (no dilation at the top, 10-pixel dilation in the middle, 20-pixel dilation at the bottom). The resulting coded images are presented in the left column, with the respective ROI definitions shown on the right.

# C

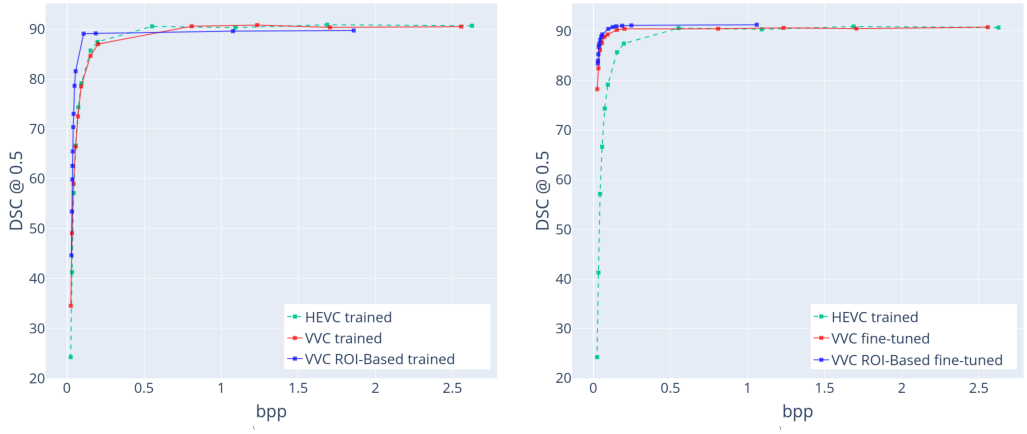
## APPENDIX C

This appendix presents the rate-performance curves for each dataset and task. Each figure includes graphs for both standard and ROI-based coding, as well as for the trained and fine-tuned versions of the models.




(a) Trained model on HEVC-coded images.

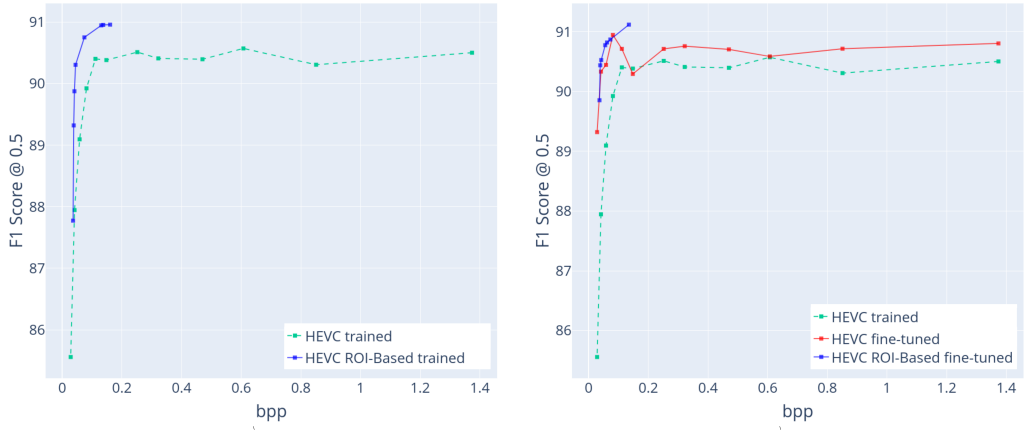
(b) Fine-tuned model on HEVC-coded images.




(c) Trained model on VVC-coded images.

(d) Fine-tuned model on VVC-coded images.

Figure C.1: Segmentation performance comparison between Lucchi++ images coded using standard coding (red line) and ROI-based coding (blue line) against the baseline (green line; trained model on standard HEVC-coded images). The top row shows results for the HEVC codec, while the bottom row corresponds to the VVC codec. The left column represents trained models, and the right column represents fine-tuned models.



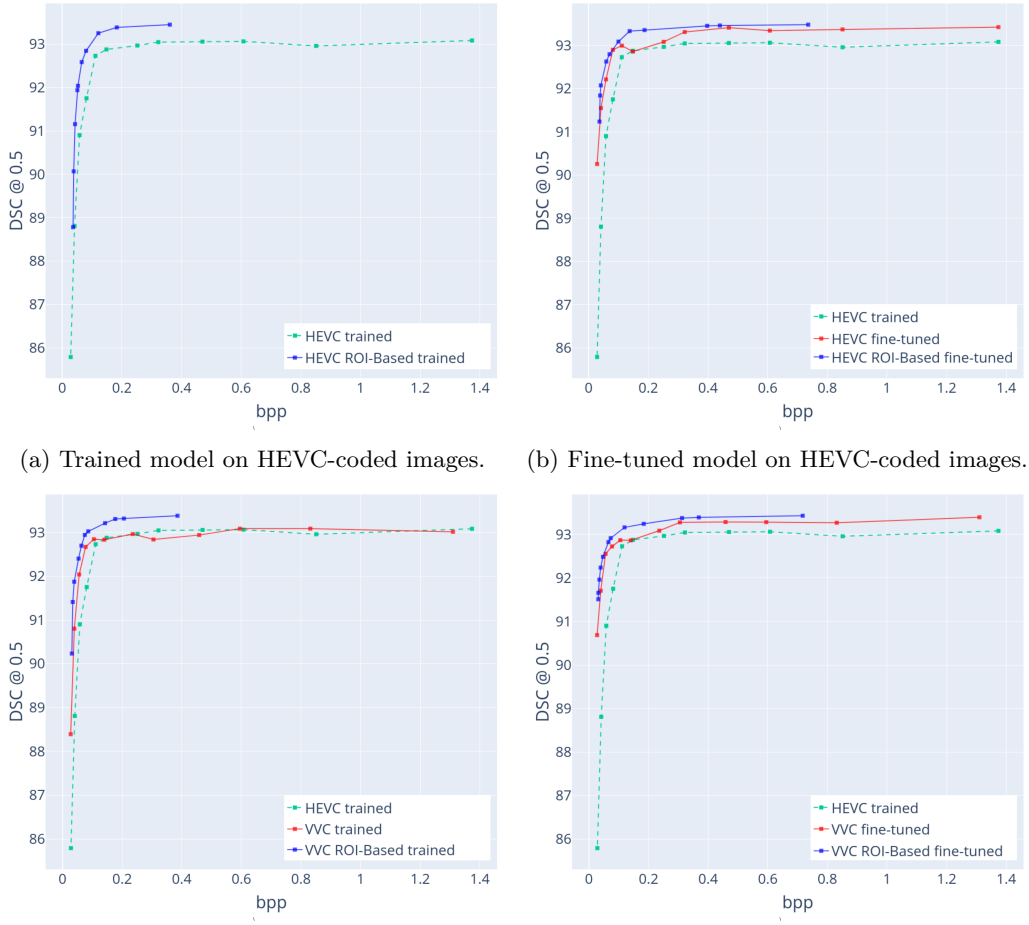

(a) Trained model on HEVC-coded images.

(b) Fine-tuned model on HEVC-coded images.

(c) Trained model on VVC-coded images.

(d) Fine-tuned model on VVC-coded images.

Figure C.2: Detection performance comparison between Kasthuri++ images coded using standard coding (red line) and ROI-based coding (blue line) against the baseline (green line; trained model on standard HEVC-coded images). The top row shows results for the HEVC codec, while the bottom row corresponds to the VVC codec. The left column represents trained models, and the right column represents fine-tuned models.




(a) Trained model on HEVC-coded images.

(b) Fine-tuned model on HEVC-coded images.

(c) Trained model on VVC-coded images.

(d) Fine-tuned model on VVC-coded images.

Figure C.3: Segmentation performance comparison between Kasthuri++ images coded using standard coding (red line) and ROI-based coding (blue line) against the baseline (green line; trained model on standard HEVC-coded images). The top row shows results for the HEVC codec, while the bottom row corresponds to the VVC codec. The left column represents trained models, and the right column represents fine-tuned models.

# D

## APPENDIX D

In this appendix, the parameters for each point selected through the proposed Pareto Frontier optimization are specified for each model and ROI-based codec.

Table D.1: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the trained detection model on the ROI-based HEVC encoder for the Lucchi++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | **Mask Type** | **Dilation** | $\mathbf{F_1}$ | **bpp** |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 41.401 | 0.030 |
| 51 | 47 | non-rectangular | 0 | 54.484 | 0.032 |
| 51 | 45 | non-rectangular | 0 | 64.337 | 0.034 |
| 51 | 43 | non-rectangular | 0 | 71.315 | 0.036 |
| 51 | 41 | non-rectangular | 0 | 76.297 | 0.038 |
| 51 | 37 | non-rectangular | 0 | 86.900 | 0.047 |
| 51 | 35 | non-rectangular | 0 | 90.992 | 0.058 |
| 51 | 32 | non-rectangular | 0 | 91.482 | 0.094 |
| 51 | 27 | non-rectangular | 0 | 91.544 | 0.173 |
| 30 | 27 | rectangular | 15 | 91.807 | 1.290 |

Table D.2: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the trained detection model on the ROI-based VVC encoder for the Lucchi++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | **Mask Type** | **Dilation** | $\mathbf{F_1}$ | **bpp** |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 40.242 | 0.029 |
| 51 | 47 | rectangular | 5 | 52.112 | 0.033 |
| 51 | 45 | rectangular | 5 | 61.674 | 0.036 |
| 51 | 41 | rectangular | 0 | 70.069 | 0.041 |
| 51 | 41 | rectangular | 5 | 73.998 | 0.044 |
| 51 | 37 | rectangular | 0 | 80.234 | 0.050 |
| 51 | 37 | rectangular | 5 | 83.361 | 0.054 |
| 51 | 35 | rectangular | 0 | 84.480 | 0.056 |
| 51 | 35 | rectangular | 5 | 86.518 | 0.062 |
| 51 | 35 | rectangular | 10 | 87.338 | 0.069 |
| 51 | 32 | non-rectangular | 0 | 91.578 | 0.108 |
| 51 | 22 | non-rectangular | 0 | 91.642 | 0.246 |
| 32 | 30 | rectangular | 5 | 91.901 | 1.015 |
| 27 | 22 | non-rectangular | 0 | 92.004 | 2.021 |

Table D.3: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the fine-tuned detection model on the ROI-based HEVC encoder for the Lucchi++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | **Mask Type** | **Dilation** | $\mathbf{F_1}$ | **bpp** |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 74.381 | 0.030 |
| 51 | 49 | rectangular | 0 | 75.704 | 0.030 |
| 51 | 47 | non-rectangular | 0 | 78.690 | 0.032 |
| 51 | 45 | non-rectangular | 0 | 81.295 | 0.034 |
| 51 | 41 | non-rectangular | 0 | 84.183 | 0.038 |
| 49 | 37 | non-rectangular | 0 | 86.770 | 0.055 |
| 47 | 35 | non-rectangular | 0 | 88.400 | 0.077 |
| 43 | 35 | rectangular | 5 | 89.344 | 0.119 |
| 43 | 32 | non-rectangular | 0 | 89.832 | 0.143 |
| 41 | 30 | non-rectangular | 0 | 90.261 | 0.196 |
| 41 | 30 | rectangular | 0 | 90.304 | 0.219 |
| 37 | 30 | rectangular | 0 | 90.402 | 0.299 |

Table D.4: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the fine-tuned detection model on the ROI-based VVC encoder for the Lucchi++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | Mask Type | Dilation | $\mathbf{F_1}$ | bpp |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 82.745 | 0.029 |
| 51 | 47 | non-rectangular | 0 | 84.533 | 0.032 |
| 51 | 45 | rectangular | 0 | 86.364 | 0.035 |
| 49 | 43 | non-rectangular | 0 | 87.541 | 0.045 |
| 51 | 41 | rectangular | 10 | 87.654 | 0.046 |
| 47 | 45 | rectangular | 10 | 87.978 | 0.050 |
| 47 | 43 | non-rectangular | 20 | 88.644 | 0.060 |
| 45 | 41 | rectangular | 20 | 89.217 | 0.074 |
| 43 | 37 | rectangular | 5 | 89.854 | 0.097 |
| 41 | 37 | rectangular | 5 | 90.310 | 0.115 |
| 37 | 35 | rectangular | 20 | 90.686 | 0.181 |
| 41 | 30 | rectangular | 20 | 91.011 | 0.374 |
| 35 | 32 | non-rectangular | 20 | 91.060 | 0.410 |
| 32 | 27 | non-rectangular | 0 | 91.401 | 1.051 |
| 27 | 22 | non-rectangular | 20 | 91.525 | 2.121 |

Table D.5: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the trained segmentation model on the ROI-based HEVC encoder for the Lucchi++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | Mask Type | Dilation | DSC | bpp |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 44.095 | 0.030 |
| 51 | 47 | non-rectangular | 0 | 55.700 | 0.032 |
| 51 | 45 | non-rectangular | 0 | 64.887 | 0.034 |
| 51 | 43 | non-rectangular | 0 | 70.792 | 0.036 |
| 51 | 41 | non-rectangular | 0 | 75.234 | 0.038 |
| 51 | 37 | non-rectangular | 0 | 84.336 | 0.047 |
| 51 | 35 | non-rectangular | 0 | 87.963 | 0.058 |
| 49 | 35 | rectangular | 15 | 89.150 | 0.098 |
| 37 | 35 | non-rectangular | 20 | 89.929 | 0.224 |
| 35 | 32 | non-rectangular | 20 | 90.719 | 0.418 |
| 32 | 30 | non-rectangular | 10 | 90.798 | 0.829 |
| 30 | 27 | rectangular | 0 | 90.805 | 1.240 |

Table D.6: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the trained segmentation model on the ROI-based VVC encoder for the Lucchi++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | Mask Type | Dilation | DSC | bpp |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 44.653 | 0.029 |
| 51 | 47 | non-rectangular | 0 | 53.433 | 0.032 |
| 51 | 45 | non-rectangular | 0 | 59.863 | 0.035 |
| 51 | 45 | rectangular | 5 | 62.577 | 0.036 |
| 51 | 43 | rectangular | 0 | 65.467 | 0.038 |
| 51 | 41 | rectangular | 0 | 70.348 | 0.041 |
| 51 | 41 | rectangular | 5 | 73.004 | 0.044 |
| 51 | 37 | rectangular | 0 | 78.639 | 0.050 |
| 51 | 35 | rectangular | 0 | 81.562 | 0.056 |
| 51 | 32 | non-rectangular | 0 | 89.081 | 0.108 |
| 41 | 32 | non-rectangular | 0 | 89.138 | 0.187 |
| 32 | 27 | rectangular | 10 | 89.587 | 1.077 |
| 30 | 22 | non-rectangular | 20 | 89.713 | 1.860 |

Table D.7: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the fine-tuned segmentation model on the ROI-based HEVC encoder for the Lucchi++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | Mask Type | Dilation | DSC | bpp |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 72.309 | 0.030 |
| 51 | 49 | rectangular | 0 | 73.262 | 0.030 |
| 51 | 43 | non-rectangular | 0 | 78.129 | 0.034 |
| 51 | 41 | non-rectangular | 0 | 80.712 | 0.038 |
| 51 | 37 | non-rectangular | 0 | 82.151 | 0.047 |
| 47 | 35 | non-rectangular | 0 | 84.707 | 0.077 |
| 43 | 37 | non-rectangular | 0 | 85.225 | 0.095 |
| 43 | 35 | rectangular | 5 | 85.739 | 0.119 |
| 43 | 35 | non-rectangular | 15 | 85.929 | 0.131 |
| 37 | 32 | non-rectangular | 0 | 87.039 | 0.246 |
| 35 | 32 | rectangular | 15 | 87.185 | 0.411 |
| 32 | 27 | rectangular | 10 | 87.239 | 0.915 |

Table D.8: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the fine-tuned segmentation model on the ROI-based VVC encoder for the Lucchi++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | Mask Type | Dilation | DSC | bpp |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 83.404 | 0.029 |
| 51 | 49 | non-rectangular | 5 | 83.955 | 0.030 |
| 51 | 47 | non-rectangular | 0 | 85.281 | 0.032 |
| 51 | 45 | rectangular | 0 | 86.770 | 0.035 |
| 51 | 43 | non-rectangular | 0 | 87.364 | 0.038 |
| 49 | 43 | non-rectangular | 0 | 88.162 | 0.045 |
| 49 | 41 | rectangular | 5 | 88.813 | 0.051 |
| 47 | 41 | non-rectangular | 0 | 89.271 | 0.058 |
| 43 | 37 | rectangular | 5 | 90.402 | 0.097 |
| 41 | 35 | rectangular | 5 | 90.759 | 0.125 |
| 41 | 35 | non-rectangular | 20 | 90.932 | 0.149 |
| 41 | 32 | non-rectangular | 0 | 91.028 | 0.187 |
| 37 | 32 | non-rectangular | 0 | 91.094 | 0.247 |
| 32 | 30 | non-rectangular | 20 | 91.231 | 1.060 |

Table D.9: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the trained detection model on the ROI-based HEVC encoder for the Kasthuri++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | Mask Type | Dilation | $\mathbf{F_1}$ | bpp |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 87.774 | 0.036 |
| 51 | 47 | non-rectangular | 0 | 89.322 | 0.039 |
| 51 | 45 | non-rectangular | 0 | 89.877 | 0.041 |
| 51 | 45 | rectangular | 5 | 90.307 | 0.045 |
| 47 | 45 | rectangular | 15 | 90.753 | 0.074 |
| 43 | 41 | rectangular | 0 | 90.949 | 0.131 |
| 45 | 37 | non-rectangular | 20 | 90.956 | 0.137 |
| 43 | 37 | rectangular | 15 | 90.960 | 0.161 |

Table D.10: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the trained detection model on the ROI-based VVC encoder for the Kasthuri++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | Mask Type | Dilation | $\mathbf{F_1}$ | bpp |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 89.050 | 0.031 |
| 51 | 49 | non-rectangular | 10 | 89.540 | 0.032 |
| 51 | 47 | non-rectangular | 0 | 89.948 | 0.034 |
| 51 | 47 | rectangular | 5 | 90.124 | 0.036 |
| 51 | 47 | rectangular | 10 | 90.191 | 0.037 |
| 37 | 35 | rectangular | 15 | 90.349 | 0.043 |
| 49 | 43 | non-rectangular | 20 | 90.728 | 0.062 |
| 47 | 41 | non-rectangular | 10 | 90.826 | 0.078 |
| 45 | 41 | rectangular | 15 | 90.925 | 0.102 |
| 45 | 32 | rectangular | 0 | 90.937 | 0.134 |

Table D.11: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the fine-tuned detection model on the ROI-based HEVC encoder for the Kasthuri++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | Mask Type | Dilation | $\mathbf{F_1}$ | bpp |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 89.854 | 0.036 |
| 51 | 49 | rectangular | 20 | 90.440 | 0.039 |
| 51 | 47 | non-rectangular | 15 | 90.527 | 0.042 |
| 49 | 45 | rectangular | 0 | 90.776 | 0.056 |
| 49 | 47 | rectangular | 20 | 90.821 | 0.062 |
| 47 | 45 | non-rectangular | 10 | 90.871 | 0.073 |
| 43 | 41 | rectangular | 15 | 91.120 | 0.135 |

Table D.12: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the fine-tuned detection model on the ROI-based VVC encoder for the Kasthuri++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | Mask Type | Dilation | $\mathbf{F_1}$ | bpp |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 90.179 | 0.031 |
| 51 | 49 | rectangular | 0 | 90.283 | 0.032 |
| 51 | 49 | rectangular | 5 | 90.392 | 0.032 |
| 51 | 47 | rectangular | 20 | 90.787 | 0.039 |
| 47 | 43 | non-rectangular | 15 | 90.978 | 0.073 |
| 45 | 43 | non-rectangular | 5 | 91.031 | 0.088 |

Table D.13: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the trained segmentation model on the ROI-based HEVC encoder for the Kasthuri++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | Mask Type | Dilation | DSC | bpp |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 88.782 | 0.036 |
| 51 | 47 | non-rectangular | 0 | 90.067 | 0.039 |
| 51 | 47 | rectangular | 15 | 91.156 | 0.043 |
| 51 | 45 | rectangular | 20 | 91.938 | 0.051 |
| 51 | 43 | rectangular | 10 | 92.039 | 0.053 |
| 51 | 41 | non-rectangular | 20 | 92.585 | 0.065 |
| 47 | 43 | rectangular | 10 | 92.845 | 0.080 |
| 49 | 35 | non-rectangular | 20 | 93.251 | 0.121 |
| 47 | 32 | rectangular | 20 | 93.383 | 0.183 |
| 47 | 22 | non-rectangular | 20 | 93.448 | 0.361 |

Table D.14: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the trained segmentation model on the ROI-based VVC encoder for the Kasthuri++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | Mask Type | Dilation | DSC | bpp |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 90.232 | 0.031 |
| 51 | 47 | non-rectangular | 0 | 91.411 | 0.034 |
| 51 | 47 | rectangular | 20 | 91.873 | 0.039 |
| 51 | 43 | rectangular | 20 | 92.401 | 0.054 |
| 51 | 41 | rectangular | 20 | 92.696 | 0.063 |
| 47 | 43 | non-rectangular | 20 | 92.942 | 0.075 |
| 47 | 41 | rectangular | 20 | 93.023 | 0.086 |
| 45 | 35 | non-rectangular | 20 | 93.210 | 0.143 |
| 45 | 32 | non-rectangular | 20 | 93.305 | 0.177 |
| 45 | 30 | non-rectangular | 20 | 93.316 | 0.206 |
| 45 | 22 | rectangular | 20 | 93.380 | 0.386 |

Table D.15: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the fine-tuned segmentation model on the ROI-based HEVC encoder for the Kasthuri++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | Mask Type | Dilation | DSC | bpp |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 91.237 | 0.036 |
| 51 | 47 | non-rectangular | 0 | 91.841 | 0.039 |
| 51 | 45 | non-rectangular | 0 | 92.075 | 0.041 |
| 51 | 43 | rectangular | 20 | 92.627 | 0.059 |
| 49 | 43 | rectangular | 20 | 92.796 | 0.070 |
| 45 | 43 | rectangular | 10 | 93.088 | 0.100 |
| 45 | 37 | non-rectangular | 20 | 93.331 | 0.137 |
| 41 | 37 | rectangular | 10 | 93.355 | 0.187 |
| 35 | 32 | rectangular | 15 | 93.453 | 0.397 |
| 35 | 22 | non-rectangular | 0 | 93.461 | 0.440 |
| 30 | 22 | non-rectangular | 5 | 93.481 | 0.736 |

Table D.16: Selected points by the Pareto Frontier optimisation parameters and the corresponding performance of the fine-tuned segmentation model on the ROI-based VVC encoder for the Kasthuri++ dataset.

| $\mathbf{QP_B}$ | $\mathbf{QP_F}$ | Mask Type | Dilation | DSC | bpp |
|---|---|---|---|---|---|
| 51 | 49 | non-rectangular | 0 | 91.510 | 0.031 |
| 51 | 49 | rectangular | 0 | 91.659 | 0.032 |
| 51 | 47 | rectangular | 0 | 91.962 | 0.035 |
| 51 | 47 | rectangular | 20 | 92.238 | 0.039 |
| 49 | 47 | non-rectangular | 20 | 92.486 | 0.047 |
| 47 | 45 | rectangular | 15 | 92.824 | 0.066 |
| 47 | 43 | non-rectangular | 15 | 92.915 | 0.073 |
| 43 | 41 | rectangular | 5 | 93.162 | 0.120 |
| 41 | 35 | non-rectangular | 5 | 93.244 | 0.184 |
| 37 | 32 | non-rectangular | 15 | 93.377 | 0.312 |
| 35 | 32 | rectangular | 15 | 93.391 | 0.369 |
| 30 | 27 | non-rectangular | 20 | 93.430 | 0.717 |

# APPENDIX E

This appendix discusses in detail the challenge found related to the interpolation scheme used when calculating the integral of curves for BD metrics. Initial tests employed a third-order polynomial interpolation, rather than the PCHIP method described in Section 4.2, and corresponding results are presented in Tables E.1 and E.2. However, inconsistencies were observed between the numerical results in the tables and the corresponding rate-performance curves. For example, in Table E.1, the detection model trained on images coded with the ROI-based HEVC encoder shows an average performance decrease of 3.69% compared to standard HEVC for the same model. Yet, when examining the corresponding curves in Figure E.1, the ROI-based HEVC outperforms standard HEVC.

This discrepancy arises because the third-order polynomial interpolation defines a global third-order function for the entire set of points. For curves with sparse and unevenly spaced data points, this global approach introduces oscillations, leading to inconsistent numerical results. Figure E.2 illustrates this issue, with the blue line representing the third-order polynomial interpolation. In contrast, the PCHIP interpolation algorithm (red line) calculates independent functions between each pair of points, reducing oscillations and producing more accurate results. The results obtained using the more adequate approach (PCHIP), are discussed in Section 4.4.2.

Table E.1: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for trained and fine-tuned models using standard and ROI-based HEVC and VVC codecs on the Lucchi++ dataset. All metrics use as baseline the trained model with standard HEVC encoding and are calculated using a third-order polynomial fit for curved integration.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| | | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Trained Model** | **HEVC** | - | - | - | - |
| | **VVC** | 5.83 | 1.65 | 7.62 | 2.22 |
| | **HEVC-ROI** | -26.15 | -3.69 | -22.77 | 12.09 |
| | **VVC-ROI** | -17.10 | 9.17 | -12.55 | 6.44 |
| **Fine-tuned Model** | **HEVC** | -53.55 | 11.66 | -44.89 | 8.58 |
| | **VVC** | -75.96 | 14.96 | -82.37 | 15.31 |
| | **HEVC-ROI** | -78.12 | **20.73** | -66.62 | 9.14 |
| | **VVC-ROI** | **-84.26** | 13.32 | **-90.30** | **15.80** |

Table E.2: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for trained and fine-tuned models using standard and ROI-based HEVC and VVC codecs on the Kasthuri++ dataset. All metrics use as baseline the trained model with standard HEVC encoding and are calculated using a third-order polynomial fit for curved integration.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| | | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Trained Model** | **HEVC** | - | - | - | - |
| | **VVC** | -68.99 | 0.46 | 11.36 | 0.57 |
| | **HEVC-ROI** | -37.84 | 1.38 | -26.91 | 0.86 |
| | **VVC-ROI** | -68.49 | 1.79 | -46.67 | **1.25** |
| **Fine-tuned Model** | **HEVC** | **-92.36** | 0.84 | -35.06 | 0.93 |
| | **VVC** | -84.23 | 0.95 | -48.02 | 0.98 |
| | **HEVC-ROI** | -83.11 | 1.60 | -60.94 | 1.01 |
| | **VVC-ROI** | -89.98 | **2.68** | **-68.25** | 1.21 |

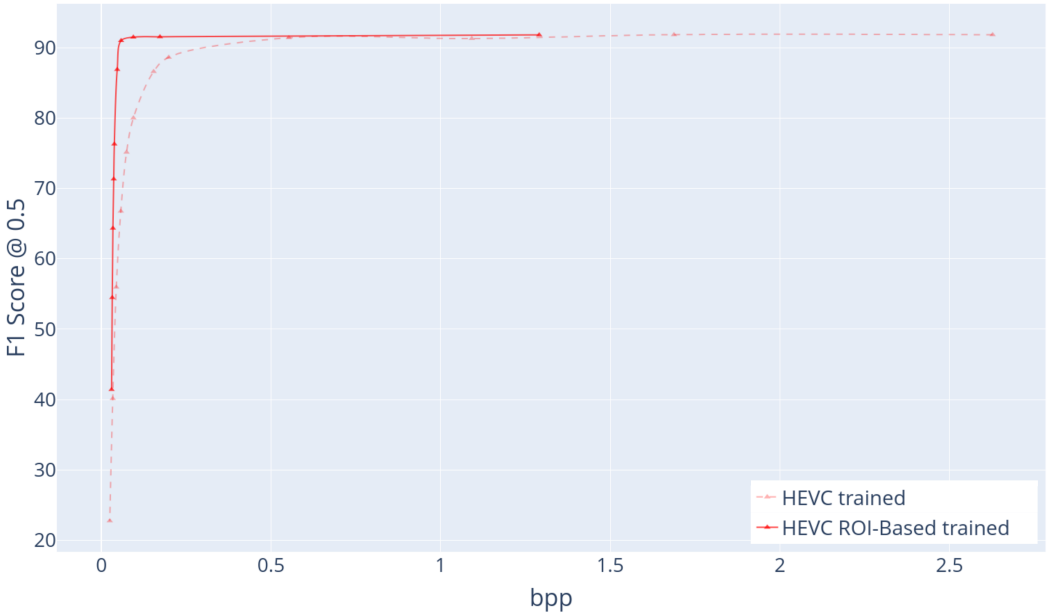



Figure E.1: Performance comparison of the YOLOv8 model trained on uncompressed images with two different approaches: standard HEVC encoding (red dashed line) and ROI-based HEVC encoding (red solid line).

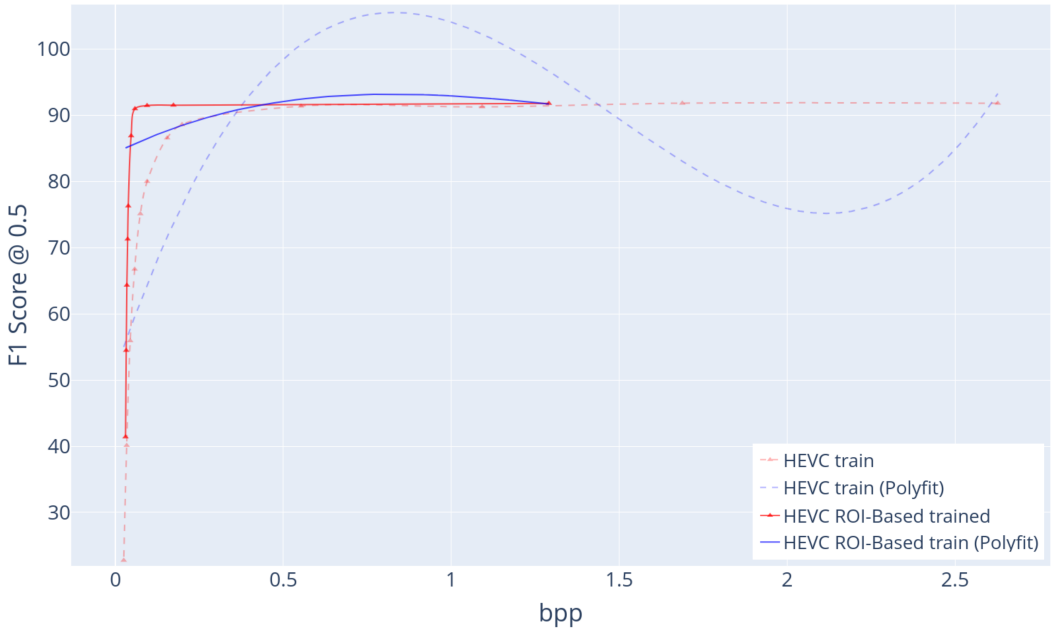



Figure E.2: Comparison between PCHIP interpolation (red lines) and third-order polynomial interpolation (blue lines), for two rate-performance curves: standard HEVC encoding (dashed line) and ROI-based HEVC encoding (solid line). Results are presented for the YOLOv8 model trained on uncompressed images.

# APPENDIX F

This appendix presents additional results on the influence of ROI-based coding parameters for different datasets and model versions (trained or fine-tuned).

Table F.1: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for fine-tuned detection and segmentation models using the ROI-based HEVC codec on the Lucchi++ dataset. Results are evaluated for varying mask types (rectangular and non-rectangular) and dilation levels (in pixels) of the binary mask. All metrics use as baseline the trained model with standard HEVC encoding.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| **Mask Type** | **Dilation** | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Rectangular** | **0** | -62.79 | 20.94 | -51.27 | 9.81 |
| | **5** | -60.72 | 15.68 | -50.55 | **13.66** |
| | **10** | -59.34 | 17.84 | -47.48 | 8.61 |
| | **15** | -58.54 | 16.10 | -46.70 | 12.28 |
| | **20** | -55.26 | 16.99 | -44.00 | 6.94 |
| **Non-rectangular** | **0** | **-64.49** | 20.91 | **-54.30** | 9.88 |
| | **5** | -61.84 | 16.69 | -50.22 | 8.81 |
| | **10** | -60.06 | 14.87 | -48.00 | 7.27 |
| | **15** | -58.67 | **21.92** | -45.57 | 7.59 |
| | **20** | -56.42 | 17.48 | -44.72 | 6.31 |

Table F.2: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for trained detection and segmentation models using the ROI-based VVC codec on the Lucchi++ dataset. Results are evaluated for varying mask types (rectangular and non-rectangular) and dilation levels (in pixels) of the binary mask. All metrics use as baseline the trained model with standard HEVC encoding.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| **Mask Type** | **Dilation** | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Rectangular** | **0** | -32.99 | 8.46 | -27.43 | 5.63 |
| | **5** | **-33.74** | **9.66** | -26.45 | 6.23 |
| | **10** | -31.98 | 7.38 | -24.46 | 5.52 |
| | **15** | -30.06 | 6.73 | -24.72 | 5.33 |
| | **20** | -28.31 | 6.15 | -24.13 | 4.39 |
| **Non-rectangular** | **0** | -33.65 | 8.19 | **-33.47** | **6.81** |
| | **5** | -28.28 | 6.83 | -20.27 | 3.59 |
| | **10** | -27.53 | 5.94 | -20.08 | 3.83 |
| | **15** | -26.11 | 5.61 | -19.30 | 3.38 |
| | **20** | -24.60 | 5.05 | -19.38 | 3.32 |

Table F.3: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for fine-tuned detection and segmentation models using the ROI-based VVC codec on the Lucchi++ dataset. Results are evaluated for varying mask types (rectangular and non-rectangular) and dilation levels (in pixels) of the binary mask. All metrics use as baseline the trained model with standard HEVC encoding.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| **Mask Type** | **Dilation** | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Rectangular** | **0** | **-73.51** | **15.35** | -82.59 | 13.94 |
| | **5** | -72.33 | 12.90 | -84.00 | 13.15 |
| | **10** | -72.02 | 12.46 | -82.32 | 15.14 |
| | **15** | -70.59 | 11.58 | -82.63 | 14.59 |
| | **20** | -72.37 | 11.63 | -82.66 | 14.96 |
| **Non-rectangular** | **0** | -71.77 | 15.04 | **-84.30** | **16.29** |
| | **5** | -70.98 | 12.33 | -83.66 | 15.52 |
| | **10** | -71.09 | 12.03 | -82.32 | 12.96 |
| | **15** | -69.96 | 11.23 | -81.67 | 14.35 |
| | **20** | -70.08 | 11.21 | -81.79 | 14.65 |

Table F.4: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for trained detection and segmentation models using the ROI-based HEVC codec on the Kasthuri++ dataset. Results are evaluated for varying mask types (rectangular and non-rectangular) and dilation levels (in pixels) of the binary mask. All metrics use as baseline the trained model with standard HEVC encoding.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| **Mask Type** | **Dilation** | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Rectangular** | **0** | -46.03 | 1.03 | -23.29 | 0.48 |
| | **5** | -49.20 | 0.84 | -26.69 | 0.70 |
| | **10** | -42.59 | 0.92 | -30.50 | 0.73 |
| | **15** | -47.96 | 1.01 | -31.38 | 0.90 |
| | **20** | **-52.03** | 0.77 | -31.43 | **1.02** |
| **Non-rectangular** | **0** | -50.03 | **1.15** | -23.69 | 0.59 |
| | **5** | -44.56 | 0.99 | -22.71 | 0.52 |
| | **10** | -48.12 | 0.81 | -28.54 | 0.74 |
| | **15** | -47.79 | 0.85 | -30.95 | 0.89 |
| | **20** | -48.14 | 0.92 | **-32.42** | 0.88 |

Table F.5: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for fine-tuned detection and segmentation models using the ROI-based HEVC codec on the Kasthuri++ dataset. Results are evaluated for varying mask types (rectangular and non-rectangular) and dilation levels (in pixels) of the binary mask. All metrics use as baseline the trained model with standard HEVC encoding.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| **Mask Type** | **Dilation** | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Rectangular** | **0** | -73.72 | 0.90 | -50.53 | 0.89 |
| | **5** | -89.10 | 0.85 | -50.93 | 0.88 |
| | **10** | -91.22 | 0.86 | -50.59 | 0.91 |
| | **15** | -91.53 | 1.44 | -51.07 | 0.92 |
| | **20** | **-93.53** | 0.85 | **-52.52** | 0.85 |
| **Non-rectangular** | **0** | -86.21 | 0.95 | -48.83 | **1.03** |
| | **5** | -87.26 | 1.76 | -52.38 | 0.91 |
| | **10** | -90.29 | **2.12** | -49.09 | 0.83 |
| | **15** | -92.43 | 1.75 | -49.66 | 0.98 |
| | **20** | -90.66 | 1.31 | -49.48 | 0.86 |

Table F.6: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for trained detection and segmentation models using the ROI-based VVC codec on the Kasthuri++ dataset. Results are evaluated for varying mask types (rectangular and non-rectangular) and dilation levels (in pixels) of the binary mask. All metrics use as baseline the trained model with standard HEVC encoding.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| **Mask Type** | **Dilation** | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Rectangular** | **0** | -71.05 | 1.52 | -42.92 | 0.90 |
| | **5** | -71.16 | 1.61 | -42.49 | 1.02 |
| | **10** | -69.88 | 1.77 | -41.65 | 1.34 |
| | **15** | -76.95 | 1.71 | -45.16 | 1.21 |
| | **20** | -75.66 | 1.70 | **-49.38** | 1.19 |
| **Non-rectangular** | **0** | -72.35 | 1.40 | -47.44 | **1.69** |
| | **5** | -71.61 | **1.84** | -40.18 | 1.00 |
| | **10** | **-79.33** | 1.35 | -43.43 | 1.42 |
| | **15** | -72.50 | 1.20 | -48.07 | 1.63 |
| | **20** | -69.80 | 1.48 | -44.96 | 1.18 |

Table F.7: BD-Rate ↓, BD-$F_1$ ↑ (detection), and BD-DSC ↑ (segmentation) performance metrics for fine-tuned detection and segmentation models using the ROI-based VVC codec on the Kasthuri++ dataset. Results are evaluated for varying mask types (rectangular and non-rectangular) and dilation levels (in pixels) of the binary mask. All metrics use as baseline the trained model with standard HEVC encoding.

| | | **Detection** | | **Segmentation** | |
|---|---|---|---|---|---|
| **Mask Type** | **Dilation** | **↓BD-Rate (%)** | **↑BD-$F_1$ (%)** | **↓BD-Rate (%)** | **↑BD-DSC (%)** |
| **Rectangular** | **0** | -93.25 | 1.09 | -56.43 | 1.11 |
| | **5** | -92.59 | 1.82 | -55.38 | 1.05 |
| | **10** | -90.41 | 0.99 | -53.03 | 1.09 |
| | **15** | -93.25 | 0.91 | -54.74 | **1.23** |
| | **20** | -93.27 | 2.08 | -56.76 | 1.00 |
| **Non-rectangular** | **0** | -93.08 | 1.04 | -55.11 | 1.04 |
| | **5** | -88.80 | 2.36 | -52.39 | 1.01 |
| | **10** | -91.73 | 0.91 | -50.77 | 0.98 |
| | **15** | -92.77 | **2.50** | **-56.93** | 1.06 |
| | **20** | **-93.44** | 1.14 | -56.68 | 1.05 |

# DECLARATION

Declaro, sob compromisso de honra, que o trabalho apresentado nesta dissertação, com o título *"ROI-Based coding of biomedical images for machine analysis"*, é original e foi realizado por Daniel Filipe da Silva Nicolau (2232623) sob orientação de Professor Sérgio M. M. Faria (sergio.faria@ipleiria.pt), Professor Lucas A. Thomaz (lucas.thomaz@ipleiria.pt) e Professor Luís M. N. Távora (luis.tavora@ipleiria.pt).

*Leiria, March 2025*

Daniel Filipe da Silva Nicolau