UCLouvain
Institute for Information
and Communication Technologies,
Electronics and Applied Mathematics

Optimized Frameworks for Deep Learning on Compressed Images

Karim El Khoury

Thesis submitted in fulfillment
of the requirements for the degree of
Docteur en sciences de l'ingénieur et technologie

Dissertation committee:

Prof. Benoît Macq (UCLouvain, Belgium) – Advisor
Prof. Christophe De Vleeschouwer (UCLouvain, Belgium)
Prof. Antonin Descampe (UCLouvain, Belgium)
Prof. Frédéric Dufaux (Université Paris-Saclay, France)
Dr. Gaël Rouvroy (intoPIX, Belgium)
Prof. David Bol (UCLouvain, Belgium) – Chair

"Demandez, et l'on vous donnera.
Cherchez, et vous trouverez.
Frappez, et l'on vous ouvrira."

— Matthieu 7:7

Acknowledgments

First and foremost, I would like to express my gratitude to my PhD advisor, **Prof. Benoît Macq**, for giving me the chance to pursue my PhD at the **Pilab**. Since our first interactions six years ago during the information theory course, we have developed a special bond that I deeply cherish. His unwavering positivity and determination to make things work are qualities I hold in the highest regard.

Second, I would also like to extend my sincere thanks to the members of the jury, namely **Prof. Christophe De Vleeschouwer**, **Prof. Antonin Descampe**, **Prof. David Bol**, **Prof. Frédéric Dufaux**, and **Dr. Gaël Rouvroy**. I am grateful for their insightful comments provided at various stages of my PhD. Their feedback has been undoubtedly instrumental in improving the quality of my work.

Third, I would like to express my thanks to the incredible colleagues who have supported me throughout my journey. Their impact has been profound, both personally and professionally. Among them, several individuals stand out. **Dr. Antoine Aspeel** has been a pillar of support and a true friend. His advice and empathy have been invaluable. **Nicolas Roisin** has been a devoted colleague and friend for the past eight years, beginning with our time as master's students and continuing through our teaching assistantships. I appreciate his many acts of kindness over the years and remain profoundly thankful for his unwavering support. **Dani Manjah** has been more than just a colleague; he has been a trusted confidant and companion throughout this challenging journey. The obstacles we have faced together have forged a bond built on mutual respect and trust. **Jonathan Samelson** has been a pleasure to work alongside and share an office with for over two years. I am grateful for the enriching discussions we have had, both professionally and personally. **Benoît Gérin** has played an big role in both my research and teaching journey. Witnessing the evolution of our relationship, from teacher-student to colleagues and eventually friends, has been a joy. **Maxime Zanella** has been a great companion during our travels for summer workshops. I appreciate the discussions and insights we have shared during my thesis and look forward to collaborating with

him in the future. **Maxence Wynen** has been an extraordinary source of comfort and camaraderie. Through our interactions, I have discovered a true friend who shares many of my values. **Tiffanie Godelaine** has left a lasting impression on my PhD. Among my 20+ co-authors, she is the person with whom I have co-authored the most papers. I hope that my guidance has made a positive impact on her, just as her growth has been a source of inspiration for me. **Manon Dausort** has been a great source of support. Her resilience is admirable, and I am confident that her determination will lead her to great success in the future. **Melanie Ghislain**'s kindness is incredibly contagious. I appreciate her continuous support and presence, especially during the latter stages of my PhD. I would also like to extend my thanks to **Adrien**, **Alain**, **Alexandre**, **Arthur**, **Clément C.**, **Clément F.**, **Colin**, **Damien**, **Dany**, **Eliott**, **Estelle**, **François**, **Frédéric**, **Gaëtan**, **Gauthier**, **Jérôme**, **Julie**, **Kaori**, **Lucas**, **Mathias**, **Nicolas B.**, **Nicolas D.**, **Pauline**, **Paul**, **Quentin**, **Rony**, **Sébastien**, **Sylvain**, **Thomas**, **Tim**, **Valentin**, **Victorien**, **Wei**, and many others for the many conversations and good times we have shared at the **Pilab**. I would like to thank **Isabelle** and **Patricia** for making our daily lives so much easier. Their work behind the scenes have been invaluable, and I am deeply appreciative of their efforts.

Fourth, I would like to extend my thanks to all the teaching staff I have had the pleasure of working with, namely **Anne-Sophie**, **Antoine**, **Benoît**, **Claude**, **David**, **Delphine**, **Francesco**, **Gilles**, **Jean-Charles**, **Jean-Pierre**, **Jérôme**, **Jonathan**, **Myriam**, **Nicolas**, **Olivier**, **Quentin**, **Rémi**, **Romain**, **Victorien**, and numerous others. I am grateful for all the collective efforts that have contributed to making teaching at the UCLouvain such a great experience.

Fifth, I would like to thank my host institute, **ICTEAM**, as well as my host faculty, **EPL**, for allowing me to have such enriching interactions over the past five years. I could not have chosen a better place to pursue my PhD.

Last but not least, I would like to express my heartfelt gratitude to my close friends and family, both in Belgium, Lebanon, and abroad. They made the PhD journey so much easier with their continuous support and encouragement. I would like to end by thanking my father **Fadi**, my mother **Rita** and my sister **Nour** for believing in me, I would not be where I am today without them.

Author's Publication List

Journal Articles

JA1 Karim El Khoury, Martin Fockedey, Eliott Brion, Benoît Macq, *Improved 3D U-Net robustness against JPEG 2000 compression for male pelvic organ segmentation in radiotherapy*," Journal of Medical Imaging — [1]

JA2 Karim El Khoury, Jonathan Samelson, Benoît Macq, *Deep Learning-Based Object Tracking via Compressed Domain Residual Frames*," Frontiers in Signal Processing — [2]

Conference Proceedings

CP1 Karim El Khoury, Tiffanie Godelaine, Simon Delvaux, Sebastien Lugan, Benoît Macq, *Streamlined Hybrid Annotation Framework using Scalable Codestream for Bandwidth-Restricted UAV Object Detection*," IEEE International Conference on Image Processing (ICIP), 2024 — [3]

CP2 Karim El Khoury, Maxence Wynen, Pietro Maggi, Meritxell Bach Cuadra, Benoît Macq, *Improving 3D Lesion Segmentation Robustness Against Image Compression in Multiple Sclerosis*," IEEE International Symposium on Biomedical Imaging (ISBI), 2024 — [4]

CP3 Karim El Khoury, Martin Fockedey, Eliott Brion, Benoît Macq, *Amélioration de la robustesse de l'U-Net 3D contre la compression JPEG 2000 pour la segmentation des organes pelviens masculins*," COmpression et REprésentation des Signaux Audiovisuels (CORESA), 2021 — [5]

CP4 Karim El Khoury, Maxime Zanella, Benoît Gérin, Tiffanie Godelaine, Benoît Macq, Saïd Mahmoudi, Christophe De Vleeschouwer, Ismail Ben Ayed, *Enhancing Remote Sensing Vision-Language Models for Zero-Shot Scene Classification*," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2025 — [6]

CP5 Manon Dausort, Tiffanie Godelaine, Maxime Zanella, Karim El Khoury, Benoît Macq, *Exploring Foundation Models Fine-Tuning for Cytology Classification*," IEEE International Symposium on Biomedical Imaging (ISBI), 2025 — [7]

Open-Source Benchmarking Tools

OS1 Martin Danhier, Karim El Khoury, Benoît Macq, *An open-source fine-grained benchmarking platform for wireless virtual reality*," International Conference on Virtual Reality and Mixed Reality, 2023 — [8]

OS2 Pauline Hermans, Tom Gautot, Karim El Khoury, Aloys du Bois d'Aische, Benoît Macq, *SVC: An open-source secure VCF compression tool with conditional random access*," Spatial Omics, 2024 — [9]

Abstract

Global data consumption has surged 100-fold over the past 15 years, aligning with the rise of deep learning and driving a massive shift in big data analysis. This growth has significantly impacted fields such as medical imaging, video surveillance, and remote sensing. However, it has also challenged existing data management infrastructures, necessitating a balance between data constraints and analysis constraints. This balancing act translates to a trade-off between data compression efficiency and deep learning task performance.

This thesis investigates strategies to better utilize data management infrastructures, aiming to minimize trade-offs between model performance and data compression. We explore this challenge through three distinct use cases: (1) improving deep learning segmentation robustness against image compression in medical imaging, (2) utilizing residual frames for object detection and tracking in video surveillance, and (3) developing a hybrid framework for bandwidth-constrained object detection in remote sensing.

Our work reveals that fine-tuning deep learning segmentation and detection models on compressed data can be highly effective in both learning and filtering out compression artifacts. Additionally, our work demonstrates that models trained on compressed data can be easily integrated into existing frameworks for medical imaging, video surveillance, and remote sensing, reducing storage and bandwidth demands while maintaining high model accuracy.

Contents

List of Figures

List of Tables

Acronyms

CBCT	Cone Beam Computed Tomography
CT	Computed Tomography
DSC	Dice Similarity Coefficient
HOTA	Higher Order Tracking Accuracy
IOU	Intersection Over Union
JPEG	Joint Photographic Experts Group
KIOU	Kalman Intersection Over Union
MPEG	Moving Picture Experts Group
MRI	Magnetic Resonance Imaging
SMBD	Symmetric Mean Boundary Distance
SORT	Simple Online and Realtime Tracking
UAV	Unmanned Aerial Vehicle
VCM	Video Coding for Machines
YOLO	You Only Look Once

1

Introduction

1.1 Thesis Motivation

We are living in the era of big data. Since 2010, global data consumption has seen a 100-fold surge, rising from 1.7 Zettabytes to an estimated 175 Zettabytes by 2025 [10]. This huge surge coincides with the rise of deep learning, driven by enhanced computational power and access to larger datasets. The rise of deep learning has brought artificial intelligence out of its second winter and has resulted in the emergence of all major large-scale models that shape the current research landscape [11–22]. Figure 1.1 shows the timeline alignment between the increase in data consumption and the release dates of popular deep learning models over the past 15 years.

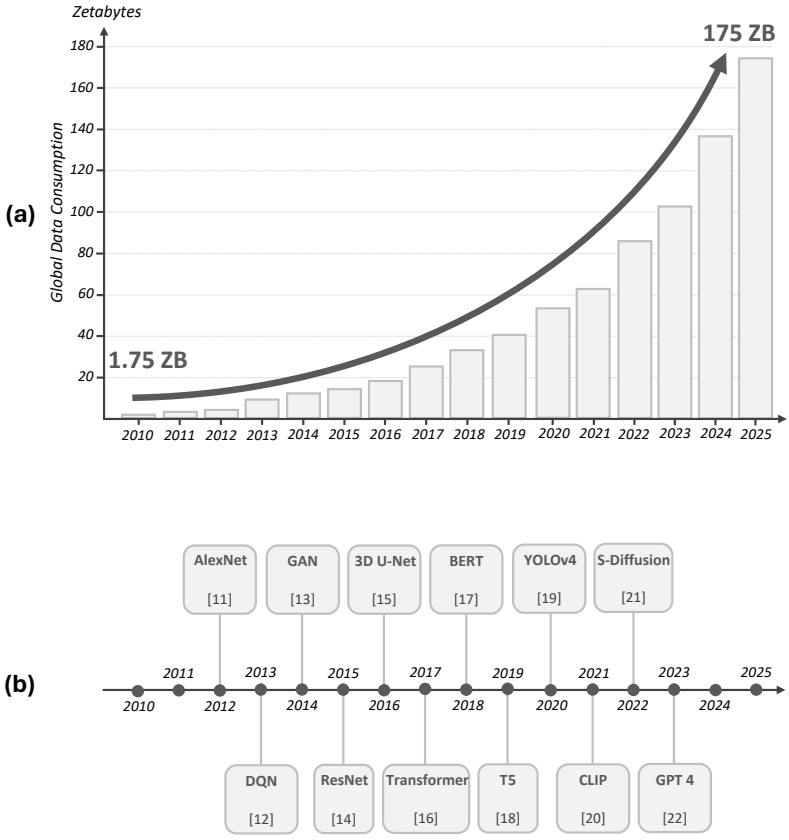

Fig. 1.1 Timeline showing (a) the increase in global data consumption and (b) the release dates of the most popular deep learning models.

This explosion of data and improved analysis capabilities has profoundly impacted various sectors. In healthcare, data-driven approaches have enhanced diagnostics and personalized medicine [23, 24]. Video surveillance has benefited from automated detection capabilities, enhancing security and monitoring [25, 26]. Progress in remote sensing has improved Earth observation, aiding environmental monitoring and disaster response [27, 28]. These advancements highlight the power of big data analysis across various fields.

However, these advancements in big data analysis require a trade-off between two constraints:

(1) Data Constraints (2) Analysis Constraints

(1) To handle the Data Constraints, the use of **Data Compression** becomes essential. It allows us to minimize storage costs and enables quick data transfer for real-time applications. These applications require substantial bandwidth, with real-time analytics accounting for up to 30% of global data consumption [10]. Additionally, as data volumes increase, storage becomes increasingly costly. Maintaining data centers requires significant energy for cooling and operations, which contributes to notable carbon emissions. For instance, data centers accounted for over 1% of global electricity use in 2018 [29].

(2) To effectively manage Analysis Constraints, deploying **Deep Learning** models is essential. These models excel in automating tasks such as classification, detection, and segmentation with high accuracy. This allows systems to operate with minimal oversight, reducing the need for human intervention. However, training deep learning models typically require large volumes of high-quality data. This can lead to challenges when systems are also faced with Data Constraints.

The trade-off between Data Constraints and Analysis Constraints becomes a balancing act between **Data Compression** efficiency and **Deep Learning** task performance.

1.2 Thesis Positioning

To handle the data compression and deep learning trade-off, the image processing community has divided itself into two research groups:

Group 1: Exploring ways to adapt data compression algorithms for machine vision in order to improve deep learning analysis.

Group 2: Exploring ways to train deep learning models on classical human vision-oriented compressed data to improve model performance.

Group 1 consists of researchers from both the image and video compression communities. The Joint Photographic Experts Group (JPEG) standardization committee is developing JPEG AI to optimize machine vision while maintaining quality for human vision [30]. This standard emphasizes efficient processing in the compressed domain with features like hardware and software efficiency and progressive decoding. Similarly, the Moving Picture Experts Group (MPEG) standardization committee has proposed Video Coding for Machines (VCM) to combine feature coding for machine vision with video coding for human vision [31]. However, while JPEG AI and MPEG VCM have shown promising results in compressing data for both human and machine vision [32,33], their widespread deployment remains quite challenging. Classical codecs are still deeply embedded in fields such as medical imaging, video surveillance, and remote sensing [34,35]. Since these new standards were launched in the late 2020s, they are still in early stages. Replacing the current data compression infrastructure requires significant effort from both standardization committees as well as hardware and software development companies.

Group 2 researchers on the other hand utilize existing data management infrastructure to fine-tune deep learning models to enhance their robustness to compression artifacts. Fine-tuning models on compressed data has shown promising results in improving robustness for various deep learning tasks such as classification, detection, and segmentation [36–41]. These works are based on the hypothesis that classical compression can serve as an effective feature extractor for both human and computer vision. While the acceptable trade-off between model performance and compression ratio is highly use case dependent, minimizing this trade-off opens up wide opportunities for the deployment of these fine-tuned models in resource-constrained fields that require both efficient data analysis and data compression [42–45].

In this thesis, we position our work alongside **Group 2** and put forward the following **Central Research Question**:

Central Research Question

Can we better utilize the existing data management infrastructure to improve model performance on compressed data?

1.3 Thesis Contributions

To address the **Central Research Question**, we explore the trade-off between model performance and data compression in **three distinct use cases** within three different research fields:

Use Case 1: Improved robustness of deep learning segmentation against image compression in medical imaging

Use Case 2: Utilizing residual frames for deep learning-based object detection and tracking in video surveillance

Use Case 3: Hybrid framework using scalable codestream for bandwidth-restricted object detection in remote sensing

The respective **Research Question** of each individual **Use Case** and their subsequent **Contributions** are introduced hereafter.

Note: Thesis Color Code

To facilitate reading, the rest of the thesis will follow a color code to distinguish each of the three use cases. The color code is as follows:

Use Case 1	Use Case 2	Use Case 3
Medical Imaging	Video Surveillance	Remote Sensing
↪ Blue	↪ Orange	↪ Green

1.3.1 Use Case 1: Improved robustness of deep learning segmentation against image compression in medical imaging

Accurate segmentation of medical images is crucial for correct patient diagnostics. Nevertheless, manual segmentation is a labor-intensive and time-consuming process. Consequently, there has been a significant shift towards utilizing deep learning techniques for automation of this process [46–48]. However, training deep learning models requires large datasets, highlighting the importance of efficient data management. Image compression is valuable in this context, as it significantly reduces storage needs and facilitates faster data transfer. This has led to the integration of various compression formats into medical imaging standards, notably JPEG 2000 into the DICOM standard [34]. Although compression effectively reduces file size, it can compromise image quality, potentially impacting segmentation accuracy. Encouraging results on natural images indicate that training models with compressed data can enhance semantic segmentation performance [37–39]. This leads us to ask **Research Question 1**:

Research Question 1

Can the robustness of deep learning-based segmentation to image compression be extended to medical imaging?

By fine-tuning the segmentation model, we show that:

Contribution A

A 3D segmentation model can learn and filter out data compression artifacts in medical imaging, even at very high compression ratios.

Contribution B

Data compression can be effectively deployed in medical data management systems with only a marginal drop in performance.

1.3.2 Use Case 2: Utilizing residual frames for deep learning-based object detection and tracking in video surveillance

Video surveillance systems are made up of extensive networks of edge-cameras deployed in urban areas. These systems utilize lightweight models on edge cameras for efficient onboard object detection and tracking, with detection results sent directly to a central server for analysis by decision-making units. Although effective, these edge models often perform below par compared to larger models that can be deployed on the server-side. These surveillance systems do not fully utilize other edge device assets onboard such as inter-frame encoders [49–51]. These encoders compress the video stream, producing motion vectors and residual frames that provide semantic information of the visual scene. Leveraging this information can improve video surveillance efficiency, maintain low bandwidth for scalability, and provide a privacy-friendly by-product. This leads us to ask **Research Question 2**:

Research Question 2

Can residual frames be utilized to improve detection accuracy in video surveillance systems?

By proposing an alternative framework that extracts compressed domain residual frames at the edge camera and sends them to a central server for inference using larger detection and tracking models, we show that:

Contribution C

A high-complexity network trained and tested on residual frames can outperform a low-complexity network trained and tested on raw frames in simple scenes.

Contribution D

A high-complexity network working with residual frames can compete with a low-complexity network trained and tested on raw frames, even in complex scenes.

1.3.3 Use Case 3: Hybrid framework using scalable codestream for bandwidth-restricted object detection in remote sensing

Hybrid detection is a time-gaining two-step process where deep learning models provide preliminary labeling of a given scene, followed by human experts refining those labels for reliability. This process is particularly useful in emergency scenarios where human annotators need to take quick and reliable decisions. However, in these emergency situations data transmission can create a bottleneck for hybrid detection frameworks due to the bandwidth restrictions. Additionally, physical barriers or infrastructure damage may disrupt communication lines even further. To tackle these challenges, utilizing existing scalable compression algorithms like JPEG 2000 can be highly effective [52–54]. In bandwidth-restricted scenarios, JPEG 2000's scalability facilitates efficient data transmission by allowing multi-resolution access to specific areas of interest. This streamlines data transmission and enhances the decision-making capabilities of first responders. This leads us to ask **Research Question 3**:

Research Question 3

Can we exploit the flexibility of the compression codestream to streamline a hybrid detection process?

By using the scalability of the JPEG 2000 codestream to address bandwidth issues and employing a fine-tuned deep learning model for initial detection at lower resolution, we show that:

Contribution E

Fine-tuning a deep learning network for object detection on low-resolution images remains effective while providing fast information access.

Contribution F

By utilizing scenario-specific constraints and the scalable codestream, the proposed hybrid framework can significantly reduce emergency response time.

1.4 Thesis Outline

A brief overview of each chapter is provided hereafter. The thesis structure follows the recommended reading flow chart shown in Figure 1.2.

- Chapter 2: Basic Principles *(optional)*

  Readers familiar with the basics of data compression and deep learning-based object detection and segmentation can feel free to skip this chapter.

  This chapter briefly reviews the working principles of compression and analysis algorithms used in the thesis. It provides a quick overview of the two main families of data compression algorithms: intra-frame and inter-frame codecs, focusing on JPEG 2000 and residual frame extraction from hybrid video compression codec. The chapter also introduces the two deep learning algorithms used for analysis, focusing on YOLO for object detection and U-Net for segmentation.

- Chapter 3: Improved Robustness of Deep Learning Segmentation Against Image compression in Medical Imaging

  In this chapter, we explore **Use Case 1** of this thesis, focusing on the challenges of data compression and analysis in medical imaging. Our work centers on fine-tuning deep learning segmentation models on JPEG 2000 compressed medical images. We demonstrate that segmentation models can effectively learn to filter out compression artifacts, even at high compression ratios. Additionally, we show that data compression can be implemented in medical data management systems with minimal performance impact.

- Chapter 4: Utilizing Residual Frames for Deep Learning-based Object Detection and Tracking in Video Surveillance

  In this chapter, we explore **Use Case 2** of this thesis, focusing on the challenges of data compression and analysis in video surveillance. Specifically, we examine the potential of utilizing compressed domain residual frames for deep learning object detection and tracking. We show that a network trained and tested on residual frames can outperform one trained and tested on raw frames in simple scenes and remains competitive in complex scenes, all while providing a privacy-friendly by-product.

- Chapter 5: Hybrid Framework using Scalable Codestream for Bandwidth Restricted Object Detection in Remote Sensing

  In this chapter, we explore **Use Case 3** of this thesis, focusing on the challenges of data compression and analysis in remote sensing. Specifically, we explore using the scalability of the JPEG 2000 codestream to address bandwidth limitations. This approach streamlines a hybrid detection process: a low-resolution mega-image is labeled by a deep learning model, while selected high-resolution tiles are refined by a human annotator. We show that by utilizing scenario-specific constraints and the scalable codestream, the proposed hybrid framework can significantly reduce emergency response time.

- Chapter 6: Conclusion

  In this chapter, we summarize the contributions of the thesis. We then address the Central Research Question and thesis perspectives. We conclude with the thesis' key take-home messages

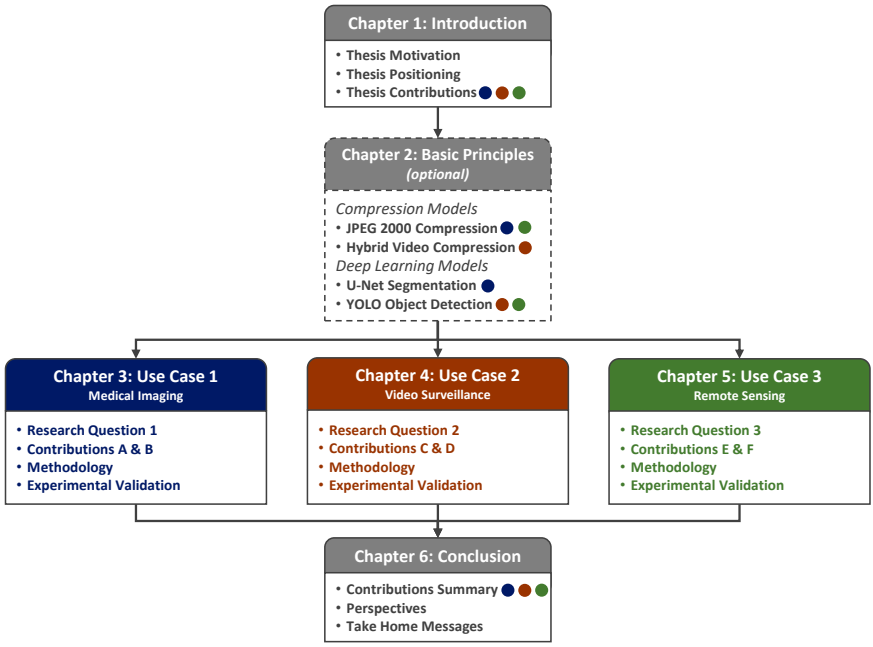

Fig. 1.2 Thesis recommended reading flow chart.

2

Basic Principles

Preliminary Note:

This chapter will briefly introduce the working principles of the compression algorithms and deep learning models used in this thesis.

Readers familiar with the following basic principles can feel free to skip this chapter:

- *JPEG 2000 compression*
- *Hybrid video compression*
- *U-Net segmentation*
- *YOLO object detection*

Basic Principles | Table Of Content

2.1 Data Compression Algorithms

2.1.1 JPEG 2000 Compression

In this work, **JPEG 2000** is used in **Use Case 1** due to its integration with the DICOM standard. It is also used in **Use Case 3** for its scalable codestream and progressive decoding capabilities.

Quick Overview

JPEG 2000 falls within the **intra-frame compression** family as it exploits spatial redundancy within an individual frame for its compression. Even though this standard dates back to the turn of the century, it is still embedded in a variety of applications and industries such as medical imaging, remote sensing and digital cinema [34, 52]. JPEG 2000's widespread use today is not only due to its efficiency, but also its scalability, as it allows for progressive decoding of its codestream which is particularly useful to improve data storage and data transmission constraints [53, 54].

Algorithm Procedure

The JPEG 2000 algorithm, as shown in Figure 2.1, involves both an encoding and a decoding phase. In the encoding process, the original image undergoes: (1) image tiling (2) a discrete wavelet transform, (3) quantization,

and (4) entropy coding. The resulting codestream is structured to simplify decoding, featuring a main header at the start that details the codestream structure and its encoding parameters. This header is essential for locating, extracting, decoding, and reconstructing the image according to specified criteria, such as the number of decomposition levels (i.e. custom resolution selection) and regions of interest (i.e. custom tiles selection). In the decoding process, the codestream is subjected to all the inverse operations of the encoder: (5) entropy decoding, (6) de-quantization, and (7) an inverse discrete wavelet transform and (8) tile restructuring.

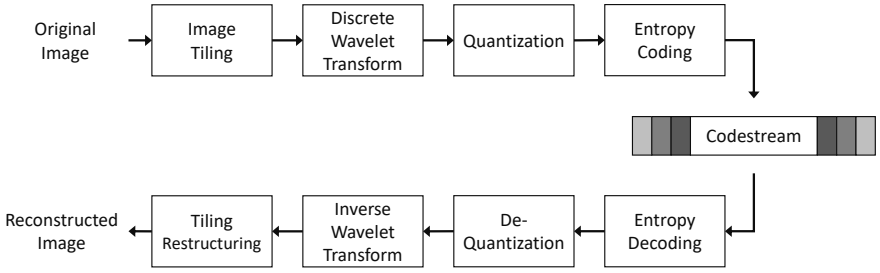

Fig. 2.1 Block diagram showing the JPEG 2000 algorithm procedure.

Scalability and Progressive Decoding

The JPEG 2000 codestream scalability and progressive decoding capabilities are due to the use of the *Discrete Wavelet Transform* as well as the *Image Tiling* approach employed [55].

JPEG 2000 compression utilizes the 2D *Discrete Wavelet Transform* to achieve a multi-resolution representation of the image. The original image is processed with high-pass and low-pass filters, resulting in three detail sub-bands (LH, HL, and HH) and one approximation sub-band (LL). These sub-bands contain coefficients that represent the image's vertical and horizontal frequency components. Figure 2.2 illustrates a single level of wavelet decomposition. The LL sub-band serves as the new base image and can be decomposed further, depending on the desired number of levels. For multiple decomposition levels, the process shown in Figure 2.2a is reapplied to the LL sub-band, further subdividing the LL block depicted in Figure 2.2b.

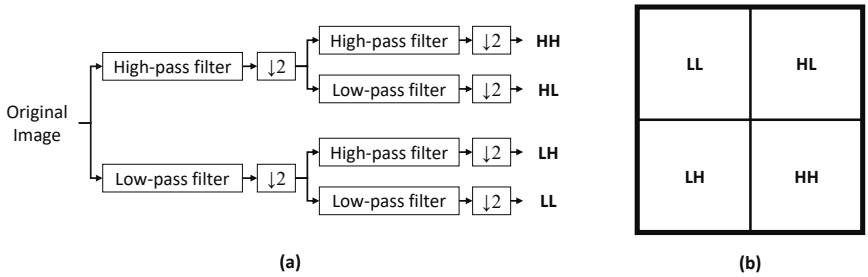

Fig. 2.2 Discrete wavelet transform: (a) block diagram showing single level of 2D wavelet decomposition into the different sub-bands and (b) the arrangement of sub-band on a single image.

In JPEG 2000, *Image Tiling* is the process of dividing the image into separate non-overlapping tiles, with each tile being compressed separately. The JPEG 2000 tiling scheme is illustrated in Figure 2.3. Figure 2.3a shows the image organization in the wavelet domain when considering 2 wavelet decomposition levels and dividing the image into 4 tiles. Figure 2.3b shows how the tiles are rearranged to form the complete wavelet representation of the image. This process is equivalent to applying the wavelet transform on the whole image.

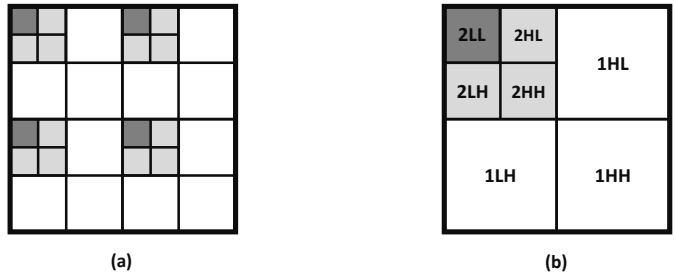

Fig. 2.3 JPEG 2000 tiling scheme: (a) image organization in the wavelet domain with two wavelet decomposition levels (b) tile rearrangement to form the complete wavelet representation of the image

⇒ Combining the *Discrete Wavelet Transform* and the *Image Tiling* features in JPEG 2000 allows for images to be encoded at various resolution levels with the ability to decode any tiles at any desired resolution level.

2.1.2 Hybrid Video Compression

In this work, **hybrid video compression** is used in **Use Case 2** because of its extensive deployment in edge cameras within video surveillance systems. It also provides semantic information contained within its codestream in the shape of residual frames and motion vectors.

Quick Overview

Unlike still image compression, hybrid video compression extends the intra-frame compression algorithms by analyzing not only the spatial redundancy within individual frames but also the temporal redundancy between consecutive frames. This process is known as **inter-frame compression**. Video frames are organized into structures called Groups of Pictures, which consist of a reference frame (I-frame) followed by several predicted frames (P or B frames). P-frames contain information such as **motion vectors** and **residual frames** (also known as the prediction error) to reconstruct the frame based on the reference I-frame. The P-frames are less costly to store than I-frames given that they only contain semantic information on the changes between two consecutive frames. This is particularly interesting at high frame rates where consecutive frames are considered very similar. Exploiting this similarity is key for efficient data compression.

Motion Vectors and Residual Frames

Hybrid compression formats such as HEVC, VVC, and VP9 [49–51] employ a common approach to inter-frame compression. Given that in our work (**Use Case 2**) we focused on **extracting residual frames**, we will limit our context to a simplified generic hybrid video compression encoding procedure and not go into details of the HEVC, VVC, or VP9 implementations.

Residual frames generation can be divided in a three step process: *Motion Estimation*, *Motion Compensation* and *Residual Frame Generation*.

The step by step procedure to obtain the compressed domain residual frames is detailed hereafter and is depicted in Figure 2.4. It should be noted that reference frames are stored using classical intra-frame compression.

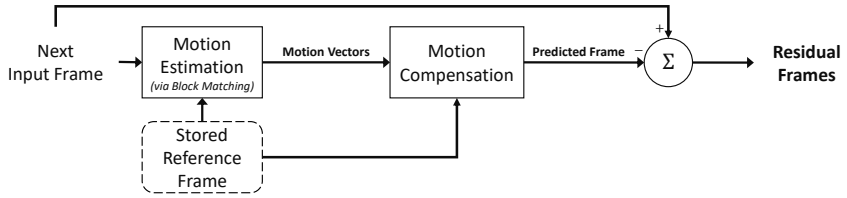

Fig. 2.4 Block diagram depicting the procedure to obtain the residual frames within a generic hybrid video compression encoder.

Step 1: Motion Estimation (via Block Matching)
The initial step in generating residual frames is motion estimation. This involves comparing the reference frame with the subsequent input frame to create a motion vector map. The process divides the reference frame into non-overlapping blocks and analyzes the position of each block in the reference frame against its position in the next frame. This is known as block matching and results in a set of motion vectors associated to each block that quantify the movement of each block between the two frames.

Step 2: Motion Compensation
The second step, known as motion compensation, involves applying the motion vectors to the reference frame. Each block in the reference frame is transformed according to these vectors, generating a predicted frame. This frame is called the predicted frame because it predicts the next input frame based on the motion vectors and the stored reference frame.

Step 3: Residual Frame Generation
The third step is residual frame generation. This involves subtracting the predicted frame from the next input frame to obtain residual frames, also known as the prediction errors. These frames highlight the differences between the actual input frame and the prediction, which is based on the motion estimation and compensation relative to the stored reference frame.

> **Note: Adaptive Block Matching**
>
> In our work (**Use Case 2**, Section 4.3.3), we used a custom adaptive block matching algorithm based on a hierarchical tree structure, similar to that of the other hybrid compression formats [49–51]. This helps us view finer details within the foreground in residual frames.

2.2 Deep Learning Algorithms

2.2.1 U-Net Segmentation

In this work, the **U-Net** model was utilized in **Use Case 1** because of its popularity in medical imaging segmentation. Specifically, the 3D U-Net model was chosen to segment 3D CT, CBCT, and MRI scans.

Quick Overview

U-Net is a convolutional neural network architecture commonly used for image segmentation. It has an encoder-decoder structure with skip connections. The encoder extracts features from the input image through downsampling, while the decoder upsamples intermediate features to produce the final segmentation mask. The architecture is called "U-Net" due to the symmetrical shape of the encoder-decoder structure. The skip connections help preserve spatial information by directly linking corresponding layers in the encoder and decoder.

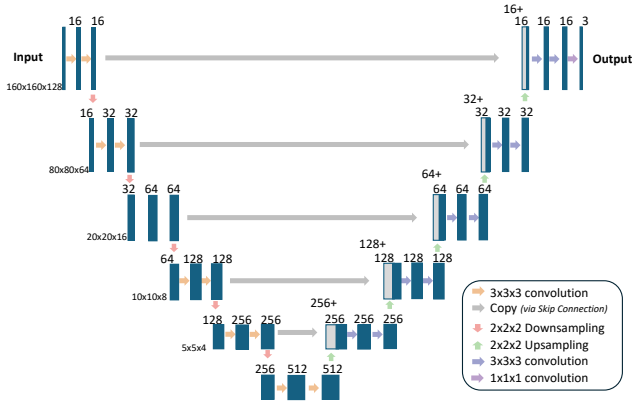

Fig. 2.5 3D U-Net Architecture with six-level depth.

U-Net Architecture

Referring to Figure 2.5, the blue rectangles represent the results of convolution operations, with the number of feature maps specified above them. White rectangles indicate feature maps that are transferred via skip connections from the encoder to the decoder, where they are concatenated. In the final layer, a 1×1×1 convolution reduces the number of output channels to correspond with the number of labels.

2.2.2 YOLO Object Detection

In this work, the **You Only Look Once (YOLO)** object detection model is utilized in **Use Case 2** due to its various model architectures and its suitability for real-time video surveillance scenarios. It is also applied in **Use Case 3** for its quick inference performance as well as its strong fine-tuning potential. It should be noted that the most recent YOLO iteration available at the time of the work was used. This led to the use of YOLOv4 and TinyYOLOv4 in **Use Case 2** and YOLOv8 in **Use Case 3**.

Quick Overview

YOLO is an open-source framework that uses a single pass through a convolutional neural network to process the entire image. It divides the image into a grid, and each grid cell predicts a set number of bounding boxes, along with confidence scores and class probabilities for each box. The confidence scores reflect the accuracy of the bounding box containing an object, while the class probabilities indicate which class the object belongs to. This unified approach allows YOLO to quickly and accurately detect multiple objects within an image, making it highly efficient for tasks that require real-time detection.

YOLO Models Family

The YOLO family of models has evolved significantly over the past eight years, progressing up to YOLOv10, which was released this past year [56]. Essentially, the model efficiency has increased with every new release [57]. Starting from YOLOv4, a scaled-down version of the YOLO framework was proposed [58]. Named TinyYOLOv4, it was designed for low-end GPUs and resource-constrained devices such as edge cameras in video surveillance systems. The trade-off in complexity and performance between YOLOv4 and tinyYOLOv4 evaluated on the COCO dataset [59] is shown in Table 2.1.

Table 2.1 Performance evaluation of YOLOv4 and TinyYOLOv4 on the COCO dataset tested on an NVIDIA RTX 1080Ti GPU.

Model	Average Precision (%)	Frames per Second
YOLOv4	**65.8**	45
Tiny YOLOv4	40.2	**375**

3

Improved Robustness of Deep Learning Segmentation Against Image Compression in Medical Imaging

Preliminary Note:

This chapter expands on conference proceedings [4, 5] and a journal publication [1], studying the impact of JPEG 2000 compression on U-Net segmentation in medical imaging.

Before starting the chapter, readers are encouraged to familiarize themselves with the following basic principles in Section 2: JPEG 2000 compression (2.1.1) and U-Net segmentation (2.2.1).

3.1 Introduction

3.1.1 Motivation

Accurate segmentation of medical images is essential for diagnosing and predicting the progression of various conditions [60,61]. It plays a key role in applications like radiotherapy—where it prevents irradiating healthy tissue—and in detecting tiny cortical lesions in the brain for multiple sclerosis diagnosis [62,63].

Traditionally, this process has been performed manually, which is both labor-intensive and time-consuming. Consequently, there has been a significant shift towards employing deep learning techniques to automate segmentation tasks [46–48]. Nonetheless, these advanced methods require access to large volumes of data, prioritizing efficient data transfer and storage. Moreover, hospitals are legally obliged to retain medical data for extended periods to meet regulatory standards and ensure long-term patient care, making effective data management crucial.

Image compression is a valuable tool in this context, as it can significantly reduce storage requirements and facilitate faster data transfer. This has led to the integration of many compression formats into medical imaging standards: the most notable being the integration of JPEG 2000 into the DICOM standard [34]. It is worth noting that compressing medical images is permitted by governing bodies in North America, the European Union and Australia provided they do not affect the diagnostic capabilities of medical practitioners. However, while compression effectively decreases file size, it can compromise image quality, potentially affecting the accuracy of deep learning models used for medical image analysis. [64].

3.1.2 Research Positioning

Assessing the diagnostic capabilities of physicians on compressed images is challenging due to the subjective nature of evaluations and variability based on different tasks. Several studies have proposed varying compression ratio limits tailored to specific medical imaging *classification* scenarios. Compression ratios higher than 32:1 have been shown to affect pathologists' ability to differentiate breast cancer development stages [65]. Con-

versely, a 200:1 compression ratio was acceptable for measuring the HER2 score in breast carcinoma images [66]. Research involving ten datasets of pathological images, primarily carcinomas and adenocarcinomas, evaluated the impact of data compression on telepathology, showing 95% accuracy with 10 medical practitioners using both uncompressed and 90:1 JPEG compressed images [67]. Additionally, a compression ratio of 20:1 was deemed adequate for detecting Helicobacter pylori gastritis from histopathological images [64]. Researchers have also pushed JPEG 2000 compression ratios up to 50:1 while maintaining a classification accuracy of 97% on cancerous CT liver images [68].

However, these studies often depend on physicians' subjective evaluations, which, although valid, can be costly and impractical for large datasets and complex deep learning tasks due to the limited availability of medical experts. As deep learning becomes more prevalent in medical imaging, there is a need for updated evaluation criteria specific to tasks such as *classification* and *segmentation*. Deep learning networks have demonstrated significant resilience to distortions from image compression for deep learning-based *classification*. For example, convolutional neural networks used for classifying metastases in breast lymph nodes with histopathological whole-slide images have shown effectiveness even when trained on JPEG 2000 compressed data. Training the network on uncompressed images and testing with compressed ones showed strong performance up to a 24:1 compression ratio, while training with images compressed at 48:1 maintained accurate detection for ratios of 48:1 and below [42].

While encouraging results from the image processing community have shown that training semantic *segmentation* neural networks with compressed natural images can maintain robust performances compared to neural networks trained on uncompressed natural image [37–39], the specific impact of image compression on deep learning-based *segmentation* of medical images has not been thoroughly investigated. This intriguing gap leads us to ask the compelling **"Research Question 1"**:

Research Question 1
Can the robustness of deep learning-based segmentation to image compression be extended to medical imaging?

3.1.3 Contribution

This chapter examines the impact of image compression on deep learning-based segmentation in medical imaging. It specifically focuses on enhancing the robustness of JPEG 2000 compression for 3D U-Net segmentation in two medical imaging scenarios:

- **Scenario 1:** Male Pelvic Organ Segmentation [1, 5]
- **Scenario 2:** Multiple Sclerosis Lesion Segmentation [4]

By adopting an iterative fine-tuning approach inspired from similar methods used in natural image processing [38–40], we show that:

Contribution A

A 3D U-Net can effectively learn and filter out data compression artifacts in medical imaging, even at very high compression ratios.

Contribution B

Data compression can be effectively deployed in medical data management systems with only a marginal drop in performance.

Quantitatively we show that:

- We can increase the robustness of the 3D U-Net to handle compression ratios of up to 48:1 in the case of male pelvic organ segmentation.
- We can extend this observation to multiple sclerosis lesion segmentation, maintaining a robustness to compress artifacts up to 64:1.

3.1.4 Chapter Organization

The chapter is structured as follows. We start by describing the materials used, including the dataset, data compression algorithm, deep learning segmentation model and the evaluation metrics. We put forward the iterative fine-tuning approach employed and we present two main experiments of our work. Next, we validate the results of the proposed method, highlighting the two key contributions: the model's robustness to compression artifacts and the potential of the fine-tuned model in practical applications. Following this, we address the limitations of the proposed work. Finally, we conclude by answering **"Research Question 1"** and discussing future work.

3.2 Materials

Materials | Table Of Content

3.2.1 Datasets

Scenario 1: Male Pelvic Organ Segmentation

The dataset is composed of 74 patients and handles two medical imaging modalities: computed tomography (CT) and cone beam computed tomography (CBCT). 74 CT and 56 CBCT scans were collected from CHU-Namur and CHU-Charleroi in Belgium. CHU-Charleroi contributed 56 CBCT and 22 CT volumes, while CHU-Namur provided 52 CT volumes. Initially stored in DICOM format [34], the 2D CT slices were combined to create volumes. All resampled volumes and binary mask volumes were cropped to 160 × 160 × 128 voxels containing the bladder and the rectum, ensur-

ing consistent size and scale across CT and CBCT volumes. These matrices utilized a grayscale range with integer values from 0 to 255. Each patient's data includes manually annotated masks by a trained expert for two male pelvic organs: the bladder and the rectum. Sample slices of CBCT scans of the bladder and the rectum, alongside their respective target segmentation mask, are shown in Figure 3.1. The dataset was divided into a training set with 60 CTs and 45 CBCTs and a test set with 14 CTs and 11 CBCTs, maintaining an 80:20 ratio for both CTs and CBCTs.

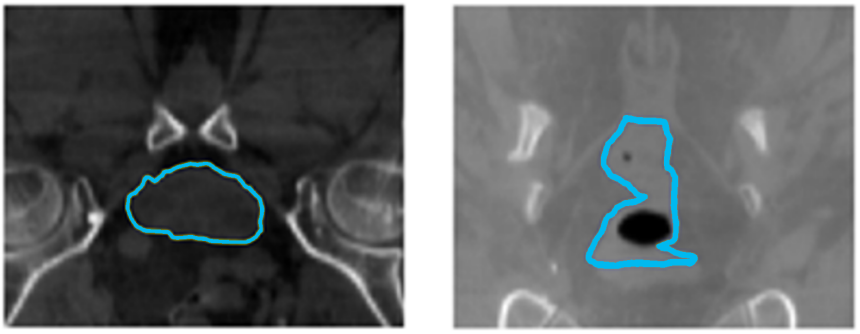

Fig. 3.1 Samples slices of CBCT scans of the bladder and the rectum alongside their respective target segmentation mask for Scenario 1.

Scenario 2: Multiple Sclerosis Lesion Segmentation

The dataset includes 63 Magnetic Resonance Imaging (MRI) scans from multiple sclerosis patients aged 22 to 66 years, with brain scans captured using a 3T Signa Premier MRI scanner at CHU-Saint-Luc in Brussels. All volumes were pre-processed using BMAT [69], binary mask volumes were cropped to 196 × 260 × 216 voxels. Manual lesion annotation was performed by two experts using 3D-FLAIR images, identifying 1104 lesions, with 221 classified as confluent lesions [70]. A sample slice of MRI scans, showing several multiple sclerosis lesions and their respective target segmentation mask, is shown in Figure 3.2. We divided the dataset into 47 subjects for training, 13 for validation, and 13 for testing, ensuring balanced lesion characteristics.

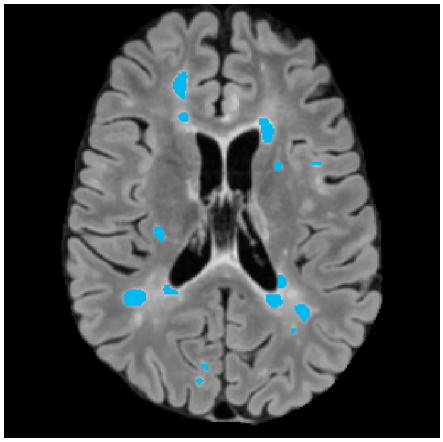

Fig. 3.2 A sample slice of MRI scans showing multiple sclerosis lesions and their corresponding target segmentation mask for Scenario 2.

> **Note: Additional Dataset Information**
>
> Both datasets for Scenario 1 and Scenario 2 are private and were obtained through collaborations with UCLouvain partners. The Scenario 1 dataset, developed with CHU-Namur and CHU-Charleroi, and the Scenario 2 dataset, acquired with CHU-Saint-Luc, has been used in the development of several other research contributions at UCLouvain [48, 70–74]

3.2.2 Data Compression

In both scenarios, we utilize the JPEG 2000 compression algorithm, noted for its high compression efficiency and compatibility with the DICOM standard in medical imaging [34, 52], to achieve visually lossless image compression. The 3D volumes (CT and CBCT scans for scenario 1 and MRI scans for scenario 2) were compressed as individual 2D slices which were reassembled into 3D volumes. This approach helps the 3D segmentation model to mitigate compression artifacts across slices, enhancing its ability to accurately identify and reconstruct anatomical structures. The OpenJPEG library[1] was used to process all 3D volumes.

[1] https://github.com/uclouvain/openjpeg

Due to the nature of male pelvic organs being much larger in size and much easier to segment than multiple sclerosis lesions, we opted to use a higher compression ratio for Scenario 1 compared to Scenario 2. Seven compression ratios, $\{1, 24, 32, 48, 64, 96, 128\}$:1, were used for Scenario 1 compared to seven compression ratios, $\{1, 8, 24, 32, 48, 56, 64\}$:1, for Scenario 2. In medical imaging, the use of high compression ratios is uncommon due to the potential loss of critical diagnostic information. This work investigates the limits of image compression by identifying the maximum compression ratio at which optimal performance can still be achieved.

3.2.3 3D Segmentation Base Model

Scenario 1: Male Pelvic Organ Segmentation

The base model for deep learning segmentation used in Scenario 1 is the fully convolutional 3D U-Net [15, 75] based on the previous work in automated male pelvic organ segmentation [1, 5, 48, 76] [2].

The 3D input was processed through a contracting path for context and an expanding path for precise localization. The final layer used a softmax function to determine the probability of each voxel belonging to the bladder, rectum, or neither. Each voxel was assigned the class with the highest probability to create a binary mask for each organ, resulting in single connected regions without disjointed areas. Fully convolutional neural networks provide the benefit of making predictions at the same resolution as the input, with each organ having its own output channel. The network used the Adam optimizer during training with a learning rate of 10^{-4}. Hyperparameters were consistent with prior research[3] [48, 76] and effective for this data. Two data augmentations techniques were used during training: shifts (ranging from -5 to +5 pixels) and rotations (ranging from -5 to +5 degrees). The batch size was set to two, the maximum that a 12 GB GPU can accommodate.

The loss function $\mathcal{L}_1$ for Scenario 1 is defined in Eq. 3.1 and uses a weighted sum of Dice Similarity Coefficient (DSC) for each organ proportional to their size difference ($\lambda = 3$ in the case bladder versus rectum):

$$\mathcal{L}_1 = (1 - DSC_0) + \lambda \cdot (1 - DSC_1) \tag{3.1}$$

[2] A note in the Supplementary Materials (3.7.2) explains our choice of 3D over 2D U-Net
[3] https://github.com/eliottbrion/pelvis_segmentation

The DSC formulation is defined as follows:

$$DSC_i = \frac{2 \cdot |P_i \cap T_i|}{|P_i| + |T_i|} \tag{3.2}$$

$$DSC_i = \frac{2 \cdot \sum_{x,y,z} p_{i,(x,y,z)} \cdot t_{i,(x,y,z)}}{\sum_{x,y,z} (p_{i,(x,y,z)} + t_{i,(x,y,z)})} \tag{3.3}$$

Where the target and predicted masks are denoted as T_i and P_i, respectively, and the voxel values as $t_{i,(x,y,z)}$ and $p_{i,(x,y,z)}$. The target is the area segmented by a medical specialist, while the prediction is the network output. The index i represents the segmentation class (0 = bladder, 1 = rectum).

Scenario 2: Multiple Sclerosis Lesion Segmentation

The base model for deep learning segmentation used in Scenario 2 is a patch-based 3D U-Net architecture for semantic segmentation of white matter lesion [77] and is based on previous work in automated multiple sclerosis lesion segmentation [4, 70].

The reduction from four resolution levels to three significantly cut the number of trainable parameters from 18.3 million to 3.8 million, lowering the likelihood of overfitting. Skip connections are included as in the original 3D U-Net design [15]. The model processes 3D patches sized 96×96×96 voxels, which are randomly sampled during training using a sliding window approach with Gaussian weighting. Training utilized the Adam optimizer, with a learning rate of 10^{-5}. Hyperparameters were aligned with previous studies [4] [70]. Data augmentation was not employed during training because it did not significantly enhance segmentation performance and increased training time.

The loss function $\mathcal{L}_2$ for Scenario 2 is defined in Eq. 3.4, using a weighted sum between the DSC and Focal Loss (FL) adapting them to high class imbalance [78, 79] :

$$\mathcal{L}_2 = \mathcal{L}_1 + \mathcal{L}_{FL} \tag{3.4}$$

where ($\lambda = 0$ in $\mathcal{L}_1$ given that we only consider one organ to segment in Scenario 2)

[4] https://github.com/maxencewynen/ConfLUNet

3.2.4 Evaluation Metrics

In both scenarios, we use the DSC, defined in Eq. 3.2 to evaluate segmentation quality by comparing the overlap of the target mask and prediction mask with their combined volume.

For Scenario 1, we observed an additional evaluation metric: the Symmetric Mean Boundary Distance (SMBD). SMBD, defined in Eq. 3.7 measures the agreement between the surfaces (borders) of two segmented volumes, rather than their entire volumes. This is particularly useful for assessing segmentation quality in cases where volume overlap metrics like the Dice Similarity Coefficient can be misleading. In large volumes like the bladder, even small errors in segmentation can result in a high DSC due to the large overlap of the overall volume. SMBD provides a more nuanced view by focusing on the boundary accuracy, which is crucial for evaluating the precision of the segmentation[5].

We consider the distance from a predicted border point $p_{i,b}$ to the target border $T_{i,b}$:

$$d(p_{i,b}, T_{i,b}) = \min_{t \in T_{i,b}} \|s \odot (p_{i,b} - t)\|_2 \tag{3.5}$$

Here, $\|.\|_2$ denotes the Euclidean norm, and $s = (1.2, 1.2, 1.5)$ represents pixel spacing in mm. The mean boundary distance is:

$$D(P_{i,b}, T_{i,b}) = \frac{1}{n} \sum_{p_{i,b}} d(p_{i,b}, T_{i,b}) \tag{3.6}$$

Since this metric is asymmetric ($D(P_{i,b}, T_{i,b}) \neq D(T_{i,b}, P_{i,b})$), we use the symmetric mean boundary distance:

$$SMBD = \frac{D(P_{i,b}, T_{i,b}) + D(T_{i,b}, P_{i,b})}{2} \tag{3.7}$$

[5] We use the SMBD solely for bladder segmentation, as the rectum's irregular shape renders the SMBD an unsuitable metric.

3.3 Methods

Methods | Table Of Content

3.3.1 Proposed Iterative Fine-Tuning Methodology

In previous works, it has been demonstrated that training models on datasets with a specific compression ratio can enhance robustness against artifacts at that ratio [38, 40]. However, other researchers argue that fine-tuning pre-trained models, initially trained on uncompressed data, using compressed datasets is more beneficial [39]. Fine-tuning, rather than directly training on compressed data, is also utilized in related research on deep learning-based classification of medical images [42]. This approach highlights the importance of understanding fundamental patterns before addressing compression artifacts, which is crucial for improving the robustness of deep learning-based segmentation models.

Building on this hypothesis, we proposed an iterative fine-tuning on a 3D U-Net to enhance resilience across various compression levels. This strategy enables the model to effectively learn and mitigate compression artifacts at higher levels, thereby improving its generalization capabilities. The methodology consists of two primary steps: initially training a base model on uncompressed data to capture essential patterns, followed by iterative fine-tuning to specifically target and address compression artifacts. The step-by-step training procedure for the proposed iterative fine-tuning approach is outlined in Algorithm 1. Further details on base model for both Scenario 1 and Scenario 2 are provided in Subsection 3.2.3.

Algorithm 1 Iterative Fine-Tuning Training Procedure

Definitions:

Tr_{cr}: Training set compressed at a compression ratio of cr:1
W: Model weights
CR: List of compression ratios used for training
BM_{epc}: Number of training epochs on base model
IFT_{epc}: Number of training epochs during each fine-tuning iteration

Functions:

$J2K_{cr}()$: Apply JPEG 2000 at a compression ratio of cr:1

Procedure:

```
// Initialize compression ratio list
```
Initialize CR
```
// Initialize model weights
```
Initialize W
```
// Set base model and iterative fine-tuning epochs
```
Initialize BM_{epc}, IFT_{epc}
```
// Train 3D U-Net base model on uncompressed data
```
Train W for BM_{epc} epochs on Tr_1
```
// Store Base Model weights
```
$W_{BM} = W$
```
// Store Iterative Fine-Tuning model starting weights
```
$W_{IFT} = W$
for each $cr \in$ CRs **do**
```
  // Iterative fine-tuning procedure
```
  $Tr_{cr} = J2K_{cr}(Tr_1)$
  Train W_{IFT} for IFT_{epc} epochs on Tr_{cr}
  Update W_{IFT}
end for
```

### Parameter Initialization Settings

- W is initialized with random weights.
- $J2K$ set to default parameters of OpenJPEG implementation v.2.5.0.
- $BM_{epc} = 300$ epochs
- $FT_{epc} = 15$ epochs
- $CR = \begin{cases} [1, 24, 32, 48, 64, 96, 128] & \text{for Scenario 1} \\ [1, 8, 24, 32, 48, 56, 64] & \text{for Scenario 2} \end{cases}$
```

3.3.2 Experiments

Two experiments are used to support the two contributions of this chapter.

Experiment 1: Model Robustness to Compression Artifacts

Contribution A
A 3D U-Net can effectively learn and filter out data compression artifacts in medical imaging, even at very high compression ratios.

To validate Contribution A, we compare the performance of the base model with our proposed iteratively fine-tuned model on various increasing compression ratios. Testing on compressed data will help us determine whether the iterative fine-tuning enables the model to learn the basic patterns of the target organs while adequately filtering out compression noise.

It should be noted that for a fair comparison, the iteratively fine-tuned model is tested after each iteration on a test set compressed at the last compression ratio used in that iteration. The detailed evaluation procedure for Experiment 1 is described in Algorithm 2 which integrate the evaluation in the iterative fine-tuning training procedure detailed in subsection 3.3.1.

Algorithm 2 Evaluation Procedure for Experiment 1 built atop of the Iterative Fine-Tuning Training Procedure in 1

Additional Definition:

Ts_{cr}: Test set compressed at a compression ratio of cr:1

Procedure:

Initialize CR, W, BM_{epc}, IFT_{epc}
Train W for BM_{epc} epochs on Tr_1
$W_{BM} = W$
$W_{IFT} = W$
for each $cr \in$ CRs **do**
  $Tr_{cr} = J2K_{cr}(Tr_1)$
  Train W_{IFT} for IFT_{epc} epochs on Tr_{cr}
  Update W_{IFT}
  $Ts_{cr} = J2K_{cr}(Ts_1)$
  Evaluate W_{IFT} and W_{BM} on Ts_{cr}*
end for

* Refer to Subsection 3.2.4 for evaluation metrics used for Scenario 1 and Scenario 2

Experiment 2: Model Deployment with Minimal Performance Drop

Contribution B

Data compression can be effectively deployed in medical data management systems with only a marginal drop in performance.

To validate Contribution B, we evaluate the performance of the base model and the iteratively fine-tuned model on an uncompressed test set. This approach simulates deploying a model trained on a hospital's compressed database, which is then evaluated daily on new uncompressed data during clinical practice.

We follow a similar setup to Experiment 1 but test exclusively on uncompressed data. The detailed evaluation procedure for Experiment 2 is described in Algorithm 3, integrating the evaluation with the training procedure outlined in subsection 3.3.1.

Algorithm 3 Evaluation Procedure for Experiment 2 built atop of the Iterative Fine-Tuning Training Procedure in 1

Additional Definition:

Ts_{cr}: Test set compressed at a compression ratio of cr:1

Procedure:

Initialize CR, W, BM_{epc}, IFT_{epc}
Train W for BM_{epc} epochs on Tr_1
$W_{BM} = W$
$W_{IFT} = W$
for each $cr \in$ CRs **do**
  $Tr_{cr} = J2K_{cr}(Tr_1)$
  Train W_{IFT} for IFT_{epc} epochs on Tr_{cr}
  Update W_{IFT}
  $Ts_{cr} = Ts_1$
  Evaluate W_{IFT} and W_{BM} on Ts_{cr}*
end for

* Refer to Subsection 3.2.4 for evaluation metrics used for Scenario 1 and Scenario 2

3.4 Results and Discussion

3.4.1 Experiment 1: Model robustness to compression artifacts

Experiment 1 was conducted on both Scenario 1 and Scenario 2, evaluating deep learning-based segmentation on three target organs (**bladder**, **rectum**, and **multiple sclerosis lesions**) using three medical imaging modalities (CT, CBCT, and MRI).

Scenario 1: Male Pelvic Organ Segmentation

In this scenario, we focus on segmenting the bladder and rectum in male pelvic CT and CBCT scans using a 3D U-Net model. We evaluate the model's performance both before and after iterative fine-tuning using two evaluation metrics: the DSC and SMBD. We set an empirical DSC cut-off threshold of 0.7 for both organs[6].

First, we examine **bladder segmentation**, which is comparatively easier than rectum segmentation. The DSC results are illustrated in Figures 3.3 and 3.4 for CT and CBCT scans, respectively[7]. The SMBD results are shown in Table 3.1.

We notice enhanced DSC performance across all compression ratios when evaluated using the iteratively fine-tuned model. The base model maintains satisfactory segmentation performance up to compression ratios of 96:1 for CT scans and 64:1 for CBCT scans with a mean DSC threshold of 0.7. In contrast, iterative fine-tuning enables compression up to 128:1 for CT scans and 96:1 for CBCT scans, demonstrating a 1.5-fold increase in robustness while preserving bladder segmentation quality.

When observing the SMBD performance, we also notice this improvement. We obtain a two-fold increase in the case of both CT and CBCT scans at very high compression ratios of 128:1.

[6] Refer to Subsection 3.5 for further explanations on the threshold selection
[7] Detailed DSC results for bladder segmentation are noted in Supplementary Material 3.7.1

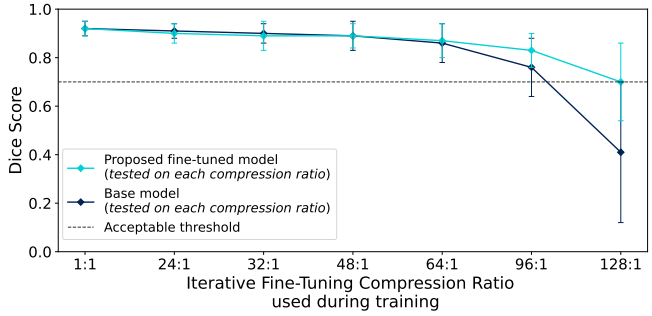

Fig. 3.3 Mean DSC performances before and after iterative fine-tuning tested at high compression ratios for **bladder** segmentation on **CT** scans.

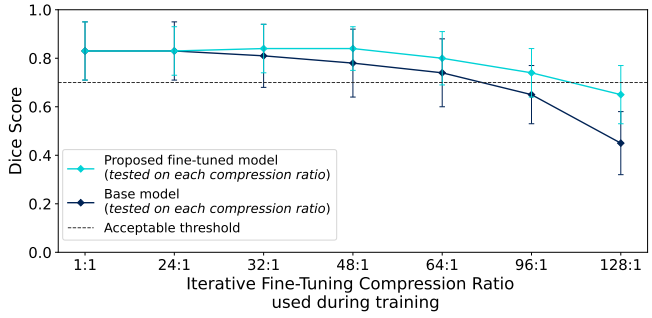

Fig. 3.4 Mean DSC performances before and after iterative fine-tuning tested at high compression ratios for **bladder** segmentation on **CBCT** scans.

Table 3.1 SMBD performance boost using iterative fine-tuning tested at high compression ratios for **bladder** segmentation on **CT** and **CBCT** scans.

		Compression Ratio						
	Model	**1:1**	**24:1**	**32:1**	**48:1**	**64:1**	**96:1**	**128:1**
CT	3D U-Net	1.45±0.58	1.55±0.62	1.83±0.81	2.08±1.12	2.46±1.36	3.90±2.00	8.21±4.19
	+ Fine-Tuning	**1.45**±0.58	**1.53**±0.55	**1.75**±1.09	**1.90**±0.94	**2.17**±1.40	**2.68**±1.33	**4.70**±3.03
CBCT	3D U-Net	2.58±1.96	2.70±2.03	2.96±2.17	3.36±2.36	3.86±2.37	4.90±1.92	7.81±2.15
	+ Fine-Tuning	**2.58**±1.96	**2.19**±1.55	**2.53**±1.78	**2.54**±1.60	**2.97**±1.98	**3.57**±1.58	**4.79**±1.81

Second, we examine **rectum segmentation**, which presents more challenges than bladder segmentation. The DSC performance are illustrated in Figures 3.5 and 3.6 for CT and CBCT scans, respectively[8]. We observe an improved DSC performance with the iteratively fine-tuned model across all compression ratios. The base model performs adequately up to compression ratios of 48:1 for both CT and CBCT scans, maintaining a mean DSC threshold of 0.7. Conversely, iterative fine-tuning supports compression ratios up to 64:1, achieving a 1.5-fold increase in robustness while maintaining rectum segmentation quality.

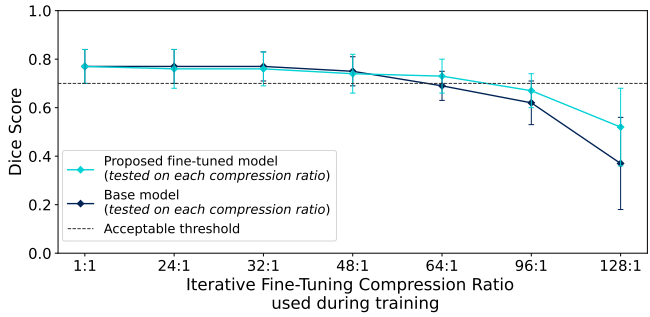

Fig. 3.5 Mean DSC performances before and after iterative fine-tuning tested at high compression ratios for **rectum** segmentation on **CT** scans.

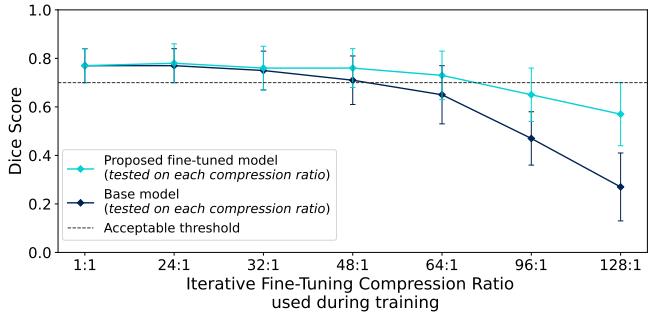

Fig. 3.6 Mean DSC performances before and after iterative fine-tuning tested at high compression ratios for **rectum** segmentation on **CBCT** scans.

[8]Detailed DSC results for rectum segmentation are noted in Supplementary Materials 3.7.1

Scenario 2: Multiple Sclerosis Lesion Segmentation

In this scenario, we focus on segmenting multiple sclerosis lesions in brain MRI scans using a 3D U-Net model. We assess the model's performance before and after iterative fine-tuning. For this scenario, we set an DSC cut-off threshold of 0.5, given the smaller size of multiple sclerosis lesions[9].

Multiple sclerosis lesion segmentation is more challenging due to their smaller size than the male pelvic organs considered in Scenario 1. Figure 3.7 illustrates the DSC results[10]. They clearly show enhanced DSC performance with the iteratively fine-tuned model across all compression ratios. The base model maintains satisfactory segmentation performance up to a compression ratio of 32:1 with a mean DSC threshold of 0.5. In contrast, iterative fine-tuning supports compression up to 64:1, demonstrating a two-fold increase in robustness while preserving lesion segmentation quality.

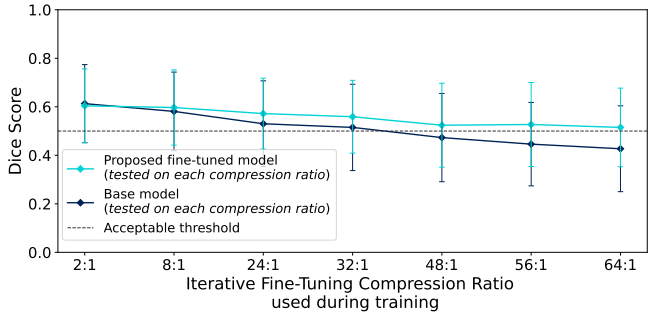

Fig. 3.7 Mean DSC performances before and after iterative fine-tuning tested at high compression ratios for **multiple sclerosis lesion** segmentation on **MRI** scans.

[9] Refer to Subsection 3.5 for further explanations on the threshold selection
[10] Detailed DSC results for lesion segmentation are noted in Supplementary Materials (3.7.1)

3.4.2 Experiment 2: Model Deployment with Minimal Performance Drop

Building on the consistent results from Experiment 1, Experiment 2 narrows the focus to Scenario 1, examining the segmentation of the bladder and rectum with CT scans. We evaluate the model's performance exclusively on uncompressed data using the DSC as evaluation metric. Similarly to Experiment 1, we set an empirical DSC cut-off threshold of 0.7 for both organs[11]. The DSC performance for bladder and rectum segmentation are illustrated in Figures 3.8 and 3.9 respectively.

The model shows less than a 5% performance decrease for bladder segmentation for the fine-tuned model, even when trained on compression ratios of up to 48:1. Even with a more extreme compression ratio of 128:1, the performance drop stays below 10%. Although the overall performance for rectum segmentation is generally lower than that of the bladder, which is understandable given the difficulty of segmenting this organ, it still maintains high accuracy. The performance drop for rectum segmentation is under 5% as well at a 48:1 and only reaches 10% at 128:1.

Crucially, all results surpass the 0.7 DSC acceptable threshold, confirming the model's reliability in accurately segmenting the bladder, thus ensuring dependable outcomes suitable for clinical deployment. This makes it a viable option for integration into medical data management systems. The minimal performance decline, alongside the consistent DSC scores, underscores the potential for deploying this model in real-world clinical environments, where data compression is often necessary for efficient storage and handling.

> **Note: Base Model Performance**
>
> The base model's performance evidently remains consistent at the 1:1 compression ratio DSC, as it was never fine-tuned on compressed data and was only trained on uncompressed data.

[11] Refer to Subsection 3.5 for further explanations on the threshold selection

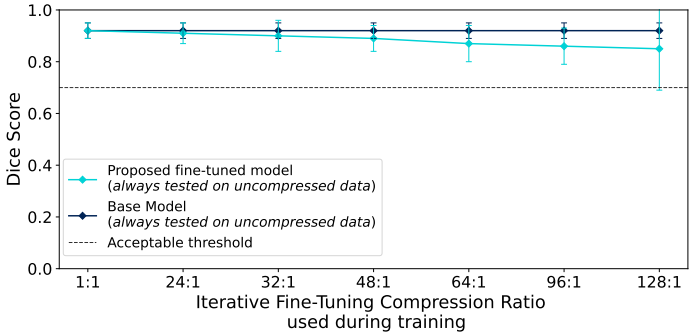

Fig. 3.8 Mean DSC performance of iterative fine-tuning when tested on uncompressed data for **bladder** segmentation on **CT** scans.

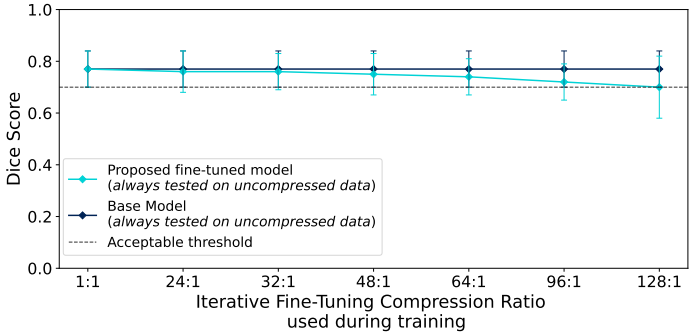

Fig. 3.9 Mean DSC performance of iterative fine-tuning when tested on uncompressed data for **rectum** segmentation on **CT** scans.

3.5 Limitations

3.5.1 Acceptable Segmentation Threshold

The primary challenge we encountered was establishing suitable evaluation criteria for assessing our results. While the DSC is a mathematical metric, it can sometimes yield the same score for two images of the same organ, yet result in different clinical evaluations by medical professionals. For Scenario 1, we set an empirical minimum DSC threshold of 0.7, focusing on male pelvic organ segmentation. This threshold was validated by both our research team and the original annotation experts after reviewing several visual outcomes of U-Net organ segmentation on our dataset. Related works also adopted a 0.7 DSC threshold in their study of organ segmentation in the male pelvic region [80]. Their quantitative assessment, involving 15 physicians who contoured the penile bulb on CT scans of 10 patients with urinary toxicity post-radiotherapy for prostate cancer, found 0.7 to be an acceptable threshold. Although DSC thresholds may vary between organs for accurate segmentation, the similarity in our research focus on CT scans in the male pelvic area reassures us of the validity of our threshold. For the MS lesion segmentation in Scenario 2, we chose a 0.5 DSC as an acceptable threshold. This decision aligns with related works, corresponding to a peak performance drop of 13% [42].

3.5.2 Reproducibility

To enhance the validation of our proposed work, we plan to test the iterative fine-tuning approach on additional public 3D segmentation datasets. Positive results have already been observed across a range of medical imaging modalities and challenging organ segmentation tasks. Utilizing diverse datasets will further assess the model's adaptability across various scenarios. There exists a plethora of 3D medical imaging challenges, with dozens added each year in medical imaging conference challenges and workshops[12]. Our code for iterative fine-tuning, applied to multiple sclerosis lesion, is publicly available on GitHub[13] and can be used as a base to apply it to other 3D medical data.

[12] https://github.com/openmedlab/Awesome-Medical-Dataset
[13] https://github.com/maxencewynen/compressed_WMLS

3.6 Conclusion

3.6.1 Summary

This work highlights significant improvements in the robustness of deep learning-based segmentation models applied to compressed medical imaging data. By adopting an iterative fine-tuning approach inspired by natural image processing methods, we demonstrate two key contributions:

Contribution A

A 3D U-Net can effectively learn and filter out data compression artifacts in medical imaging, even at very high compression ratios.

Contribution B

Data compression can be effectively deployed in medical data management systems with only a marginal drop in performance.

Focusing on JPEG 2000 compression, we show that a 3D U-Net model can adapt through iterative fine-tuning. Initially training the model on uncompressed data captures essential patterns for effective segmentation. This foundational training is followed by iterative fine-tuning on datasets with progressively higher compression ratios, allowing the model to maintain accuracy at compression levels up to 48:1 for male pelvic organ segmentation and 64:1 for multiple sclerosis lesion segmentation.

The findings suggest that high levels of data compression can facilitate efficient storage and faster data transfer without significant segmentation performance drop, providing a practical framework for enhancing deep learning models in real-world applications. The successful integration of training on compressed and testing on uncompressed data with minimal performance drop underscores the potential for widespread adoption in clinical settings.

3.6.2 Perspectives

Future work should broaden this research to address various applications in medical imaging that rely on 3D U-Net segmentation of organs. This includes, but is not limited to, segmentation of lung nodules for detecting lung carcinoma [81], kidney segmentation for treating renal cell carcinoma [82], and whole-heart segmentation to aid early detection of critical cardiovascular diseases [83]. By expanding the scope of this research, we can continue to support the use of image compression in medical data management and demonstrate its effective deployment with minimal performance loss.

Furthermore, while this research indicates that data compression can be integrated into medical systems with minimal performance impact, some doctors may have concerns about its effect on diagnostic accuracy. It is helpful to note that the noise introduced by compression is negligible compared to noise from various sensors in medical imaging, such as beam hardening in CT [84, 85], ring artifacts in CBCT [86, 87], and gradient noise in MRI [88]. The outcome of our research should help in addressing concerns and building trust in the clinical application of compressed data.

The findings of this work indicate a positive direction towards answering **Research Question 1**. However, we have barely unlocked the potential of data compression in medical imaging. Although we explored segmentation in three medical imaging tasks (bladder, rectum, and lesion segmentation) using three medical imaging modalities (CT, CBCT, and MRI), the challenges of medical imaging segmentation extend beyond these tasks and modalities. This work should be seen as a positive result that further encourages the use of data compression in medical imaging.

3.7 Chapter Supplementary Materials

3.7.1 Detailed Results for Experiment 1 (Section 3.4.1)

Table 3.2 DSC performance boost using iterative fine-tuning tested at high compression ratios for **bladder** segmentation on **CT** and **CBCT** scans.

		Compression Ratio						
	Model	**1:1**	**24:1**	**32:1**	**48:1**	**64:1**	**96:1**	**128:1**
CT	Base Model	0.92±0.03	**0.91**±0.03	**0.90**±0.04	0.89±0.06	0.86±0.08	0.76±0.12	0.41±0.29
	+ Fine-Tuning	**0.92**±0.03	0.90±0.04	0.89±0.06	**0.89**±0.05	**0.87**±0.07	**0.83**±0.07	**0.70**±0.16
CBCT	Base Model	0.83±0.12	0.83±0.12	0.81±0.13	0.78±0.14	0.74±0.14	0.65±0.12	0.45±0.13
	+ Fine-Tuning	**0.83**±0.12	**0.83**±0.10	**0.84**±0.10	**0.84**±0.09	**0.80**±0.11	**0.74**±0.10	**0.65**±0.12

Table 3.3 DSC performance boost using iterative fine-tuning tested at high compression ratios for **rectum** segmentation on **CT** and **CBCT** scans.

		Compression Ratio						
	Model	**1:1**	**24:1**	**32:1**	**48:1**	**64:1**	**96:1**	**128:1**
CT	Base Model	0.77±0.07	**0.77**±0.07	**0.77**±0.06	**0.75**±0.06	0.70±0.06	0.62±0.09	0.37±0.19
	+ Fine-Tuning	**0.77**±0.07	0.76±0.08	0.76±0.07	0.74±0.08	**0.73**±0.07	**0.67**±0.07	**0.52**±0.16
CBCT	Base Model	0.77±0.07	0.77±0.07	0.75±0.08	0.71±0.10	0.65±0.12	0.47±0.11	0.27±0.14
	+ Fine-Tuning	**0.77**±0.07	**0.78**±0.08	**0.76**±0.09	**0.76**±0.08	**0.73**±0.10	**0.65**±0.11	**0.57**±0.13

Table 3.4 DSC performance boost using iterative fine-tuning tested at high compression ratios for **multiple sclerosis lesion** segmentation on **MRI** scans.

		Compression Ratio						
	Model	**1:1**	**8:1**	**24:1**	**32:1**	**48:1**	**56:1**	**64:1**
MRI	3D U-Net	0.61±0.16	0.58±0.16	0.53±0.18	0.52±0.18	0.47±0.18	0.45±0.17	0.43±0.18
	+ Fine-Tuning	**0.61**±0.16	**0.60**±0.16	**0.55**±0.17	**0.57**±0.15	**0.56**±0.15	**0.52**±0.17	**0.52**±0.16

3.7.2 Discussion on 2D VS 3D U-Net Robustness to Compression

We further evaluated the decision to use a 3D U-Net over a 2D U-Net, despite employing a 2D slice-based approach for data compression. We conducted experiments for a comparison between the 3D U-Net and a basic 2D-Net model, without Iterative Fine-Tuning. Our focus was on male pelvic organ segmentation in Scenario 1. We also only considered CT scans for this experiment. The detailed detailed experimental results are provided in Tables 3.5 and 3.6 and illustrated in Figures 3.10 and 3.11. The 3D U-Net model shows enhanced DSC performance across all compression ratios for both organs. The 2D U-Net model maintains satisfactory performance up to compression ratios of 64:1, with a mean DSC threshold of 0.7. However, the 3D U-Net model's robustness is up to 96:1, demonstrating a 1.5-fold improvement. Similarly, for rectum segmentation, the 2D U-Net model performs adequately up to 32:1, while the 3D U-Net model handles up to 48:1, also achieving a 1.5-fold increase.

The primary reason for opting for a 3D U-Net lies in its ability to leverage spatial context across slices. A possible explanation for these observations can be found by analyzing the distortion caused by JPEG 2000 artifacts in both 2D and 3D compression scenarios. When performing 2D compression, we compress the 3D volume slice by slice. The distortion introduced by JPEG 2000 is independent for each slice, leading to varying results between adjacent slices. Conversely, in 3D compression by slice, the 3D U-Net can effectively average out the noise across slices, helping to recognize the original shape of the organ.

Table 3.5 DSC performances for 2D and 3D U-Net without fine-tuning tested at high compression ratios for **bladder** segmentation on **CT** scans.

		Compression Ratio						
	Model	**1:1**	**24:1**	**32:1**	**48:1**	**64:1**	**96:1**	**128:1**
CT	2D U-Net	0.92±0.04	0.91±0.04	0.90±0.04	0.87±0.07	0.81±0.09	0.58±0.18	0.30±0.26
	3D U-Net	**0.92**±0.03	**0.91**±0.03	**0.90**±0.04	**0.89**±0.06	**0.86**±0.08	**0.76**±0.12	**0.41**±0.29

Table 3.6 DSC performances for 2D and 3D U-Net without fine-tuning tested at high compression ratios for **rectum** segmentation on **CT** scans.

		Compression Ratio						
	Model	**1:1**	**24:1**	**32:1**	**48:1**	**64:1**	**96:1**	**128:1**
CT	2D U-Net	0.75±0.09	0.76±0.09	0.73±0.09	0.67±0.11	0.58±0.16	0.37±0.16	0.14±0.09
	3D U-Net	**0.77**±0.07	**0.77**±0.07	**0.77**±0.06	**0.75**±0.06	**0.71**±0.06	**0.62**±0.09	**0.37**±0.19

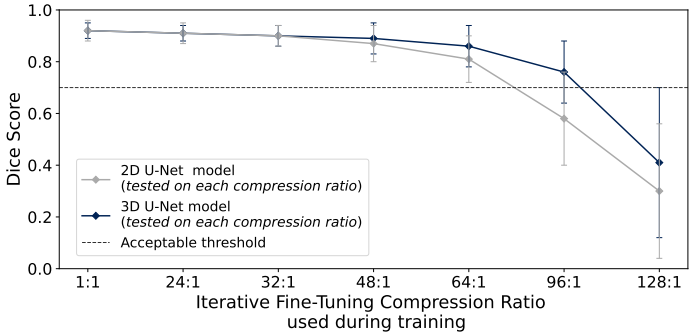

Fig. 3.10 Mean DSC performances for 2D and 3D U-Net without fine-tuning tested at high compression ratios for **bladder** segmentation on **CT** scans.

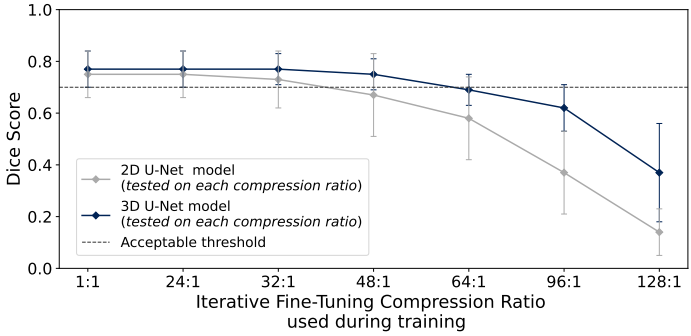

Fig. 3.11 Mean DSC performances for 2D and 3D U-Net without fine-tuning tested at high compression ratios for **rectum** segmentation on **CT** scans.

4

Utilizing Residual Frames for Deep Learning-based Object Detection and Tracking in Video Surveillance

Preliminary Note:

This chapter expands on a journal publication [2], studying the potential of deep learning-based object tracking using compressed domain residual frames.

Before starting the chapter, readers are encouraged to familiarize themselves with the following basic principles in Section 2: Hybrid video compression (2.1.2) and YOLO object detection (2.2.2).

4.1 Introduction

4.1.1 Motivation

Video surveillance systems deploy extensive networks of cameras strategically placed throughout urban landscapes. These surveillance systems employ lightweight models on edge cameras to efficiently perform onboard object detection and tracking. These detection results are then transmitted to a central server, where they undergo analysis by decision-making units. While effective, these edge models are often subpar in performance compared to larger models that can be deployed on the server side.

These surveillance systems, however, do not fully exploit other potential assets on edge devices. These devices usually employ onboard inter-frame encoders [49–51] to compress the incoming video stream. Motion vectors and residual frames are byproducts of these encoders, providing semantic information about the observed scene. Leveraging this semantic information provides an opportunity to enhance the efficiency of video surveillance while maintaining low-bandwidth consumption to ensure system scalability. Representations such as compressed domain residual frames can also offer privacy-friendly benefits. Specifically, they retain only essential motion information in a frame while discarding key re-identification features such as vehicle shapes, colors, number plates, and the location of the observed scene. This motivates the use of a surveillance framework where only compressed domain residual frames are made public, while the rest of the encoded bitstream is encrypted.

4.1.2 Related Works

The integration of video compression and deep learning-based tasks has recently attracted interest from the image processing community. The Moving Picture Experts Group (MPEG) has taken significant steps by establishing a group dedicated to the standardization of Video Coding for Machines (VCM). This initiative emerged from the realization that traditional video codecs are not optimized for deep learning feature extraction. The VCM group aims to develop a codec specifically for machine vision, achieving higher detection accuracy at lower bit rates compared to other inter-frame codes, while also improving the visual quality of raw videos [31].

Additionally, several frameworks have been developed that integrate deep learning-based object detection with traditional background subtraction techniques. These frameworks generally follow a two-step process: first, identifying regions of interest using background subtraction, and then applying a deep learning-based detector to these regions [89–93]. While they achieve notable object detection performance, they require extra computational resources for background extraction.

Moreover, using compressed domain motion vectors has been proposed to enhance the efficiency of deep learning networks. By integrating deep learning-based detectors with these motion vectors, researcher have significantly reduced power consumption and improved detection efficiency, even surpassing several multi-object tracking benchmarks trained on raw frames in some cases [94, 95].

These advancements highlight the potential of using existing video compression data to enhance computer vision applications. This leads us to ask **"Research Question 2"**:

Research Question 2

Can residual frames be utilized to improve detection accuracy in video surveillance systems?

4.1.3 Contribution

This chapter explores how residual frames can enhance detection accuracy in video surveillance systems. We propose an alternative framework that extracts compressed domain residual frames at the edge camera and sends them to a central server for inference using larger detection and tracking models. By maintaining low-bandwidth consumption, the proposed framework ensures system scalability while achieving higher task performance. By introducing the proposed framework to existing automated video surveillance systems, we demonstrate that:

Contribution C

A high-complexity network trained and tested on residual frames can outperform a low-complexity network trained and tested on raw frames in simple scenes.

Contribution D

A high-complexity network working with residual frames can compete with a low-complexity network trained and tested on raw frames, even in complex scenes.

Quantitatively, as shown in Figure 4.1, we demonstrate that:

- The proposed framework achieves a tracking accuracy up to 8% higher than the baseline framework in simple scenes.
- The proposed framework maintains a competitive advantage in highly complex scenes, with less than a 3% performance drop compared to the baseline framework.

4.1.4 Chapter Organization

The chapter is organized as follows. We begin by outlining the methods employed, including both the baseline and proposed frameworks and the experiments used for validation. Next, we describe the materials used, covering the object detectors and trackers, the data compression algorithm, the evaluation metrics, and the video surveillance dataset for multi-object tracking. We then present and discuss our results, including a stress test to ensure robustness of the proposed framework. Afterward, we address the limitations of the proposed work as well as the potential privacy benefits of using compressed domain residual frames. Finally, we conclude by answering **"Research Question 2"** and discussing future work.

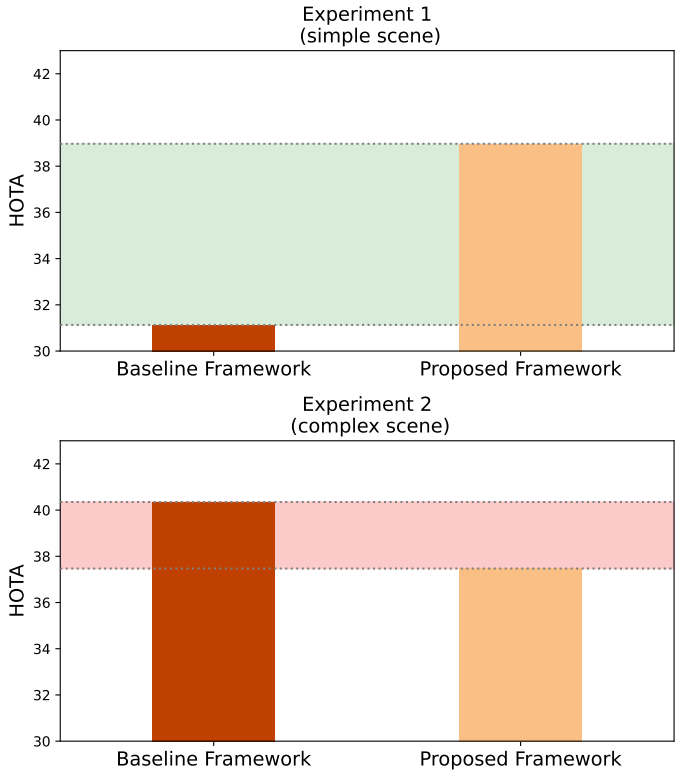

Fig. 4.1 Combined detection and tracking results highlighting the advantages of using residual frames in simple scenes and showing the slight limitation of residual frames in more complex scenes

4.2 Methods

Methods | Table Of Content

4.2.1 Experimental Frameworks

In this work, we consider two experimental frameworks for automated deep learning object detection in camera surveillance systems, as illustrated in Figure 4.2.

- The baseline framework involves deploying **low-complexity detection and tracking models** on the edge camera. Here, object detection is performed directly on the **raw video frames**, with only the detection labels being transmitted to the central server.
- The proposed framework encodes the video stream on the edge device and extracts compressed domain residual frames, **encrypts the rest of the encoded stream** and sends them to the central server. A **high-complexity detection and tracking models** trained on **residual frames** processes these frames for analysis on the server side.

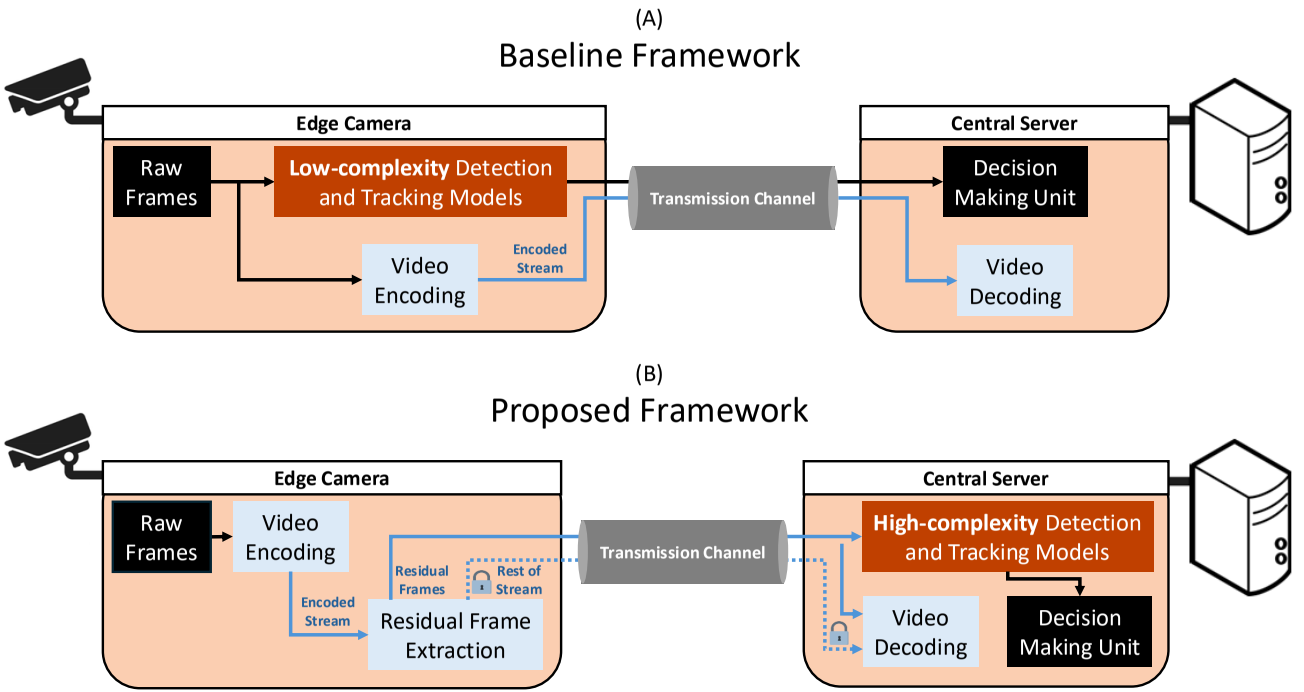

Fig. 4.2 Block diagram of (A) the Baseline framework working with raw frames and (B) the Proposed framework working with residual frames.

4.2.2 Experiments

We set up two experiments on each of two frameworks to show that an object tracker trained and tested on residual frames can be just as effective as an object tracker trained and tested on raw frames.

Experiment 1: Simple Traffic Monitoring Scene Analysis

Contribution C

A high-complexity network trained and tested on residual frames can outperform a low-complexity network trained and tested on raw frames in simple scenes.

The scene for Experiment 1 is shown in Figure 4.3(A). One camera observed a double-lane two-way street with uninterrupted traffic flow. This scene is used for vehicle counting and wrong-way driving violation detection. The constant traffic flow allows the block matching algorithm to generate motion vectors, and their consequent residual frames, consistently. This would lead to uninterrupted and clearer residual frames in return.

Experiment 2: Complex Traffic Monitoring Scene Analysis

Contribution D

A high-complexity network working with residual frames can compete with a low-complexity network trained and tested on raw frames, even in complex scenes.

The scene for Experiment 2 is shown in Figure 4.3(B). Three cameras observed the same intersection between a double-lane two-way street and a single-lane two-way street. This scene is indeed very complex, with overlapping number of vehicles, and can be used for various video surveillance tasks such as traffic light violation detection, vehicle counting, and traffic jam monitoring. The overlapping vehicules, stop signs and traffic lights are a more challenging setup for the block matching algorithm to generate motion vectors, and their consequent residual frames, consistently.

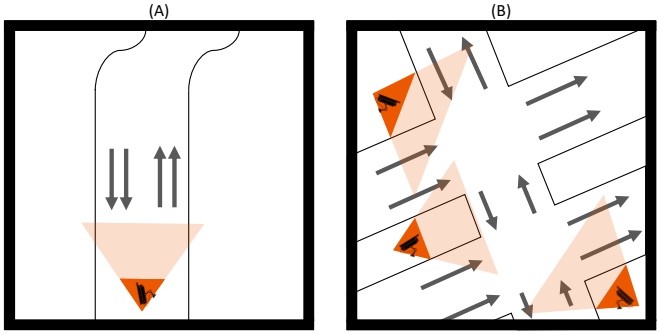

Fig. 4.3 Visualization of the two experimental scenes. Experiment 1 (A) shows a simple scene of a two-way street viewed by 1 camera. Experiment 2 (B) shows a complex scene of an intersection viewed by 3 cameras.

4.3 Materials

Materials | Table Of Content

4.3.1 Object Detectors

In this work, we utilized two object detectors: YOLOv4 [19] and Tiny YOLOv4 [58]. YOLOv4 is denoted as the high-complexity object detector and Tiny YOLOv4 as the low-complexity counter part. At the time of this work, YOLOv4 was the latest iteration in the "You Only Look Once" series available on the Ultralytics open-source library[1].

You Only Look Once (YOLO) has been a state-of-the-art single-stage detector [97]. As its name implies, the entire frame is analyzed in a single evaluation, enabling faster inference and real-time performance. The Tiny version follows the same principles but with a significantly reduced network size. It scales down the number of convolutional layers in the backbone and reduces the number of anchor boxes, resulting in quicker inference but generally lower accuracy.

4.3.2 Object Trackers

In this work, we utilized two object trackers: Kalman intersection-over-union (KIOU) [98] and Simple Online and Realtime Tracking (SORT) [99]. KIOU is denoted as the low-complexity object tracker and SORT as the high-complexity counter part.

The role of an object tracker is to associate the object detections with the same identities over successive frames. We first considered IOU tracker [98], a very simple algorithm that relies on the assumption that detections of an object highly overlap on successive frames, resulting in a high Intersection Over Union (IOU) score. Although this is true for high refresh-rate video sequences, this is less the case for videos from traffic surveillance cameras, which run at lower refresh-rates and where vehicles travel a larger distance between two frames. To address this constraint, we chose the Kalman-IOU tracker (KIOU) instead, where a Kalman filter is added to better estimate object location and speed. This filter also lets you retain a history of the objects and re-identify them in case of missing detections. It was slightly adapted in order to work in an online tracking context, i.e., to work simultaneously with the detector.

[1] https://github.com/ultralytics/ultralytics

We then considered Simple Online and Realtime Tracking (SORT) [99]. This algorithm also includes a Kalman filter to predict existing targets' locations. It computes an assignment cost matrix between those predictions and the provided detections on the current frame using the IOU distance and then solves it optimally using the Hungarian algorithm. Both trackers are localization-based trackers as they use a fast statistical approach based on localization of bounding boxes.

4.3.3 Data Compression Algorithm

For this work we developed our own open-source generic adaptive block matching algorithm[2] inspired by the works of [100, 101]. Our algorithm is a simplified version of the block matching techniques used in widespread inter-frame compression formats such as HEVC, VVC and VP9 [49–51]. The simplified adaptive block matching algorithm used is shown in Algorithm 4.

The algorithm operates on two consecutive images, denoted as I_t and image I_{t+1}. It requires three key parameters: the maximum block size, the minimum block size, and a sensitivity threshold.

Initially, motion vectors are computed for the largest possible blocks. A motion vector represents the movement of a block of pixels from one frame to another, indicating the direction and distance the block has shifted. The algorithm processes each motion vector sequentially from left to right and top to bottom, evaluating the neighboring blocks. If the absolute difference between a motion vector and the average of its neighbors exceeds the sensitivity threshold, the block undergoes splitting. This involves dividing the block into four smaller blocks, and the process continues recursively until the smallest block size is achieved. If the difference does not exceed the threshold, the motion vector is accepted as final. These finalized motion vectors are employed to perform motion compensation on image I_t, generating a prediction I'_{t+1}. The predicted image I'_{t+1} is then subtracted from the actual image I_{t+1}, resulting in the prediction error, i.e. residual frame.

A sample of the adaptive block matching map and residual frames, generated from two consecutive raw frames, are shown in Figure 4.4.

[2] https://github.com/JonathanSamelson/ResidualsTracking

Algorithm 4 Simplified Adaptive Block Matching Algorithm

Definitions:

B_{max}: Maximum block size
B_{min}: Minimum block size
τ: Calibrated sensitivity threshold
I_t: Image at time t
I'_t: Predicted image at time t
$v(b)$: Motion vector associated with block b
V: Set of all motion vectors
RS_t: Residual frames between t and $t+1$

Procedure:

// Initialize block sizes and sensitivity threshold
Initialize B_{max}, B_{min} and τ
// Compute block count
Compute *block_count* by dividing I_t into blocks of size B_{max}
// Process each block
for each block b in *block_count* **do**
  // Compute V by adaptive block matching
  Compare b in I_t and I_{t+1} to compute $v(b)$
  Compute average v_{avg} of neighboring blocks
  if $|v(b) - v_{avg}| > \tau$ **and** size$(b) > B_{min}$ **then**
    Split b into 4 sub-blocks $\{b_1, b_2, b_3, b_4\}$
    Recursively process each sub-block b_i with the same criteria
  else
    Confirm $v(b)$
    Store $v(b)$ in V
  end if
end for
// Apply motion compensation and generate residual frames
Generate I'_{t+1} by applying V to I_t
$RS_t = I'_{t+1}$ - I_{t+1}

Parameter Initialization Settings

- B_{max} = 128x128
- B_{min} = 8x8
- τ is calibrated per camera in each experiment

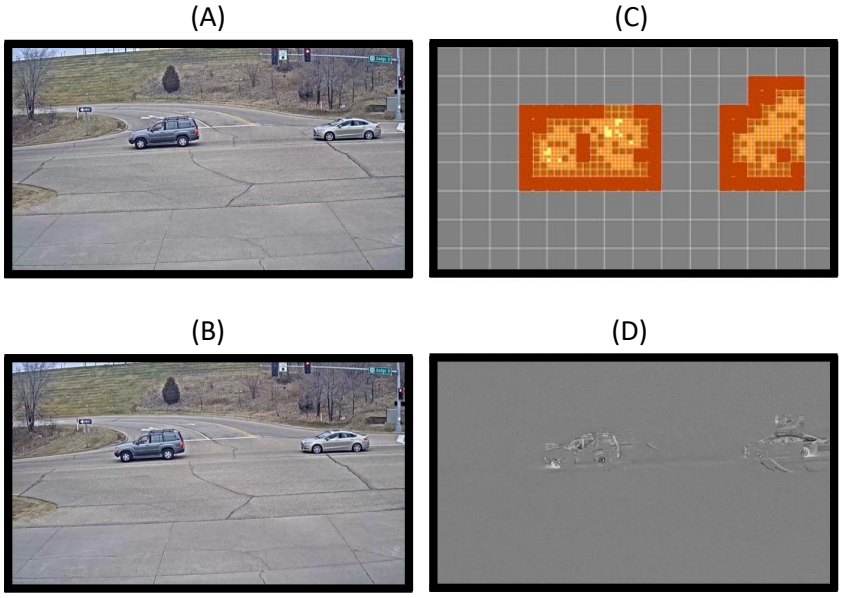

Fig. 4.4 Samples of two consecutive raw frames at time t (A) and $t+1$ (B), along with the adaptive block matching map (C) and the resulting residual frames (D).

4.3.4 Evaluation Metrics

In this work, we use the Higher Order Tracking Accuracy (HOTA) as our evaluation metric [102]. This metric was used to assess the performance of our detector/tracker combinations on the multi-object tracking task. HOTA allows to measure the performance of the both object detection and object tracking in a single metric. The HOTA metric was preferred to the classic Multiple Object Tracking Accuracy (MOTA) [103] metric because the latter is biased towards detection performance.

The HOTA score is computed by means of two sub-metrics that can also be used for deeper analysis. The matching is done at the detection level in each frame based on the similarity score. It divides the task of evaluating tracking into two main subtasks: **Detection** and **Association**.

Detection

Detection measures alignment between predicted and ground-truth detections using Intersection over Union. A localization threshold (e.g., IoU > 0.5) determines overlap. The Hungarian algorithm assigns one-to-one matches, identifying True Positives (TP). Unmatched predictions are False Positives (FP), and unmatched ground truths are False Negatives (FN). We can measure the overall Detection Accuracy (DetA) by calculating the Det-IoU using the count of TPs, FNs, and FPs over the whole dataset:

$$\text{DetA} = \text{Det-IoU} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}| + |\text{FP}|} \tag{4.1}$$

Additionally, Detection Recall and Detection Precision are useful for analysis. Detection Recall indicates how many true positives were correctly identified, while Detection Precision shows how many of the true positives were correct.

$$\text{DetRe} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FN}|} \tag{4.2}$$

$$\text{DetPr} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FP}|} \tag{4.3}$$

Association

Association evaluates how effectively a tracker connects detections across time into consistent identities, based on a set of ground-truth identity links. This is measured by comparing a predicted detection with its ground-truth counterpart, using Hungarian matching. The alignment between the predicted and ground-truth tracks is expressed as an IoU. The overlap between two tracks is defined by the number of True Positive Associations

(TPA). Detections in the predicted track that don't match the ground-truth are False Positive Associations (FPA), and unmatched detections in the ground-truth track are False Negative Associations (FNA). The Association IoU (Ass-IoU) is calculated as follows:

$$\text{Ass-IoU} = \frac{|\text{TPA}|}{|\text{TPA}| + |\text{FNA}| + |\text{FPA}|} \tag{4.4}$$

Overall Association Accuracy (AssA) is determined by averaging the Ass-IoU across all matched detection pairs in the dataset.

$$\text{AssA} = \frac{1}{|\text{TP}|} \sum_{c \in \text{TP}} \frac{|\text{TPA}(c)|}{|\text{TPA}(c)| + |\text{FNA}(c)| + |\text{FPA}(c)|} \tag{4.5}$$

Additionally, Association Recall and Association Precision are useful for analysis. Association Recall indicates how many true positive associations were correctly identified, while Association Precision shows how many of the identified associations were correct.

$$\text{AssRe} = \frac{1}{|\text{TPA}|} \sum_{c \in \text{TP}} \frac{|\text{TPA}(c)|}{|\text{TPA}(c)| + |\text{FPA}(c)|} \tag{4.6}$$

$$\text{AssPr} = \frac{1}{|\text{TP}|} \sum_{c \in \text{TP}} \frac{|\text{TPA}(c)|}{|\text{TPA}(c)| + |\text{FNA}(c)|} \tag{4.7}$$

HOTA

The final HOTA score can be computed the geometric mean of the detection and the association scores. The use of the geometric mean for combining detection and association ensures that both are evenly weighted in the final score:

$$\text{HOTA} = \sqrt{\text{DetA} \cdot \text{AssA}} \tag{4.8}$$

4.3.5 Datasets

For the raw frames, the MIO-TCD Localization dataset [96] was used to train the detectors for raw frames. This public dataset acquired 140,000 annotated images captured at different times of the day and periods of the year by 8,000 traffic cameras deployed over North America. We then fine-tuned the detection models on AICity Challenge 2021 Track1 [104, 105]. In this work, all predictions were grouped into a single mobile object class to match the residual frames' annotations.

We then applied our compression algorithm on video sequences from AICity Challenge 2021 Track 1 [104, 105] to obtain the residual frames dataset. We then trained our detection models this dataset. For this purpose, we manually annotated 14,000 residual frames from six different points of view to make it appropriate for training. Indeed, there are some visibility discrepancies between raw and residual frames so all annotation needed to be adjusted. The latter representation relies on movement in the observed scene, leading to stationary vehicles often not being visible. The model weight are made publicly available [3].

For the test set, we used AICity Challenge 2021 Track 3 [104, 106]. We selected two full HD video sequences (recorded at 10 frames per second), resulting in a total of 10,000 frames to test the performance of the two frameworks. All ground-truths with vehicle IDs are provided.

> **Note: Additional Dataset Information**
>
> The AICity Challenge 2021 targets multi-camera tracking. Therefore, only objects that travel across at least two cameras were annotated. Also, vehicles whose bounding boxes were smaller than 1,000 square pixels (smaller than 0.05% of the native resolution) were not annotated. To keep a fair comparison, predictions smaller than this threshold were also removed. Otherwise, detectors would have been wrongly penalized, since they are capable of detecting further objects resulting in false positives.

[3] https://github.com/JonathanSamelson/ResidualsTracking

4.4 Results

4.4.1 Experiment 1: Simple Traffic Monitoring Scene Analysis

The objective of Experiment 1 was to demonstrate the advantages of using residual frames in traffic monitoring. We evaluated both the baseline and proposed frameworks using footage from the street shown in Figure 4.5. The dataset consisted of 4,000 frames captured at 10 frames per second. We assessed the performance using the HOTA metric and its sub-metrics. The detailed results are presented in Table 4.1.

In terms of the HOTA metric, the proposed framework achieved a score of 38.97%, compared to 31.13% for the baseline. For the DetA sub-metric, the score was 37.87% with residual frames, versus 27.85% with raw frames. Regarding the AssA sub-metric, residual frames yielded a score of 40.33%, while raw frames resulted in 34.95%.

Overall, the HOTA score improved by over 7.5% with residual frames. This improvement is mainly due to the DetA sub-metric, which showed a difference of more than 10%. The enhancement is attributed to higher detection recall (DetRe), as the movement of vehicles makes them more visible in residual frames, and improved precision (DetPr) due to fewer false positives. Additionally, the association accuracy (AssA) was over 5% better with residual frames. The camera's proximity to the ground made it challenging to distinguish distant vehicles, which explains the generally lower scores for the tracking performances.

Table 4.1 Results of Experiment 1 (simple scene) for proposed versus baseline frameworks evaluated with HOTA metric and sub-metrics. Bold values highlight the overall HOTA scores and the underlined values show the best scores for each metric between the two frameworks.

		Evaluation Metric						
Framework	**Detector/Tracker**	**HOTA**	**DetA**	**DetRe**	**DetPr**	**AssA**	**AssRe**	**AssPr**
Baseline	TinyYOLOv4/KIOU	**31.13**	27.85	44.93	32.85	34.95	41.76	49.12
Proposed	**YOLOv4/SORT**	**38.97**	37.87	50.67	46.86	40.33	50.25	55.18
Δ	/	**+7.84**	+10.02	+5.74	+14.01	+5.38	+8.49	+6.06

4.4.2 Experiment 2: Complex Traffic Monitoring Scene Analysis

The aim of Experiment 2 is to demonstrate that a network trained and tested on residual frames can perform comparably to a network using raw frames, even in complex scenes. Both frameworks were evaluated using footage of an intersection, as depicted in Figure 4.6. The dataset consisted of 6,000 frames captured by three cameras at 10 frames per second. We assessed the performance using the HOTA metric and its sub-metrics. The results are detailed in Table 4.2.

The HOTA score was 37.47% for residual frames, compared to 40.35% for raw frames. Specifically, for the DetA sub-metric, residual frames scored 32.96% versus 34% for raw frames. The AssA sub-metric showed a score of 44.79% for residual frames, while raw frames achieved 49.75%.

Overall, the HOTA score was nearly 3% higher with raw frames. This discrepancy primarily arises from the AssA sub-metric, which showed a 5% difference, largely due to the association recall (AssRe). AssRe measures how object tracks are split into multiple tracks, which is affected by traffic light stop lines. In residual frames, stationary vehicles temporarily disappear because these frames rely on motion vectors from block matching. Consequently, vehicles are assigned new IDs when they move again. In contrast, the DetA sub-score showed only a 1% difference. The detection recall (DetRe) is impacted by stop lines, affecting the completeness of detections. However, this is offset by higher detection precision (DetPr) in residual frames, resulting in fewer false positives due to effective background suppression.

Table 4.2 Results of Experiment 2 (complex scene) for proposed versus baseline frameworks evaluated with HOTA metric and sub-metrics. Bold values highlight the overall HOTA scores and the underlined values show the best scores for each metric between the two frameworks.

		Evaluation Metric						
Framework	**Detector/Tracker**	**HOTA**	**DetA**	**DetRe**	**DetPr**	**AssA**	**AssRe**	**AssPr**
Baseline	TinyYOLOv4/KIOU	**40.35**	34.00	62.77	36.22	49.75	63.93	60.78
Proposed	**YOLOv4/SORT**	**37.47**	32.96	42.37	47.43	44.79	57.14	61.30
Δ	/	**-2.88**	-1.04	-20.40	+11.21	-4.96	-6.79	+0.52

Fig. 4.5 Visual results on residual frames (A) versus raw frames (B) in Experiment 1. White bounding boxes represent the ground truth annotations, colored boxes represent the detections together with their respective IDs and confidence score.

Fig. 4.6 Visual results on residual frames (A) versus raw frames (B) in Experiment 2. White bounding boxes represent the ground truth annotations, colored boxes represent the detections together with their respective IDs and confidence score.

4.5 Discussion

4.5.1 Raw frames versus Residual Frames

We tested different sequence snippets from the AI City Challenge 2021 to analyze discrepancies in object detection performance between residual and raw frames. This analysis helped identify scenarios where using residual frames might be beneficial or not recommended for deployment.

The first observation is illustrated in Figure 4.7, highlighting both advantages and disadvantages of using residual frames. Residual frames can cause detection mergers due to uniform color distribution. However, they also provide image smoothing, which helps reduce false positives that deep learning models might predict due to confusing shapes or colors. Additionally, residual frames can enhance the clarity of moving objects by reducing background noise.

Furthermore, Figure 4.8 demonstrates another advantage of residual frame representation. It effectively handles backgrounds with detectable but irrelevant objects, like a full parking lot. This capability is particularly beneficial in dynamic environments where static objects are not of interest.

Another observation is depicted in Figure 4.9. In scenarios with continuously interrupted traffic flow due to traffic lights, compression codecs rely on movement to create residual frames, causing stationary objects to not appear. Nonetheless, residual frames still allow for counting vehicles entering or exiting intersections for statistical purposes.

In summary, in complex scenes, the frequent interruption of targets reduces the advantages of residual frames' background subtraction. Conversely, in simpler scenes with uninterrupted traffic flow, the drawbacks of residual frames are minimized, enhancing the benefits of background subtraction. Therefore, for object tracking, the choice of image representation should be based on the complexity of the scene under evaluation. Ultimately, understanding the specific needs of the application can guide the decision on whether residual frames are appropriate.

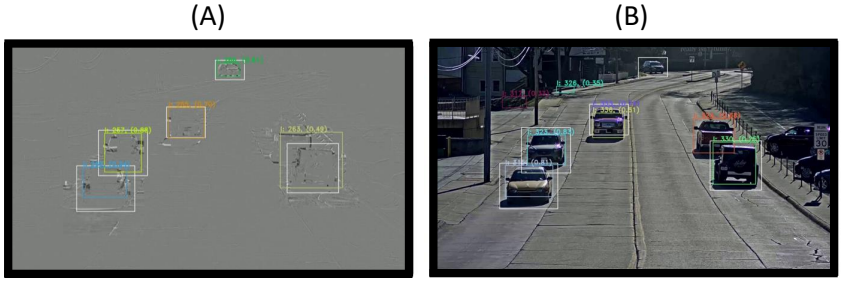

Fig. 4.7 Visual results on residual frames (A) versus raw frames (B) showing false positives in the background and foreground of the raw frame and detection mergers in the residual frame.

Fig. 4.8 Visual results on residual frames (A) versus raw frames (B) showing a parking with irrelevant cars in the background.

Fig. 4.9 Visual results on residual frames (A) versus raw frames (B) showing false negatives in residual frames when cars are stopped.

4.5.2 Impact of Object Tracker

In terms of framework complexity, the difference between the object detectors (YOLOv4 and Tiny YOLO) is significantly greater than the difference between the two trackers (KIOU and SORT). The complexity gap between these detectors is substantial due to the difference in architecture depth. YOLOv4 has around 64 million parameters, making it a complex model designed for high accuracy and detailed object detection. In contrast, Tiny YOLO, with about 6 million parameters, is optimized for speed to be deployed on lightweight micro-controllers. On the other hand, KIOU and SORT can both be considered lightweight tracking algorithms in their respect. We conducted additional experiments by swapping the tracking algorithms between the frameworks. The goal was to analyze the impact of different object trackers and observe any variations compared to the results from Experiments 1 and 2. Given their minimal difference in complexity, this could be a feasible option for deployment.

The results for Experiment 1 and Experiment 2 with the swapped trackers are shown in Tables 4.3 and 4.4, respectively. We observe that the trends remain consistent with previous experiments: residual frames perform better in simple scenarios, while complex scenarios still favor raw frames. However, the advantage of residual frames has diminished, as illustrated in Figure 4.10. The HOTA metric decreased from 7.84% in favor of residual frames to just 3.34% in Experiment 1. In Experiment 2, it shifted from a 2.88% advantage for raw frames to 5.01%. The main difference in HOTA scores before and after tracker swapping is due to the impact on Association scores, with performance drops of around 9% in both experiments.

Table 4.3 Results of Experiment 1 (simple scene) for proposed versus baseline frameworks when swapping the tracking algorithm, evaluated with HOTA metric and sub-metrics. Bold values highlight the overall HOTA scores and the underlined values show the best scores for each metric between the two frameworks.

		Evaluation Metric						
Framework	**Detector/Tracker**	**HOTA**	**DetA**	**DetRe**	**DetPr**	**AssA**	**AssRe**	**AssPr**
Baseline	TinyYOLOv4/SORT	**32.23**	27.69	44.09	32.93	37.69	45.29	48.78
Proposed	**YOLOv4/KIOU**	**35.57**	37.96	51.11	46.60	33.60	41.15	54.92
Δ	/	**+3.34**	+10.27	+7.02	+13.67	-4.09	-4.14	+6.14

Table 4.4 Results of Experiment 2 (complex scene) for proposed versus baseline frameworks when swapping the tracking algorithm, evaluated with HOTA metric and sub-metrics. Bold values highlight the overall HOTA scores and the underlined values show the best scores for each metric between the two frameworks.

		Evaluation Metric						
Framework	**Detector/Tracker**	**HOTA**	**DetA**	**DetRe**	**DetPr**	**AssA**	**AssRe**	**AssPr**
Baseline	TinyYOLOv4/SORT	**41.29**	33.38	61.06	35.92	52.95	66.86	60.49
Proposed	**YOLOv4/KIOU**	**36.28**	34.04	44.24	47.83	39.95	51.11	60.89
Δ	/	**-5.01**	+0.66	-16.82	+11.91	-13.00	-15.75	+0.40

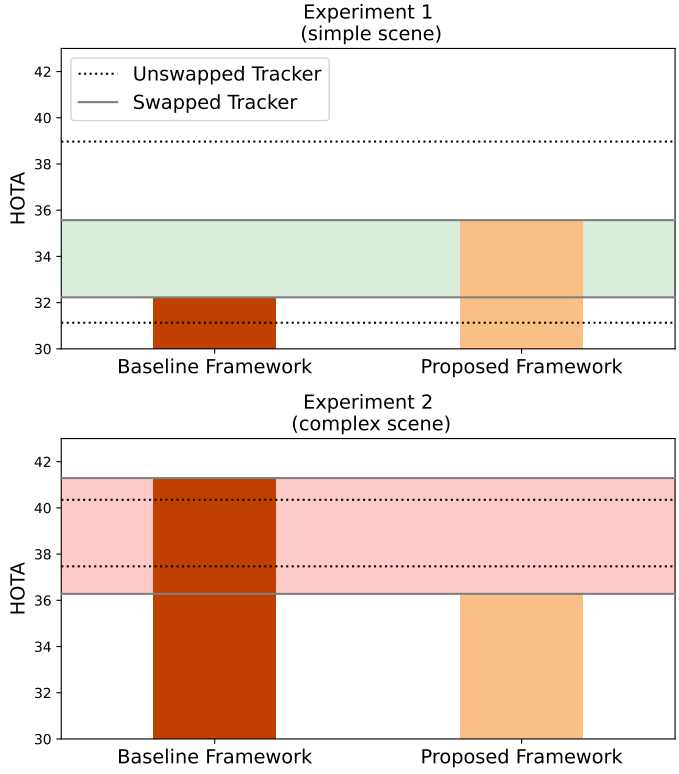

Fig. 4.10 Results on both frameworks for both experiments when swapping the tracking algorithms.

4.6 Limitations

4.6.1 Parameter Choices

We opted to use the same parameters for all video sequences for the detectors and trackers. We decided to set parameters that worked well for all sequences because optimizing parameters for each one would have been arbitrary and could lead to biased results. We used consistent initialization parameters for the adaptive block matching algorithm across all training and test sequences. However, the sensitivity threshold τ was calibrated for each sequence based on camera placement to account for variations in motion vector norms.

4.6.2 Model Training

Our detector for raw frames was trained using two datasets: MIO-TCD and the AI City Challenge. Despite being trained on these datasets, the detectors demonstrated strong generalization capabilities on other video sequences. Even though the detectors for residual frames were trained on AICity footage alone, there is less need for variety in the observed scene to obtain a generic model given the simple appearance of the moving objects in the residual representation.

4.6.3 Dataset Annotations

An important external factor that impacted our results was the AICity annotations. The fact that vehicles have to travel across at least two cameras to be annotated results in missing ground truths for vehicles passing in front of a single camera. Moreover, to ensure full coverage of the vehicles, these ground truths were annotated larger than normal. Furthermore, only annotations larger than two-thirds of the visible vehicle bodies were kept [104, 106]. These factors affected the HOTA scores across all sequences, but since they applied consistently to both types of detectors, they did not affect our comparative analysis.

4.7 Conclusion

4.7.1 Summary

This work introduces a new framework for managing video surveillance systems. By using compressed domain residuals and analyzing them with larger models on the server side, instead of processing raw frames on edge devices with lightweight models, we enhance object detection and tracking performance while maintaining system scalability. This work presents two key contributions:

Contribution C

A high-complexity network trained and tested on residual frames can outperform a low-complexity network trained and tested on raw frames in simple scenes.

Contribution D

A high-complexity network working with residual frames can compete with a low-complexity network trained and tested on raw frames, even in complex scenes.

The proposed framework improves detection accuracy by nearly 8% over the baseline in simple scenes when using a combined detection and tracking evaluation metric. It remains competitive in complex scenes with less than a 3% performance decrease.

These results quantitatively demonstrate that the proposed framework is suitable for deployment in basic urban surveillance tasks. These tasks, such as vehicle counting and identifying wrong-way driving violations, are indeed easily detectable in simple environments. It should be worth noting as well that this is the first work to provide publicly available weights for training and testing deep learning models using compressed domain residual frames [4].

[4] https://github.com/JonathanSamelson/ResidualsTracking

4.7.2 Perspectives

Several promising directions for future work could enhance the validation and applicability of our current results. The migration of the detection model within our proposed framework to a computationally robust central server opens avenues for further exploration with larger and more sophisticated object detection models and trackers. By integrating these advanced models, we anticipate a substantial increase HOTA performance, thereby widening the performance gap from the baseline framework.

Moreover, an intriguing research path involves the implementation of residual frames for object detection under night time conditions [107] as residual frames provide an increased resilience to variations in illumination and color compared to raw frames. This intrinsic robustness could be pivotal in improving night-time surveillance accuracy, offering a more consistent tracking performance despite challenging lighting conditions. Furthermore, employing residual frames necessitates a comprehensive scalability study to assess the bandwidth implications of processing multiple streams concurrently, ensuring that the system remains efficient and cost-effective [108–110].

Overall, while further work is needed to strengthen our claims of a more efficient video surveillance model deployment system, this work affirmatively answers **Research Question 2** and is to be seen as an encouraging step towards using compressed domain representations in deep learning-based video analysis.

4.8 Chapter Supplementary Materials

4.8.1 Discussion on Privacy-Friendliness of Residual Frames

Residual frames provide benefits beyond data compression and analysis; they also have the potential to provide a certain level of data privacy.

Related work tackles the challenge of combining data compression and data privacy. Researchers have attempted to simplify the problem by proposing to exclude specific features from frames as a binary decision. A transform-domain image scrambling method for privacy-friendly video surveillance demonstrated that it is possible to scramble a frame's region of interest (ROI) to hide critical information in the observed scene [111]. Other works have developed custom license plate and facial recognition software to encrypt specific ROIs before encoding and sending out the video sequence [112]. While these works are interesting, they do not address the main issues related to data privacy: "Defining a clear evaluation metric for measuring data privacy is quite challenging". Additionally, every country has different thresholds for the amount of information that may be leaked from video surveillance footage. In the European Union, the General Data Protection Regulation (GDPR) ensures individuals' right to access any information held about them, including CCTV footage [113]. It is extremely challenging to guarantee with 100% accuracy that an image, regardless of its representation, does not reveal any private information about individuals in the visual scene.

All in all, although residual frames help filter information by removing backgrounds and distorting foregrounds, guaranteeing complete privacy is challenging. Our method does not ensure that no private information is revealed. It provides a mean to ensure with high probability that a targeted identification would be unsuccessful.

5

Hybrid Framework using Scalable Codestream for Bandwidth-Restricted Object Detection in Remote Sensing

Preliminary Note:

This chapter expands on conference proceedings [3], studying the integration of hybrid detection framework in bandwidth-restricted UAV object detection.

Before starting the chapter, readers are encouraged to familiarize themselves with the following basic principles in Section 2: JPEG 2000 compression (2.1.1) and YOLO object detection (2.2.2).

5.1 Introduction

5.1.1 Motivation

Unmanned Aerial Vehicles (UAVs) have become an integral part of emergency response operations. These devices offer crucial real-time visual information, which significantly enhances capabilities in disaster-stricken areas. UAVs are instrumental in conducting damage assessments, detecting hazards, and facilitating search and rescue missions [114, 115]. By providing an aerial perspective, UAVs enable responders to quickly evaluate the situation and allocate resources more efficiently.

A significant advancement in utilizing UAV data is the implementation of hybrid detection methods. Hybrid detection enhances the accuracy and reliability of data interpretation, which is vital in the complex and often chaotic environments of emergency situations. This method combines automated deep learning-based object detection with human expertise. Initially, deep learning algorithms analyze data to identify objects and patterns. Human annotators then review and refine these results, focusing on areas where the automated system is uncertain or may have made errors. This collaboration between machines and humans aims to streamline the annotation process, achieving two critical objectives: (1) Allowing first responders to make fast and accurate decisions and (2) Saving human expert annotators' valuable time during emergencies.

Despite these advantages, the efficacy of hybrid detection frameworks is heavily reliant on reliable data transmission. In emergency scenarios, several factors can impede this transmission. Limited bandwidth, for example, can constrain the amount of data that can be sent from UAVs to ground stations. Additionally, physical obstacles or damage to infrastructure can disrupt the line of sight necessary for communication, leading to delays in data reception and processing.

To address these challenges, employing a scalable compression algorithm such JPEG 2000 can be highly effective. In bandwidth constrained scenarios, JPEG 2000's scalability allows for efficient data transmission by enabling multiresolution access to specific regions of interest [52–54]. This capability ensures the streamlining of the transmission process, enhancing the overall decision making ability of first responders.

This has prompted us to raise **"Research Question 3"**:

Research Question 3

Can we exploit the flexibility of the compression codestream to streamline a hybrid detection process?

5.1.2 Contribution

In this work, we present a streamlined hybrid detection framework for UAV object detection in bandwidth-restricted environments. Our approach integrates the JPEG 2000 compression algorithm within a hybrid detection workflow.

Initially, a deep learning network, fine-tuned for low-resolution images, is employed for an initial labeling step on the low-resolution mega-images, minimizing computational demands while maintaining accuracy. For detailed examination, the scalability of the JPEG 2000 codestream is utilized to selectively enhance these regions at high resolution with the remaining bandwidth budget. This subsequent step enables human experts to focus on critical areas with high-resolution details.

By integrating deep learning for initial analysis with JPEG 2000 for selective enhancement, the framework enhances decision speed and detection accuracy, making it suitable for complex image labeling tasks in emergency response scenarios with limited computational resources. In this work we show that:

Contribution E

Fine-tuning a deep learning network for object detection on low-resolution images remains effective while providing fast information access.

Contribution F

By utilizing scenario-specific constraints and the scalable codestream, the proposed hybrid framework can significantly reduce emergency response time.

Quantitatively we observe that the proposed framework can reduce the response time by a factor 1.5 up to 33 compared to a baseline approach in time critical missions while maintain good detection recall. Sample results highlighting the benefits of the proposed framework under various scenario-specific constraints are shown in Figure 5.1.

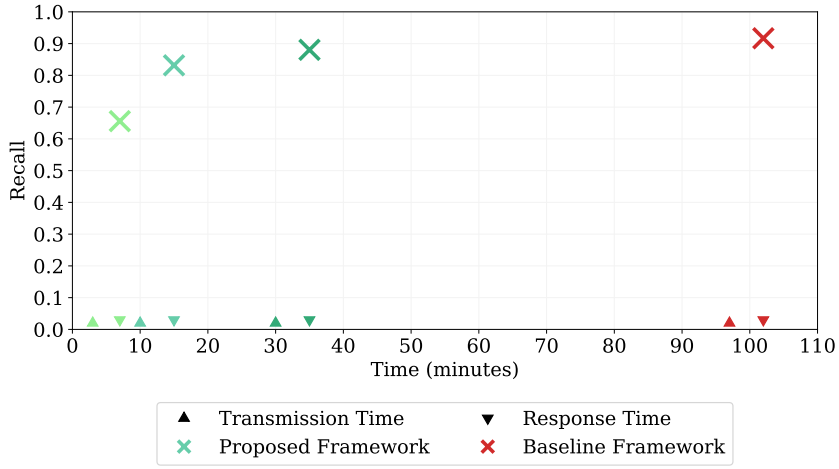

Fig. 5.1 Sample results illustrating the benefits of the proposed streamlined framework under low-bandwidth constraints compared to a conventional baseline. The mission transmission times are 3, 10, and 30 minutes, with a data rate of 88 kbps using Iridium Certus 100.

5.1.3 Chapter Organization

The chapter is structured as follows: We first related work addressing similar challenges. We then outline the methods, including the baseline and proposed frameworks, and the experiments used for validation. Next, we describe the materials used, covering the object detector, the data compression algorithm, dataset and evaluation metrics. We then present and discuss the results. Following this, we discuss the limitations of the proposed hybrid framework. Finally, we conclude by addressing **"Research Question 3"** and discussing perspectives.

5.2 Related Works

In order to categorize the contribution of each of the related works, we define the following four research challenges:

(I) High-resolution image compression

(II) Deep learning-based object detection

(III) Hybrid detection

(IV) Applicability to bandwidth-restricted UAV scenarios

We focus on contributions that address at least two of the four challenges. To the best of our knowledge, the proposed streamlined hybrid detection framework using scalable codestream for bandwidth-restricted UAV object detection is the first contribution to offer a comprehensive solution to all four challenges. Table 5.1 provides a summarized comparison of the related works as per the aforementioned challenges.

5.2.1 Related works merging (I) and (IV):

Zhou et al. [35] present a comprehensive review of remote sensing image compression algorithms. They highlight the JPEG 2000 codec's ability to achieve high compression ratios without significant loss of image quality which is crucial for efficient transmission of high-resolution images in limited bandwidth scenarios. Indradjad et al. [116] compare four wavelet-based methods for compressing satellite imagery. They show that JPEG 2000 delivers better image quality than other wavelet-based codecs at lower bit rates. Descampe et al. [53] highlight the potential of the JPEG 2000 codestream for use in remote settings. Building on their previous work on coarse-to-fine texture retrieval [54], they propose a feature extractor that enables effective classification using a traditional, non-deep learning-based, cascaded classifier, avoiding the need for full decompression of the codestream.

5.2.2 Related works merging (I) and (II):

Ehrlich et al. [117] introduce a self-supervised artifact correction approach that improves object detection accuracy on JPEG images. Ghazvinian Zanjani et al. [42] establish that deep learning models can accurately identify metastatic cancer in JPEG 2000 compressed histopathological images. El

Khoury et al. [1,4] show that fine-tuning a 3D U-Net improves its ability to handle JPEG 2000 compression artifacts while maintaining precise organ detection and segmentation on radiotherapy scans. Marie et al. [40] evaluate the impact of different image compression parameters on deep-based semantic segmentation. They show that image resolution is the most critical parameter for an optimal compression rate-to-accuracy trade-off.

5.2.3 Related works merging (I), (II) and (IV):

Zhang et al. [43] develop a deep learning-based mechanism for UAV-aided networks that compresses media data into constant-sized latent codes for efficient real-time transmission, successfully maintaining the quality of visual data for small objects despite bandwidth constraints and compression artifacts. However, their proposed system relies on having uninterrupted line-of-sight communication which limits its applicability in obstructed environments. Preethy Byju et al. [44,45] design a deep learning framework for efficiently classifying scenes in JPEG 2000 compressed remote-sensing images, which balances computational efficiency with detection accuracy. Given that they work on UAV scenarios with large compressed data repositories, their contribution is not tailored for real-time use. Both contributions do not work within a hybrid detection framework.

5.2.4 Related works merging (II), (III) and (IV):

Kellenberger et al. [118] present an active learning method that requires limited labeled data from human annotators to improve animal detection in UAV imagery. Yamani et al. [119] introduce a single-stage object detection active learning framework for UAV imagery that minimizes annotation expenses through an innovative uncertainty aggregation technique while mitigating class imbalance. Both contributions do not consider the image compression requirements imposed by low-bandwidth constraints.

Table 5.1 Comparison of related works considering four challenges: (I) High-resolution compression, (II) Deep learning object detection, (III) Hybrid detection, (IV) Applicability to bandwidth restricted UAV scenarios.

	[1]	[4]	[35]	[40]	[42]	[43]	[44]	[45]	[53]	[54]	[116]	[117]	[118]	[119]	**Ours**
(I)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓
(II)	✓	✓		✓	✓	✓	✓	✓				✓	✓	✓	✓
(III)													✓	✓	✓
(IV)			✓			✓	✓	✓	✓	✓	✓		✓	✓	✓

5.3 Methods

Methods | Table Of Content

5.3.1 Baseline Hybrid Detection Framework

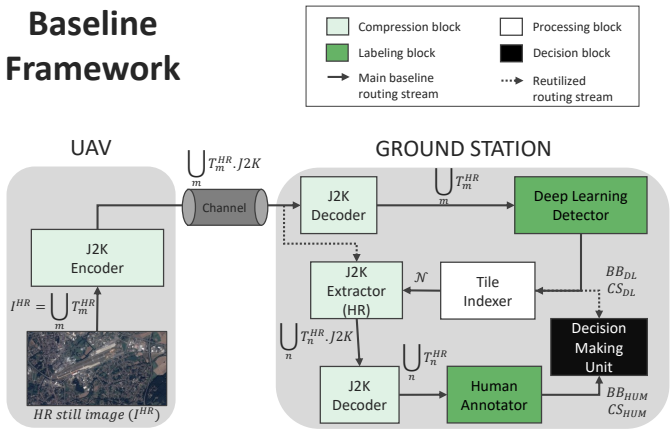

Fig. 5.2 Block diagram of the baseline hybrid detection framework.

The baseline framework aims to emulate the standard hybrid detection approach used by UAVs in emergency situations. Figure 5.2 illustrates a block diagram of the framework. A step-by-step pseudo-code of the baseline hybrid detection framework is presented in Algorithm 5.

The basic working principles of the baseline framework are described as follows. The framework takes as input a high-resolution still image (I^{HR}), a predefined high-resolution level (HR), the high-resolution tile size for the JPEG 2000 encoding ($SIZE(T^{HR})$), and a predetermined human annotation budget ($|\mathscr{N}|$), which represents how many tiles a human annotator can afford to annotate in time-limited situations. The first step involves calculating the total number of tiles ($|\mathscr{M}|$) in I^{HR}. The high-resolution tiles (T^{HR}) from I^{HR} are then encoded using JPEG 2000 and sent via a lossless communication channel to the ground station.

The tiles are then decoded and processed by a deep learning model to produce preliminary labels. The *indexer* block, detailed in Algorithm 6, processes the resultant labels, made up of the bounding boxes (BB_{DL}) and their confidence scores (CS_{DL}). It identifies a set of tiles that require inspection by a human expert annotator and stores their indices ($\mathscr{N}$). We employed a simple *Least Confidence* tile selection strategy for human annotation within the indexer block of both the baseline and the proposed framework.

The framework then calls the *extractor* block, detailed in Algorithm 7, to process the JPEG 2000 scalable codestream and extracts the specific $\mathscr{N}$ tiles in high resolution. The $\mathscr{N}$ selected tiles (T_n^{HR}) are then annotated by the human expert. The resultant bounding boxes (BB_{HUM}) and confidence scores (CS_{HUM}) are sent out along with the deep learning labels (BB_{DL}, CS_{DL}) to the decision-making unit where critical decisions are taken to respond to emergency situations.

Essentially, the main limitation of the baseline framework is due to its constant transmission of high-resolution tiles from the UAV to the ground station, regardless of the scenario-specific constraints. This becomes problematic when data rates and time demands are low, thereby impeding the framework's effectiveness in time-sensitive situations.

Baseline Framework Main Function

Algorithm 5 Baseline Framework Pseudocode

Definitions:

1: I^{HR}: High-resolution still image
2: $SIZE(T^{HR})$: High-resolution tile size
3: $\mathscr{N}$: Set of tile indices selected for human annotation
4: $\mathscr{M}$: Set of all tile indices in the high-resolution image I^{HR}.
5: BB, CS: Bounding Boxes, Confidence Scores

Functions:

6: $J2K()$: JPEG 2000 encoding
7: $TR()$: Lossless UAV-to-ground station transmission channel
8: $DL()$: Deep Learning Detector
9: $HUM()$: Human Annotator
10: $EXT()$: Tile extractor from JPEG 2000 codestream
11: $IND()$: Tile index selector from human annotation

Require: $I^{HR}, HR, SIZE(T^{HR}), |\mathscr{N}|$

Procedure:

// Get number of tiles
12: $|\mathscr{M}| \leftarrow SIZE(I^{HR})/SIZE(T^{HR})$
// Encode all tiles using JPEG 2000
13: $\bigcup_m T_m^{HR}.J2K \leftarrow J2K(I^{HR}, \mathscr{M})$, where $\mathscr{M} = \{1, ..., |\mathscr{M}|\}$
// Send tiles via channel
14: $\bigcup_m T_m^{HR}.J2K \leftarrow TR(\bigcup_m T_m^{HR}.J2K)$
// Decode all tiles using JPEG 2000
15: $\bigcup_m T_m^{HR} \leftarrow J2K^{-1}(\bigcup_m T_m^{HR}.J2K, \mathscr{M})$
// Label high-resolution tiles using deep learning model
16: $BB_{DL}, CS_{DL} \leftarrow DL(\bigcup_m T_m^{HR}, \mathscr{M}, HR)$
// Spot indices of tiles that need human annotation
17: $\mathscr{N} \leftarrow IND(BB_{DL}, CS_{DL}, |\mathscr{N}|)$
// Extract chosen tiles from JPEG 2000
18: $\bigcup_n T_n^{HR}.J2K \leftarrow EXT(\bigcup_m T_m^{HR}.J2K, \mathscr{N}, HR)$
// Decode chosen high-resolution tiles using JPEG 2000
19: $\bigcup_n T_n^{HR} \leftarrow J2K^{-1}(\bigcup_n T_n^{HR}.J2K, \mathscr{N})$
// Annotate chosen tiles by human annotator
20: $BB_{HUM}, CS_{HUM} \leftarrow HUM(\bigcup_n T_n^{HR}, \mathscr{N})$
// Make decision based on hybrid labels
21: $Decision \leftarrow BB_{HUM}, CS_{HUM}, BB_{DL}, CS_{DL}$

Baseline Framework Sub-functions

Algorithm 6 Detailed Indexer Block

Additional Definitions:

1: $\mathscr{X}$: Temporary set of tile indices selected for human annotation

Function: $IND(BB, CS, |\mathscr{N}|)$

Sort BB with respect to CS values in ascending order*

2: $i \leftarrow 0$
3: $\mathscr{X} \leftarrow \emptyset$
4: **while** $i < |\mathscr{N}|$ **do**
5: $\mathscr{X}$.append$(BB[i])$
6: $i \leftarrow i + 1$
7: $\mathscr{N} \leftarrow \mathscr{X}$
8: **return** $\mathscr{N}$

* Employ a *Least Confident* tile selection strategy where we prioritize the tile that contain the objects where the deep learning detector provides the lowest confidence score.

Algorithm 7 Detailed Extractor Block

Function: $EXT(\bigcup_m T_m^{HR}.J2K, \mathscr{N}, R)$

1: **for each** $n \in \mathscr{N}$ **do**
2: Extract T_n^R from $\bigcup_m T_m^{HR}.J2K$ at resolution R
3: **return** $\bigcup_n T_n^R.J2K$

5.3.2 Proposed Streamlined Hybrid Detection Framework

The objective of the proposed framework is to ensure that the decision-making process is rapid and reliable, meeting the critical needs of emergency response operations. To achieve this objective, we propose to modify the baseline framework by streamlining the hybrid detection process to reduce the overall bandwidth consumption of the system. Figure 5.3 presents a block diagram of the proposed streamlined hybrid detection framework, highlighting the main differences with the baseline framework. A step-by-step pseudo-code of the proposed framework is shown in Algorithm 8.

The streamlining process is achieved by implementing four improvements to the baseline framework:

(1) We define a *budget calculator* block, detailed in Algorithm 9, that takes into account two scenario-specific inputs, the transmission time limit ($t_{TRlimit}$) and available data rate ($data\_rate$) from the UAV, as well as the predetermined resolution levels ($Rlvls$) of the JPEG 2000 encoding. The *budget calculator* block then determines the highest resolution available (LR) at which we can send our images and sets the human annotation budget ($|\mathcal{N}|$) with the remaining bandwidth budget.

(2) We fine-tune a deep learning model on low-resolution images to maintain high detection performance at the initial labeling step.

(3) We exploit the JPEG 2000 scalable codestream by utilizing the tile extraction process at the UAV level, instead of at the ground station level, for both low-resolution and high-resolution tile extraction.

(4) We reroute the framework to ensure that only the tiles selected for human annotation are sent in high-resolution, effectively reducing the overall bandwidth consumption of the framework.

Essentially, the proposed framework utilizes JPEG 2000's scalable codestream to reduce the overall bandwidth consumption by first sending a low-resolution image for initial labeling via deep learning network fine-tuned on that specific resolution and then only sends the selected high-resolution tiles for human annotation.

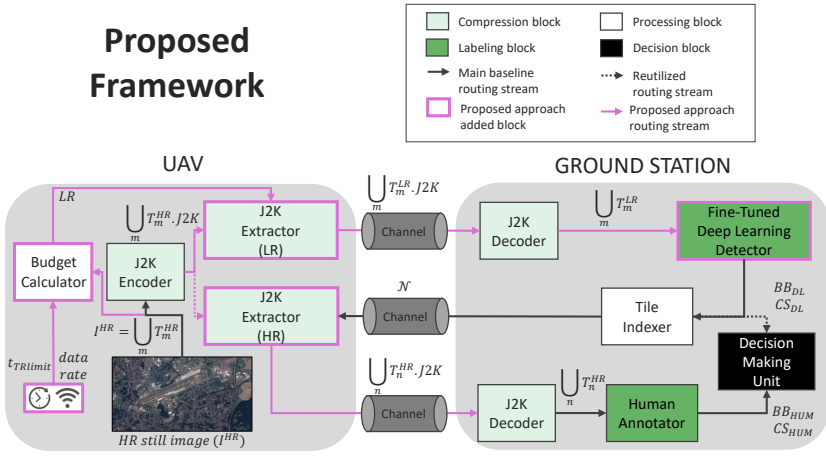

Fig. 5.3 Block diagram of the proposed hybrid detection framework.

Proposed Framework Main Function

Algorithm 8 Proposed Framework Pseudocode

Additional Definitions:

1: $data\_rate$: Scenario-specific available data rate
2: $t_{TRlimit}$: Scenario-specific mission transmission time limit
3: $Rlvls$: Predetermined resolution levels for JPEG 2000 encoding

New Functions:

4: $BC()$: Budget Calculator

Require: $I^{HR}, SIZE(T^{HR}), data\_rate, t_{TRlimit}, Rlvls$

Procedure:

// Get number of tiles and chosen low resolution
5: $LR, HR, |\mathscr{N}|, |\mathscr{M}| \leftarrow BC(I^{HR}, SIZE(T^{HR}), data\_rate, t_{TRlimit}, Rlvls)$
// Encode all tiles using JPEG 2000
6: $\bigcup_m T_m^{HR}.J2K \leftarrow J2K(I^{HR}, \mathscr{M})$, where $\mathscr{M} = \{1, ..., |\mathscr{M}|\}$
// Extract all tiles in low resolution from JPEG 2000
7: $\bigcup_m T_m^{LR}.J2K \leftarrow EXT(\bigcup_m T_m^{HR}.J2K, \mathscr{M}, LR)$
// Send low-resolution tiles via channel
8: $\bigcup_m T_m^{LR}.J2K \leftarrow TR(\bigcup_m T_m^{LR}.J2K)$
// Decode all low-resolution tiles using JPEG 2000
9: $\bigcup_m T_m^{LR} \leftarrow J2K^{-1}(\bigcup_m T_m^{LR}.J2K, \mathscr{M})$
// Label low-resolution tiles using deep learning model
10: $BB_{DL}, CS_{DL} \leftarrow DL(\bigcup_m T_m^{LR}, \mathscr{M}, LR)$
// Spot indices of tiles that need human annotation
11: $\mathscr{N} \leftarrow IND(BB_{DL}, CS_{DL}, |\mathscr{N}|)$
// Send chosen tile indices via channel
12: $\mathscr{N} \leftarrow TR(\mathscr{N})$
// Extract chosen tiles in high resolution from J2K
13: $\bigcup_n T_n^{HR}.J2K \leftarrow EXT(\bigcup_m T_m^{HR}.J2K, \mathscr{N}, HR)$
// Send chosen high-resolution tiles via channel
14: $\bigcup_n T_n^{HR}.J2K \leftarrow TR(\bigcup_n T_n^{HR}.J2K)$
// Decode chosen high-resolution tiles using JPEG 2000
15: $\bigcup_n T_n^{HR} \leftarrow J2K^{-1}(\bigcup_n T_n^{HR}.J2K, \mathscr{N})$
// Annotate chosen tiles by human annotator
16: $BB_{HUM}, CS_{HUM} \leftarrow HUM(\bigcup_n T_n^{HR}, \mathscr{N})$
// Make decision based on hybrid labels
17: $Decision \leftarrow BB_{HUM}, CS_{HUM}, BB_{DL}, CS_{DL}$

Proposed Framework Sub-functions

Algorithm 9 Detailed Budget Calculator Block

Function: $BC(I^{HR}, SIZE(T^{HR}), data\_rate, t_{TRlimit}, Rlvls)$

$LR \leftarrow -1$
$|\mathscr{N}| \leftarrow 0$
$|\mathscr{M}| \leftarrow 0$
$BW \leftarrow data\_rate \cdot t_{TRlimit}$
$HR \leftarrow MAX(Rlvls)$
for $R \leftarrow HR$ **to** 1 **do**
  if $SIZE(I^R) \leq BW$ **then**
    $LR \leftarrow R$
    $|\mathscr{N}| \leftarrow (BW - SIZE(I^R))/SIZE(T^{HR})$
    break
end if
end for
if $LR = -1$ **then**
  Insufficient BW Budget, must relax constraints
end if
$|\mathscr{M}| \leftarrow SIZE(I^{HR})/SIZE(T^{HR})$
return LR, HR, $|\mathscr{N}|$, $|\mathscr{M}|$

5.4 Materials

Materials | Table Of Content

5.4.1 Datasets

The dataset used is private dataset [120] developed by the Institut Scientifique de Service Public (ISSeP) in partnership with the Belgian Science Policy Office (BELSPO). The dataset used is composed of 136 high-resolution satellite still images from the Pleiades4Belgium[1] database representing five airports in different locations around the world. These images were enhanced using the Airbus aircraft detection[2] dataset to include multiple classes of airplanes and helicopters. The training and test set were split as to contain an equal amount of images from each of the airports.

[1] https://www.pleiades4belgium.be/
[2] https://www.kaggle.com/datasets/airbusgeo/airbus-aircrafts-sample-dataset

5.4.2 Image Compression Algorithm

In this work, we utilize the JPEG 2000 algorithm [52] from OpenJPEG[3]. JPEG 2000's scalable codestream efficiently manages image data by encoding it into smaller tiles, which allows for both spatial and resolution scalability while decoding. This enables access to specific regions of images quickly at varying levels of resolutions, complying with the needs of both the baseline and the proposed frameworks. For our implementation, we used a tile size of 512x512 pixels and perform five levels of 2D wavelet decomposition, resulting in five resolution levels (R5 to R1).

> **Note: Resolution Level Interpretation**
>
> The resolution levels correspond to the number of DWT decomposition levels retained. For example, At resolution level 5, all decomposition levels are preserved during decompression, providing the highest resolution of the original image. At level 4, one decomposition level is discarded, removing mostly detailed coefficients. This pattern continues, with each subsequent level discarding an additional decomposition level, until level 1, which retains only the coarsest representation of the image.

5.4.3 Object Detector

We choose the YOLOv8 model as our object detector for the multi-class aircraft detection task. At the time of this work, YOLOv8 was the latest iteration in the "You Only Look Once" series available on the Ultralytics open-source library[4]. YOLO has become the benchmark for object detection tasks and is particularly used in the remote sensing community [121]. A key advantage of YOLOv8 is its ability to maintain strong detection performance across varying image resolutions [122]. YOLO has also been shown to work well for fine-tuning in previous studies [123–126].

[3] https://github.com/uclouvain/openjpeg
[4] https://github.com/ultralytics/ultralytics

5.4.4 Evaluation metrics

Two evaluation metrics are used to compare the baseline and the proposed frameworks: **response time ratio** (t_{RS}_ratio) and **recall difference** ($recall\_diff$).

Response time ratio

t_{RS}_ratio is used to compare the response times between the proposed and the baseline frameworks. For each framework, the response time (t_{RS}), the time at which the decision is taken, is defined as:

$$t_{RS} = t_{TR} + t_{HUM} \tag{5.1}$$

where t_{TR} is the transmission time and t_{HUM} is the human annotator time.

The transmission time (t_{TR}) is the time needed for data to be transmitted through the channel. We distinguish between the transmission times of the two frameworks as follows.

The baseline framework transmission time is defined as:

$$t_{TR} = \frac{\text{SIZE}(\bigcup_m T_m^{HR}.J2K)}{data\_rate} \tag{5.2}$$

where $data\_rate$ varies depending on the scenario at hand.

The proposed framework transmission time is defined as:

$$t_{TR} = \frac{\text{SIZE}(\bigcup_m T_m^{LR}.J2K) + SIZE(\bigcup_n T_n^{HR}.J2K)}{data\_rate} \tag{5.3}$$

where the framework adapts the tile resolution (LR) depending on the $data\_rate$ and transmission time limit $t_{TRlimit}$ of the scenario at hand.

It should be noted that, for both frameworks, we consider that any additional transmitted metadata is minor in size compared to the image data and therefore has a negligible impact on the total data size.

The human annotator time (t_{HUM}), the time required for human experts to annotate uncertain areas of an image, is defined as:

$$t_{HUM} = \mu_{t_{\text{HUM}}} \cdot |\mathcal{N}| \tag{5.4}$$

where $\mu_{t_{\text{HUM}}}$ is the average time a human expert needs to annotate a single tile and $|\mathcal{N}|$ is the number of tiles requiring expert annotation.

The **response time ratio** (t_{RS}_ratio) is thus defined as:

$$t_{RS}\_ratio = \frac{t_{RS}(Baseline)}{t_{RS}(Proposed)} \tag{5.5}$$

Recall difference

$recall\_diff$ quantifies the difference in object detection performance between the proposed and the baseline frameworks. We consider the *recall* as the metric for detection accuracy because, in an emergency situation, the priority is to ensure that we have as few false negatives as possible to avoid missing objects of interest.

The *recall* is defined as:

$$recall = \frac{TP}{TP + FN} \tag{5.6}$$

where TP is the number of true positive and FN is the number of false negative, a detection being considered as a TP if the intersection over union with the ground truth is higher than 0.1 to prevent missing true positive at low resolution, objects being represented by only few pixels.

The **recall difference** ($recall\_diff$) is thus defined as:

$$\begin{aligned} recall\_diff = {} & recall(t_{RS}(Baseline)) \\ & - recall(t_{RS}(Proposed)) \end{aligned} \tag{5.7}$$

5.4.5 Experiments

Two experiments are used to validate the two contributions of this chapter.

Experiment 1: Fine-tuned Object Detector Validation

Contribution E

Fine-tuning a deep learning network for object detection on low-resolution images remains effective while providing fast information access.

The proposed framework is based on the premise that deep learning models can effectively detect objects in low-resolution images. To this end, we propose to use an aggregated model consisting of several fine-tuned object detectors as our deep learning detector. We conducted test on the aggregated model and a base model (trained only on high-resolution images) at each resolution level. The training and evaluation procedures to obtain all these models are detailed in Algorithm 10.

Algorithm 10 Object Detectors Training and Evaluation Procedure

Definition:

Tr^R, Ts^R: Training and Test sets at resolution R
W_{BM}: Base model weights.
W_{FT}^R: Fine-tuned model weights trained at resolution R.
W_{FT}: Aggregated fine-tuned model
$Rlvls$: List of resolution levels in descending order

Procedure:

// Initialize weights to YOLOv8 pre-trained weights on COCO [59]
Initialize W_{BM} and W_{FT}^R $\forall R$
// Base and fine-tuned model Training
for each $R \in Rlvls$ **do**
  Update Tr^R and Ts^R
  if $R = HR$ **then**
    Train W_{BM} for 150 epochs on Tr^{HR} and store model weights
  end if
  Train W_{FT}^R for 150 epochs on Tr^R and store model weights
end for
Aggregate W_{FT}^R $\forall R$ into W_{FT}
// Base and fine-tuned model Evaluation
Test W_{BM} on Ts^{HR} and W_{FT} on Ts^R $\forall R$

Experiment 2: Proposed Framework Validation

Contribution F

By utilizing scenario-specific constraints and the scalable codestream, the proposed hybrid framework can significantly reduce emergency response time.

The proposed framework aims to reduce response times in emergency situations compared to traditional methods, while ensuring high-quality object detection. In our case the object detection task is to identify aircraft in multiple airport settings. To achieve this, we configure JPEG 2000 compression with five resolution levels and fine-tune the YOLOv8 models at these same levels to obtain the aggregated model described in Experiment 1. We evaluate the framework using high-resolution images of 60 megapixels across nine emergency scenarios. Two scenario-specific parameters are varied: the transmission time limit ($t_{TRlimit}$) and the available data rate ($data\_rate$).

Scenarios-specific Constraints Considered

- $t_{TRlimit}$ used to simulate degrees of urgency: 3, 10, and 30 minutes.
- $data\_rate$ considered are: 22, 88, and 176 kbps.[5]
- The average time for human annotation of a single tile ($\mu_{t_{\text{HUM}}}$) is fixed at 0.5 minutes/tile.
- The human annotator time (t_{HUM}) is limited to a maximum 5 minutes, i.e. we have time to manually annotate a maximum of 10 high-resolution tiles ($|\mathcal{N}| \leq 10$).

Note: Project Setting

This work was done in collaboration with the Secure Active Learning for Territorial Observations (SALTO) project in collaboration with the Belgian Science Policy Office (BELSPO). The object detection use-case, dataset and scenario-specific constraints were determined by the project partners.

[5]These rates correspond to the range available for the Iridium Certus satellite network, commonly used for low-bandwidth UAV communications (https://www.iridium.com/iridiumcertus/).

5.5 Results

5.5.1 Experiment 1: Fine-tuned Object Detector Validation

The objective of Experiment 1 is to assess the effectiveness of fine-tuning deep learning models for object detection on low-resolution images. We achieved this by fine-tuning a separate object detector for each resolution level and then aggregating these models to create our fine-tuned object detector.

Figure 5.4 illustrates the detection performance of the aggregated fine-tuned model. We observe a detection recall of 0.92 for W_{FT}^{R5}, 0.88 for W_{FT}^{R4} and 0.83 for W_{FT}^{R3}. This underscores the robustness of the aggregated model even at lower resolutions. The performance of drops of W_{FT} remains minimal compared to W_{BM} all while gaining massively in terms of bandwidth consumption.

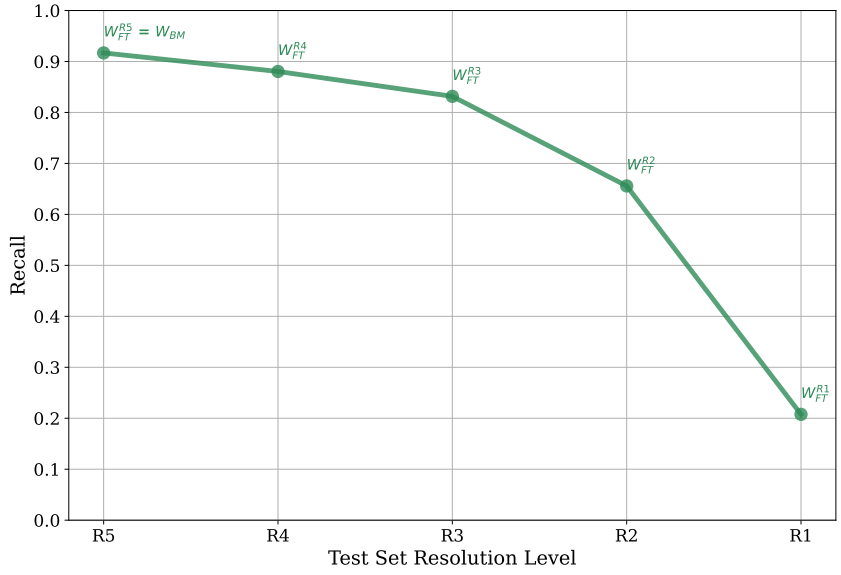

Fig. 5.4 Detection performance of the aggregated fine-tune models showing each of the underlined models specialized at each resolution level.

5.5.2 Experiment 2: Proposed Framework Validation

The evaluation of the proposed framework against the baseline across nine scenarios is depicted in Figure 5.5.

Initially, we examine the scenario with the least constraints, characterized by a *data_rate* of 176 kbps (Figure 5.5a). In this scenario, for $t_{TRlimit}$ values of 3, 10, and 30 minutes, the $recall\_diff$ values are 0.261, 0.085, and 0.036, respectively. Concurrently, there is an observed improvement in the t_{RS}_ratio of 7.08, 3.67, and 1.54. This indicates a trade-off between t_{RS}_ratio and $recall\_diff$ when contrasting the baseline with the proposed framework. This trade-off results from the proposed framework's streamlining approach, which takes into account scenario-specific bandwidth constraints. By prioritizing the transmission of lower-resolution images for initial labeling, the framework significantly enhances t_{RS}_ratio, despite a minor penalty in $recall\_diff$. Conversely, the baseline framework, which lacks a streamlining strategy, consistently transmits images at the highest resolution available. This results in a slight improvement in $recall\_diff$, but at the expense of a substantial decrease in t_{RS}_ratio.

A similar pattern emerges at a *data_rate* of 88 kbps (Figure 5.5b). Here, for $t_{TRlimit}$ values of 3, 10, and 30 minutes, the $recall\_diff$ worsens to 0.269, 0.098, and 0.036, respectively, while the t_{RS}_ratio shows improvements of 25.44, 7.27, and 2.95. The proposed framework's streamlining strategy demonstrates robustness in more constrained scenarios by maintaining low-resolution transmissions for initial labeling. The downside is a reduction in t_{HUM} due to the limited bandwidth, slightly decreasing $recall(t_{RS})$. In contrast, the baseline framework's performance in t_{RS} declines as it fails to adapt to scenario-specific constraints.

Figure 5.5c illustrates the scenario with the lowest *data_rate* of 22 kbps, challenging the limits of the proposed framework. At a $t_{TRlimit}$ of 3 minutes, the bandwidth is too limited to even send the lowest resolution, resulting in a $recall(t_{RS})$ of 0. For $t_{TRlimit}$ values of 10 and 30 minutes, the $recall\_diff$ is 0.271 and 0.105, respectively, but there are significant gains in t_{RS} of 32.8 and 11.28. These very low bandwidth scenarios highlight the substantial t_{RS} benefits of the proposed streamlined framework over the baseline.

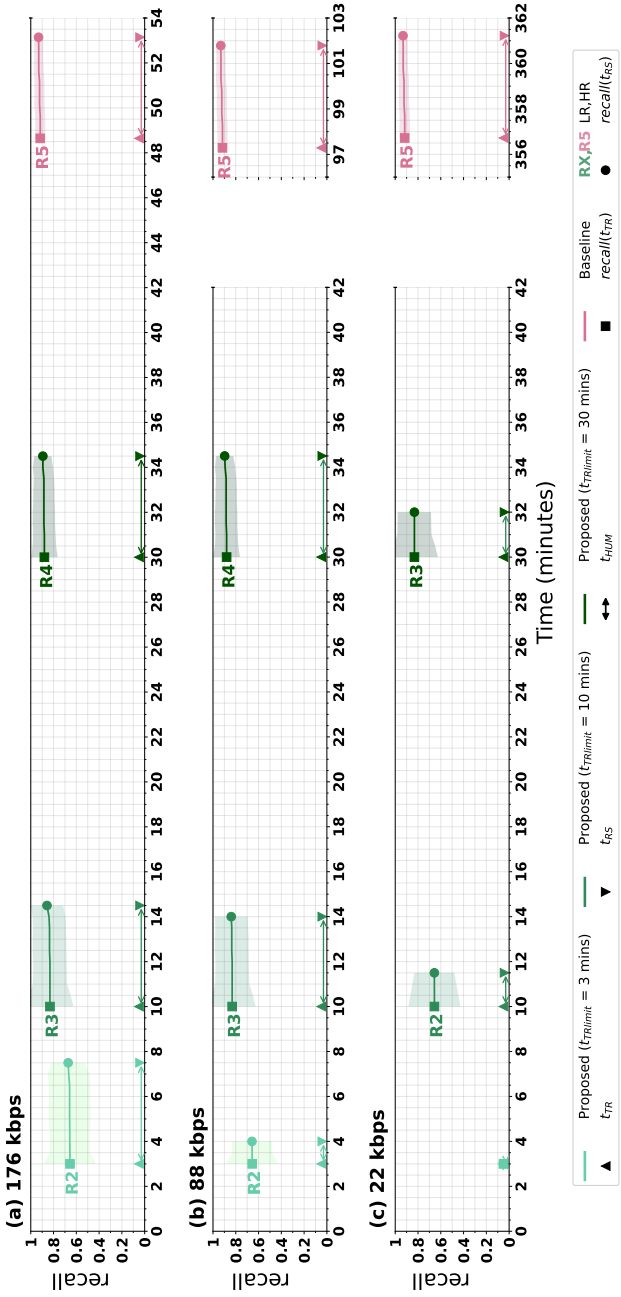

Fig. 5.5 Recall versus time plot for the baseline framework and the proposed framework for high-resolution image size of 60 MP under a data rate limit of (a) 176 kbps (b) 88 kbps and (c) 22 kbps, with a maximum human annotator time of 5 minutes and considering emergency scenarios with a transmission time limit of: 3, 10, and 30 minutes.

5.6 Limitations

5.6.1 Budget Optimization

In our proposed framework, we prioritize the fine-tuned deep learning detector over the human annotator to optimize bandwidth allocation. This decision is based on its superior efficiency in detection accuracy and processing speed, particularly under the constraint of $|\mathcal{N}| = 10$. The *budget calculator* block is responsible for managing resources by determining the highest feasible image resolution for transmission and allocating any remaining bandwidth for human annotation.

We conducted an experiment to determine the amount of human annotation necessary to improve the deep learning detector's performance at Resolution 4, enabling it to surpass its performance at Resolution 5. These resolutions represent the highest levels of detail in our scenario. Our findings, illustrated in Figure 5.6, indicate that using the *Least Confident* tile selection strategy, 54 additional human annotations are required at Resolution 4 to improve recall from 88% to 91.5%. Although this results in a 3.5% increase, it needs 27 minutes of additional human effort, which is considerable in time-sensitive scenarios. While we understand that the naive *Least Confident* strategy can affect result interpretation, it remains robust enough to support our optimization approach. We prioritize transmitting lower resolution images, achieving an immediate 88% recall and gaining 1% additional accuracy with $|\mathcal{N}| = 10$ annotation rather than consuming additional human annotator time for a marginal 3.5% recall improvement.

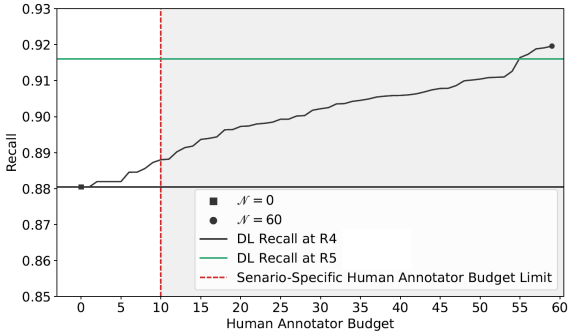

Fig. 5.6 Human Annotator efficiency using a Least Confident tile selection strategy atop of the deep learning model inferred at Resolution 4.

5.6.2 Tile Selection Strategy

A significant limitation of our work is the use of a naive tile selection strategy. Although the *Least Confident* strategy is relatively effective, its efficiency can greatly influence the choice of the budget optimization approach. It is important to study various tile selection strategies, both dependent and independent of the initial deep learning model detections. In our dataset, most objects are concentrated around specific regions of interest, such as airports. Therefore, our tile selection strategy should include this prior knowledge to avoid unnecessary queries on irrelevant tiles. This would enhance the efficiency of the human annotation process and promote a dynamic back-and-forth between the deep learning model and the human annotator, instead of the current streamlined two-step approach.

5.6.3 Project-Imposed Dataset

The primary limitations of our study arise from project-imposed constraints, particularly concerning the dataset. The dataset involved synthetically enhanced images from the Pléiades4Belgium platform, where custom aircraft were added to existing airport imagery. This augmentation improved model training by increasing the diversity of aircraft classes. However, it introduces potential bias favoring the detection of synthetic aircraft over real ones, which could compromise the model's applicability in real-world scenarios.

Moreover, the distribution of objects in high-resolution images results in a significant class imbalance, affecting training and inference processes. Although the training set was filtered to include tiles with objects, intra-image class imbalance persists during testing, causing false positives due to the detector's limited exposure to the "no object" class.

Additionally, the spatial concentration of objects of interest poses challenges. While this concentration aids in tile selection strategies, it also risks misdirecting focus due to false positives, leading to the examination of irrelevant regions within the image.

5.7 Conclusion

5.7.1 Summary

We present a streamlined hybrid detection framework to enhance UAV object detection in bandwidth-restricted scenarios. The framework successfully combines automated deep learning-based detection with human expertise to refine detections in uncertain areas, managing to cut the decision-making time for emergency response operations. This work presents two key contributions:

Contribution E

Fine-tuning a deep learning network for object detection on low-resolution images remains effective while providing fast information access.

Contribution F

By utilizing scenario-specific constraints and the scalable codestream, the proposed hybrid framework can significantly reduce emergency response time.

These findings show that the proposed framework effectively operates in bandwidth-limited environments, reducing response times by up to 33 times compared to a baseline, while maintaining detection performance. This research encourages the remote sensing community to adopt scalable compression codecs for enhanced multiresolution capabilities across all bandwidth-constrained analysis tasks beyond object detection.

5.7.2 Perspectives

Several promising directions for future work could enhance the validation and applicability of our current results. Future efforts should focus on optimizing each component of the framework, particularly the fine-tuning approach of the deep learning model and the tile selection strategy for human annotation.

Extending the current work to other use cases in remote sensing, such as scene classification and region-of-interest segmentation, would be highly beneficial to the community. With the multi-resolution capabilities of JPEG 2000, a coarse-to-fine approach using a fine-tuned deep learning model could facilitate the detection of regions of interest in large-scale satellite images [53, 127].

Additionally, exploring more advanced tile selection strategies would be a significant upgrade to the proposed system. This could lead to a "proposed framework 2.0" where instead of merely streamlining the detection process into a two-step procedure as we did in this work, we develop a dynamic active detection framework. In this framework, the deep learning model's detection is updated with each human annotation quicker convergence. An active learning layer could also be added in an offline setting to be refreshed after each mission, further improving the deep learning model and reducing the need for human annotation [128–130].

Given the significant success remote sensing foundational models have had [131–133], exploring zero-shot or few-shot settings with these large pre-trained models to minimize human intervention time is also quite an interesting idea[6].

Overall, while further work is needed to strengthen our approach in reducing response time in emergency scenarios, this work affirmatively answers **Research Question 3** and is an encouraging step for the remote sensing community to utilize scalable compression codecs for their enhanced multiresolution capabilities across all bandwidth-constrained analysis tasks beyond object detection.

[6] A small discussion ongoing research publication in that area of research is note in Supplementary Materials 5.8.1

5.8 Chapter Supplementary Material

5.8.1 Publication on Zero-Shot Classification in Remote Sensing

Recognizing that human annotation time posed a critical challenge to our bandwidth budget optimization, we sought solutions to minimize this dependency. Initially focused on object detection, we simplified the deep learning task to tile classification in a zero-shot setting, i.e., without requiring any human annotations.

Inspired by research on zero-shot transductive learning in vision-language models [134], we extended our focus to transductive zero-shot classification in remote sensing. The other benefit the proposed transductive approach offers, aside from requiring no labeled data, is its extremely low computational complexity, requiring less than a second to classify over 10^3 tiles.

Experiments on various remote sensing benchmark datasets demonstrated significant accuracy improvements over classical zero-shot classification models. The proposed solution has been accepted to the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2025) and is publicly available as a preprint on ArXiv[7] [6]. The source code is publicly available on GitHub [8].

Relation to Chapter Contributions

While not directly related to the main contributions of this chapter, this work offers a promising research direction in zero-shot learning, completely mitigating the need for human annotation all whilst requiring minimal computational cost.

[7] https://arxiv.org/abs/2409.00698
[8] https://github.com/elkhouryk/RS-TransCLIP

6

Conclusion

6.1 Contributions Summary

In this thesis, we explored the trade-off between model performance and data compression in three distinct use cases across three different research fields and put forward the following contributions:

Use Case 1: Medical Imaging

Contribution A

A 3D segmentation model can learn and filter out data compression artifacts in medical imaging, even at very high compression ratios.

Contribution B

Data compression can be effectively deployed in medical data management systems with only a marginal drop in performance.

Use Case 2: Video Surveillance

Contribution C

A high-complexity network trained and tested on residual frames can outperform a low-complexity network trained and tested on raw frames in simple scenes.

Contribution D

A high-complexity network working with residual frames can compete with a low-complexity network trained and tested on raw frames, even in complex scenes.

Use Case 3: Remote Sensing

Contribution E

Fine-tuning a deep learning network for object detection on low-resolution images remains effective while providing fast information access.

Contribution F

By utilizing scenario-specific constraints and the scalable codestream, the proposed hybrid framework can significantly reduce emergency response time.

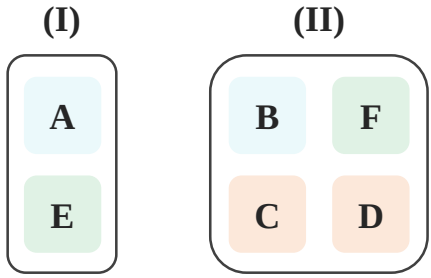

Fig. 6.1 Thesis Contributions grouped into two categories: (I) Contributions highlighting deep learning models capabilities to learn on compressed data. (II) Contributions highlighting the deployability of these models in existing infrastructures.

As show in Figure 6.1, the six contributions of this thesis can be grouped into two categories:

(I) **Contributions putting forward deep learning models ability to learn on compressed data.**

We showed that in the cases of deep learning-based segmentation (Contribution A) and object detection (Contribution E), fine-tuning models on compressed data can be highly effective in both learning and filtering out compression artifacts. Both direct and iterative fine-tuning provide significant benefits for segmentation and detection models, demonstrating that classical human vision-oriented compression can serve as a strong feature extractor for both human and computer vision.

(II) **Contributions showing the deployability of these models in existing infrastructures.**

We demonstrated that in all three use cases — Medical Imaging (Contribution B), Video Surveillance (Contributions C and D), and Remote Sensing (Contribution F) — deep learning models trained on compressed data can be effectively deployed within existing frameworks. These models successfully enable us to reduce storage constraints (Contribution B) and alleviate bandwidth stress (Contributions C, D, and F), all while maintaining high task accuracy.

6.2 Perspectives

6.2.1 Central Research Question

At the start of this thesis, we challenged ourselves to answer the following:

Central Research Question
Can we better utilize the existing data management infrastructure to improve model performance on compressed data?

Overall, the thesis contributions encourages us to **positively** answer the **Central Research Question**. However, we have barely scratched the surface in terms of applications needing to handle a trade-off between model performance and data compression. Extending this work to other use cases — beyond medical imaging, video surveillance, and remote sensing — is crucial to answer the **Central Research Question** more affirmatively.

6.2.2 Promising Use Cases

Three highly encouraging use cases that need to handle the trade-off between model performance and data compression are **genome sequencing**, **space imaging** and **compressed language modeling**.

In **genome sequencing**, the sheer volume of data generated poses significant challenges for storage, transmission, and analysis. Sequencing a single human genome produces approximately 200 GB of raw data [135], primarily due to the fact that a single human's genome consists of 3 billion nucleotide base pairs. This data consists of raw reads from sequencing machines, metadata, quality scores, and intermediate outputs from various preprocessing steps. If applied to large-scale studies involving thousands or millions of genomes, such as in population genomics or personalized medicine, the data size quickly scales to petabytes, making efficient data handling critical. Automating the genome analysis process is crucial to detect mutations as small as a single nucleotide [136].

In **space imaging**, satellites often generate vast amounts of images and sensor data as they orbit Earth, capturing critical information about our planet, distant celestial objects, and various astrophysical phenomena. However, transmitting this enormous volume of data to ground stations poses a significant challenge due to limited transmission windows. These windows are constrained by the satellite's orbital path and its proximity to Earth, during which high-speed communication is possible. As a result, only a fraction of the collected data can be transmitted back before the satellite moves out of optimal range. Compressing the data can significantly increase the sampling rate, allowing for more data acquisition and transmission [137–139]. A key benefit of this approach is the enhanced ability to detect and analyze rare astrophysical events, such as pulsating stars or transient phenomena, which require high-frequency monitoring [140]. By improving data acquisition and transfer, compression increase the likelihood of capturing these fleeting events.

Another highly interesting domain is the training of **compressed language models**. These models are trained directly on compressed byte streams rather than raw, uncompressed data [141]. This approach eliminates the need for decompression, enabling models to understand the syntax, semantics, and structure of compressed file formats. By operating on compressed data, compressed-language models benefit from its inherent compactness, allowing them to process larger datasets within the same computational and storage constraints. This efficiency also removes the need for preprocessing steps like tokenization or pooling, as compressed formats already reduce redundancy. Compressed language models have demonstrated strong performance across tasks such as file recognition, anomaly detection and correction, and compressed file generation.

6.3 Take Home Messages

We conclude this thesis by outlining three key take-home messages:

Take Home Message 1

Classical compression codecs embedded in existing data management infrastructure can serve for both human and computer vision.

Take Home Message 2

Deep learning models trained on compressed data maintain their effectiveness even at high compression ratios.

Take Home Message 3

Encouraging the use of deep learning models trained on compressed data offers a promising solution to overcome current critical bottlenecks in various research fields.

Bibliography

[1] Karim El Khoury, Martin Fockedey, Eliott Brion, and Benoît Macq. Improved 3D U-Net robustness against JPEG 2000 compression for male pelvic organ segmentation in radiotherapy. *Journal of Medical Imaging*, 8(4):041207, 2021.

[2] Karim El Khoury, Jonathan Samelson, and Benoît Macq. Deep learning-based object tracking via compressed domain residual frames. *Frontiers in Signal Processing*, 1:765006, 2021.

[3] Karim El Khoury, Tiffanie Godelaine, Simon Delvaux, Sebastien Lugan, and Benoît Macq. Streamlined hybrid annotation framework using scalable codestream for bandwidth-restricted uav object detection. *arXiv preprint arXiv:2402.04673*, 2024.

[4] Karim El Khoury, Maxence Wynen, Pietro Maggi, Meritxell Bach Cuadra, and Benoît Macq. Improving 3d lesion segmentation robustness against image compression in multiple sclerosis. In *2024 IEEE International Symposium on Biomedical Imaging*, page 1, 2024.

[5] Karim El Khoury, Martin Fockedey, Eliott Brion, and Benoît Macq. Amélioration de la robustesse de l'u-net 3d contre la compression jpeg2000 pour la segmentation des organes pelviens masculins. In *CORESA 2021*, 2021.

[6] Karim El Khoury, Maxime Zanella, Benoît Gérin, Tiffanie Godelaine, Benoît Macq, Saïd Mahmoudi, Christophe De Vleeschouwer, and Ismail Ben Ayed. Enhancing remote sensing vision-language models for zero-shot scene classification. *arXiv preprint arXiv:2409.00698*, 2024.

[7] Manon Dausort, Tiffanie Godelaine, Maxime Zanella, Karim El Khoury, Isabelle Salmon, and Benoît Macq. Exploring foundation models fine-tuning for cytology classification. *arXiv preprint arXiv:2411.14975*, 2024.

[8] Martin Danhier, Karim El Khoury, and Benoît Macq. An open-source fine-grained benchmarking platform for wireless virtual reality. In *International Conference on Virtual Reality and Mixed Reality*, pages 115–121. Springer, 2023.

[9] Pauline Hermans, Tom Gautot, Karim El Khoury, Aloys du Bois d'Aische, and Benoît Macq. Svc: An open-source secure vcf compression tool with conditional random access. In *Spatial Omics*, 2024.

[10] David Reinsel-John Gantz-John Rydning, John Reinsel, and John Gantz. The digitization of the world from edge to core. *Framingham: International Data Corporation*, 16:1–28, 2018.

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

[12] Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Advances in neural information processing systems*, 27, 2014.

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

[15] Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: Learning dense volumetric segmentation from sparse annotation, 2016.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Atten-

tion is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[17] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[18] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[19] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[21] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[22] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[23] Isabella Castiglioni, Leonardo Rundo, Marina Codari, Giovanni Di Leo, Christian Salvatore, Matteo Interlenghi, Francesca Gallivanone, Andrea Cozzi, Natascha Claudia D'Amico, and Francesco Sardanelli. Ai applications to medical images: From machine learning to deep learning. *Physica medica*, 83:9–24, 2021.

[24] Mohd Javaid, Abid Haleem, Ravi Pratap Singh, Rajiv Suman, and Shanay Rab. Significance of machine learning in healthcare: Features, pillars and applications. *International Journal of Intelligent Networks*, 3:58–73, 2022.

[25] Mohamed Elhoseny. Multi-object detection and tracking (modt) machine learning model for real-time video surveillance systems. *Circuits, Systems, and Signal Processing*, 39(2):611–630, 2020.

[26] Chen Chen, Bin Liu, Shaohua Wan, Peng Qiao, and Qingqi Pei. An edge traffic flow detection scheme based on deep learning in an intelligent transportation system. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1840–1852, 2020.

[27] Qiangqiang Yuan, Huanfeng Shen, Tongwen Li, Zhiwei Li, Shuwen Li, Yun Jiang, Hongzhang Xu, Weiwei Tan, Qianqian Yang, Jiwen Wang, et al. Deep learning in environmental remote sensing: Achievements and challenges. *Remote sensing of Environment*, 241:111716, 2020.

[28] Ayesha Shafique, Guo Cao, Zia Khan, Muhammad Asad, and Muhammad Aslam. Deep learning-based change detection in remote sensing images: A review. *Remote Sensing*, 14(4):871, 2022.

[29] Eric Masanet, Arman Shehabi, Nuoa Lei, Sarah Smith, and Jonathan Koomey. Recalibrating global data center energy-use estimates. *Science*, 367(6481):984–986, 2020.

[30] Joao Ascenso, Elena Alshina, and Touradj Ebrahimi. The jpeg ai standard: Providing efficient human and machine visual data consumption. *Ieee Multimedia*, 30(1):100–111, 2023.

[31] Lingyu Duan, Jiaying Liu, Wenhan Yang, Tiejun Huang, and Wen Gao. Video coding for machines: A paradigm of collaborative compression and intelligent analytics. *IEEE Transactions on Image Processing*, 29:8680–8695, 2020.

[32] Ruoyu Feng, Xin Jin, Zongyu Guo, Runsen Feng, Yixin Gao, Tianyu He, Zhizheng Zhang, Simeng Sun, and Zhibo Chen. Image coding for machines with omnipotent feature learning. In *European Conference on Computer Vision*, pages 510–528. Springer, 2022.

[33] Shuai Li, Yanbo Gao, Chuankun Li, and Hui Yuan. Aligned intra prediction and hyper scale decoder under multistage context model for jpeg ai. *IEEE Signal Processing Letters*, 2024.

[34] Markusd Herrmann, Davida Clunie, Andriy Fedorov, Seanw Doyle, Steven Pieper, Veronica Klepeis, Longp Le, Georgel Mutter, Davids

Milstone, Thomasj Schultz, and et al. Implementing the dicom standard for digital pathology. *Journal of Pathology Informatics*, 9(1):37, 2018.

[35] S. Zhou, C. Deng, B. Zhao, Y. Xia, Q. Li, and Z. Chen. Remote sensing image compression: A review. In *2015 IEEE International Conference on Multimedia Big Data*, pages 406–410, 2015.

[36] Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. Effects of degradations on deep neural network architectures. *CoRR*, abs/1807.10108, 2018.

[37] Jonas Löhdefink, Andreas Bär, Nico M. Schmidt, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. On low-bitrate image compression for distributed automotive perception: Higher peak snr does not mean better semantic segmentation. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 424–431, 2019.

[38] Suvash Sharma, Christopher Hudson, Daniel Carruth, Matt Doude, John E. Ball, Bo Tang, Chris Goodin, and Lalitha Dabbiru. Performance analysis of semantic segmentation algorithms trained with JPEG compressed datasets. In Nasser Kehtarnavaz and Matthias F. Carlsohn, editors, *Real-Time Image Processing and Deep Learning 2020*, volume 11401, pages 1 – 12. International Society for Optics and Photonics, SPIE, 2020.

[39] Kristian Fischer, Christian Blum, Christian Herglotz, and André Kaup. Robust deep neural object detection and segmentation for automotive driving scenario with compressed image data. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2021.

[40] Alban Marie, Karol Desnos, Luce Morin, and Lu Zhang. Video coding for machines: Large-scale evaluation of deep neural networks robustness to compression artifacts for semantic segmentation. In *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, pages 01–06. IEEE, 2022.

[41] Vajira Thambawita, Inga Strümke, Steven A Hicks, Pål Halvorsen, Sravanthi Parasa, and Michael A Riegler. Impact of image resolution on deep learning performance in endoscopy image classification: An

experimental study using a large dataset of endoscopic images. *Diagnostics*, 11(12):2183, 2021.

[42] Farhad Ghazvinian Zanjani, Svitlana Zinger, Bastian Piepers, Saeed Mahmoudpour, and Peter Schelkens. Impact of jpeg 2000 compression on deep convolutional neural networks for metastatic cancer detection in histopathological images. *Journal of Medical Imaging*, 6(02):1, 2019.

[43] Jiajie Zhang, Jian Weng, Weiqi Luo, Jia-Nan Liu, Anjia Yang, Jiancheng Lin, Zhijun Zhang, and Hailiang Li. Remt: A real-time end-to-end media data transmission mechanism in uav-aided networks. *IEEE network*, 32(5):118–123, 2018.

[44] Akshara Preethy Byju, Gencer Sumbul, Begüm Demir, and Lorenzo Bruzzone. Remote-sensing image scene classification with deep neural networks in jpeg 2000 compressed domain. *IEEE Transactions on Geoscience and Remote Sensing*, 59(4):3458–3472, 2021.

[45] Akshara Preethy Byju, Gencer Sumbul, Begüm Demir, and Lorenzo Bruzzone. Approximating JPEG 2000 wavelet representation through deep neural networks for remote sensing image scene classification. In Lorenzo Bruzzone and Francesca Bovolo, editors, *Image and Signal Processing for Remote Sensing XXV*, volume 11155, page 111550S. International Society for Optics and Photonics, SPIE, 2019.

[46] Chenyi Zeng, Lin Gu, Zhenzhong Liu, and Shen Zhao. Review of deep learning approaches for the segmentation of multiple sclerosis lesions on brain mri. *Frontiers in Neuroinformatics*, 14:610967, 2020.

[47] Jan Schreier, Angelo Genghi, Hannu Laaksonen, Tomasz Morgas, and Benjamin Haas. Clinical evaluation of a full-image deep segmentation algorithm for the male pelvis on cone-beam ct and ct. *Radiotherapy and Oncology*, 145:1–6, 2020.

[48] Jean Léger, Eliott Brion, Paul Desbordes, Christophe De Vleeschouwer, John A. Lee, and Benoit Macq. Cross-domain data augmentation for deep-learning-based male pelvic organ segmentation in cone beam ct. *Applied Sciences*, 10(3):1154, 2020.

[49] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) stan-

dard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012.

[50] Yu-Wen Huang, Chih-Wei Hsu, Ching-Yeh Chen, Tzu-Der Chuang, Shih-Ta Hsiang, Chun-Chia Chen, Man-Shu Chiang, Chen-Yen Lai, Chia-Ming Tsai, Yu-Chi Su, Zhi-Yi Lin, Yu-Ling Hsiao, Olena Chubach, Yu-Cheng Lin, and Shaw-Min Lei. A vvc proposal with quaternary tree plus binary-ternary tree coding block structure and advanced coding techniques. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(5):1311–1325, 2020.

[51] Debargha Mukherjee, Jim Bankoski, Adrian Grange, Jingning Han, John Koleszar, Paul Wilkins, Yaowu Xu, and Ronald Bultje. The latest open-source video codec vp9 - an overview and preliminary results. In *2013 Picture Coding Symposium (PCS)*, pages 390–393, 2013.

[52] A. Skodras, C. Christopoulos, and T. Ebrahimi. The jpeg 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58, 2001.

[53] Antonin Descampe, Christophe De Vleeschouwer, Pierre Vandergheynst, and Benoit Macq. Scalable feature extraction for coarse-to-fine jpeg 2000 image classification. *IEEE Transactions on Image Processing*, 20(9):2636–2649, 2011.

[54] Antonin Descampe, Pierre Vandergheynst, Christophe De Vleeschouwer, and Benoit Macq. Coarse-to-fine textures retrieval in the jpeg 2000 compressed domain for fast browsing of large image databases. In *Multimedia Content Representation, Classification and Security: International Workshop, MRCS 2006, Istanbul, Turkey, September 11-13, 2006. Proceedings*, pages 282–289. Springer, 2006.

[55] Derrick S Campbell and William D Reynolds Jr. Wavelet-based image registration with jpeg2000 compressed imagery. In *Visual Information Processing XVII*, volume 6978, pages 82–93. SPIE, 2008.

[56] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024.

[57] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4):1680–1716, 2023.

[58] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. *CoRR*, abs/2011.08036, 2020.

[59] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[60] Alan J Thompson, Brenda L Banwell, Frederik Barkhof, William M Carroll, Timothy Coetzee, Giancarlo Comi, Jorge Correale, Franz Fazekas, Massimo Filippi, Mark S Freedman, et al. Diagnosis of multiple sclerosis: 2017 revisions of the mcdonald criteria. *The Lancet Neurology*, 17(2):162–173, 2018.

[61] Mike P Wattjes, Olga Ciccarelli, Daniel S Reich, Brenda Banwell, Nicola de Stefano, Christian Enzinger, Franz Fazekas, Massimo Filippi, Jette Frederiksen, Claudio Gasperini, et al. 2021 magnims–cmsc–naims consensus recommendations on the use of mri in patients with multiple sclerosis. *The Lancet Neurology*, 20(8):653–670, 2021.

[62] Jean Léger, Eliott Brion, Umair Javaid, John Lee, Christophe De Vleeschouwer, and Benoit Macq. Contour propagation in ct scans with convolutional neural networks. In *Advanced Concepts for Intelligent Vision Systems: 19th International Conference, ACIVS 2018, Poitiers, France, September 24–27, 2018, Proceedings 19*, pages 380–391. Springer, 2018.

[63] Maxence Wynen, Pedro M. Gordaliza, Anna Stolting, Pietro Maggi, and Meritxell Bach Cuadra. Lesion instance segmentation in multiple sclerosis: Assessing the efficacy of statistical lesion splitting. In *Proceedings of the ISMRM*, 2024.

[64] Thomas Kalinski, Ralf Zwönitzer, Florian Grabellus, Sien-Yi Sheu, Saadettin Sel, Harald Hofmann, and Albert Roessner. Lossless compression of jpeg2000 whole slide images is not required for diagnostic virtual microscopy. *American Journal of Clinical Pathology*, 136(6):889–895, 2011.

[65] Elizabetha Krupinski, Stacey Jaw, Ronalds Weinstein, Jeffreyp Johnson, and Annar Graham. Compressing pathology whole-slide images using a human and model observer evaluation. *Journal of Pathology Informatics*, 3(1):17, 2012.

[66] Liron Pantanowitz, Chi Liu, Yue Huang, Huazhang Guo, and Gustavok Rohde. Impact of altering various image parameters on human epidermal growth factor receptor 2 image analysis data quality. *Journal of Pathology Informatics*, 8(1):39, 2017.

[67] Alvin Marcelo, Paul Fontelo, Miguel Farolan, and Hernani Cualing. Effect of image compression on telepathology: a randomized clinical trial. *Archives of pathology & laboratory medicine*, 124(11):1653–1656, 2000.

[68] K. Steinhofel, C.F. Dewey, D. Janssens, and B. Macq. Classification of jpeg2000 compressed ct images. In *2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No.02TH8628)*, volume 1, pages 381–385 vol.1, 2002.

[69] Colin Vanden Bulcke, Maxence Wynen, Jules Detobel, Francesco La Rosa, Martina Absinta, Laurence Dricot, Benoît Macq, Meritxell Bach Cuadra, and Pietro Maggi. Bmat: An open-source bids managing and analysis tool. *NeuroImage: Clinical*, 36:103252, 2022.

[70] Maxence Wynen, Maxime Istasse, Pedro M. Gordaliza, Anna Stölting, Pietro Maggi, Benoît Macq, and Meritxell Bach Cuadra. Conflunet: Improving confluent lesion identification in multiple sclerosis with instance segmentation. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1–5, 2024.

[71] Eliott Brion, Jean Léger, A.M. Barragán-Montero, Nicolas Meert, John A. Lee, and Benoit Macq. Domain adversarial networks and intensity-based data augmentation for male pelvic organ segmentation in cone beam ct. *Computers in Biology and Medicine*, 131:104269, 2021.

[72] Jean Léger, Eliott Brion, Paul Desbordes, Christophe De Vleeschouwer, John A. Lee, and Benoit Macq. Cross-domain data augmentation for deep-learning-based male pelvic organ segmentation in cone beam ct. *Applied Sciences*, 10(3), 2020.

[73] Maxence Wynen, Pedro M Gordaliza, Anna Stölting, Pietro Maggi, Meritxell Bach Cuadra, and Benoit Macq. Multi-modal segmentation for paramagnetic rim lesion detection in multiple sclerosis. In *Medical Imaging 2024: Imaging Informatics for Healthcare, Research, and Applications*, volume 12931, pages 241–246. SPIE, 2024.

[74] Colin Vanden Bulcke, Anna Stölting, Dragan Maric, Benoît Macq, Martina Absinta, and Pietro Maggi. Comparative overview of multi-shell diffusion mri models to characterize the microstructure of multiple sclerosis lesions and periplaques. *NeuroImage: Clinical*, 42:103593, 2024.

[75] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[76] Eliott Brion, Jean Léger, Umair Javaid, John Lee, Christophe De Vleeschouwer, and Benoit Macq. Using planning CTs to enhance CNN-based bladder segmentation on cone beam CT. In Baowei Fei and Cristian A. Linte, editors, *Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 10951, page 109511M. International Society for Optics and Photonics, SPIE, 2019.

[77] Francesco La Rosa, Ahmed Abdulkadir, Mário João Fartaria, Reza Rahmanzadeh, Po-Jui Lu, Riccardo Galbusera, Muhamed Barakovic, Jean-Philippe Thiran, Cristina Granziera, and Meritxell Bach Cuadra. Multiple sclerosis cortical and wm lesion segmentation at 3t mri: a deep learning method based on flair and mp2rage. *NeuroImage: Clinical*, 27:102335, 2020.

[78] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[79] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M. Jorge Cardoso. *Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations*, page 240–248. Springer International Publishing, 2017.

[80] Viviana Carillo, Cesare Cozzarini, Lucia Perna, Mauro Calandra, Stefano Gianolini, Tiziana Rancati, Antonello Enrico Spinelli, Vittorio Vavassori, Sergio Villa, Riccardo Valdagni, and Claudio Fiorino. Contouring variability of the penile bulb on ct images: Quantitative assessment using a generalized concordance index. *International Journal of Radiation Oncology*Biology*Physics*, 84(3):841–846, 2012.

[81] Zhitao Xiao, Bowen Liu, Lei Geng, Fang Zhang, and Yanbei Liu. Segmentation of lung nodules using improved 3d-unet neural network. *Symmetry*, 12(11):1787, 2020.

[82] Wenshuai Zhao, Dihong Jiang, Jorge Peña Queralta, and Tomi Westerlund. Mss u-net: 3d segmentation of kidneys and tumors from ct images with a multi-scale supervised u-net. *Informatics in Medicine Unlocked*, 19:100357, 2020.

[83] Marija Habijan, Hrvoje Leventic, Irena Galic, and Danilo Babin. Whole heart segmentation from ct images using 3d u-net architecture. *2019 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2019.

[84] Emir Benca, Morteza Amini, and Dieter H Pahr. Effect of ct imaging on the accuracy of the finite element modelling in bone. *European Radiology Experimental*, 4:1–8, 2020.

[85] Risa Kanatani, Takashi Shirasaka, Tsukasa Kojima, Toyoyuki Kato, and Masateru Kawakubo. Influence of beam hardening in dual-energy ct imaging: phantom study for iodine mapping, virtual monoenergetic imaging, and virtual non-contrast imaging. *European radiology experimental*, 5:1–8, 2021.

[86] KP Mahesh, Prasannasrinivas Deshpande, and S Viveka. Prevalence of artifacts in cone-beam computed tomography: A retrospective study. *Journal of Indian Academy of Oral Medicine and Radiology*, 34(4):428–431, 2022.

[87] Jianran Sha and Jianwu Li. Removing ring artifacts in cbct images via transformer with unidirectional vertical gradient loss. *Medical Physics*, 2024.

[88] Chantal MW Tax, Matteo Bastiani, Jelle Veraart, Eleftherios Garyfallidis, and M Okan Irfanoglu. What's new and what's next in diffusion mri preprocessing. *NeuroImage*, 249:118830, 2022.

[89] Suman Kumar Choudhury, Pankaj Kumar Sa, Sambit Bakshi, and Banshidhar Majhi. An evaluation of background subtraction for object detection vis-a-vis mitigating challenging scenarios. *IEEE Access*, 4:6133–6150, 2016.

[90] Rudrika Kalsotra and Sakshi Arora. Background subtraction for moving object detection: explorations of recent developments and challenges. *The Visual Computer*, 38(12):4151–4178, 2022.

[91] Ma'moun Al-Smadi, Khairi Abdulrahim, Kamaruzzaman Seman, and Rosalina Abdul Salam. A new spatio-temporal background–foreground bimodal for motion segmentation and detection in urban traffic scenes. *Neural Computing and Applications*, 32(13):9453–9469, 2020.

[92] Midhula Vijayan and Mohan Ramasundaram. Moving object detection using vector image model. *Optik*, 168:963–973, 2018.

[93] Chulyeon Kim, Jiyoung Lee, Taekjin Han, and Young-Min Kim. A hybrid framework combining background subtraction and deep neural networks for rapid person detection. *Journal of Big Data*, 5, 07 2018.

[94] Takayuki Ujiie, Masayuki Hiromoto, and Takashi Sato. Interpolation-based object detection using motion vectors for embedded real-time tracking systems. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

[95] Qiankun Liu, Bin Liu, Yue Wu, Weihai Li, and Nenghai Yu. Real-time online multi-object tracking in compressed domain. *IEEE Access*, 7:76489–76499, 2019.

[96] Zhiming Luo, Frédéric Branchaud-Charron, Carl Lemaire, Janusz Konrad, Shaozi Li, Akshaya Mishra, Andrew Achkar, Justin Eichel, and Pierre-Marc Jodoin. Mio-tcd: A new benchmark dataset for vehicle classification and localization. *IEEE Transactions on Image Processing*, 27(10):5129–5141, 2018.

[97] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[98] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2017.

[99] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016.

[100] F. Vermaut, Y. Deville, X. Marichal, and B. Macq. A distributed adaptive block matching algorithm: Dis-abma. *Signal Processing: Image Communication*, 16(5):431–444, 2001.

[101] Aroh Barjatya. Block matching algorithms for motion estimation. *IEEE Transactions Evolution Computation*, 8:225–239, 01 2004.

[102] Jonathon Luiten, Ošep Aljoša, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129(2):548–578, 2020.

[103] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[104] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Yue Yao, Liang Zheng, Pranamesh Chakraborty, Christian E. Lopez, Anuj Sharma, Qi Feng, Vitaly Ablavsky, and Stan Sclaroff. The 5th ai city challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021.

[105] Milind Naphade, Ming-Ching Chang, Anuj Sharma, David C. Anastasiu, Vamsi Jagarlamudi, Pranamesh Chakraborty, Tingting Huang, Shuo Wang, Ming-Yu Liu, Rama Chellappa, Jenq-Neng Hwang, and Siwei Lyu. The 2018 nvidia ai city challenge. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 53–537, 2018.

[106] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and

Jenq-Neng Hwang. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8789–8798, 2019.

[107] Yu Jiang, Yuehang Wang, Minghao Zhao, Yongji Zhang, and Hong Qi. Nighttime traffic object detection via adaptively integrating event and frame domains. *Fundamental Research*, 2023.

[108] F. Ziliani. The importance of 'scalability' in video surveillance architectures. In *The IEE International Symposium on Imaging for Crime Detection and Prevention, 2005. ICDP 2005.*, pages 29–32, 2005.

[109] Antonio C Nazare Jr and William Robson Schwartz. A scalable and flexible framework for smart video surveillance. *Computer Vision and Image Understanding*, 144:258–275, 2016.

[110] Amal Ben Hamida, Mohamed Koubaa, Henri Nicolas, and Chokri Ben Amar. Video surveillance system based on a scalable application-oriented architecture. *Multimedia Tools and Applications*, 75:17187–17213, 2016.

[111] F. Dufaux and T. Ebrahimi. Scrambling for video surveillance with privacy. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 160–160, 2006.

[112] Paula Carrillo, Hari Kalva, and Spyros Magliveras. Compression independent object encryption for ensuring privacy in video surveillance. In *2008 IEEE International Conference on Multimedia and Expo*, pages 273–276, 2008.

[113] European-Council. Regulation 2016/679, general data protection regulation, 2016.

[114] Vladan Papić, Petar Šolić, Ante Milan, Sven Gotovac, and Miljenko Polić. High-resolution image transmission from uav to ground station for search and rescue missions planning. *Applied Sciences*, 11(5), 2021.

[115] N. Tuśnio and W. Wróblewski. The efficiency of drones usage for safety and rescue operations in an open area: A case from poland. *Sustainability*, 14(1):327, 2021.

[116] Andy Indradjad, Ali Syahputra. Nasution, Hidayat Gunawan, and Ayom Widipaminto. A comparison of satellite image compression methods in the wavelet domain. *Earth and Environmental Science*, 280(1), 2019.

[117] Max Ehrlich, Larry Davis, Ser-Nam Lim, and Abhinav Shrivastava. Analyzing and mitigating jpeg compression defects in deep learning. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2357–2367, 2021.

[118] Benjamin Kellenberger, Diego Marcos, Sylvain Lobry, and Devis Tuia. Half a percent of labels is enough: Efficient animal detection in uav imagery using deep cnns and active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12):9524–9533, 2019.

[119] Asma Yamani, Albandari Alyami, Hamzah Luqman, Bernard Ghanem, and Silvio Giancola. Active learning for single-stage object detection in uav images. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1849–1858, 2024.

[120] Palmaerts Benjamin, Beaumont Benjamin, Graur Dimitri, Swinnen Gérard, and Hallot Eric. Oriented aircraft object detector using scaled yolov4 on very high resolution satellite and synthetic datasets. In *2023 Joint Urban Remote Sensing Event (JURSE)*, pages 1–4, 2023.

[121] Muhammad Hussain. Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*, 11(7), 2023.

[122] Qizhi Xu, Yuan Li, and Zhenwei Shi. Lmo-yolo: A ship detection model for low-resolution optical satellite imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:4117–4131, 2022.

[123] Beibei Cui, Liang Liang, Baofeng Ji, Lei Zhang, Liang Zhao, Kunpeng Zhang, Fengzheng Shi, and Jean-Charles Créput. Exploring the yolo-ft deep learning algorithm for uav-based smart agriculture detection in communication networks. *IEEE Transactions on Network and Service Management*, 2024.

[124] Mohammad Hossein Hamzenejadi and Hadis Mohseni. Fine-tuned yolov5 for real-time vehicle detection in uav imagery: Architectural

improvements and performance boost. *Expert Systems with Applications*, 231:120845, 2023.

[125] Dani Manjah, Davide Cacciarelli, Baptiste Standaert, Mohamed Benkedadra, Gauthier Rotsart de Hertaing, Benoît Macq, Stéphane Galland, and Christophe De Vleeschouwer. Stream-based active distillation for scalable model deployment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4999–5007, June 2023.

[126] Dani Manjah, Davide Cacciarelli, Christophe De Vleeschouwer, and Benoit Macq. Camera clustering for scalable stream-based active distillation, 2024.

[127] Xuan Zhu Guang Xu and Nigel Tapper. Using convolutional neural networks incorporating hierarchical active learning for target-searching in large-scale remote sensing images. *International Journal of Remote Sensing*, 41(11):4057–4079, 2020.

[128] Devis Tuia, FrÉdÉric Ratle, Fabio Pacifici, Mikhail F. Kanevski, and William J. Emery. Active learning methods for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 47(7):2218–2232, 2009.

[129] Andre Stumpf, Nicolas Lachiche, Jean-Philippe Malet, Norman Kerle, and Anne Puissant. Active learning in the spatial domain for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5):2492–2507, 2014.

[130] Claudio Persello and Lorenzo Bruzzone. Active learning for domain adaptation in the supervised classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 50(11):4468–4483, 2012.

[131] Zhecheng Wang, Rajanie Prabha, Tianyuan Huang, Jiajun Wu, and Ram Rajagopal. Skyscript: A large and semantically diverse vision-language dataset for remote sensing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5805–5813, 2024.

[132] Zilun Zhang, Tiancheng Zhao, et al. Rs5m and georsclip: A large scale vision-language dataset and a large vision-language model for remote sensing. *arXiv preprint arXiv:2306.11300*, 2024.

[133] Fan Liu, Delong Chen, et al. Remoteclip: A vision language foundation model for remote sensing. *IEEE Trans. Geosci. Remote Sens.*, 2024.

[134] Maxime Zanella, Benoît Gérin, and Ismail Ben Ayed. Boosting vision-language models with transduction. *arXiv preprint arXiv:2406.01837*, 2024.

[135] Vivien Marx. The big challenges of big data. *Nature*, 498(7453):255–260, 2013.

[136] Alex V Nesta, Denisse Tafur, and Christine R Beck. Hotspots of human mutation. *Trends in Genetics*, 37(8):717–729, 2021.

[137] Yong Xu, Jionghui Li, Xiangyu Lin, and Fan Bai. An optimal bit-rate allocation algorithm to improve transmission efficiency of images in deep space exploration. *China Communications*, 17(7):94–100, 2020.

[138] Yong Zhang, Jie Jiang, and Guangjun Zhang. Compression of remotely sensed astronomical image using wavelet-based compressed sensing in deep space exploration. *Remote Sensing*, 13(2):288, 2021.

[139] Weicheng Zhang, Yajing Liu, Lingyu Chen, Jianghong Shi, Xuemin Hong, and Xianbin Wang. Semantically-disentangled progressive image compression for deep space communications: Exploring the ultra-low rate regime. *IEEE Journal on Selected Areas in Communications*, 2024.

[140] J. Audenaert, J. S. Kuszlewicz, R. Handberg, A. Tkachenko, D. J. Armstrong, M. Hon, R. Kgoadi, M. N. Lund, K. J. Bell, L. Bugnet, D. M. Bowman, C. Johnston, R. A. García, D. Stello, L. Molnár, E. Plachy, D. Buzasi, C. Aerts, and The T'DA collaboration. Tess data for asteroseismology (t'da) stellar variability classification pipeline: Setup and application to the kepler q9 data. *The Astronomical Journal*, 162(5):209, oct 2021.

[141] Juan C Pérez, Alejandro Pardo, Mattia Soldan, Hani Itani, Juan Leon-Alcazar, and Bernard Ghanem. Compressed-language models for understanding compressed file formats: a jpeg exploration. *arXiv preprint arXiv:2405.17146*, 2024.