RELEVANCE BASED VIDEO COMPRESSION OF SURVEILLANCE VIDEOS USING DEEP LEARNING METHODS

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Science & Engineering

By

Mohod Nikita Prabhakar

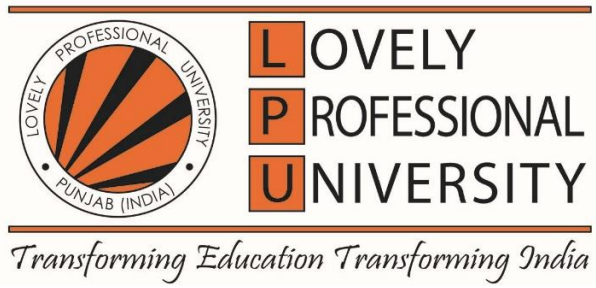
Registration Number: 42000365

Supervised By

Dr. Prateek Agrawal (13714)

Computer Science & Engineering (Professor & Associate Dean)

Lovely Professional University



LOVELY PROFESSIONAL UNIVERSITY, PUNJAB

2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LOVELY PROFESSIONAL UNIVERSITY

Punjab, India-144411

DECLARATION

I, hereby declared that the presented work in the thesis entitled "Relevance based video compression of surveillance videos using deep learning methods" in fulfillment of degree of **Doctor of Philosophy (Ph.D.)** is the outcome of research work carried out by me under the supervision of Dr. Prateek Agrawal, working as Professor & Associate Dean, in the School of Computer Science Engineering, of Lovely Professional University, Punjab, India. In keeping with general practice of reporting scientific observations, due acknowledgments have been made whenever work described here has been based on findings of other investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.

(Signature of Scholar)

Name of the scholar: Mohod Nikita Prabhakar

Registration No.: 42000365

Department/School: School of Computer Science and Engineering

Place: Lovely Professional University, Punjab, India

Date: 28.06.2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LOVELY PROFESSIONAL UNIVERSITY

Punjab, India-144411

CERTIFICATE

This is to certify that the work reported in the Ph.D. thesis entitled "Relevance based video compression of surveillance videos using deep learning methods" submitted in fulfillment of the requirement for the award of degree of **Doctor of Philosophy (Ph.D.)** in the School of Computer Science and Engineering, is a research work carried out by Nikita P. Mohod, 42000365, is bonafide record of her original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.

(Signature of Supervisor)

Name of the Supervisor: Dr. Prateek Agrawal

Designation: Professor & Associate Dean

Department: School of Computer Science and Engineering

University: Lovely Professional University, Punjab, India

Date: 28.04.2025

Abstract

The prevalence of closed-circuit television cameras (CCTV) has witnessed a rapid escalation over recent decades. CCTV serves the purposes of safety, security, and monitoring across various sectors. Though the requirement for CCTV increases, it faces the major challenges of surveillance video storage. Surveillance cameras yield a substantial data payload, particularly within high-resolution systems. The accumulation of copious amounts of high-definition video data can swiftly saturate local hard drives or cloud storage repositories, resulting in expenses associated with the acquisition and ongoing maintenance of adequate storage capacity. This not only increases the expenses of its maintenance but also compels the owners to delete the recorded video after a certain interval of time. These data deletions sometimes lead to the loss of relevant information that might be useful for future perspectives. To address this issue, we proposed three models: *(i)* Object Detection based Surveillance Video Compression (ODSC) Model. *(ii)* Relevant Frame Detection and Compression (D&C) Model and *(iii)* Frame Relevance-based Video Compression (FRVC).

The ODSC model is divided into two steps: -i) depending on the objects in the video, determine the relevant and irrelevant frames of surveillance video using the neural network approach YOLOv5, YOLOv7 and YOLOv8 ii) construct the video of relevant frames. Following a comprehensive analysis of the experimental outcomes, it is noted that YOLOv8 stands out with a remarkable detection accuracy of 98.7% on the COCO dataset. Our ODSC approach is reducing the storage space greatly and achieving an average compression ratio of up to 66.23% on seven ATM surveillance videos.

In the D&C model, we used the COCO dataset to train relevant frame detec-

tion modules and tested them on six different ATM surveillance videos for two different scenarios. In the detection module, the YOLOv5, YOLOv7, YOLOv8 and YOLOv9 modules of object detection are deployed to predict the relevant and irrelevanr frames of surveillance video. Later, the relevant frames are forwarded to the similarity identification module to perform further compression. Experimental results show that in the pertinent frame detection model YOLOvp surpasses YOLOv5, YOLOv7 and YOLOv9 in terms of speed and accuracy. Using the proposed D&C model, the highest compression of 98.96% and 68.64% lowest compression is achieved by maintaining the same resolution and frame per second at 90% threshold value.

In the FRVC model, we present the ATM Surveillance Video (ASV) dataset to train the object detection module YOLOv9 and MASk-RCNN. These object detection module detects the relevant and irrelevant frames of surveillance and relevant frames are given as input to further video compression module. We applied our approach to seven distinct CCTV surveillance under three different scenarios. In the relevance frame classification module, YOLOv9 surpasses the Mask-RCNN in terms of accuracy and speed. Using the proposed FRVC framework, we achieved the highest 96.3% compression for scenario-I and 63.5% for scenario-II. While the 79.6% of average compression is received on surveillance video, by preserving the video quality, particularly in terms of resolution and frame per second (FPS). Our FRVC framework surpasses the existing state-of-the-art approach.

All the models are implemented in GPU-enabled Intel Xeon®Gold 5222 3.8GHz processor workstation. It contains 1TB SATA hard disk and 128GB DDR4 RAM with Windows 11 Pro operating system. Spyder IDE with Python 3.11 is used for various deep-learning operations.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
LOVELY PROFESSIONAL UNIVERSITY
Punjab, India-144411

ACKNOWLEDGEMENT

I wish to express my sincerest gratitude to my supervisor Dr. Prateek Agrawal for the continuous guidance and mentorship. The journey from conceptualization to completion has been both challenging and rewarding, and I am thankful for the support and assistance that made it possible. Their insights and constructive feedback were invaluable in shaping the direction and quality of this thesis. Your belief in my abilities kept me motivated, and your encouragement supported me throughout my Ph.D. research work journey.

I offer my heartfelt thanks to my god, the guiding force behind every step of my Ph.D. journey. I would like to express my deepest gratitude to my family for their unwavering support and understanding throughout my Ph.D. journey. To my beloved husband, Dr. Amar Sable, your encouragement, patience, and love have been my source of strength and motivation. To my dear son, Devansh Sable, your boundless energy and innocent smile have been a constant source of joy and inspiration. I am also profoundly grateful to my parents and in-laws for their love, guidance, and sacrifices.

(Signature of Scholar)

Name of the scholar: Mohod Nikita Prabhakar

Registration No.: 42000365

Department/School: School of Computer Science and Engineering

Place: Lovely Professional University, Punjab, India

Date: 28.06.2025

Contents

Declaration i

Certificate ii

Abstract iv

Acknowledgement vi

Table of Contents ix

List of Tables xi

List of Figures xiv

List of Symbol xv

List of Abbreviation xvii

1 INTRODUCTION 1

1.1 Introduction 1

1.2 What is Compression ? 3

1.2.1 Lossy Compression 4

1.2.2 Lossless Compression 5

1.3 Evolution of Video Compression 7

1.4 Significance of Surveillance Video Compression 8

1.5 Motivation 10

1.6 An Overview of Deep Learning 11

1.6.1 Deep Learning Use Cases 12

1.6.2 Deep Learning Mechanism 14
1.7 Approach 16
1.8 Thesis Contribution 17
1.9 Thesis Organization 19

2 STATE-OF-THE-ART SURVEY 21
2.1 Object Detection 22
2.1.1 Traditional Object Detection 23
2.1.2 Region-Based Approach 25
2.1.3 Region-Free Approach 28
2.1.4 Findings 31
2.2 Video Compression 32
2.2.1 Intra-Frame Compression 33
2.2.2 Inter-Frame Compression 35
2.2.3 Transform, Quantization and Entropy Coding 38
2.2.4 Post-Loop and In-Loop Filtering 39
2.2.5 Down and Up-Sampling 41
2.2.6 Encoder Optimization 42
2.2.7 End-to-End Compression 44
2.3 Surveillance Video Compression 46
2.4 Open Challenges in Video Compression 47
2.5 Thesis Objectives 47
2.6 Research Methodology 48

3 DEVELOPMENT OF RELEVANCE BASED OBJECT DETECTION AND VIDEO COMPRESSION MODEL 50
3.1 Object Detection Based Surveillance Video Compression 53
3.1.1 Dataset 53
3.1.2 Methodology 55
3.1.3 Workflow and Algorithm 63
3.2 Relevant Video Frame Detection and Compression Model 65

3.2.1 Methodology 66
3.2.2 Workflow and Algorithm 73
3.3 Frame Relevance Based Video Compression 76
3.3.1 ATM Surveillance Video Dataset 76
3.3.2 Methodology 79
3.3.3 Workflow and Algorithm 86
3.4 Summary 90

4 RESULT ANALYSIS AND COMPARISON 91
4.1 Evaluation Metrics 91
4.1.1 Evaluation Metrics in Object Detection 92
4.1.2 Evaluation Metrics in Compression 93
4.2 Object Detection Based Surveillance Video Compression Model . . 94
4.2.1 Comparison of Object Detection Modules 94
4.2.2 Result Analysis of Compression Module 98
4.3 Deep Learning-Based Relevant Video Frame Detection and Compression Model 100
4.3.1 Comparison of Object Detection Modules 100
4.3.2 Result Analysis of Compression Module 102
4.4 Frame Relevance Based Video Compression Model 108
4.4.1 Comparison of Object Detection Modules 108
4.4.2 Result Analysis of Compression Module 112
4.5 Comparison of ODSC, D&C and FRVC Model 121
4.6 Comparison of Proposed Model with State-of-the-Art Models 123
4.7 Summary 124

5 CONCLUSION AND FUTURE SCOPE 126
5.1 Future Work 128

A Research Outcomes 149

List of Tables

3.1 Comparison of YOLOv5, YOLOv7, YOLOv8, and YOLOv9 70

4.1 Confusion Matrix 92
4.2 Original Characteristic of Seven Tested Videos 95
4.3 Parameter Specification of ODSC Model 96
4.4 Comparison of YOLOv5, YOLOv7 and YOLOv8 Modules 96
4.5 Frame Count Using YOLOv8 Model 97
4.6 Output of ODSC Model with Resolution 1280 × 720 on YOLOv8 . 98
4.7 Original Characteristic of Six Tested Videos. 100
4.8 Comparison of YOLO Modules 101
4.9 Frames Count Using YOLOv9 Module 102
4.10 Count of Primary and Similar Frames at Various Threshold Values for Tested Six Videos of D&C Model 103
4.11 Compressed Video Characteristics at Threshold Value = 90, FPS = 15, and Resolution = 1920×1080 for D&C Model 106
4.12 Compression Achieved in MB at Various Threshold (TH) Values . 107
4.13 Original Characteristic of Seven Tested Videos 109
4.14 Parameter Specification 109
4.15 Comparison of YOLOv9 and Mask-RCNN Module 110
4.16 Frames Count Using YOLOv9 Module 112
4.17 Count of Primary and Similar Frames at Various Threshold Values for Tested Seven Videos of FRVC Model 115
4.18 Compressed Video Characteristics at Threshold Value = 90%, FPS = 15, and Resolution = 1920×1080 116

4.19 Compression Achieved in Megabytes (MB) at Various Threshold (TH) Values in FRVC Model 119
4.20 Comparison of Approaches on Evaluation Metrics 122
4.21 Proposed Model Comparison with Existing Approach on FRVC Dataset 123

List of Figures

1.1 Domain-Wise Utilization of Closed-Circuit Television (CCTV) Systems. 2
1.2 Overview of Real-World Applications of Deep Learning Across Various Domains. 13
1.3 Architecture of Convolution Neural Network. 15
1.4 Block Diagram of Thesis Contribution. 18
1.5 Organization of Thesis. 19

2.1 Year-wise Distribution of Published Research Papers From 2019 to 2024 [1] (Source: Google Scholar). 23
2.2 Object Detection's Popular Techniques. 24
2.3 This Figure Presents a Timeline for the Evolution of Object Detection Approaches Using the Neural Network From 2014. 26
2.4 Taxonomy of Object Detection using Neural Network. 28
2.5 General Architecture of YOLO Modules. 30
2.6 Hybrid Video Coding in H.265. 33
2.7 Flowchart of Research Methodology. 48

3.1 Sample Relevant Frame From the Collected ATM Surveillance Video Dataset. 51
3.2 Sample Irrelevant Frame From the Collected ATM Surveillance Video Dataset. 52
3.3 Object Detection Based Surveillance Video Compression Model. 54
3.4 Figure Illustrate the Architecture of YOLOv5 Module Which Consist of Three Stages: Backbone, Neck and Head [29]. 56

3.5 Output of YOLOv5 Module Trained on COCO Dataset (Relevant Frame). 58
3.6 Output of YOLOv5 Module Trained on COCO Dataset (Irrelevant Frame). 59
3.7 This Figure Showcase the YOLOv7 Architecture Module, Which Features an Optimized Backbone, Enhanced Feature Fusion, and an Advanced Detection Head [30]. 60
3.8 Output of YOLOv7 Module Trained on COCO Dataset (Relevant Frame). 61
3.9 The Architecture of the YOLOv8 Module Which Highlights Key Components such as CBS (Conv+BN+SiLU), Bottleneck, and C2F structures. The Data Flow Illustrates a Feature Fusion Through Add and Concat Operations that Optimize Detection Performance [31]. 61
3.10 Output of YOLOv8 Module Trained on COCO Dataset (Relevant Frame). 62
3.11 Sequence Flow Diagram of ODSC Model. 63
3.12 An Overview of Deep Learning-based Relevant Frame Detection and Compression (D&C) model, which consists of three modules: (i) Data Engineering Module, (ii) Relevant Frame Detection Module and (iii) Similarity Identification Module. 65
3.13 The Figure Showcases the Architecture of YOLOv9 Module Which Comprises Three Primary Components: the Backbone, Neck, and Head [32]. 69
3.14 Output of YOLOv9 Module Trained on COCO Dataset (Relevant Frame). 70
3.15 Sequence Flow Diagram of D&C Model. 74
3.16 An Overview of Frame Relevance Based Video Compression model, which consists of three modules: (i) Dataset Preparation Module, (ii) Relevance Frame Classification Module and (iii) Video Compression Module. 77
3.17 Person Annotation Process in Make Sense AI Tool for Dataset Preparation. 79
3.18 Output of YOLOv9 Trained on ASV Dataset (Relevant Frame). 81
3.19 Architecture of Mask R-CNN Module [33]. 82
3.20 Output of Mask R-CNN Module Trained on ASV Dataset (Relevant Frame). 85
3.21 Output of Mask R-CNN Module Trained on ASV Dataset (Irrelevant Frame). 86

3.22 Sequence Flow Diagram of FRVC Model. 87

4.1 Time Taken to Detect Relevant and Irrelevant Frames of Surveillance Video using YOLOv5, YOLOv7 and YOLOv8 Module in ODSC Model. 97

4.2 Graphical Representation Original and Compressed Video Sizes Across Seven Videos of ODSC Model. 99

4.3 Time Taken to Detect Relevant and Irrelevant Frames of Surveillance Video using YOLOv5, YOLOv7, YOLOv8 and YOLOv9 Module in D&C Model. 102

4.4 Count of Frames vs. Similarity Threshold in D&C Model. 105

4.5 Size of Compressed Videos at Different Threshold of D&C Model. 108

4.6 Graphical Performance of YOLOv9 Module in FRVC Model. 110

4.7 Time Taken to Detect Relevant and Irrelevant Frames of Surveillance Video Using YOLOv9 and Mask R-CNN in FRVC Model. 111

4.8 Sample Surveillance video Frame for Scenario I. 113

4.9 Sample Surveillance Video Frame for Scenario II. 113

4.10 Sample Surveillance Video Frame for Scenario III. 114

4.11 Count of Frames vs. Similarity Threshold in FRVC Model. 117

4.12 Size of Compressed Videos at Different Threshold of FRVC Model. . 120

4.13 Pareto Chart for Average % of Compression on Seven Tested Videos. . 121

List of Symbols

$*$	Convolution Opeeration
$+$	Addition
$-$	Subtraction
$\times$	Multiplication
$\in$	Belongs To
$\forall$	For Every
$\sum$	Summation
$<$	Less Than
$>$	Greater Than
$\|\|$	Modulus
$\%$	Percentage
α	Leakage Factor
μ	Mean Intensity
σ	Standard Deviation

List of Abbreviations

AI	Artificial Intelligence
CV	Computer Vision
DL	Deep Learning
ML	Machine Learning
OD	Object Detection
FPS	Frame Per Second
Sec	Seconds
CCTV	Closed Circuit Television
MPEG	Moving Picture Expert Group
FFV1	FFmpeg Video Codec 1
H.265	Highly Efficient Video Coding
VVC	Versatile Video Coding
ATM	Automated Teller Machine
NLP	Natural Language Processing
CNN	Convolutional Neural Network
BN	Batch Normalization
YOLO	You Only Look Once
RoI	Region of Interest
SVM	Support Vector Machine
SPP	Spatial Pyramid Pooling
IoU	Intersection over Union
SSD	Single Shot Detector
CSP	Cross Stage Partial Connection
Bos	Bag of Specials
BoF	Bag of Freebies
CTU	Control Tree Unit
GAN	Generative Adveserial Network
MC	Motion Compensation
ME	Motion Estimation
MP	MaxPool
MSE	Mean SquareError
COCO	Common Objects in Context
SiLU	Sigmoid Linear Unit
ReLU	Rectified Linear Unit
ELAN	Extended Efficient Layer Aggregation Network

FPN	Feature Pyramid Network
PAN	Path Aggregation Network
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
SSIM	Structural Similarity Index Measure
CR	Compression Ratio
BB	Bounding Box
LSTM	Long Short-Term Memory
IoT	Internet of Things
ODSC	Object Detection Based Surveillance Video Compression
D&C	Deep Learning-Based Relevant Video Frame Detection and Compression
FRVC	Frame Relevance Based Video Compression
CIoU	Complete Intersection over Union
DIoU	Distance Intersection over Union
NMS	Non-Maximum Suppression
TH	Threshold
ASV	ATM Surveillance Video Dataset
AP	Average Precision
BCE	Binary Cross Entropy
DFL	Distribution Focal Loss

Chapter 1

INTRODUCTION

1.1 Introduction

"Every great and deep difficulty bears in itself its own solution. It forces us to change our thinking in order to find it [1]."

–Niels Bohr

The pervasive integration of technology into our daily lives is not only on the rise but is also rapidly advancing. Among these technological strides, one pivotal innovation in the realm of security is Closed-Circuit Television [2]. In 1942, German engineer Walter Bruch developed the first Closed-Circuit Television (CCTV) system. By the 1980s and 1990s, the use of CCTV systems increased for surveillance and security purposes in public spaces, businesses, and even residential areas. The use of CCTV became more common in the late 20^{th} century as technology advanced and became more affordable. The proliferation of CCTV systems continued into the 21^{st} century, with advancements in digital technology making them even more prevalent and effective [3]. CCTV surveillance has become an integral part of modern security systems, significantly impacting various aspects of our lives. The widespread adoption of surveillance cameras reflects a growing awareness of the need for enhanced security measures. The use of surveillance has expanded exponentially, driven by technological advancements, increased affordability, and a heightened awareness of security concerns [4]. Functioning as a type of security camera, CCTV creates the perception of an ever-watchful additional set of eyes [5]. These surveillance systems find application in diverse settings, encompassing

both public and private domains. However, as the utilization of CCTV for security purposes continues to expand, the deployment and maintenance of these systems encounter escalating challenges [6]. Figure 1.1 shows the utilization of CCTV in various sectors.

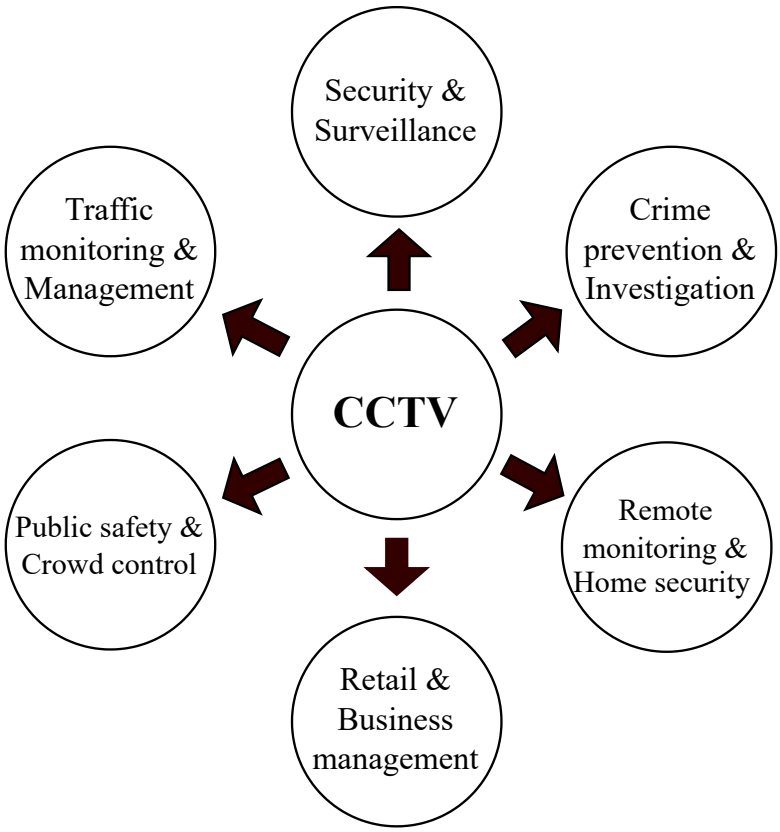


Figure 1.1: Domain-Wise Utilization of Closed-Circuit Television (CCTV) Systems.

A primary obstacle faced by CCTV systems pertains to the substantial storage space consumed by video content. This challenge becomes particularly pronounced due to the prevalence of hard discs as the primary storage medium for recorded footage. To address this issue and optimization of storage efficiency is needed. Various compression techniques are employed to mitigate the storage space requirements of CCTV systems [7]. Relying solely on expanding storage capacity to address the challenges in video surveillance is a simplistic approach. Typically, CCTV surveillance footage is stored selectively, either on cloud platforms or local devices like microSD cards and hard drives. The continuous recording of

round-the-clock surveillance videos quickly depletes storage on local devices. Consequently, videos are routinely deleted after a set time interval. This method may lead to the loss of pertinent information from users view. Considering this necessity a domain-specific relevant data compression techniques should be developed. The thesis concentrate on this topic.

1.2 What is Compression ?

"Compression is an art of reducing the size of data without losing its essential information, akin to squeezing a sponge to extract every last drop of water [8]."

–Archibald MacLeish

In information security, the reduction of data is achieved through compression. It makes the data more efficient to store, transmit, and process [9]. The overarching goal of compression is to minimize redundancy and eliminate unnecessary information within data, without compromising its essential content. This process facilitates faster transmission speeds, reduces storage requirements, and optimizes computational resources. Compression techniques are widely employed in various domains, including text, images, audio, and video, each with specialized algorithms tailored to their unique characteristics. In thesis, the focus is on the compression of the surveillance system.

Video compression reduces the redundant information within video data, which typically consists of a sequence of images displayed in rapid succession [10]. The intricacies of video compression lie in its ability to exploit temporal and spatial redundancies inherent in consecutive frames, which results in substantial data size reduction without compromising visual quality. To achieve effective video compression a balance between fidelity and reduced file size is maintained [11]. lossless and lossy are the two types of video compression and are explained in further subsection 1.2.1 and 1.2.2.

1.2.1 Lossy Compression

For significant data reduction lossy compression discards certain information that is less crucial to the overall perception of the video [12]. This type of compression is particularly prevalent in scenarios where slight quality degradation is acceptable, such as video streaming and multimedia applications. For broadcasting, video conferencing, and online video platforms H.264 as well as Moving Picture Experts Group (MPEG) standards are used. One of the fundamental techniques in lossy compression is quantization, which involves reducing the precision of the data. This can be thought of as rounding off values to a coarser level. For example, in image compression, color values or pixel intensities may be quantized to a limited set of levels. Lossy compression often leverages insights from human perception.

In image and video compression, the human eye has lower sensitivity to certain subtle changes in color and detail, allowing for the removal of less noticeable information [13]. Similarly, in audio compression, the psychoacoustic model is used to identify and discard audio components that are less likely to be perceived by the human ear. These algorithms often use entropy coding techniques to represent frequently occurring patterns with shorter codes, to acheive compression. Huffman coding and arithmetic coding are common entropy coding methods.

Pros of Lossy Compression:

i. Lossy compression can achieve significantly higher compression ratios compared to lossless compression. This makes it suitable for applications where reducing file size is a priority.

ii. Lossy compression is well-suited for multimedia applications, where the human perception system allows for the removal of certain details without a significant loss in perceived quality.

iii. The reduced file sizes resulting from lossy compression contribute to lower storage requirements and make it more efficient for transmitting data over networks, especially in bandwidth-constrained scenarios.

Cons of Lossy Compression:

i. The most significant drawback of lossy compression is the loss of data and quality. This may not be acceptable for applications where maintaining the highest quality is crucial, such as medical imaging.

ii. Compression artifacts, such as blurring, blocking, or ringing, may be introduced in the compressed data. These artifacts can be noticeable, especially at higher compression levels, and may impact the user experience.

As technology continues to advance, the development of more sophisticated lossy compression algorithms aims to further optimize the compromise between compression ratios and preserved quality in diverse data types.

1.2.2 Lossless Compression

To retain all original data without any quality loss, lossless compression is a helpful approach [14]. This method is crucial in applications where data integrity is paramount, such as medical imaging and professional video editing. While lossless compression typically results in less compact files compared to lossy methods, it ensures a perfect reconstruction of the original data. The FFmpeg Video Codec 1 (FFV1) and H.265 are examples of lossless compression standards. The techniques that are generally used to obtain Lossless Compression are *(i)* Run-Length Encoding:- Represents consecutive identical elements (runs) with a single value and a count [15]. For example, the sequence "AAAAAABBCCCDAAA" can be encoded as "6A2B3C1D3A." *(ii)* Dictionary-based Compression:- Lempel-Ziv and its variants are widely used in lossless compression [16]. These algorithms build a dictionary of frequently occurring patterns and replace them with shorter codes. The dictionary is typically updated dynamically during compression. *(iii)* Huffman Coding:- It is an entropy encoding process that allows variable-length codes to input characters depending on their frequencies [17]. More frequently occurring characters are assigned shorter codes, resulting in efficient compression. *(iv)* Burrows-Wheeler Transform (BWT):- BWT reorganizes the characters in a string

to exploit redundancy before applying additional compression techniques so that it is used in two-way mode [18]. It is often used in combination with other algorithms, such as the Move-to-Front (MTF) transform.

Pros of Lossless Compression:

i. With lossless compression, no data is lost in the process of compression, providing you a video that retains all of its original quality.

ii. Since no data is lost, losslessly compressed videos are ideal for editing purposes as they can be decompressed without any loss of quality, allowing for seamless editing workflows.

iii. Lossless compression is well-suited for archiving purposes where maintaining the original quality of the video is essential for future use or reference.

iv. Lossless compression is particularly beneficial for high-quality video content, such as professional video production, where preserving the highest level of detail is crucial.

Cons of Lossless Compression:

i. Losslessly compressed videos tend to have larger file sizes compared to lossy compression methods since all the original data is retained. This can pose challenges for storage and bandwidth requirements.

ii. Encoding and decoding of losslessly compressed videos needed high sources compared to lossy compression. This can impact both hardware requirements and processing time.

iii. Poorer compression ratios are usually obtained when lossless and lossy compression techniques are compared.Due to which the reduction in file size is not significant.

iv. Due to their larger file sizes, losslessly compressed videos are not suitable for streaming applications where bandwidth efficiency is crucial. They may result in buffering issues and slower streaming speeds.

1.3 Evolution of Video Compression

The evolution of video compression is a dynamic journey marked by continuous innovation and adaptation to emerging technological landscapes. Below, we explore the key milestones in the evolution of video compression, highlighting the transformative developments that have shaped the current state of the field.

i. *MPEG-1 and MPEG-2 (1988-1994):* The inception of video compression standards can be traced from the 1980s with the development of MPEG-1 which was ratified in 1992. It aims to compress video data for storage on CDs and revolutionize video distribution. MPEG-2, developed in the early 1990s, extended these capabilities for broadcast television, DVDs, and digital satellite transmission. These standards laid the groundwork for subsequent advancements in video compression [19].

ii. *Rise of Digital Television: MPEG-4 (1999-2003):* The emergence of digital television and the need for more efficient compression led to the development of MPEG-4. Ratified in 1999, MPEG-4 brought about significant improvements in compression efficiency, to enable the transmission of higher-quality video over bandwidth-constrained networks. This standard found applications in video conferencing, mobile multimedia, and internet streaming, contributing to the proliferation of digital content delivery [20].

iii. *High-Definition Era: H.264 (2003-Present):* H.264 standard, finalized in 2003, marked a watershed moment in video compression. Its efficient compression algorithms to exploit spatial and temporal redundancies, significantly enhance video quality while reduces bitrates. H.264 became the cornerstone for high-definition video distribution for platforms such as Blu-ray discs, video streaming services, and video conferencing applications [21].

iv. *Toward Ultra High Definition: H.265/HEVC (2013-Present):* In 2013, the demand for higher resolution content, including Ultra High Definition (UHD) and 4K video, prompted the progress of ratified H.265 and introduced im-

proved compression efficiency. Compared to its predecessor, it allows high-quality video to be sent at lower bitrates. HEVC is integral to the streaming of UHD content and is poised to play a crucial role in the advent of 8K video [22].

v. *The Era of Versatility: VP9 and AV1 (2014-Present):* As video consumption diversified across a myriad of devices and platforms, open-source codecs like VP9 [23] and AV1 [24] emerged to address the need for versatile, royalty-free compression standards. VP9, developed by Google, and AV1, a product of the Alliance for Open Media, prioritize efficiency and flexibility. These codecs are particularly relevant in web-based video streaming, providing alternatives to proprietary standards.

vi. *Immersive Experiences: Future Directions:* With the advent of virtual and augmented reality the demand for immersive video experiences has intensified. Video compression standards are evolving to accommodate these emerging technologies, ensuring efficient delivery of high-quality content. Newer standards, including Versatile Video Coding (VVC) [25], are under development to support enhanced features like 360-degree video and augmented reality applications.

1.4 Significance of Surveillance Video Compression

Surveillance video compression plays a pivotal role in modern security systems to address several critical needs inherent to the surveillance industry. A few of the are summarized as follows.

i. *Storage Efficiency:* Surveillance systems capture vast amounts of video data, often operating 24/7. Without compression, storing this data in its raw, uncompressed form would be impractical due to the immense storage requirements. Surveillance video compression significantly reduces the size of video

files while preserving essential information, enabling efficient storage and management of surveillance footage. This is particularly crucial for organizations and industries that require long-term retention of video data for security, compliance, or evidentiary purposes.

ii. *Cost Savings:* The significance of video compression in surveillance lies in its potential for cost savings. Storing uncompressed video data requires substantial investments in storage infrastructure, including servers, hard drives, and data centers. By compressing video footage, organizations can significantly reduce storage costs by minimizing the amount of storage space required. This cost-effectiveness makes surveillance systems more accessible and sustainable for a wide range of applications, including public safety, law enforcement, transportation, and commercial security.

iii. *Bandwidth Conservation:* Surveillance systems often rely on networks to transmit live video feeds or transfer recorded footage. Uncompressed video data consumes significant bandwidth that leads to network congestion and potential performance issues. Video compression conserves bandwidth by reducing the size of video files, ensuring smooth and efficient transmission of video data over networks. This is essential for real-time monitoring, remote access to surveillance feeds, and sharing video data across distributed locations or devices.

iv. *Real-time Monitoring:* Compression enables real-time streaming and monitoring of video feeds without significant delays. It ensures that surveillance operators can access live video streams efficiently, to enhance their ability to respond to events on time.

v. *Extended Storage Duration:* The significance of video compression extends to the extended storage duration it enables for surveillance footage. By reducing the size of video files, compression allows organizations to store video data for longer durations within available storage capacity. This is essential for compliance with regulatory requirements, such as data retention policies

and legal mandates, as well as for retrospective analysis of security incidents, identification of patterns and trends, and forensic investigations.

vi. *Scalability:* Surveillance systems often need to scale to accommodate additional cameras and increased resolution. Compression techniques help maintain scalability by efficiently managing the increased data load without requiring substantial upgrades to storage or network infrastructure.

1.5 Motivation

CCTV cameras have become integral components of security infrastructure, strategically placed across diverse locations which include educational institutions, commercial establishments, city roads, highways, financial institutions, Automated Teller Machines (ATM), and corporate offices. While these surveillance systems effectively monitor and record events, they grapple with a significant challenge of limited storage capacity. The conventional practice of deleting surveillance data at predefined intervals, such as one month for school and college footage, six months for banks, offices, and ATM, and three months for city roads and highways, results in the loss of relevant data embedded in the video content. We prefer to use ATM surveillance videos for research purpose. The CCTV camera in an ATM room records video 24/7. However, only a few hours of transaction activity occur, and the remainder of the video captures the ATM in a steady state. In reality, the majority of this surveillance video contains static scenes featuring a fixed ATM, with only a fraction of the recorded time-capturing dynamic activities. This phenomenon results in the inefficient utilization of storage space and the situation necessitates the progress of domain-specific compression to selectively retain solely segments containing transaction activity in the surveillance video. To overcome this limitation and preserve crucial data over extended periods, we propose techniques to perform compression of ATM surveillance video. This research is particularly significant in the context of addressing the storage conundrum associated with surveillance videos. This method offers a solution that ensures the

retention of relevant information for more extended durations.

For the detection of relevant intervals of ATM surveillance video, a deep learning approach is used. This process leads to the classification of frames as either relevant or irrelevant. Specifically, frames containing the presence of humans or animals in the ATM room are categorized as relevant frames, whereas frames solely depicting the ATM without such presence are designated as irrelevant frames. Hence, to detect the relevant and irrelevant frames of surveillance video, deep learning-based object detection (OD) frameworks are used in this thesis.

1.6 An Overview of Deep Learning

"In the realm of computer vision, object detection serves as the cornerstone, enabling machines to perceive and understand the visual world around them [26]."

–A. Courville

Deep learning (DL) is a branch of Artificial Intelligence (AI) that focuses on training artificial neural networks to gain knowledge from data representations, enabling them to perform tasks without explicit programming instructions [27]. DL is fundamentally similar to the way the human brain's visual cortex work, which consist of interconnected layers of processing units known as neurons [28]. Each layer processes input data, progressively extracting higher-level features and patterns. DL models are characterized by their depth, they consist of many layers which allows them to learn complex representations of data. During training, these models adjust their internal parameters through a process known as backpropagation. wherein in order to update the parameters and reduce the discrepancy between the expected and actual outputs, errors are propagated backward through the network. DL models are especially well-suited for CV tasks like identifying and recognizing objects in an image, data processing etc. because of their iterative learning process, which enables them to automatically find complex patterns and relationships within the data. Recent advances in DL have been driven by breakthroughs in model architectures, training methods, and the accessibility of enormous amounts of data and computer resources. As a result, DL has become

a cornerstone of modern AI systems, powering a wide range of applications across industries and revolutionizing how we solve complex problems and interact with technology. Following are the major use cases of DL.

1.6.1 Deep Learning Use Cases

DL has found extensive applications across various domains, revolutionizing industries and enabling advanced solutions to complex problems. Here are some major use cases of DL and shown in Figure 1.2 :

i. **Computer Vision (CV):**

   a. *Object Detection and Recognition:* DL models are used for detecting and recognizing objects in frames, which enables applications like self-driving cars, surveillance systems, and facial recognition.

   b. *Image Classification:* DL algorithms classify images into predefined categories, useful in medical imaging for diagnosing diseases, quality control in manufacturing, and content moderation on social media platforms.

   c. *Segmentation:* It involves the process of dividing the picture into multiple segments or sub-regions to simplify its representation. It involves identifying and delineating objects or regions of interest within an image. An example is Segmenting medical images to identify and delineate tumors for accurate diagnosis and treatment.

ii. **Natural Language Processing (NLP):**

   a. *Machine Translation:* DL models like transformers have significantly improved machine translation systems, allowing services like google translator to offer more accurate translations in a variety of languages.

   b. *Sentiment Analysis:* DL techniques are employed to assess sentiment using text data from surveys, social media, and customer reviews, which is valuable for market research, brand monitoring, and customer feedback analysis.

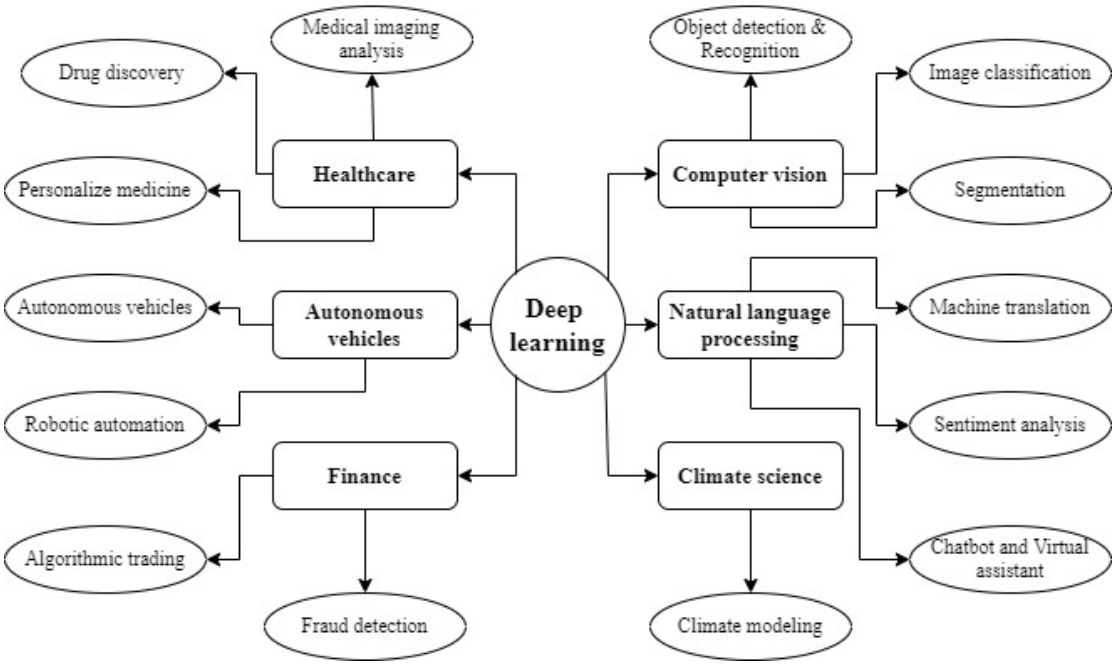


Figure 1.2: Overview of Real-World Applications of Deep Learning Across Various Domains.

c. *Chatbots and Virtual Assistants:* DL-driven chatbots and virtual assistants provide natural language interaction for customer support, information retrieval, and task automation in various industries.

iii. **Healthcare:**

a. *Medical Imaging Analysis:* DL models use medical images like X-rays, CT scans etc. to assist radiologists in the early detection of several crucial illnesses, such as cancer, pneumonia, tumors, and neurological problems.

b. *Drug Discovery:* Drug discovery can be accelerated by the application of DL approaches, which also anticipate drug-target interactions and analyze molecular structures to provide new medications and treatments.

c. *Personalized Medicine:* DL models analyze patient data, including genetic information, health records, and lifestyle to tailor treatments and interventions based on individual characteristics and risk factors.

iv. **Autonomous Systems:**

a. *Autonomous Vehicles:* DL algorithms enable self-driving cars to perceive their surroundings, navigate traffic, and make decisions in real-time, enhancing safety and efficiency on the roads.

b. *Robotic Automation:* DL-powered robots are used in manufacturing, logistics, and healthcare for tasks such as object manipulation, assembly, and surgical procedures, improving productivity and precision.

v. **Finance:**

a. *Algorithmic Trading:* DL models analyze financial data, market trends, and news sentiment to make automated trading decisions, optimizing investment strategies and minimizing risks.

b. *Fraud Detection:* DL algorithms detect fraudulent activities in banking transactions, insurance claims, and e-commerce transactions by identifying patterns and anomalies in data, reducing financial losses and improving security.

vi. **Climate Science:**

a. *Climate Modeling:* DL techniques are applied to analyze large-scale climate data, satellite imagery, and weather patterns to improve climate models, enhance weather forecasting accuracy, and study the impact of climate change on ecosystems and societies.

1.6.2 Deep Learning Mechanism

A DL model typically is composed of several layers of interconnected nodes called neurons. These layers fall into three categories.: input, hidden and output layers. Let's understand the detailed mechanism of DL with the help of Convolutional Neural Network (CNN) which is capable of taking input images for assigning importance to many objects of the image and helping to differentiate one image from another. CNN's requirement for preprocessing is less when compared to

other algorithms. These algorithms give high performance concerning image and video inputs. In CNN, there are three layers:

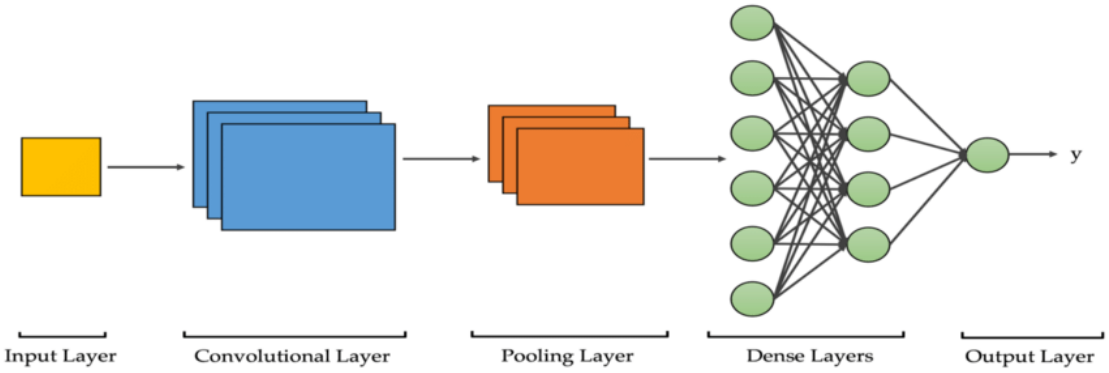


Figure 1.3: Architecture of Convolution Neural Network.

i. *Convolutional layer:* CNN is a foundational layer followed by additional CNN or pooling layers. With the increase in each layer, there is an increase in the complexity of CNN, thus identifying more image portions. The starting layers will focus on superficial features like edges and colors, whereas as image data progresses in the CNN layers, it will start recognizing larger object shapes and elements until it finally identifies the object. The convolutional layer also has a filter that will move in the receptive image field to check the feature's presence, known as convolution. Figure 1.3 shows the general architecture of CNN.

ii. *Pooling layer:* The goal of using a pooling layer is to reduce the dimensionality of input and parameters in the network termed as down-sampling. It slides the filter across the input in two ways: *(a)* when the filter selects the most prominent features (having high-value pixels) called as max pooling. It screens the complete image, prefers good quality features and sends it to the next layer while *(b)* when the filter computes and transmits the average value then average pooling occurs. This layer adds efficiency, reduces complexity and contributes in avoiding the overfitting of the network by focusing on important temporal and spatial features.

iii. *Dense layer:* Every node in the output layer is directly connected to every

other node in the earlier layer, hence it is also called a fully connected layer. It means that each node receives input from all the nodes in the previous layer and produces a single output value. As this layer uses the retrieved features and filters from earlier levels it performs a classification task efficiently.

Pros of CNN:

i. CNN automatically learns spatial anf semantic features from raw data.

ii. CNN uses parameter sharing in its convolutional layers, applying the same set of weights to various input spatial regions.

iii. Detects images with high accuracy

Cons of CNN:

i. It requires a large dataset for processing.

ii. Training of deep CNN with multiple layers and parameters is computationally intensive.

iii. Overfitting is a common problem in deep neural network topologies with plenty of parameters.

1.7 Approach

The ubiquity of surveillance systems in modern society has generated vast amounts of video data, which necessitates innovative approaches to address storage challenges without compromising on relevance. This Chapter sets the stage for exploring the intricate interplay between relevance and video compression in the context of surveillance. Leveraging the competency of DL methods, this study aims to propel the field forward by introducing a relevance-based video compression paradigm. The ensuing exploration delves into the implementation of a novel framework, seeking to identify and compress relevant frames within surveillance videos, thus optimizing storage utilization while preserving critical information.

This research not only contributes to the evolving landscape of video compression but also holds implications for the practical enhancement of surveillance systems in an era of burgeoning data and technological advancements.

In the initial phase of our study, DL-based object detection's popular techniques like YOLOv5 [29], YOLOv7 [30] and YOLOv8 [31], YOLOv9 [32] and Mask-RCNN [33] are used to predict the relevant and irrelevant frames of surveillance video using different combination of dataset. The ATM surveillance footage is divided into frames in this instance, and these frames are then sent into OD modules, which classify the frames into relevant and irrelevant categories based on the objects in the frames. For example, If frames contain a person/human then the frame is recognized as a relevant frame otherwise irrelevant. After the categorization of frames, we consider relevant frames for the next step where we propose two methods to execute the compression process. The details of implementation with result analysis and comparison are explained in Chapter 3 and Chapter 4.

1.8 Thesis Contribution

The accomplishments of the proposed research work are summed up as follows:

i. A thorough investigation of the literature review and its analysis are explored to identify relevant frames of video is explained is Chapter 2.

ii. Collection of ATM surveillance video through authenticated channels is performed and mentioned in Chapter 3.

iii. The surveillance videos are segmented into individual frames to facilitate detailed analysis. Here, each frame is annotated to label objects of interest to create a high-quality training dataset using the FFmpeg software and Make Sense AI tool.

iv. Implemented various object detection techniques which include YOLO modules and Mask R-CNN to classify the frames as either relevant or irrelevant

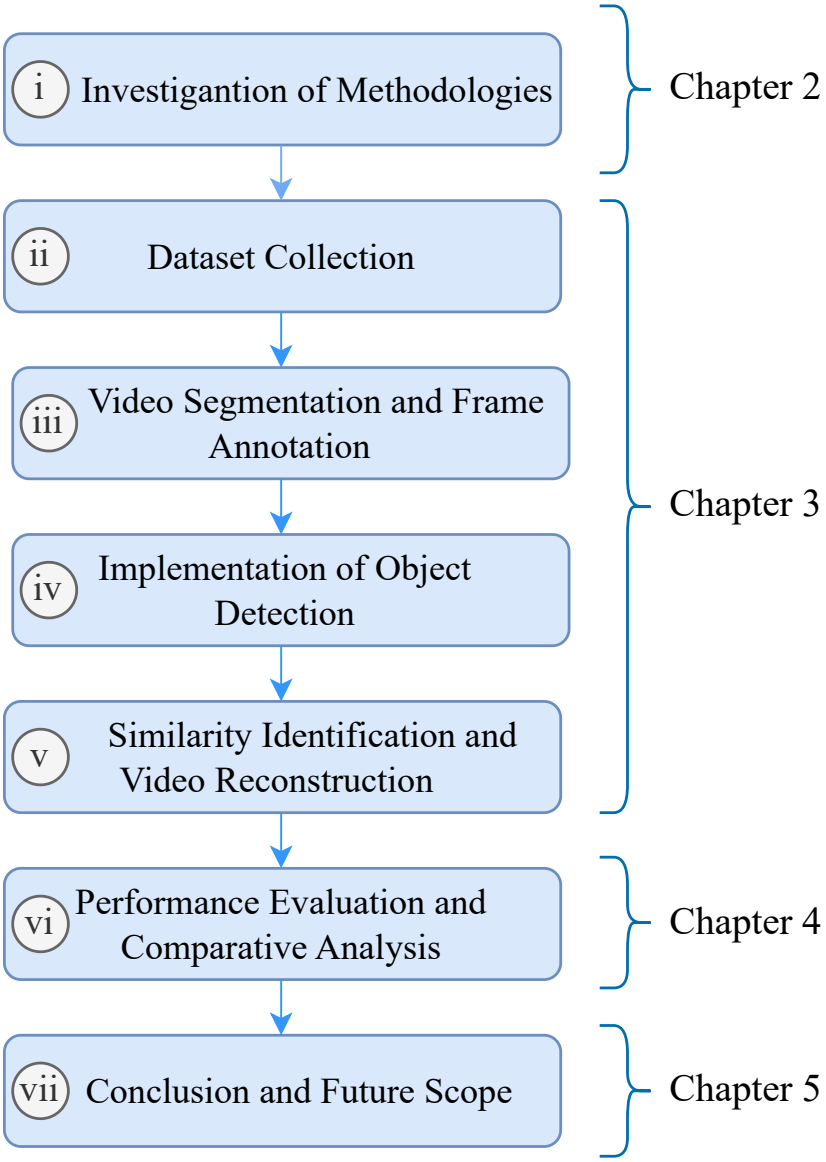


Figure 1.4: Block Diagram of Thesis Contribution.

in surveillance videos. The details about methodology, dataset and algorithm is mentioned in Chapter 3.

v. Focus exclusively on the relevant frames in the surveillance video and assess the similarity between frames at different threshold values. Further, categorizes frames into primary and similar. Lastly, construct a video using relevant primary frames where the codec fourcc is used for encoding. Then, it initializes a VideoWriter object (video_writer) from the OpenCV library to merge all frames and elaborate in Chapter 3.

vi. Compare the proposed model with existing approaches to ascertain their effectiveness.

vii. The conclusion of the thesis and future scope to extend the current research work is mentioned in Chapter 5.

1.9 Thesis Organization

The remaining part of this research work is organized into five chapters as shown in Figure 1.5.

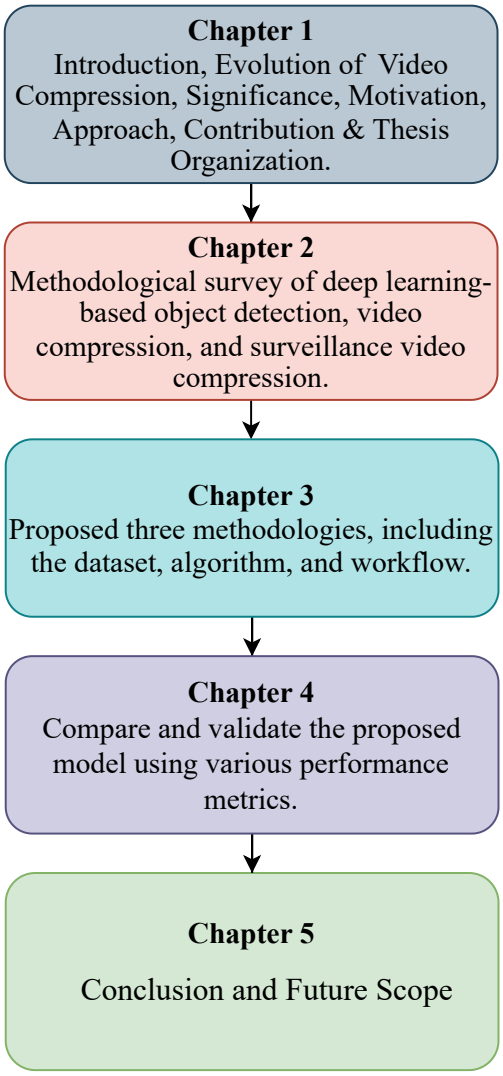


Figure 1.5: Organization of Thesis.

Chapter 1 provides an overview of compression, including its types and evolu-

tion. It highlights the importance of surveillance video compression and outlines the motivation behind the proposed research. Additionally, it discusses the development and mechanism of the deep learning framework and concludes with a research methodology.

Chapter 2 explains the background of deep learning-based object detection, video compression, and surveillance video compression techniques. It also list the findings in object detection and challenges in video compression.

Chapter 3 provides the methodology of the proposed three approaches. This chapter also explains the dataset, required to train and test the model along with workflow and algorithm.

Chapter 4 explains the result analysis of the developed three model. It also illustrates the various evaluation metrics for object detection and video compression framework. At last, the best approach is compared with existing approaches.

Finally, *Chapter 5* summarizes this research work and also discusses open perspectives.

Chapter 2

STATE-OF-THE-ART SURVEY

"Every great advancement begins as the state-of-the-art and becomes the new standard. [34]"

–Arthur C. Clarke

Deep learning (DL) based object detection (OD) and video compression frameworks are explained in this chapter. In recent decades, significant progress has been achieved in Computer Vision (CV) with DL frameworks for OD. Identifying items in an image that belong to particular target classes and labeling each one based on its exact location are the main objectives of OD. In contrast to conventional, handcrafted OD modules, DL-based approaches become capable of extracting both spatial and semantic features from images. Section 2.1 offers a comprehensive review of DL-based OD frameworks starting from 2014 which is categorized into two different methods: *(i)* the region-free method and *(ii)* the region-based method and offers some recommendations. While Section 2.2 unlocked the potential of DL techniques for Video Compression. Video compression methods becoming popular due to their significant contribution to minimizing network traffic, providing higher bandwidth, and solving storage space issues. Earlier used traditional and handcrafted video compression modules cannot cope with the demand of developing technologies. There are growing obstacles in achieving further improvement in compression using conventional methods. This section gives a deep review of various video compression approaches and their methodology proposed from year 2017. The video compression survey for the thesis is divided into two categories: *(i)* where each tool in a HEVC framework incorporates a neural network (NN) and

(ii) where the NN is applied to the entire video compression process and hence termed as end-to-end compression. With ever ever-increasing demand for video surveillance, the need for surveillance compression is also explained, and listed few challenges in surveillance video compression.

2.1 Object Detection

"In the realm of computer vision, object detection serves as the cornerstone, enabling machines to perceive and understand the visual world around them [35]."

–Sarah Chen

OD plays an essential role in CV to identify the objects in an image. DL accelerates CV development and leads to cutting-edge improvements in object classification, detection, and segmentation. In general, image classification refers to identifying the object in an image and giving a label to the identified object. OD is a technique that uses a Bounding Box (BB) to precisely locate items inside an image. On the other hand, semantic segmentation is aimed at categorizing every pixel into a predefined set of classes between various object instances, whereas instance segmentation, at the pixel level, uses segmentation masks to identify multiple object instances [36]. In other words, instance segmentation entails merging the principles of OD and semantic segmentation. To perform OD, classification, and segmentation, neural networks require a large quantity of training data and powerful computing resources. In 2014, Girshick *et al.* [37], introduced RCNN to perform OD using a Convolution Neural Network (CNN). R-CNN achieved a 30% gain in average precision as compared to previous models on the PascalVOC dataset. Therefore, in this thesis, the focus is on reviewing the key initiatives in the DL-based object recognition model from 2014. Figure 2.1 gives a graphical representation of increasing work in the DL-based OD area. Since, the 2024 paper publication count is considered only till October, it is less than that of other years.

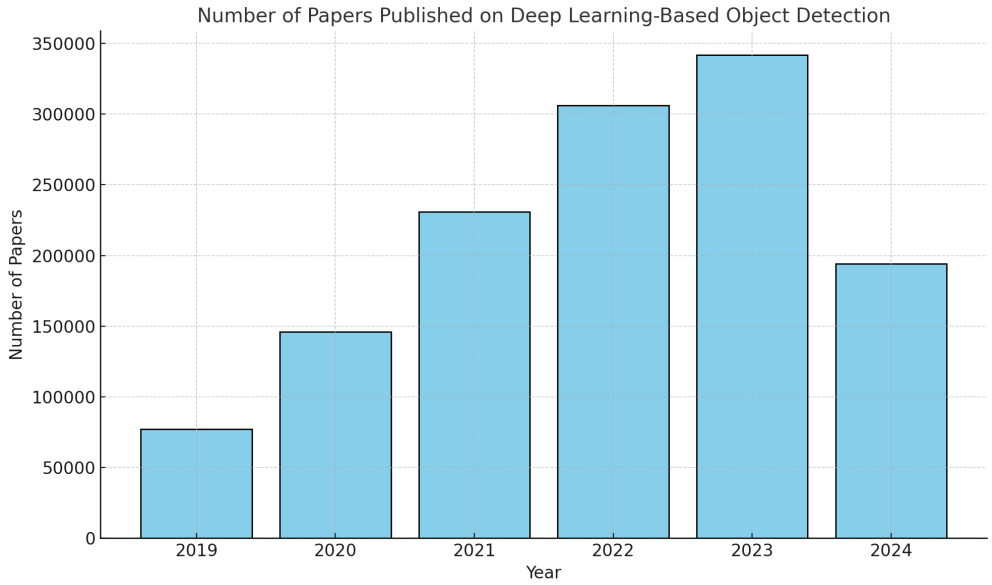


Figure 2.1: Year-wise Distribution of Published Research Papers From 2019 to 2024 [1] (Source: Google Scholar).

2.1.1 Traditional Object Detection

Before the advent of DL, feature extraction, classification, and proposal creation were the three processes in the object identification framework. The process of creating a proposal which aims to locate potential areas in the image that might contain objects is known as "Regions of Interest" (ROI) for these areas. A simple concept is to use sliding windows to examine the complete image [38]. The input image was scaled into various sizes and deployed to scroll through these images to gather data on multi-scale and various aspect ratios of objects. In the second phase, fixed-length feature vectors were generated on each ROI using the sliding window technique [39]. Low-level descriptors were frequently used to express this feature vector like the histogram of gradients [40], Haar [41] and speeded-up robust features [42], which exhibited resilience against changes in illumination, rotation, and scale. Lastly, in the third step, Support Vector Machines (SVM) and other region classifiers assign categorical labels to the covered regions [43]. Lastly, in the classification step classification methods like AdaBoost and cascaded learning were applied. The majority of the effective conventional approaches for OD centered on meticulously creating feature descriptors to acquire embedding for an ROI. For

the Pascal VOC dataset [44], outstanding results were obtained with the aid of appropriate feature representations and region classifiers. Figure 2.2 shows the popular traditional and DL approaches for OD.

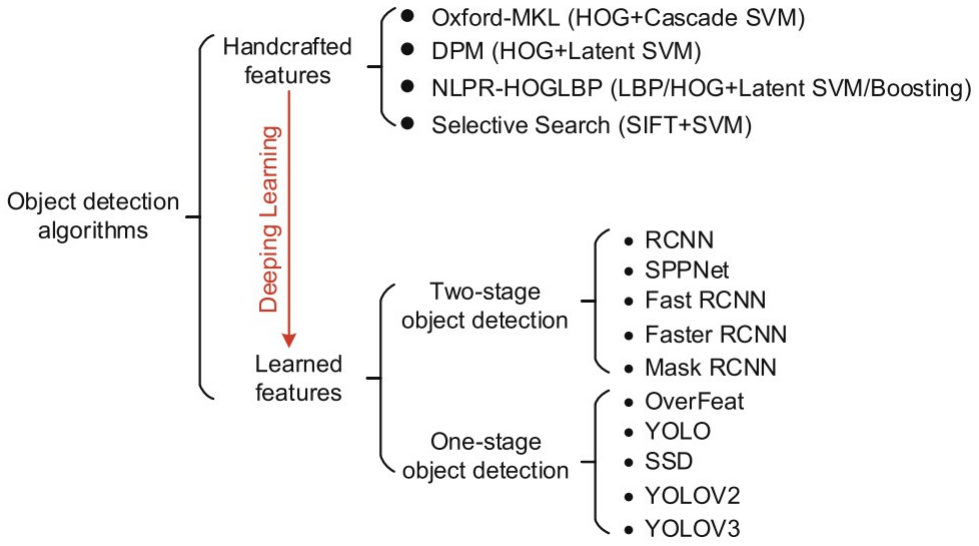


Figure 2.2: Object Detection's Popular Techniques.

Although classical OD is an established technology, it still has several inherent flaws. Initially, the method of region selection based on sliding windows is characterized by its elevated computational complexity and the presence of redundancy in the window placement. Secondly, many duplicate proposals were produced during proposal creation, which led to an enormous number of false positives during categorization. Lastly, feature descriptors were manually created based on basic visual signals, and it was challenging to accurately express semantic information in intricate situations [36]. Neural networks, which are very reliable has the potential to retrieve distinct characteristics of images. Thus, the researchers eventually focused on OD using neural networks. Neural networks convert unprocessed images into high-level contextual information via hierarchical feature representations instead of manually creating descriptors, as with conventional detectors. These representations are automatically learned from training data and exhibit a greater ability to discriminate in complex contexts. Standard visual descriptors comprise a fixed learning capacity that does not expand with increasing amounts of data.

On the other hand, an NN can leverage its high learning capacity to produce stronger feature representations with larger amounts of data. Due to these characteristics, end-to-end optimization of algorithms is possible using an NN-based OD framework and has stronger feature representation capabilities. This thesis will act as a timely assessment for researchers and industry professionals to further promote research on detection in surveillance systems.

Currently, deep learning-based OD is categorized into two techniques: i) Region-Based Techniques ii) Region-Free Techniques. Due to the separation of the detection problem into two stages, the region-based approach is also known as a two-stage detector. This process consists of two steps *(a)* Proposal Generation and *(b)* Feature Extraction and Object Classification. At the same time, the region-free approach is called a one-stage detector where entire images are considered and classify every region as a foreground or background object. Details of these approaches and popular algorithms are summarized as follows.

2.1.2 Region-Based Approach

Girshick *et al.* [37], present OD for the first time using a neural network approach named RCNN which achieves rapid gain in Mean Average Precision (mAP) by 30%. This approach consists of three modules: - i) Using selective search technique [38], RCNN develops approximately 2000 proposed regions, ii) Feature extraction through CNN and iii) SVM classifier. Here, the five CNN and two dense layers are fed a fixed size, 227 $\times$ 227 RGB image as input, and SVM is used to predict the object inside the image. Though this approach gained remarkable success but suffers from two major drawbacks: - expensive training and slow detection. Spatial Pyramid Pooling (SPP) is used to solve the problem of fixed-size input data in RCNN and does away with the idea of image wrapping and cropping. Hence, He *et al.* [45], developed SPP which generates firmed dimension output regardless of the amount of the input dimension. Because of this versatility, SPP integrates extracted features at different scales. Like R-CNN, SPP also has multi-stage pipeline training which slows down the process. To address the drawback of

R-CNN and SPP, Girshick *et al.* [46], designed Fast R-CNN which enhances the speed in terms of training and testing. In Fast R-CNN a set of object proposals or entire image is taken as input for CNN and max pooling layers, these networks initially developed a feature map for the complete image. With the help of the mapped feature RoI is generated on the other hand pooling layer develops a firmed feature vector for every object proposition. A series of dense layers that eventually split into two sister output layers receive each feature vector. Though fast R-CNN is 143 $\times$ faster than R-CNN there is only a slight change in accuracy. Hence, Ren *et al.* [47], introduced Faster R-CNN to enhance the accuracy of NN which has two modules: - i) Generation of region proposal network using anchor box method instead of selective search ii) Used RoI pooling to obtain features map and perform classification as well as bounding box regression. Lin *et al.* [48], developed a Feature Pyramid Network (FPN) to extricate spatial and semantic characteristics from images. Generally, CNN typically retrieves features from images using an up-down approach where semantic features are gathered from the images top layers, while spatial features are obtained from the images bottom layers. Consequently, FPN integrates low-quality features with high-quality features using both top-down as well as bottom-up methods.

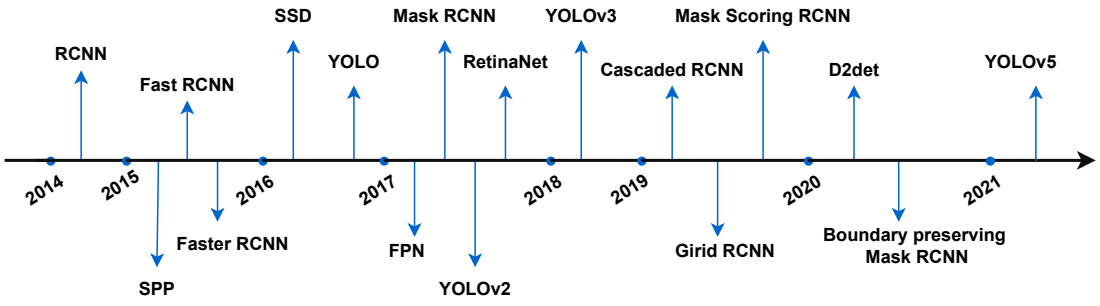


Figure 2.3: This Figure Presents a Timeline for the Evolution of Object Detection Approaches Using the Neural Network From 2014.

He *et al.* [33], introduced the concept of instance segmentation using a neural network in 2017 named Mask R-CNN which uses ResNet50 and ResNet101 architecture with FPN to perform classification, detection, and segmentation simultaneously. Mask R-CNN is intuitive and follows the path of Faster R-CNN.

Without any bells and whistles it surpasses existing models on the COCO dataset. The pixel-to-pixel alignment concept is included in Mask R-CNN to address the issue of ROI misalignment. We used Mask R-CNN to determine the relevant and irrelevant interval of video footage and compare its performance with YOLOv4 [49]. Chen *et al.* [50], developed MaskLab, which does semantic segmentation and direction prediction, to achieve foreground/background classification across every predicted box. This method used ResNet101 architecture to produce three different outputs i.e. detection, segmentation, and prediction.

The Intersection over Union (IoU) threshold in OD is widely used to differentiate among positive and negative results. Though larger thresholds typically impair detection performance, the traditional threshold of 0.5 frequently results in noisy, low-quality detections. To address this problem, Cai *et al.* [51], developed a multistage OD cascaded R-CNN that is comprised of several trained detectors with progressively higher IoU thresholds. The output of one detector serves as the training set for the ones that follow as they are trained one after the other. Figure 2.3 summarizes the significant turning points in Region-Based Techniques. Cao *et al.* [52], present D2Net which is a two-stage detection technique that handles both accurate categorization and precise localization of the object. For accurate localization, D2Net uses dense local regression to locate an object by predicting a number of dense box offsets. For categorization introduced RoI pooling technique which selects from several sub-regions of a proposal and applies adaptive weighting to produce discriminative features. Lu *et al.* [53], developed Grid-RCNN to effectively maintain spatial information by replacing traditional regression networks with fully convolutional networks. Huang *et al.* [54], add MaskIoU in Mask R-CNN, which is often ignored in most instance segmentation systems. Currently, instance segmentation techniques use fully convolutional networks to classify pixels instead of object boundaries and shapes, which results in coarse and fuzzily defined mask prediction outputs and inaccurate localization. To address this issue, Cheng *et al.* [55], introduced Boundary-Preserving Mask R-CNN (BMask R-CNN) to increase mask localization precision by making use

of object boundary information. Figure 2.4 gives an idea about the taxonomy of OD models where researchers split contributions of DL into two categories: Region-Based approach and Region-Free approach and listed popular benchmark. Region-Based OD addresses the issue of memory management by optimizing GPU memory, adjusting batch sizes, and employing memory-efficient layers. Efficient data loader configurations, quantization, and dynamic shape inference help reduce memory requirements. Gradient clipping and on-device inference optimizations like model pruning enhance memory management. Regular monitoring and profiling are essential for identifying and addressing potential memory issues in the complex architecture of region-based models.

Object Detection				
Region Based		Region Free Approach		Popular Dataset
Method	Architecture	Method	Architecture	
RCNN	VGG-16/ AlexNet	YOLO	CNN	CIFAR 10/100
SPP	ZF-5	SSD	VGG-16	COCO
Fast RCNN	VGG-16	YOLOv2	DarkNet-16	Pascal VOC
Faster RCNN	ZFNet VGG-16	YOLOv3	DarkNet-53	ILSVRC
Mask RCNN	ResNetxt	YOLOv4	CSP DarkNet-53	Open Image

Figure 2.4: Taxonomy of Object Detection using Neural Network.

2.1.3 Region-Free Approach

The goal of Region-Free OD approaches is to identify objects from images without using region proposals or predefined anchor boxes. These methods predict the existence and placement of objects over the entire image, instead of segmenting the image into sections and estimating bounding boxes for every part. In 2016, the first successful DL-based detector that has the ability to detect an object from a picture instantaneously was presented by Redmon *et al.* [56], and named as You

Only Look Once (YOLO) which outperforms RCNN. YOLO partitions the whole area into a predetermined set of 7×7 grid cells. The object is located by searching every cell; if the object's center is in a grid cell, that cell is in the position of locating the object. BB and confidence ratings will be projected for each cell. YOLO may anticipate 45 Frames Per Second (FPS) and achieve up to 155 FPS with a more streamlined backbone. However, there were some difficulties with YOLO: *(i)* it detects only two objects and is difficult to find small objects and *(ii)* the most recent feature map was used for prediction, but it was insufficient to predict objects with varying sizes and aspect ratios. Liu *et al.* [57], present a Single-Shot multi-box Detector (SSD) to overcome YOLO challenges. To assess the BB output matrix, SSD divides images into a grid cell. A set of anchors is generated for each grid cell, varying in size and dimension. Compared to the current real-time YOLO, the SSD300 model operates at 59 FPS, producing noticeably better detection accuracy. It is also observed that during training of the detector class imbalance occurs for foreground and background in the central cause, Lin*et al.* [58], address this imbalance by developing RetinaNet. In order to address class inequality, RetinaNet modifies the standard cross-entropy loss such that samples with accurate classifications receive a down-weighted loss.

After YOLO, Redmon *et al.* [59], present an enhanced version of YOLO named YOLOv2 or YOLO9000, which greatly increased detection performance. This approach applied a far more effective deep convolutional backbone framework that pre-trained on higher quality pictures using ImageNet and as a result, the learned weights were more adept at capturing fine-grained data. In the end, YOLOv2 incorporates multi-scale training techniques and batch normalization (BN) to yield detection outcomes that were cutting-edge at the time. In 2018, Redmon *et al.* [60], present YOLOv3 which integrates residual block, FPN, and binary cross entropy loss. YOLOv3 performs extraction using DarkNet-53, a 53-layer CNN architecture that was developed using the Imagenet dataset. YOLOv3 uses a k-means clustering technique to calculate the initial width and height of the predicted BB. This method is time-consuming to analyze large-scale datasets because the

predicted width and height depend on the original cluster centers. To estimate the dimensions of the BB, Zhao et al. YOLOv3 [61], suggested the AFK-MC2 technique, which outperforms the original.

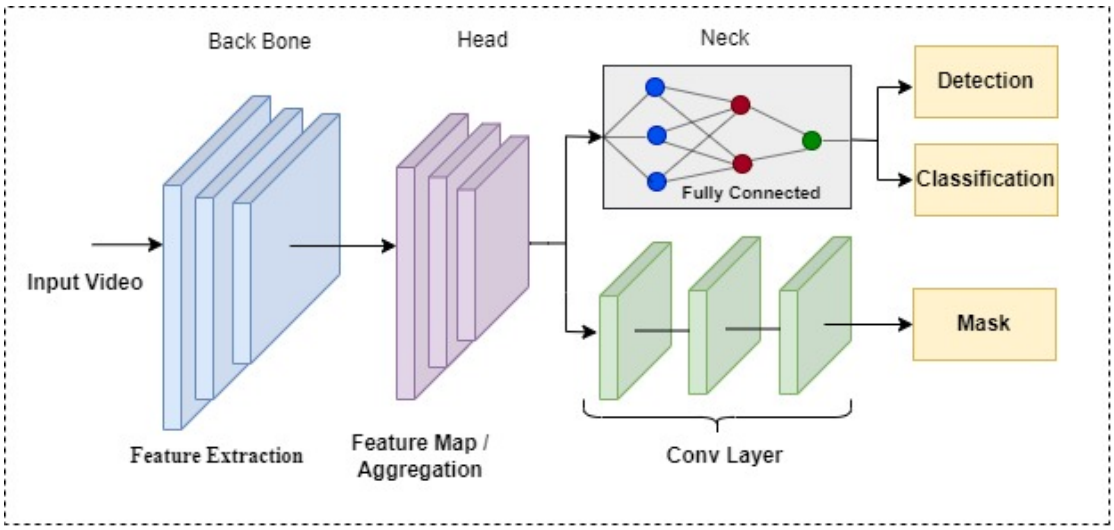


Figure 2.5: General Architecture of YOLO Modules.

Later, Redmon quit to perform further enhancement in YOLO due to security reasons, Bochkovskiy *et al.*, took the opportunity and introduced YOLOv4 [62]. Figure 2.5 indicates the general architectures of YOLO layers which is divided into three steps:- backbone, head, and neck. YOLOv4 adopts the use of Cross-Stage Partial Connections (CSP) Darknet-53 as its backend. In YOLOv4, the neck section incorporates SPP, while the head comprises various YOLO layers. The performance of YOLOv4 was substantially enhanced with the use of detector training strategies known as Bag-of-Specials (BoS) and Bag-of-Freebies (BoF). In this configuration, through CSP the input data is split into two groups: the first group is processed by CNN (DenseNet), while the second group bypasses CNN and is used as input for CSPDarkNet and lastly the output of both CNN and CSPDarknet is integrated to pass it to further layers. This technique resolved the vanishing gradient issue by using DenseNet architecture, which greatly enhances CNN learning capabilities.

Ultralytics developed YOLOv5 on PyTorch framework and presented four different models YOLOV5s, YOLOv5m, YOLOv5l and YOLOv5x, each with differ-

ent layers and parameters [63]. Like YOLOv4, YOLOv5 uses CSP-darknet-53 as a backbone, PaNet for feature integration, and YOLO layers to perform classification and detection at different scales. Bochkovskiy *et al.* [30], developed three different models YOLOv7-tiny, YOLOv7, and YOLOv7-W6 in 2022, for various types of GPUs respectively. Using trainable BoF methods, without sacrificing inference cost YOLOv7 optimizes the training process rather than focusing on architecture optimization. Yolov7 addresses the issue of model reparameterization and proposes a new technique for dynamic label assignment named "coarse-to-fine lead guided" assignment.

2.1.4 Findings

i. *Region-based vs Region-free approach:* DL-based OD frameworks may now be broadly categorized into two families: i)Region-based CNN (R-CNN) and its variations, and (ii) Region-free CNN, such as YOLO and its variants. Using a proposal generator, Region-Based detectors produce a sparse set of proposals, which are subsequently classified using region classifiers. On the other hand, Region-Free detectors forecast how each object will be categorized in the feature maps. Region-free detectors prioritize time efficiency and are better suited for real-time object identification, whereas region-based detectors typically deliver superior detection performance.

ii. *Gap between efficiency and accuracy:* While assessing a semantic segmentation approach, accuracy and effectiveness are both essential. Unfortunately, for all of the available semantic segmentation techniques, the improvements in these two areas still contradict one another. In real the model with high accuracy exhibits low efficiency and vice versa.

iii. *Lack of dataset:* Now, the most widely utilized detection benchmark at the moment is MSCOCO. On the other hand, MSCOCO contains just 80 categories, which is still insufficient to comprehend more intricate real-world scenarios.

iv. *Dependency on data:* To achieve a good result in semantic segmentation, high-quality training data is required. However, gathering high-quality training data continues to be a laborious and time-consuming process due to the lack of enough labeled images with pixel-level annotation. This significant dependency has emerged as yet another issue with semantic segmentation.

2.2 Video Compression

In the past years, DL techniques have shown impressive results as they reduce the need for handcrafted modules. Consequently, DL is viewed as beneficial for analyzing unorganized data like video and audio data, which is still an unsolved challenge in AI. In the field of image/video compression, artificial NN has a long history. In the 1980s and 1990s, several studies on NN-based compression were undertaken, however at the time the networks were small and the compression performance was poor. It is now feasible to train extremely deep networks with over 1000 layers because of the availability of a huge amount of data, along with powerful computing platforms and innovative algorithms. It is thus worthwhile to reconsider the integration of DL to video compression. and it's an active research topic from 2015. At this time, research has yielded promising findings, demonstrating the viability of AI-based video coding [64].

Traditional video compression strategy adopts the concept of a hybrid video coding framework, which integrates both predictive and transform coding. As explained in Figure 2.6 input video data is split into frames, frames are divided into blocks, blocks into units, and the largest unit is termed as Control Tree Unit (CTU). CTU is also divided into the control unit, control unit divided into prediction unit, and at last prediction unit into transform unit. These frames/blocks/units are compressed in a predefined manner termed intra-frame prediction, while earlier compressed frames are used to compress the next frame or to predict the next frame which is termed inter-frame prediction, respectively. Then predicted data are transformed, quantized, and entropy-coded to gain final coded bits. As predicted data is quantized, there might be a chance to loose some

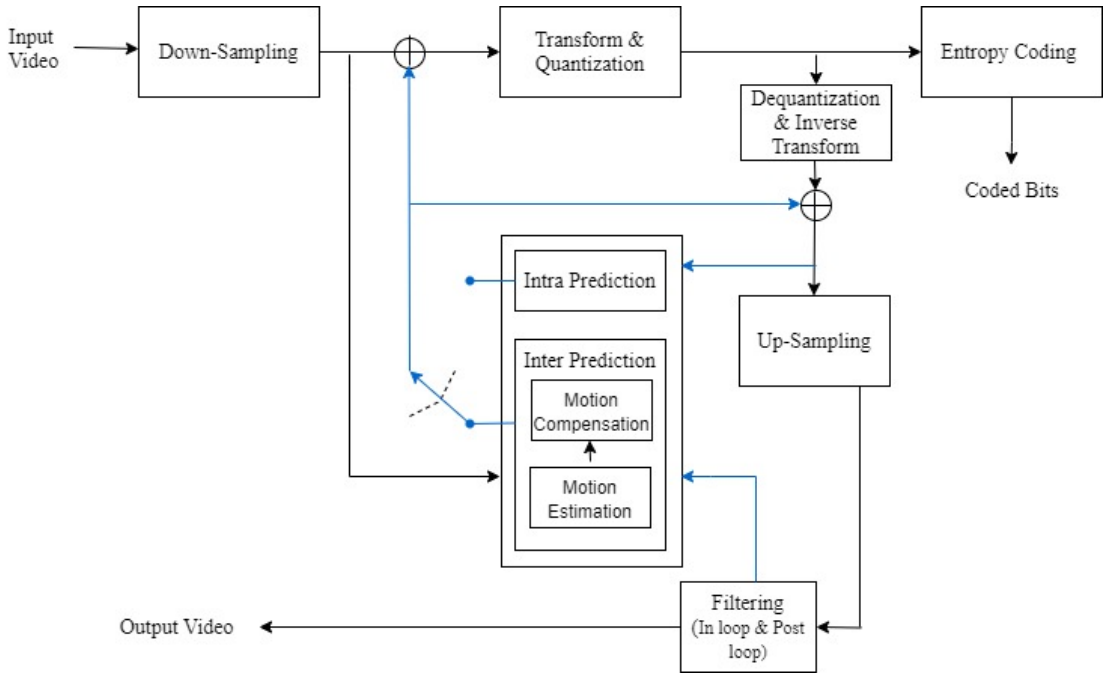


Figure 2.6: Hybrid Video Coding in H.265.
[1](blue lines indicate predicted data)

amount of data and may cause noise also. To avoid such type of loss, two filters were suggested: *(i)* In-Loop Filter and *(ii)* Out-Loop Filter. With these, for the reduction of video size, the frames undergo downsampling before compression and upsampling thereafter. Various researchers have tried to apply the concept of neural network to any tool in the HEVC framework which is explained in brief in the following subsection.

2.2.1 Intra-Frame Compression

The intra-prediction technique is used for predicting the relationships between blocks in a single image. Many researchers are working to improve video compression performance by incorporating DL techniques into HEVC frameworks. H.265 includes a 35 intra-prediction mode, but it ignores the considerably deeper features among the present and its nearer blocks, resulting in erroneous prediction when they have poor spatial connection. To overcome this problem, Cui *et al.* [65], integrate the CNN with H.265, where 16×16 new block is predicted using the three nearest reconstruction blocks called residue. Next, by deducting the

residue block from the original block, the target block will be generated. Lastly, this target block replaces the original in H.265. Zhang *et al.* [66], used a dense network to achieve end-to-end mapping with the help of nearer-developed pixels to the present block which enhanced the quality of prediction. Hu *et al.* [67], uses a Generative Adversarial Network (GAN) to reduce spatial redundancy. Liu *et al.* [68], deploy a CNN-guided spatial RNN with intra-prediction in HEVC to enhance the performance, the network which generates a prediction signal simply by looking at the correlations between pixels. This approach automatically solves the problem of asymmetry since no information from unknown locations is used before the prediction process and the performance of the network could be better than HEVC.

Liu *et al.* [69], designed progressive spatial recurrent NN which is made up of three spatial recurrent units, that create predictions in stages by transmitting data from earlier contents to the encoded block. This method suggested a novel loss function to measure the rate-distortion of encoded blocks and train the model which allows the network to give predictions that incorporate both distortion and bit rate. This method makes dynamic block size intra-prediction feasible, which is more beneficial in real coding scenarios. Existing intra-prediction algorithms can successfully predict many blocks, but still require some additional exploration of certain blocks with complicated textures. For this, Wang *et al.* [70], present a multi-scale CNN on video encoding. The angular prediction generates an expected block, which is then input into the network together with its adjacent L-shaped reconstruction block. However, multi-function extraction uses feature maps of multiple sizes to enhance prediction. The L-shape structure can be utilized with the multi-scale structure to tune the pixels in the predicted block.

Sun *et al.* [71], boost the performance of intra-prediction using multiple NN. Here, the authors present two strategies to merge NN with the traditional mode in HEVC. For the first scheme *i.e.* appended scheme Neural Network Mode (NM) is considered an additional mode to Traditional Mode (TM) while for the substitution scheme, TM is altered as NM. If several NM are added, some TM are

grouped according to a predictive error. TM selection based on probability for the substitution scheme: the authors recommend two perspectives for replacing TM *(i)* substitute the most likely TM, *(ii)* substitute the less likely TM.

Blanch *et al.* [72], worked on a new training approach which construct a model that surpasses advanced attention-based architectures and decreases the complexity of inferences. The suggested techniques included in the VVC to retained the efficiency of cutting-edge chroma intra-prediction methods based on NN. While Amna *et al.* [73], used only the luma component to extract CTU units from three deep CNN models. To minimize the changes in the CTU sample, the prepossessing model and down-sampling process is applied to achieve a good result for CU. This structure is integrated with HEVC and Joint Exploration Model (JEM) to optimize quantization parameters. For lossless coding applications Schiopu *et al.* [74], presents a CNN approach where 33 angular modes in HEVC are replaced by CNN-based prediction method and achieve bitrate reduction by 5.8% . Statistical features determine which partition to use so Yao *et al.* [75], proposed a partitioned network based on rate-distortion. This network is composed of *(i)* Partition network - for predicting CU partition and *(ii)* Target network - for improving network parameters by computing rate distortion. Zhang *et al.* [76], designed deep region segmentation that depends on intra-prediction to boost the performance for depth video compression. In order to manage the block restoration, the proposed network first extracts the segmentation result from the source frame and divides the frame division into block divisions.

2.2.2 Inter-Frame Compression

Inter-frame prediction estimates the Motion Compensation (MC) and Motion Estimation (ME) between two consecutive video frames to eliminate duplication towards the temporal axis and also determines video compression efficiency. To predict the block, MC gathers the content at the position where ME determines the point in the reference frame which is most equivalent. For boosting the efficiency of block level ME and MC researchers try to integrate the neural network

with inter-prediction.

To produce a more correctly interpolated frame with the help of motion vector Choi *et al.* [77], used triple frame-based bi-directional motion estimation with reliability. In the proposed approach motion vector refinement method is used, here authors first identify and then correct artefacts on interpolated frames using CNN. Zhang *et al.* [78], developed two Frame Rate Up-Conversion (FRUC) methods to enhance the resulting frame quality since there was no interpolation in the current frame. FRUCs residue network takes an input from the previous FRUC method and minimizes the blurring and artefacts of the violent frame, then deep residue network with weighted convolutional MC, merged Motion Compensation Interpolation (MCI) with existing MCI-FRUC and at last the output of interpolation fed to CNN to improve the efficiency. To treat extrapolation and interpolation equally, Mao *et al.* [79], created spatial-temporal CNN. This CNN uses spatial nearby pixels and temporal display orders as extra inputs. Also developed a bi-predictor using a naïve CNN structure, to enhance the accuracy of the predictor and the high co-relation between spatial current predicted block and spatial neighbouring pixels of the current unit helps to minimize artefact and prediction residue around the frames block. Meng *et al.* [80], focused on finding a reliable technique for improving the quality of HEVC compressed videos, a multi-frame directed attention network with temporal-spatial priors improves the quality with fast inference time due to its lightweight design structure.

Zhang *et al.* [81], proposed compression priors assisted CNN in enhancing performance using two additional compression priors *(i)* for the matching residual component and *(ii)* for both the rebuilt component and a high-quality collocated component. Rather than extending the reference block, the model immediately generates some more prediction options as fractional samples with identical dimensions as the present block. According to the authors, the first time compression prior concept invoked into the compression. Yu *et al.* [82], present a framework for performing fractional interpolation where multi-task training provides additional information on distortion characteristics for future interpolation performance. To

achieve fractional interpolation, a sub-network for interpolation is built by fusing numerous local features using the feature fusion module and gathering multi-scale information on compression artifacts using the distortion awareness module. Zhang *et al.* [83], focused on fractional interpolation for generation of the frame using the reference blocks integer location samples and create fractional samples closer to the present coding block. Independently, reference block prediction and residual sections are used as input to enhance CNN nonlinear learning ability. Authors integrate dual-input CNN-based interpolation technique with the HEVC framework.

Li *et al.* [84], designed a new bi-directional MC architecture to extract and interpolate information from the reference frame. The method builds a dense motion field that records complex behaviour within reference frames without requiring extra data by using optical flow data. To address the issue of deviation in interpolation, transmission offset motion vectors are added to the anticipated optical flow. The suggested speed-up method maintains most of the performance advantages while reducing complexity. Wang *et al.* [85], developed a novel segmentation-based MC, where MC zone predicts segmentation information: edge, foreground, and background zone. A CNN for creating target frames from compressed frames was proposed by Liu *et al.* [86]. The proposed approach aims to increase the network's capacity to extract more data from good-quality frames and more thoroughly investigate spatial connections, ultimately leading to an improvement in interpolation accuracy. To enhance multi-view video coding speed Lei *et al.* [87], proposed deep virtual reference frame inception where a proposed network is used to produce a better quality reference frame.

To increase the efficiency of the inter-prediction framework Wang *et al.* [88], presents three different networks where the first residue estimation network extracts residue information from the current and predicted block using spatial information. The second network integrates the features of the determined and residue block. The last redefined network takes this integrated feature as input and produces a redefined block which includes the determined block to generate

the predicted block. Authors integrate this framework with HEVC to identify its accuracy. Lee *et al.* [89], designed an inter-layer kernel prediction compression model by analyzing the weight of different CNN, which gives better accuracy than ResNet-110. Also proposed approach was combined with the quantization method to generate inter-layer kernel prediction-Quantization to give 13× compression ratio but then performance was degraded in terms of accuracy as compared to ResNet architectures.

2.2.3 Transform, Quantization and Entropy Coding

Transform, Quantization and Entropy Coding play a vital role in achieving final coded bits for H.265 [22]. Originally, video compression employed the discrete cosine transform technique, which is now replaced in H.264 and H.265 by the integer cosine transform. To enhance the quality of coded bits Zhaou *et al.* [90], introduced NN based reinforcement model for video clips. Initially, the Markov decision process problem is used to solve the rate control problem and then the reinforcement model identifies quantization parameters to train the NN. Wang *et al.* [91], proposed a Soft-Decision Quantization (SDQ) approach which surpasses hard-decision quantization using the inter-coefficient correction. By simulating SDQ completely, a coefficient-adaptive offset model was built using a DL method to modify the output of HDQ. Li *et al.* [92], present the integration of a trellis-coded quantizer into a system for image compression based on deep learning. To allow for back-propagation during training, a soft-to-hard approach was used in which they constructed a basic image compression model composed of three sub-networks (encoder, decoder, and entropy estimator) and optimized each component end-to-end.

Li *et al.* [93], used CNN for efficient and effective entropy modeling. To choose appropriate coding contexts for simultaneous entropy decoding, the authors employ a 3D code-splitting technique and a 3D zigzag scanning order. As a result, the discretized mixture of Gaussian distributions for each code is represented by the latter utilizing three context-based convolutional networks. Flamich *et al.*

[94], proposed a new technique, relative entropy coding to encode the latent representation of a single image with a code length that is near to the entropy of the image. For the first time, relative entropy coding beats previous bits-back algorithms on the Kodak data set, allowing it to be immediately used for lossy compression. Ladune *et al.* [95], suggested a strategy to improve learned image compression by using a more accurate probability model for the latent variables. The author used three binary values and one integer to indicate the latent, with various probability models inspired by binary arithmetic coding.

In HEVC, spatially predicted residues are converted to frequency domain coefficients via a discrete cosine or sine transform. Each coefficient was encoded using quantization and entropy coding in the bit stream but still, there remain linear and non-linear correlations between various coefficients after transform. Ma *et al.* [96], developed an approach for intra-predicted residues in the HEVC standard using CNN to minimize redundancy between coefficients. By combining two novel ideas: channel training and hidden residue prediction Minen and Singh [97] can reduce serial processing and produce network designs with better rate-distortion accuracy than earlier models.

2.2.4 Post-Loop and In-Loop Filtering

The quantization process increases the need for filtering in hybrid video compression, as the quality of reconstructed videos differs from that of the original. Since predicted data is taken as input to quantization, loss in the quantization process is considerable due to which artifacts like blurring, blocking, color shift, etc. are visible in the reconstructed video. Hence, filtering is essential for boosting the recreated video's quality. There are two types of filters in video compression: *(i)* In-Loop filter which creates the next frame using the predicted filtered frame as a guide and *(ii)* Out-Loop filter before output video creation. The de-block filter and adaptive offset were two new filters introduced by the H.265 codec.

Huang *et al.* [98], developed a new filter for minimizing the bit rate by more than 40% while retaining equivalent objective visual quality for VVC using CNN.

Zheng *et al.* [99], designed dual CNN to minimize artifacts of color images. To design a color-to-color network model, the proposed architecture directly predicts the discrete cosine losses without using a pair of corrector-extractors. Huang *et al.* [100], designed a frame-by-frame filter approach using deep NN to post-filter the intra-coded video. The network is trained using a novel frame-size patching approach. Here, luma and chroma both channels are filtered separately. For segmentation purposes, the authors proposed a patch generation paradigm using H.265 which is an alternative approach, for filtering in the conventional codec. As per Lu *et al.* [101], use of non-local Kalman NN, improves the quality of compressed frames, and it also helps in minimizing artifacts in the frames. This approach employs multiple deep NN to evaluate the Kalman filter's associated states and then integrates them into a deep Kalman filtering network.

Inspired from DL success, Pan *et al.* [102], present an In-Loop filter that depends on extended deep CNN which enhances in-loop filter performance in HEVC. The proposed architecture quickly removes artifacts based on statistical analysis. It employs three solutions including the weight normalization technique. While Liu *et al.* [64], presents a comprehensive evaluation technique for reducing compression artifacts, which covers both conventional and deep-learning-based algorithms. Galteri *et al.* [103], introduced a CNN residue network model for picture modification. By using a GAN the authors generate images with a higher degree of photo realism than structural similarity index measure networks. Lin *et al.* [104], developed a partition-aware CNN. While current CNN-based algorithms merely employ the decoded frame as input, the proposed methodology combines the CU size with the distorted decoded frame to effectively reduce HEVC artifacts. Xia *et al.* [105], presents an asymmetric Convolution residue network for an in-loop filter where directional information, is recovered from an asymmetric convolution block to restore the structure. The hierarchical properties of compressed frames are closely reflected by cascading residual blocks. Additionally, to reinforce the network for larger Quantisation parameters, the author used pruned dense connections. Kim *et al.* [106], introduced Spatial CNN, for the elimination

of JPEG image compression artifacts which uses down-sampling operations to estimate residual frequency from Spatial input. For each 8×8 pixel in the JPEG, the DCT domain is grouped and spatially invariant, it is feasible to downsample the input by a factor of eight to decrease the computational cost. Huang *et al.* [107], designed a multi-gradient CNN-based in-loop filter for VVC. The suggested approach makes use of potential picture structural information, such as contour information, to restore more detailed information and therefore further enhance frame quality, while Yang *et al.* [108], discovered three performance bottlenecks in the conventional approach: an amortization bottleneck, a discretization bottleneck, and a marginalization bottleneck.

2.2.5 Down and Up-Sampling

Recently, increased resolution in several dimensions such as spatial, temporal, and pixel value resolution is a common technique. As a result, data volume also increases, which creates an issue for video transmission systems. So, there is a need to reduce resolution before encoding and then enhance it again after decoding. This approach is called down / upsampling of video. Earlier, down/upsampling was implemented using a handcrafted module only. Now neural network approaches were used to enhance the down/up-sampling of video for efficient video coding.

To extent the rate-distortion of video, Bourtsoulatze *et al.* [109], presents deep NN for video distribution. In this article, an adaptive pre-coding mode selection approach that adaptively finds the optimal resolution before encoding is used. Here, high-resolution frames are downscaled over many scale factors by a multi-scale pre-coding CNN, which is trained to minimize blurring artifacts and post-processing resulting from conventional linear up-scaling filters. Lin *et al.* [110], designed a down-sampled approach to enhance bitrate in HEVC. Key and non-key frames are encoded at various resolutions. The non-key frames are upscaled by the decoder using DL. An adaptive patching-based method is used to interconnect the high-quality blocks with the low-quality non-key frame blocks.

While Yu *et al.* [111], introduced a new technique based on DL in the video reconstruction phase of scalable bitstreams. in this approach,a super-resolution triggered RNN is used to retrieve and merge information from the previous high-resolution frames as well as the current low-resolution frame. By incorporating accessible features smoothly, substantial increases in PSNR, SSIM, and VMAF were observed. Zhang *et al.* [112], present CNN-based effective bit depth adaptation for compression. It performs effective down-sampling at the decoder, it reconstructs the original bit depth using CNN up-sampling algorithm. Moreover, Feng *et al.* [113], present dual network architecture, one for upsampling and other for compression artifact reduction both are tuned step by step. For intra-frame block sampling introduced novel CNN structure [114]. The author introduced a five-layer- sample-based CNN, which contains multi-scale fusion, de-convolution of feature maps, and residue learning, to obtain a greater reconstruction quality and simpler network structure. In addition, the author used separate networks for sampling components of luma and chroma and CNN utilises luminous information to enhance its performance.

2.2.6 Encoder Optimization

The mentioned tools aim to enhance compression efficiency in terms of bitrate reduction. Some other tools also focus on various objectives. This subsection examines several tools designed for three distinct goals: ROI, rate control, and fast encoding. As these tools are exclusively utilized at the encoder side, they are collectively termed encoder optimizer tools and its popular methodology are explained as follows:

i *Fast Encoding:* In this subsection, we reviewed some novel strategies of video compression using a neural network which is applicable at the encoder side only. In HEVC, encoding is much more complicated as compared to that of decoding as 35 different modes were used in HEVC. The encoder must make a comparison in order to choose the mode for every block while the decoder has a simple choice to evaluate the given mode. Hence encoders prefer to perform

exhaustive searches, where compression efficiency is much higher but with that computational complexity also increases. Practically, the encoder should prefer a heuristic approach for the selection of mode where DL methods guide.

Liu *et al.* [115] used trained CNN to aid in predicting CU partition mode for the development of H.265 intra-encoder. Here, a quadtree structure is produced by recursively breaking CTU into CU. Taking into account the material within the CU and the designated QP, their CNN decides whether to split a CU into 32×32/16×16/8×8 or not. This is essentially an issue of binary decisions. Song et al. investigate a CNN-based technique for fast mode selection in the HEVC intra-encoder [116]. A CNN is trained to produce a list of the most likely modes for every 8×8 or 4×4 block, taking into account the given QP and content. From this list, a mode is then chosen using the conventional rate-distortion optimized procedure. Xu *et al.* [117] analyzed mode choice for traditional codec. They use a hierarchical LSTM network to forecast the CU partition mode based on information extracted from H.264 encoded bits.

Jin *et al.* [118], additionally consider the CU mode decision, for the upcoming VVC compared to, as a quadruple-tree-binary-tree structure is proposed for CU partition in VVC. On a 32×32 CU, a CNN is trained to do a 5-way classification, with different classes indicating different tree levels. For researchers to identify the CU partition mode for I frames and P/B frames of the H.265 inter-encoder, Xu *et al.* [119], presented an early-executed step-by-step Long Short-Term Memory (LSTM) and CNN.

ii *Rate Control:* A encoder attempts to generate bits that do not exceed a given transmission capacity which is a prerequisite for rate control. Hu *et al.* [120], adjust intra-frame rate control using reinforcement learning techniques. Researchers establish an association across the rate control and reinforcement learning problems, asserting that bit balance and block texture complexity represent the state of the environment. Li *et al.* [121], offer training of CNN to detect the features for all CTU. The suggested strategy achieves lower rate control error and improved compression efficiency, according to experimental

data.

iii *RoI Coding:* During compression, the ROI portion of the image must have high-quality content is the primary need of this approach. The main query arises about how to predict the ROI part of the image. Many researchers present novel techniques to determine the ROI of images using DL techniques. Prakash *et al.* [122], present CNN-based multi-scale ROI. Using an image, they deploy a trained image classification network to predict the classes and then determine which regions belong to these classes. As a result, the proposed techniques highlight noteworthy areas associated with semantic analysis. Our proposed framework for surveillance video compression comes under ROI coding.

2.2.7 End-to-End Compression

Till now researchers tried to enhance tools of H.265 codec like inter-frame prediction, intra-frame prediction, etc. using neural networks, and hence end-to-end compression area remains untouched. Some researchers directed their efforts towards improving video compression through an end-to-end way using NN, and this approach is outlined in this section.

Chen *et al.* [123], used auto-encoders to perform video compression and frames are divided into 32 x 32 blocks and the Mean Square Error (MSE) of the inter-predicted block is determined. If it is greater than the threshold then intra-prediction is preferred otherwise inter-prediction. In intra-prediction, an auto-encoder is used to minimize the block, whereas in inter-prediction, they execute ME and MC using the conventional approach before passing the residues through the auto-encoder. Both auto-encoders use the Huffman method to directly quantize and code the encoded representations. According to the authors, this technique is somewhat poor and does not surpass with H.264. Jiang *et al.* [124], introduced the concept of two CNN. Firstly, it compacts CNN to down-samples an image which is done by using a standard encoder like JPEG and then reconstructs CNN up-samples the decoded image. Because the image encoder/decoder

is not differentiable, the authors choose to tune the two CNNs separately.

Wu *et al.* [125], proposed the prediction of a complete video frame via interpolation, in which the key frames or I-frames are compressed first using an iterative DL algorithm based on convolution LSTM, and then the leftover B-frames are compressed hierarchically. By interpolating between adjacent anchor frames, the proposed architecture codec reconstructs all of the remaining frames. In addition, the author provides the interpolation network with a compact and compressible code that allows it to distinguish between various interpolations and accurately encode the original video frame. Chen *et al.* [126], present Pixel Motion CNN, where compression of the frame is performed using temporal order. Frames are separated into chunks and squeezed via raster scan order. The preceding two squeezed frames were utilized to "extrapolate" the current frame until it was compressed. Cheng *et al.* [127], introduce an image compression framework with a convolutional autoencoder, and then apply the identical strategy to video encoding using an interpolation cycle in both the encoder and decoder ends.

Lu *et al.* [128], propose a most successful compression technique name as Deep Video Compression (DVC), where an estimated optical flow module serves to collect information between the frame and earlier compressed frames for all frames that get compressed. A trained network also performs motion compensation to produce a prediction for a present frame. Prediction residues and motion information are compressed using two auto-encoders. The joint rate-distortion cost, is used to optimize the entire network. Researchers extend the same concept of compression with two new lightweight structures in [129]. Lin *et al.* [130], present the logic of using many previous frames as a reference to detect the relationship among the present and next frames to determine the value of the motion-vector. Multiple reference frames also aid in the generation of motion vector prediction, lowering the coding cost of the motion vector sector and resulting in less residual information. Djelouah *et al.* [131], suggested an interpolation approach that incorporates movement compression and synthesis of images while decreasing computation during the decoding time. This collaborative approach allows for the

reduction of motion code size. Furthermore, since the same network is used for keyframes and residuals, so residuals in latent space improve the technique of video compression.

2.3 Surveillance Video Compression

This sub-section gives a review of video compression approaches applied to surveillance video. Zonglei *et al.* [132], present a deep compression algorithm for apron surveillance. Here, Faster-RCNN approach finds the moving and fixed objects in surveillance video. Researchers perform object detection on the frames of apron surveillance and cropped the object according to its coordinates and saved it on disk in linked list format. In this way, the foreground and background parts of images are differentiated with illumination and brightness information. While performing decompression, using coordinate information cropped objects placed at their position in background images and this approach efficiently solves the space issue. This technique achieves 93.74% compression. Ghamsarian *et al.* [133], perform compression on a cataract surgery video using semantic segmentation approaches and perform categorization of a frame into active and idle frames. Later, active frames of videos are compressed with different quantization parameters under 5 different scenarios from an ophthalmologist's point of view. The compression process achieves 68% of storage space gain by removing the idle part of the video.

Wu *et al.* [134], perform compression on the foreground and background images parallelly using an NN approach. In this study, a scheme to compress foreground and background frames separately is designed. Here, the background information is shared with adjacent frames using background interpolation and updation methods. A coarse-to-fine two-step module increases picture quality by combining foreground and background during decompression. For foreground compression, motion estimation with residual encoding is used and to share the background with neighboring frames, an interpolation method is applied. Paneer and Selvan [135] identify and eliminate duplicate frames and replace them with repeated single images through a Generative Adversarial Network (GAN)+ CNN. These

changes, generated by GAN, facilitate frame-level compression. K-nearest Neighbors is used to compare pixels across frames, followed by K-means clustering and Singular Value Decomposition on each frame of RGB channels to reduce dimensions. Parameters are packed using a codec, converting frames to video format for comparison with the original which achieves 91.51% compression.

2.4 Open Challenges in Video Compression

i. Despite the surge in surveillance applications, DL-based surveillance video compression remains underexplored.

ii. Though DL-based video compression acquires success in terms of performance as compared to hybrid video coding but still computational complexity and memory are still unexplored.

iii. Deep learning methods are increasingly used to extract semantic information from images and videos as network depth increases. However, these techniques remain in the early stages and have scope for improvement.

2.5 Thesis Objectives

To achieve surveillance video compression, the objective of our research work are stated as follows:

i. **Comparative study and analysis of various techniques to identify irrelevant frames in surveillance video using semantic features**

ii. **To develop deep learning and relevance-based object detection model in surveillance videos**

iii. **To develop a content-adaptive video compression model**

iv. **To compare and validate the proposed model with the existing model based on various performance metrics**

2.6 Research Methodology

The expected outcome of the research work is to perform compression of surveillance video using deep learning-based object detection techniques. A flowchart in Figure 2.7 is a list of activities to achieve the mentioned objectives followed by research methodology for each objective.

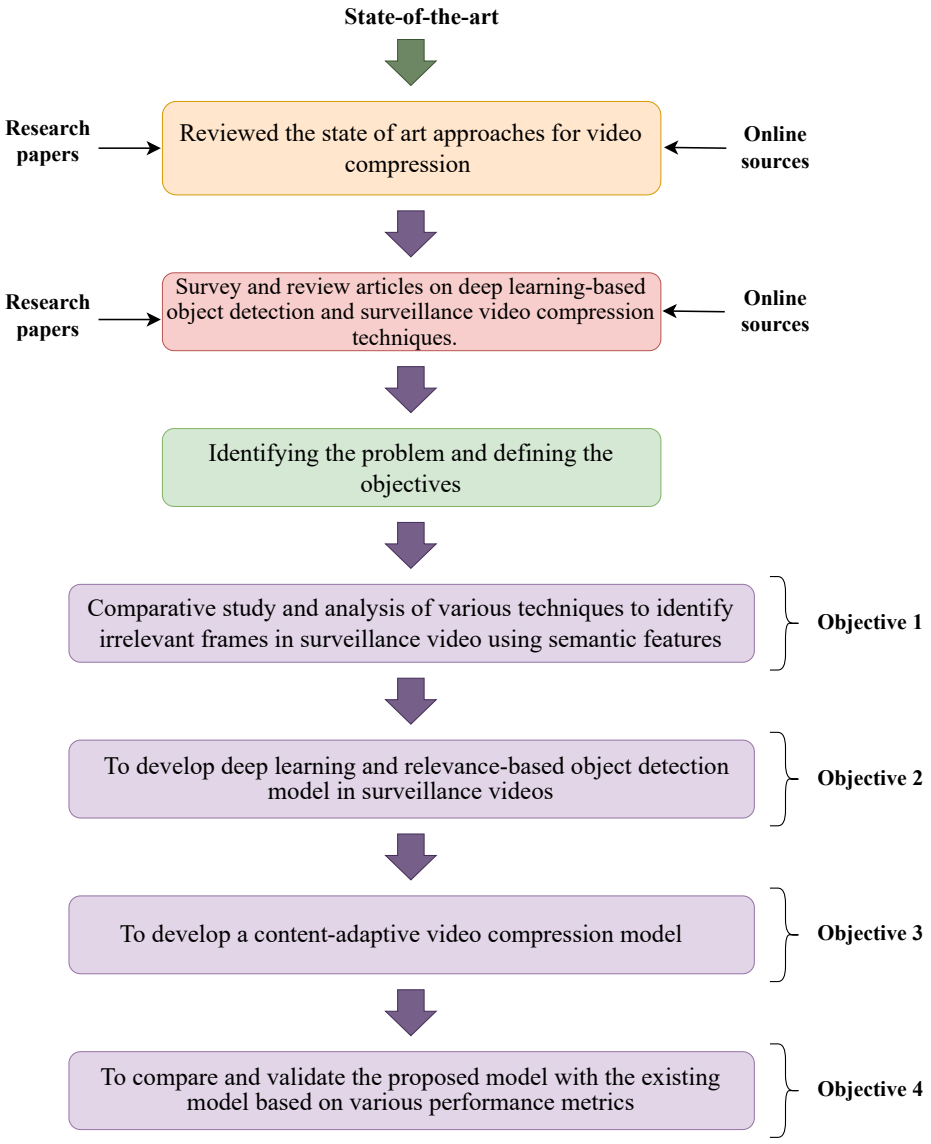


Figure 2.7: Flowchart of Research Methodology.

i. To accomplish the first objective, a systematic survey of deep learning-based object detection methods is conducted to distinguish between relevant and irrelevant frames in surveillance video using semantic features. At the same

time, since the research uses ATM surveillance video as a use case and the data is not publicly available, the data collection process is also carried out.

ii. To achieve the second objective of research work, deep learning based various object detection modules which include YOLOv5, YOLOv7, YOLOv8, YOLOv9 and Mask R-CNN are deployed to detect the relevant frames using a combination of COCO and ATM surveillance video dataset.

iii. In the third objective of research work, relevant frames detected using objective 2 are utilized to perform further compression process. Here, the similarity between consecutive relevant frames is evaluated at various threshold values (ranging from 100% to 90%). Based on similarity, relevant frames are split into primary and similar frames and at last compressed video is constructed using primary frames with appropriate video codec.

iv. Validation of proposed model with existing state-of-the-art model based on various metrics is performed.

Chapter 3

DEVELOPMENT OF RELEVANCE BASED OBJECT DETECTION AND VIDEO COMPRESSION MODEL

CCTV plays a crucial role in modern security and serve as a vigilant electronic eye. The CCTV systems provide continuous monitoring in various sectors, from public spaces to private properties [136]. Their presence deters criminal activities, aids in crime detection, and enhances overall safety [137]. The ability to capture real-time footage and the potential for retrospective analysis make CCTV an invaluable tool for ensuring the security of people and assets [138]. In an era where safety is a top priority, CCTV plays an indispensable role in bolstering surveillance measures [139]. Hence, the proliferation of surveillance cameras to over one billion units indicates its prevalent need [5]. As the need for surveillance increases, service providers consistently improve the quality of surveillance video by augmenting spatial and temporal resolutions, as well as frame rates [140]. This enhancement inevitably results in an expansion of the storage capacity required for surveillance video. It's naive to think that simply increasing the capacity of storage requirements will solve the problem. Typically, CCTV surveillance videos are stored on either cloud platforms or local storage devices, such as microSD cards and local hard drives. The continuous recording of 24×7 surveillance video results in the rapid exhaustion of storage capacity on local hard drives. As a result, surveillance video was deleted after a specific interval of time. This could

result in a loss of relevant information from the user's point of view and hence it is necessary to develop a relevance-based domain-specific compression technique. The compression technique minimizes the storage requirements, making it feasible to store video for a longer period without facing exorbitant storage costs. To deal with this issue, this thesis provides three surveillance video compression techniques which are listed as follows:-

i. Object Detection Based Surveillance Video Compression (ODSC) Model.

ii. Relevant Video Frame Detection and Compression (D&C) Model.

iii. Frame Relevance Based Video Compression (FRVC) Model.

Figure 3.1: Sample Relevant Frame From the Collected ATM Surveillance Video Dataset.

The proposed ODSC model consists of two steps: *(i)* Object Detection Module and *(ii)* Compression Module. In the object detection module, the relevant and irrelevant frames of the surveillance videos are identified using YOLOv5, YOLOv7, and YOLOv8 object detection framework. The frames containing the presence of humans or animals in the ATM room are categorized as relevant frames, whereas frames solely depicting the ATM without such presence are designated as irrelevant frames. In the compression module, the relevant frames of surveillance

Figure 3.2: Sample Irrelevant Frame From the Collected ATM Surveillance Video Dataset.

video are merged to reconstruct a compressed video using an appropriate video codec and the irrelevant frames of surveillance video are discarded. Figure 3.1 and Figure 3.2 represent a visual representation of relevant and irrelevant frames of surveillance video. After ODSC, we present the next deep learning-based relevant video frame Detection and Compression (D&C) model, which consists of three phases: *(i)* Data engineering, *(ii)* Relevant frame detection module and *(iii)* Similarity identification module. In the first phase, surveillance videos are collected and processed for the next module. The relevant frame detection module of D&C model used YOLOv5, YOLOv7, YOLOv8 and YOLOv9 frameworks to detect the relevant frames and irrelevant frames of surveillance video. Later, in the third phase, relevant frames are used to evaluate the similarity between consecutive frames at various threshold values while irrelevant frames are removed. Based on the calculated similarities between frames, the frames are further categorized into two distinct groups: primary frames and similar frames. lastly, the compressed video is constructed using a primary frame which preserves relevant data and optimizes storage efficiency greatly. Here, both the model ODSC and D&C are trained on COCO dataset and tested on ATM surveillance video. Lastly, to address the issue of COCO dataset, we proposed domain specific ATM Surveillance

Video (ASV) dataset which is used to train and test our third model i.e. Frame Relevance based Video Compression (FRVC). The presented FRVC framework is divided into three phases:- *(i)* In the first phase ASV dataset is prepared and annotated, *(ii)* we employ customized one-phase object detectors YOLOv9 and two-stage object detector Mask-RCNN to predict relevant and irrelevant frames of surveillance video. Following the frame categorization process, we turn our attention to the relevant frames only, and *(iii)* the similarity index among them is identified at different threshold levels. These relevant frames are subsequently divided into primary and similar frames. At last, the compressed video is obtained by combining the relevant primary frames. The detailed explanation of all three proposed methodologies with dataset, algorithm and workflow is explained in the following section.

3.1 Object Detection Based Surveillance Video Compression

The proposed Object Detection based Surveillance Video Compression (ODSC) model is divided into two steps *(i)* Object Detection Module and *(ii)* Compression Module. Figure 3.3 represents the architecture of proposed ODSC model. The proposed work employed two datasets: *(i)* The Common Objects in Context (COCO) dataset for training and *(ii)* The ATM Surveillance Video (ASV) for testing purposes. Hence at the earlier stage ATM surveillance videos were divided into frames using the OpenCV library and given as input to the object detection module to identify the relevant and irrelevant frames of video. The dataset, methodology and algorithm of the ODSC model are explained as follows.

3.1.1 Dataset

Data is an integral and essential part of all AI models. It serves as the bedrock upon which AI models are built. It also act as a driving force behind the widespread adoption of machine learning and deep learning technologies. A dataset, as de-

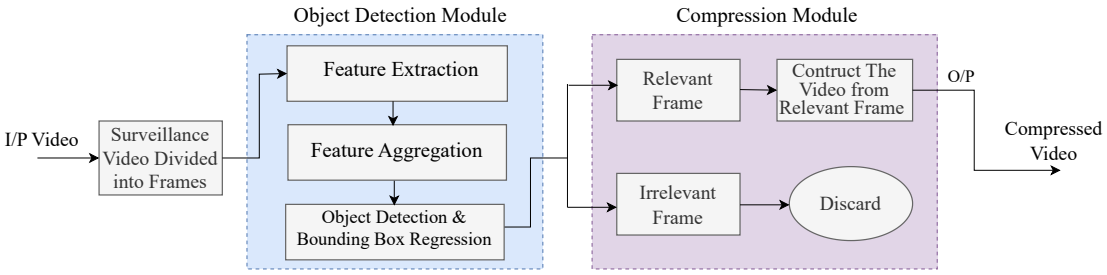


Figure 3.3: Object Detection Based Surveillance Video Compression Model.

fined by Oxford Dictionary, "is a collection of data that is treated as a single unit by a computer" [141]. It means that a dataset has different and separate parts of data that can be used to prepare an algorithm and to obtain predictable patterns inside the whole dataset. A single dataset is divided into three parts; (i) Training dataset: - to checks how well the model is trained. (ii) Testing dataset: - checks the model performance. (iii) Validation dataset: - to evaluate the performance of a model during training and fine-tune its hyperparameters to prevent overfitting. For the development of the proposed ODSC model, two datasets were used: (i) COCO and (ii) ATM Surveillance video which are explained as follows:

i. **Common Objects in Context (COCO) dataset:** COCO dataset is a prominent baseline in the field of CV. It was created to overcome the difficulties associated with OD, segmentation, and image captioning. COCO is notable for its large scale, diversity, and complexity, making it suitable for training and evaluating advanced CV algorithms [142]. The key aspects of the COCO dataset are as follows:

   a. *Image Annotations:* The COCO dataset contains over 200,000 images, each annotated with comprehensive object-level annotations. These annotations provide detailed information about the objects present in the images, including their categories, bounding boxes, and segmentation masks.

   b. *Object Categories:* The dataset includes 80 object categories, containing ordinary regular things like humans, pets, automobiles, domestic items, and more, making it appropriate for a wide range of OD and recognition

applications.

c. *Instance Segmentation:* In addition to OD, the COCO dataset also includes annotations for instance segmentation. This means that each object in the image is not only labeled with its category but also segmented at the pixel level, providing precise boundaries for each object instance.

d. *Scene Context:* The COCO dataset distinguishes itself by emphasizing the capture of items in their surroundings. Images in the dataset are diverse and include a wide range of scenes, backgrounds, and contexts, reflecting real-world scenarios encountered in everyday life.

ii. **ATM Surveillance Video Dataset:** We consider ATM surveillance video as the use case for our research work and the dataset is not publically available as it is highly confidential. Hence, the data is collected through private channels for research work purposes. We collect 15 ATM surveillance videos of duration 60 Minutes to 90 minutes from ADCC bank, Morshi, Maharashtra. The collected raw data is of large duration, so it is split into smaller clips to analyze its performance. This segmentation strategy allowed for more manageable data processing while retaining the integrity of the original footage.

3.1.2 Methodology

Step-I: Object Detection Module: To predict the relevant and irrelevant frames of ATM surveillance video, we used YOLOv5, YOLOv7, and YOLOv8 from a region-free detector family, which are summarized as follows.

i. *YOLOv5:* Till 2020 the framework of OD is split into two steps:- (i) feature extraction and (ii) categorization, YOLOv5 [29] is the first to introduce the concept of feature aggregation in the OD framework and developed on PyTorch framework. The three stages of the YOLOv5 architecture i.e. backbone, neck, and head is shown in Figure 3.4. This framework used Cross-Stage Partial connection (CSP DarkNet-53) [143] as a backbone to perform feature extraction, SPP [45] and PaNet [144] are used as a neck for feature merging,

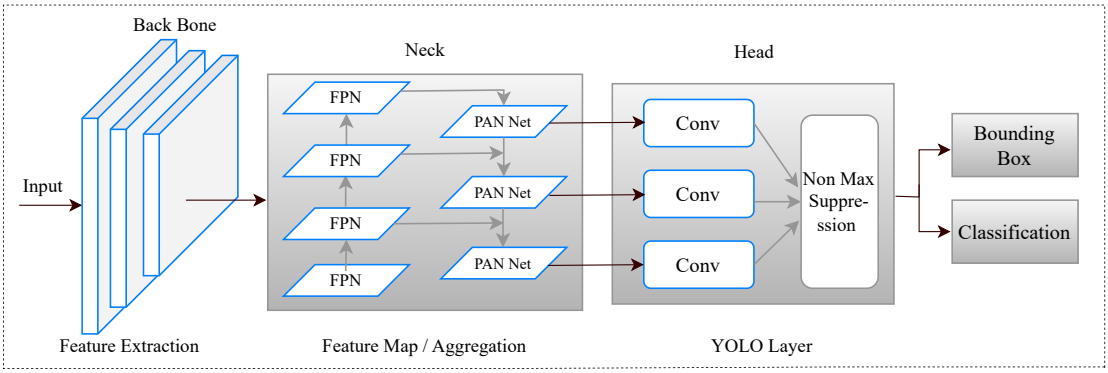


Figure 3.4: Figure Illustrate the Architecture of YOLOv5 Module Which Consist of Three Stages: Backbone, Neck and Head [29].

and YOLO layers are use to perform detection. The detailed explanation of each stage is as follows:

a. *Backbone:* YOLOv5 uses the CSP Darknet-53 as a backbone for feature extraction. CSP DarkNet-53 represents an integration of CSPNet and DenseNet architectures. Here, the feature extraction begins with convolutional layers represent in Eq. (3.1).

$$F_l = \sigma(W_l * F_{l-1} + b_l), \tag{3.1}$$

where:

- F_{l-1}: Input feature map at layer $l-1$,
- W_l and b_l: Weights and biases of the convolutional layer l,
- $*$: Convolution operation,
- σ: Activation function.

To enhance the network's precision, dense blocks incorporate residual/skip connections, where each block output get passed on to the layers that follow it. Within each dense block, there is an amalgamation of batch normalization, ReLU activation, CNN, and dropout layers. Several of these dense blocks combine to form a DenseNet, which is subsequently followed by a transition layer. Since each layer's output feeds to the next layer, this approach escalates the network's computational complexity and faces

obstacles related to gradient during backpropagation. To tackle this issue, CSPNet is integrated with DenseNet to create CSP-DenseNet or CSP-Darknet. In CSP-DarkNet, the input is divided into two parts. One part is fed into the DenseNet, and the other is merged with the output of DenseNet to extract high-quality features as shown in Eq. (3.2).

$$F_{\text{CSP}} = \text{Concat}(F_1, F_2), \tag{3.2}$$

where F_1 and F_2 are feature maps from two separate paths.

b. *Neck:* To combine semantic and spatial features of the various layers of backbone, the neck is used as a feature aggregator. In YOLOv5, FPN and PaNet perform the task of feature aggregation as shown in Eq. (3.3). As we go deeper into the network, good quality semantic features are extracted but lacking in spatial features which are present at the early layers of the network. To merge both features FPN, uses top-down and bottom-up paths. While PanNet used the lateral connection to reduce the information pathway and focused on enhancing the high-resolution feature maps with detailed information. It consists of 1x1 convolutions that transform feature maps from the backbone into a common channel dimension. These transformed feature maps are then added element-wise to good-resolution feature through Bottom-up Path. This step is crucial for aligning features across scales.

$$F_{\text{aggregated}} = \text{Concat}(\text{Upsample}(F_l), F_{l-1}), \tag{3.3}$$

where Upsample$(\cdot)$ typically involves bilinear interpolation or transposed convolution.

c. *Head:* At last, YOLO layers perform detection at multiple scales. This network adds anchor boxes to the feature map created by the previous layer and generates a vector comprising the target object's category likelihood, object score, and Bounding Box (BB) position. Every layer ultimately

produces a 21-channel vector, which includes information about classes, class probabilities, and BB coordinates as shown in Eq. (3.4) (comprising 2 classes, 1 class probability, and 4 coordinates for each of the 3 anchor boxes). At inference time, YOLOv5 applies Non-Maximum Suppression (NMS) to remove duplicate detections by eliminating bounding boxes that have a high Intersection over Union (IoU) overlap and low confidence scores. This process results in the prediction and labeling of bounding boxes and division of relevant and irrelevant frames. Figure 3.5 represents relevant frames and Figure 3.6 represents irrelevant frames using the YOLOv5 network.

$$\text{Dimensions of the Output Tensor for OD} = S \times S \times (B \times 5 + C) \quad (3.4)$$

where:

- $S \times S$ is the number of grid cells,
- B is the number of bounding boxes per grid cell,
- C is the number of classes.

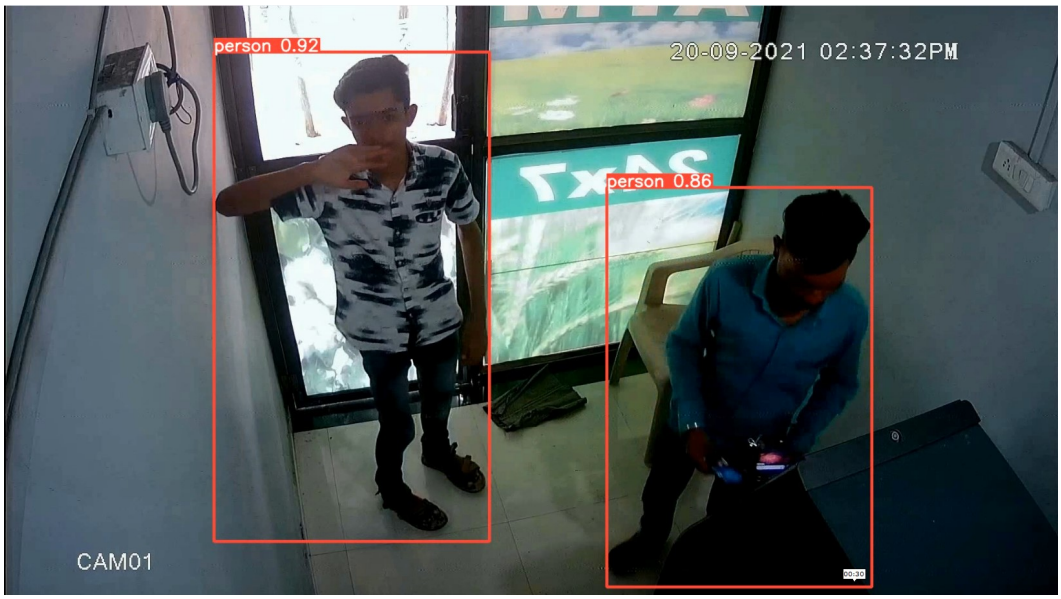


Figure 3.5: Output of YOLOv5 Module Trained on COCO Dataset (Relevant Frame).

Figure 3.6: Output of YOLOv5 Module Trained on COCO Dataset (Irrelevant Frame).

ii. *YOLOv7:* In 2023, YOLOv7 is introduced to improve the detection accuracy and speed of YOLO architectures [30]. Like earlier YOLO architectures, YOLOv7 is made up of a backbone, neck, and head component. The basic structure includes a CNN + BN + Silu (CBS) module, an ELAN module, and a MaxPool (MP) module for feature extraction. The ELAN module divides feature maps into different scales across varying depths before integrating them, which facilitates efficient learning and convergence within the deeper network. The primary purpose of the MP module is downsampling, achieved by combining the maxpool downsampling branch with the convolution downsampling branch. This fusion of feature maps obtained through different downsampling techniques preserves maximal feature information while minimizing computational overhead. The neck is composed of FPN and PAN-Net for feature aggregation which facilitates top-down and bottom-up bidirectional fusion. This structure allowed for multi-scale fusion of network features by combining characteristics from several backbone network and detection layers. The head network featured three detection heads of different dimen-

sions. Figure 3.7 shows the architecture of YOLOv7 module and Figure 3.8 indicates the output of the YOLOv7 module.

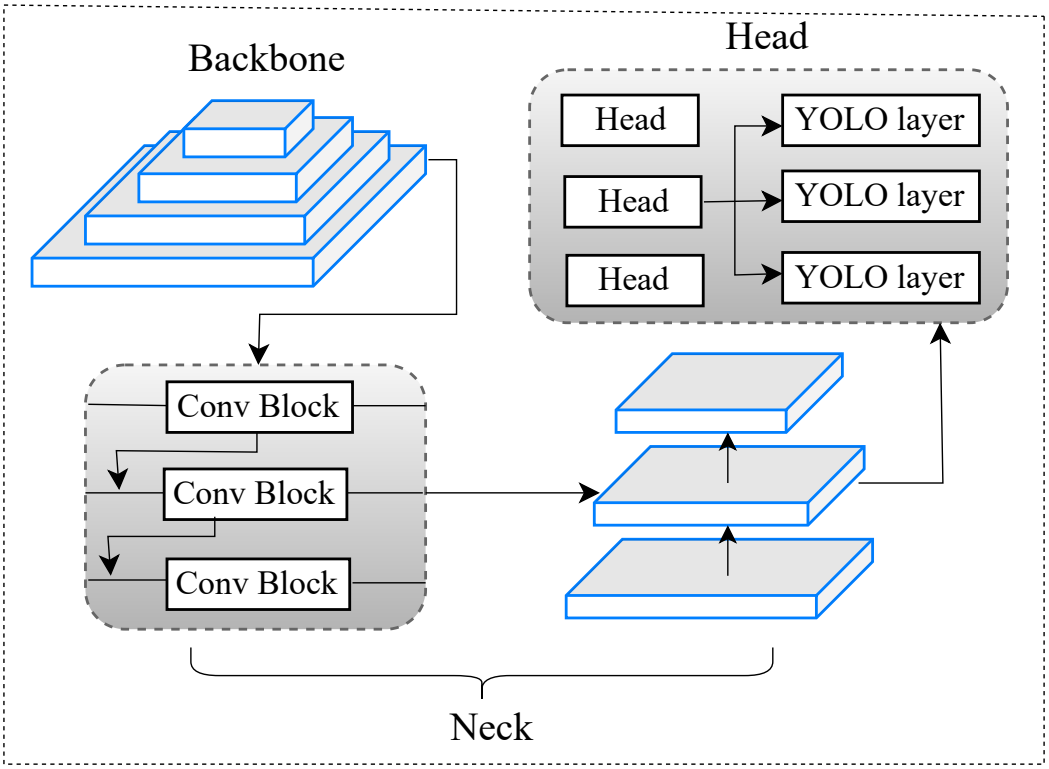


Figure 3.7: This Figure Showcase the YOLOv7 Architecture Module, Which Features an Optimized Backbone, Enhanced Feature Fusion, and an Advanced Detection Head [30].

iii. *YOLOv8:* In 2023, Ultralytics again presented YOLOv8 [31], a region-free OD framework that consists of several key components: an input, a backbone, a neck, and an output [39]. The based structure is responsible for preprocessing the input image. It applies mosaic data augmentation, adaptive anchor calculation, and adaptive grayscale padding to enhance the input data. The backbone network and neck module form the core structures of YOLOv8. The backbone network processes the input image using Conv and C2f modules to extract feature maps at different scales. It incorporates the ELAN structure from YOLOv7 [30], reducing one standard convolutional layer and enhancing gradient flow through the Bottleneck module. This approach maintains lightweight characteristics while capturing more gradient flow information.

Figure 3.8: Output of YOLOv7 Module Trained on COCO Dataset (Relevant Frame).

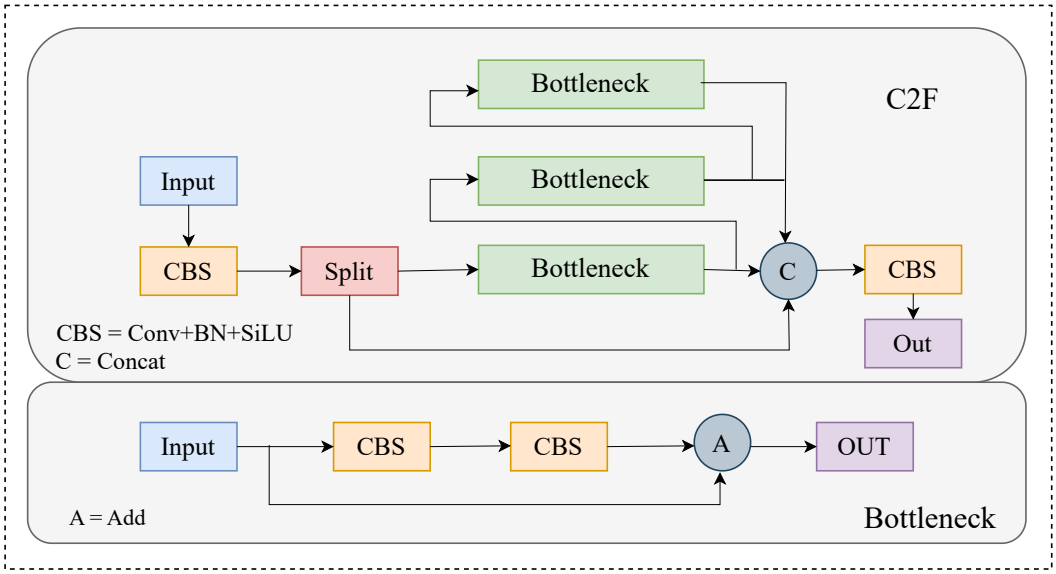


Figure 3.9: The Architecture of the YOLOv8 Module Which Highlights Key Components such as CBS (Conv+BN+SiLU), Bottleneck, and C2F structures. The Data Flow Illustrates a Feature Fusion Through Add and Concat Operations that Optimize Detection Performance [31].

The output feature maps are processed by the SPPF module, which employs pooling with varying kernel sizes to combine feature maps effectively. Finally, the results are passed to the neck layer for further processing. The neck layer of YOLOv8 is designed to improve the model's feature fusion capability

by incorporating the FPN along with the Path Aggregation Network (PAN) structure like YOLOv5 [29]. This combination enables better integration of features from different scales and enhances the model's ability to capture contextual information across the entire image. The detection head of YOLOv8 adheres to the standard approach of dividing the classification head from the detection head. This process includes performing loss computation and filtering out target detection boxes. The loss calculation consists of two main parts: classification and regression, with the Objectness branch excluded. For classification, Binary Cross-Entropy (BCE) loss is utilized, while Distribution Focal Loss (DFL) is employed for regression. Figure 3.9 represents the YOLOv8 framework and Figure 3.10 indicates the output of the YOLOv8 module..

Figure 3.10: Output of YOLOv8 Module Trained on COCO Dataset (Relevant Frame).

Step-II Compression Module:

In the compression module of the ODSC model, the output of the YOLOv8 module is used to perform compression. In these step, Compression is achieved by merging the relevant frames of surveillance video and irrelevant frames are eliminated. These frames of surveillance video are merged at the determined FPS

value of the original video using the proper video codec.

3.1.3 Workflow and Algorithm

The Sequence Flow Diagram (SFD) of the ODSC model is shown in Figure 3.11. The steps in this framework are as follows: -

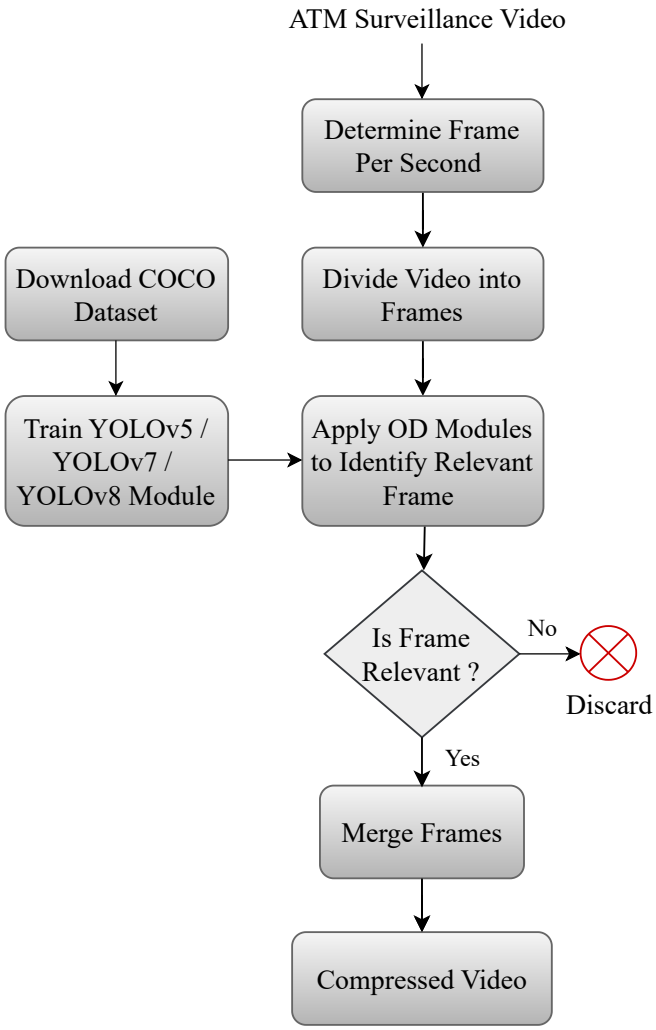


Figure 3.11: Sequence Flow Diagram of ODSC Model.

i. The ATM surveillance video is taken as input and its frame rate i.e. FPS is determined.

ii. The Surveillance video is divided into frames using OpenCV library for further process.

iii. The OD modules YOLOv5, YOLOv7, and YOLOv8 are tained on COCO dataset and used to predict relevant and irrelevant frames in the surveillance video.

iv. If the frame of Surveillance video are classified as relevant then considered it for further process else discard the irrelevant frames of video.

v. The compressed surveillance video is created in the final phase by merging all relevant frames.

Algorithm 1: ODSC algorithm

Input: ATM surveillance video
Data: COCO dataset
Output: Compressed video

```
ODSC_FPS ← Obtain FPS (Input ATM video);
ODSC_Split ← Divide video in frames;
while all frames do
    ODSC_OD ← Apply YOLOv5, YOLOv7, and YOLOv8 frameworks;
    ODSC_Relevant ← Predict frames;
    ODSC_Irrelevant ← Predict frames;
end
ODSC_Compressed ← Construct(ODSC_Relevant);
ODSC_Discard ← Delete(ODSC_Irrelevant);
```

Algorithm 1 demonstrates the step-by-step execution of the proposed ODSC model:- In this algorithm, the FPS of surveillance video is calculated using the Open CV library and saved in variable ODSC_FPS. Then the video is divided into frames and saved in ODSC_Split. Then we apply the three OD modules to identify relevant and irrelevant frames of surveillance video and relevant frames save in variable ODSC_Relevant and irrelevant into ODSC_Irrelevant variable. At last, the video is constructed using ODSC_Relevant and saved in variable ODSC_Compressed, and irrelevant frames are deleted from the network.

3.2 Relevant Video Frame Detection and Compression Model

This section outlines, the deep learning-based relevant video frame Detection and Compression (D&C) model. The developed D&C model consists of three steps as shown in Figure 3.12. The model is trained on the COCO dataset and tested on ATM Surveillance video. The details explanation of the dataset is mentioned in subsection 3.1.1. The three-step of D&C model are listed as follows:

i. Data Engineering: Collection of the dataset and split video to frames.

ii. Relevant Frame Detection Module: Used OD modules to detect relevant and irrelevant frames of surveillance video.

iii. Similarity Identification Module: Based on similarity threshold value divide relevant frames into primary frames and similar frames.

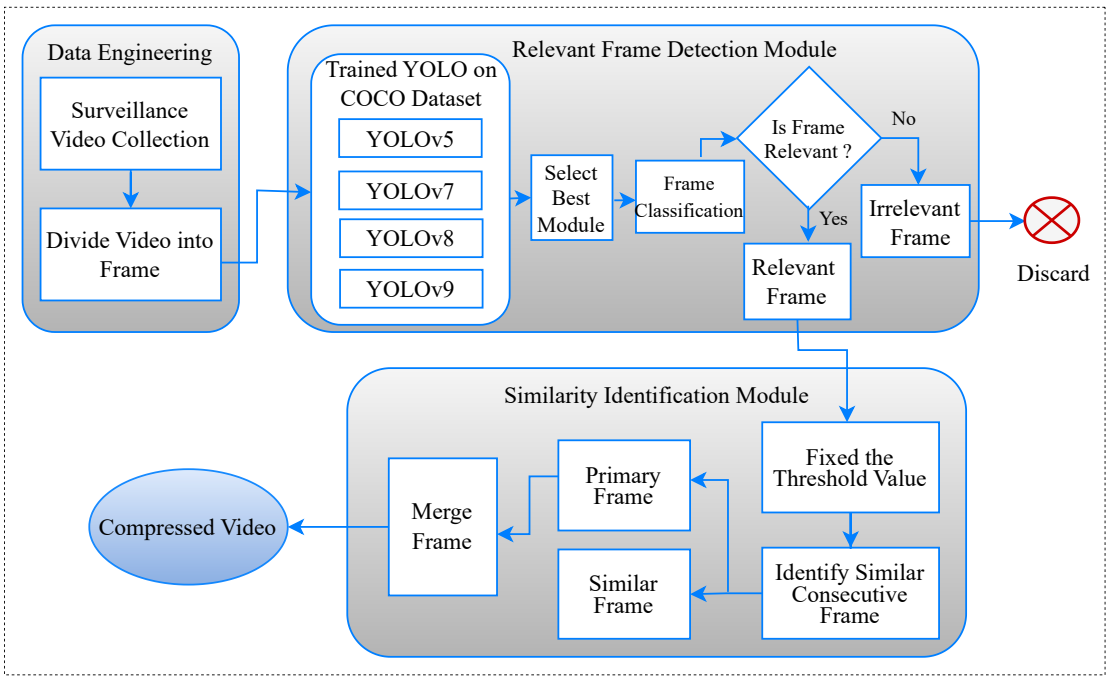


Figure 3.12: An Overview of Deep Learning-based Relevant Frame Detection and Compression (D&C) model, which consists of three modules: (i) Data Engineering Module, (ii) Relevant Frame Detection Module and (iii) Similarity Identification Module.

3.2.1 Methodology

Step-1: Data Engineering

i. *Surveillance video collection:* Our research focuses on the specific application of ATM surveillance video and the dataset is not publicly accessible. To address this limitation, we gathered a collection of 15 ATM surveillance videos, each spanning one hour plus in duration, from ADCC Bank located in Amravati, Maharashtra.

ii. *Divide video into the frame:* To facilitate the analysis process, we employed FFmpeg software to divide the original long videos into smaller segments. Using OpenCV library the video is further divided into frames and passed to YOLO modules.

Step-2: Relevant Frame Detection Module

i. *YOLOv5:* Ultalytics introduced YOLOv5 in 2020 on PyTorch framework [29]. It employs the CSP-Darknet53 as a backbone network, used its cross-stage partial network design for enhanced information flow and computational efficiency.YOLOv5 integrates a neck architecture to refine feature extraction, through a FPN [48] and PANNet [144] to enhance spatial resolution. The head of YOLOv5 is responsible for predicting BB, confidence scores, and class probabilities of objects. This section typically consists of CNN layers followed by a final layer to output predictions. During training, YOLOv5 utilizes a combination of loss functions to optimize performance which includes localization (L_{Loc}), confidence (L_{Obj}), and classification losses (L_{Cls}) shown in Eq. (3.7). Localization loss used smooth L1 loss function denoted in Eq. (3.5), Where x denotes the variance among the predicted and goal values. Objectness loss and classification loss used BCE, where p_{i} predicted the objectness score, t_{i} is the true objectness score and N_{obj} is the total number of anchor boxes shown in Eq. (3.6). The overall loss is determined using Eq. (3.7), which is a weighted sum of these individual losses, with different weights assigned to

maintain a balance during training. The function $L_{\text{Loc}}(x)$ is defined as:

$$L_{\text{Loc}}(x) = \begin{cases} 0.5x^2 & \text{if } x < 5, \\ |x| - 0.5 & \text{otherwise.} \end{cases} \tag{3.5}$$

The equation for $L_{\text{Obj}}, L_{\text{Cls}}$ is given by:

$$L_{\text{Obj}}, L_{\text{Cls}} = -\frac{1}{N_{\text{obj}}} \sum_{i=0}^{N_{\text{obj}}} [t_i \log p_i + (1 - t_i) \log (1 - p_i)] \tag{3.6}$$

$$L = L_{\text{Loc}} + L_{\text{Obj}} + L_{\text{Cls}} \tag{3.7}$$

Post-processing techniques such as NMS are applied to refine and filter the predicted bounding boxes. Overall, YOLOv5 offers a streamlined and efficient architecture for real-time object detection, making it a powerful tool for various applications. Figure 3.4 denotes the architecture of the YOLOv5 module.

ii. *YOLOv7:* In 2023 Wang *et al.* [30], introduced YOLOv7, a popular and efficient OD algorithm that predicts and classifies objects. Researchers developed fundamental models tailored for different GPU settings. YOLOv7 uses a modified YOLOv3 architecture as its backbone, but it incorporates several architectural improvements to enhance detection accuracy and speed. The architecture comprises several key elements, including compound scaling, EELAN and re-parameterized convolution. The trainable bag-of-freebies approach improves detection accuracy without incurring additional inference costs. It also focuses on the re-parameterization module which replaces the original module and the dynamic label assignment strategy which manages the assignment of labels to different output layers is a key aspect in improving object detection techniques. This approach performs stack scaling to the neck component and utilizes the proposed compound scaling method to increase the depth and width of the entire model, resulting in YOLOv7. Figure 3.7

represents the architecture of the YOLOv7 model [30].

iii. *YOLOv8:* In 2023, Ultralytics [31], revealed YOLOv8, an innovative region-free OD framework that surpasses all existing YOLO versions from V1 to V7 in terms of speed accuracy and speed. Like other YOLO algorithms, the structure of YOLOv8 also contains backbone, neck, and head parts. The c2f module (CONV Layer with 2 features) is replaced by the c3 module (CSP + CONV layers) in the backbone structure. The c3 module architecture is crafted by incorporating elements from the E-ELAN network of YOLOv7 [30] alongside the traditional C2f module and depthwise separable convolutions(DSepConv). This fusion ensures the extraction of high-quality features while maintaining a rich flow of gradient information throughout the network. The neck layer integrates the FPN [48] and PAN [144] structure to bolster the model's feature fusion capacity. By leveraging a combination of upsample and downsample techniques, FPN+PAN amalgamates high-level and low-level FM and facilitates the transfer of both semantic and localization features. The head section of YOLOv8 transitions from an anchor-based methodology to an anchor-free strategy. Consequently, it executes multi-scale predictions utilizing 8×, 16×, and 32× down-sampled features to ensure precise predictions for small, medium, and large targets. Classification and regression are the two loss functions used in YOLOv8, where BCE loss is employed for classification, while DFL is utilized for regression. Hence, YOLO algorithms are used to detect the person in surveillance video [145].

iv. *YOLOv9:* In 2024, Wang*et al.* [32], presents You Only Look Once version 9 i.e. YOLOv9 to tackle the issue of losing information in the deeper network. Unlike other YOLO, its architecture is divided into three parts: backbone, neck, and head. The backbone of YOLOv9 incorporates advanced techniques such as Programmable Gradient Information (PGI) and Generalized Efficient Layer Aggregation Networks (GELAN). PGI optimizes gradient flow and improves training stability, efficiency, and feature learning in deep networks. While GELAN combines two neural network architectures, CSPNet [143] and

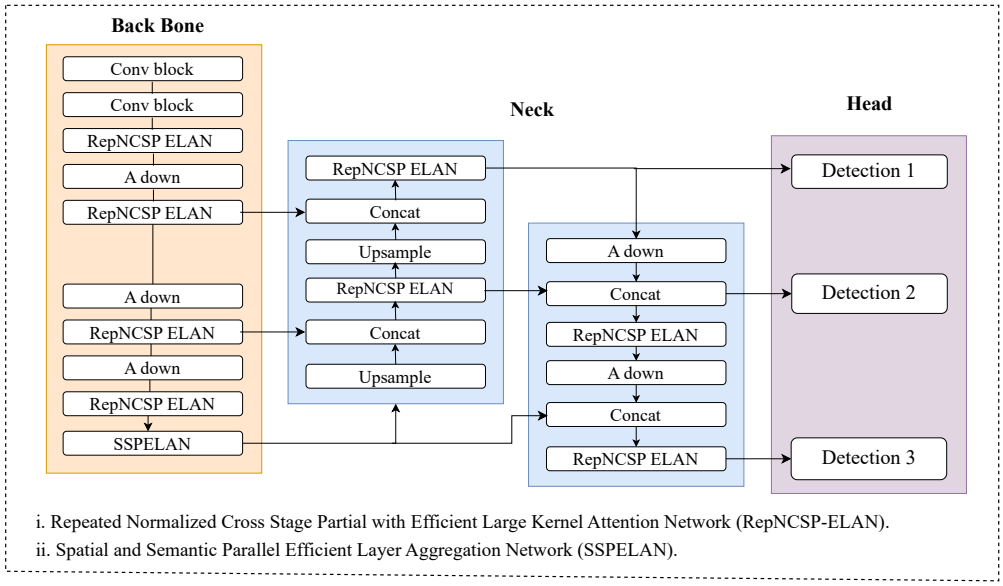


Figure 3.13: The Figure Showcases the Architecture of YOLOv9 Module Which Comprises Three Primary Components: the Backbone, Neck, and Head [32].

ELAN [146] to optimize feature extraction through lightweight architecture. Figure 3.13 denotes the architecture diagram of YOLOv9. It also incorporates gradient path planning for better performance. The neck of YOLOv9 utilizes a PANNet to integrate multi-scale features and improves spatial context and localization accuracy while preparing refined feature maps for the detection head. YOLOv9 also employs an Asymmetric Downsampling (A DOWN) block instead of max pooling, to preserve essential spatial information and to reduce the size of the feature map. Lastly, the head predicts bounding boxes and class probabilities using a custom loss function that combines localization, confidence, and classification losses. Table 3.1 highlights the major difference between implemented YOLO modules and Figure 3.14 shows the output of YOLO modules.

Figure 3.14: Output of YOLOv9 Module Trained on COCO Dataset (Relevant Frame).

Table 3.1: Comparison of YOLOv5, YOLOv7, YOLOv8, and YOLOv9

Aspect	**YOLOv5**	**YOLOv7**	**YOLOv8**	**YOLOv9**
Backbone	CSPDarkNet-53	E-ELAN	E-ELAN + DSepConv	PGI+GELAN
Neck	PANNet+ FPN	Extended PANet	PANNet + FPN with advanced fusion techniques	Improved PANNet with enhanced Transformer
Head	Unified head for classification and localization	Decoupled head for separate classification and localization	Decoupled head (separate classification and localization)	Decoupled, anchor-free, optimized for edge devices
Anchor Mechanism	Anchor-based	Anchor-based	Anchor-free	Anchor-free/fully anchorless
Loss Function	CIoU (Complete Intersection over Union) , Focal Loss	Task-aligned loss functions	CIoU/DIoU Loss, Focal Loss	CIoU, Focal Loss with advanced regularization
Post-Processing	Non-Maximum Suppression (NMS)	Improved NMS, including DIoU-NMS	NMS with slight optimizations	Soft-NMS, DIoU-NMS

Step-III: Similarity Identification Module

In the similarity identification module (Compression module) of D&C model, processing is performed on the relevant frames of surveillance video while the irrelevant frames are discarded from the network. On relevant frames, the similarity index between frames across various threshold values is calculated and further divided these frames into primaryt frames and similar frames. Relevant frames are defined as frames that exhibit similarity when compared with their consecutive frames. If consecutive frames are identical, they are categorized as similar frames. The first dissimilar frame is marked as the next primary frame. For instance, the first frame is considered as primary frame and if this frame is similar to the second, third and fourth frames but dissimilar to the fifth frame, then the first and fifth frames are considered primary frames. While the second, third and fourth frames are designated as similar frames. As the fifth frame is dissimilar to the first frame, it is subsequently identified as the next primaryt frame, and again the frames similar to the fifth frame are determined and the process continues until similarity between all frames is determined. We determine the similarity between frames at various threshold values from 100% similar to 90% similar and save the count of the similar frames corresponding to the primary frame in a data structure. Lastly, the video is constructed using primary frames, at the original FPS.

In D&C model, to identify similarity between to frames the Structural Similarity Index (SSIM) [147] is used. SSIM is a composite measure that evaluates the similarity between two images based on three components: luminance, contrast, and structure. The luminance component quantifies the similarity in average brightness, computed using the means of pixel intensities. The contrast component measures variability in the images, expressed through standard deviations. The structure component evaluates the correlation between the two images using covariance. These components are combined multiplicatively to yield the overall SSIM index. The luminance term in SSIM compares the mean intensities of the

two images x and y. Eq. (3.8) is used to calculate luminance.

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \tag{3.8}$$

where μ_x and μ_y represent the mean intensities of the two images and are determined using Eq. (3.9). In Eq. (3.9), N represents the number of pixels in the sliding window. For e.g. if the window is square with a size of $K \times K$, then: $N = K \times K$ and if $K = 11$ then 11×11 window is used where $N = 121$. Hence, Luminance evaluates the similarity in the average brightness levels between two images.

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i. \tag{3.9}$$

The contrast assesses the variability (standard deviation) in the images using Eq. (3.10), where σ_x and σ_y are the standard deviations of the two images and computed using Eq. (3.11):

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2} \tag{3.10}$$

$$\sigma_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)^2}, \quad \sigma_y = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \mu_y)^2}. \tag{3.11}$$

The structure term captures the correlation between the images. It is defined using Eq. (3.12), where σ_{xy} is the covariance between the two images and determined in Eq. (3.13).

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \tag{3.12}$$

$$\sigma_{xy} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y). \tag{3.13}$$

Finally, the overall SSIM index combines luminance, contrast, and structure which is given by Eq. (3.14) or Eq. (3.15).

$$\text{SSIM}(x,y) = l(x,y) \cdot c(x,y) \cdot s(x,y), \tag{3.14}$$

or equivalently:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}. \tag{3.15}$$

In the above equations, c_1, c_2, and c_3 are small constants used to ensure numerical stability, where $c_3 = \frac{c_2}{2}$ by convention.

3.2.2 Workflow and Algorithm

The SFD of the D&C model is shown in Figure 3.15. The model is divided into the following steps:

i. Initially surveillance videos and the COCO dataset are collected for training and testing purposes of D&C model.

ii. Determines the Frames Per Second (FPS) surveillance video and then splits the video into frames.

iii. Next, a relevant frame detection module is trained on the COCO dataset. This module is responsible for identifying frames that contain relevant information.

iv. Iterates through all frames of the video. For each frame, it uses the trained relevant frame detection module to detect relevant frames and discard irrelevant frames. Firstly, YOLO modules perform Feature Extraction (FA), then Feature Aggregation (FA) and at last Feature Mapping (FM). Using the output of FM, relevant and irrelevant frames are detected.

v. For each relevant frame, calculate the similarity index with its consecutive frames across various threshold values ranging from 100% to 90% similarity.

vi. If a frame is found to be similar to its consecutive frame above a certain threshold, it is considered a similar frame. Otherwise, it is marked as a primary frame.

vii. Constructs the video using the primary frames at the original FPS, ensuring that only the frames containing significant changes are retained.

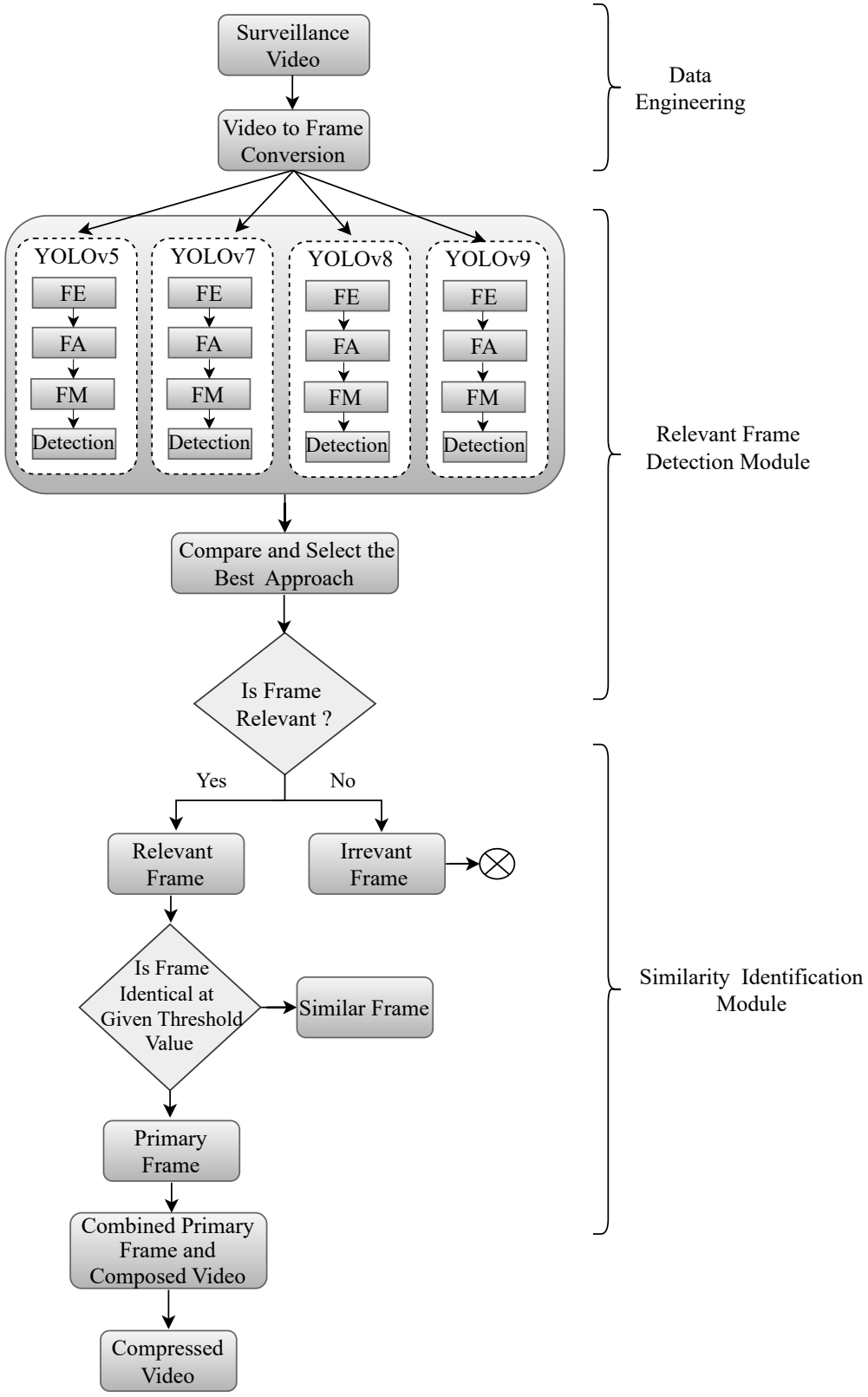


Figure 3.15: Sequence Flow Diagram of D&C Model.

viii. Finally, the compressed surveillance video is generated using the relevant primary frames.

Overall, D&C model combines object detection techniques to identify relevant frames and a similarity identification process to further compress the video by retaining only the frames that significantly differ from their consecutive frames. This approach helps reduce the size of the surveillance video while preserving important information for analysis.

Algorithm 2: D&C Algorithm

Require: Surveillance footage
Ensure: Compressed video
Input: ATM surveillance
Output: Compressed surveillance video
Steps:
$D\&C\_FPS \leftarrow$ Determine FPS (Input ATM Surveillance video)
$D\&C\_Split \leftarrow$ Split video into frames
$D\&C\_Train \leftarrow$ Train relevant frame detection module on COCO dataset
$D\&C\_Save \leftarrow$ Give $D\&C\_Split$ as input to trained $D\&C\_Train$
While all frames
    $D\&C\_Relevant \leftarrow$ Detect relevant frames
    $D\&C\_Irrelevant \leftarrow$ Detect irrlevant frames
    $D\&C\_Discard \leftarrow$ Remove the $D\&C\_Irrelevant$
Consider only relevant frames
Apply Similarity Identification in Compression Module
for each frame i in the relevant frames **do**
    for each threshold value t from 100% to 90% similarity **do**
        if frame i is similar to its consecutive frame with similarity $\geq t$ **then**
            Increment count of similar frames corresponding to primary frame i in data structure
        else
            Mark frame i as a primary frame
        end if
    end for
end for
Construct the video using key frames at the original D&C_FPS

The Algorithm 2 begins by taking the ATM surveillance footage as input and aims to produce a compressed surveillance video as output. Initially, it determines the FPS of the input video, denoted as D&C_FPS, and proceeds to split the video into frames, stored in the variable D&C_Split. Subsequently, a relevant frame

detection module is trained on the COCO dataset, denoted as D&C_Train, and the resultant model is used to identify relevant frames within the video. These relevant frames are stored in the variable D&C_relevant, while irrelevant frames are detected and discarded, represented by D&C_Irrlevant and D&C_Discard, respectively. Moving to the compression module, each relevant frame is processed to identify its similarity with consecutive frames across various threshold values, denoted as Threshold_t. For each relevant frame, if it exhibits similarity above a certain threshold with its consecutive frame, it is categorized as a similar frame; otherwise, it is marked as a primary frame. The count of similar frames corresponding to each primary frame is stored in Similarity_Count. Eventually, the compressed surveillance video is constructed using the primary frames at the original FPS, denoted as Compressed_Video.

3.3 Frame Relevance Based Video Compression

This section presents the Frame Relevance Based Video Compression (FRVC) model which is divided into three phases: *(i)* Dataset Preparation, *(ii)* Relevance Frame Classification and *(iii)* Video Compression. Figure 3.16 shows the architecture of the proposed model. The FRVC model deploys ATM Surveillance Video (ASV) dataset for training and testing purpose. The details steps involved in the development of ASV dataset is mentioned below.

3.3.1 ATM Surveillance Video Dataset

The COCO dataset becomes a standard benchmark for evaluating and comparing the performance of OD algorithms. Though it is widely used and highly regarded in the field of CV, it does have some drawbacks which are mentioned as follows:

i. *Complexity and Annotation Errors:* Due to the complexity and scale of the dataset, there may be inconsistencies or errors in the annotations. This can include inaccuracies in object bounding boxes, segmentation masks, or category labels which affects the caliber of training data and the efficacy of algorithms

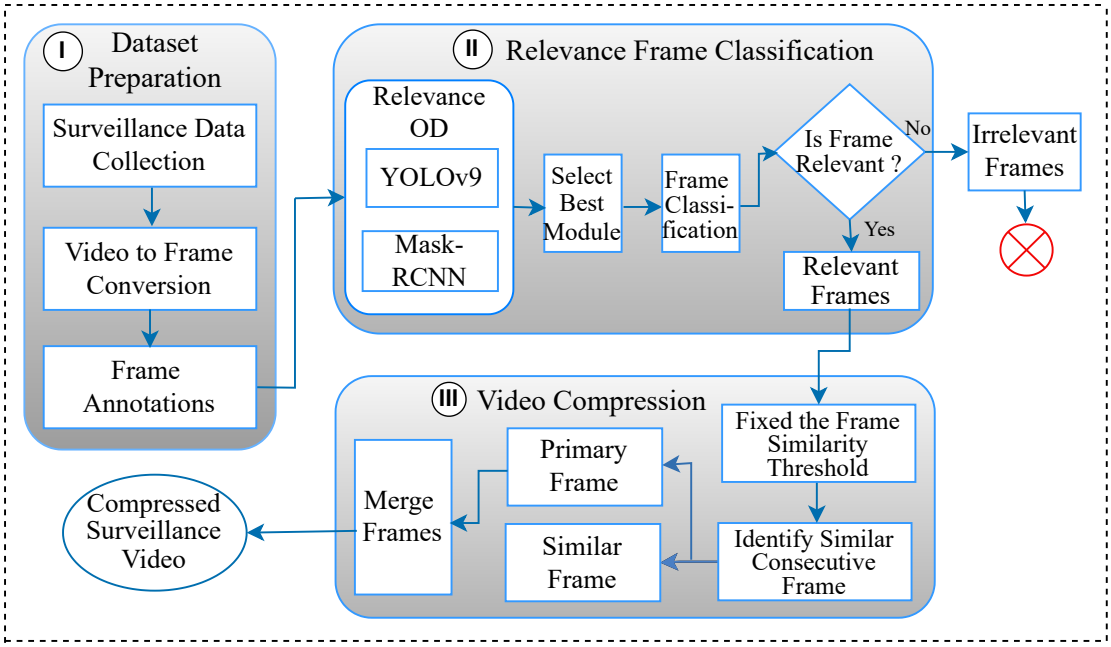


Figure 3.16: An Overview of Frame Relevance Based Video Compression model, which consists of three modules: (i) Dataset Preparation Module, (ii) Relevance Frame Classification Module and (iii) Video Compression Module.

trained on the dataset.

ii. *Lack of Fine-Grained Annotations:* While COCO provides detailed annotations for OD and segmentation tasks, it may lack fine-grained annotations for certain attributes or characteristics of objects.

iii. *Limited Contextual Information:* While COCO aims to capture objects in their context, the contextual information provided in the dataset may be limited compared to real-world scenarios.

iv. *Scalability and Computational Requirements:* Because of their large size and complexity, training models on COCO requires significant computational resources and expertise. This can pose challenges for researchers and practitioners with limited access to high-performance computing infrastructure or specialized hardware.

v. *Data Bias and Representation:* Like many large-scale datasets, COCO may suffer from biases in terms of data representation, cultural contexts, or demo-

graphic characteristics.

Considering the limitations of the COCO dataset, we chose to develop our own Dataset. For our research work, we selected ATM surveillance videos as the use case. Hence, we prefer ATM surveillance video to prepare our dataset. However, this dataset is not publicly available due to its highly confidential nature.

Initially, ATM Surveillance videos are collected through private channels for the research work. We collect 15 ATM surveillance videos of the duration of one hour plus from ADCC bank, Morshi, Maharashtra. As the collected raw data is of duration more than two hours, so it is split into smaller clips starting from 10 minutes to 40 minutes to analyze its performance. To facilitate efficient processing and analysis of the videos, FFmpeg[1] which is a powerful multimedia framework is used. FFmpeg divides each video into smaller, more manageable segments. Its flexible command-line functionality allows us to control parameters such as segment length, codec, and frame rate. It also ensures that the integrity of the video remains intact while only splits the videos for faster and parallel processing. Hence, the ATM Surveillance Videos are divided into smaller clips ranging from 10 to 40 minutes each. This segmentation strategy allowed for more manageable data processing while retaining the integrity of the original footage. The clips were then partitioned in an 80:20 ratio for testing and training reasons. For the training purpose, frames were extracted from the clips, and a subset of 6600 frames was randomly selected for annotation using the MakeSense AI tool. This annotation process involved labeling frames as either relevant or irrelevant to the research objectives, specifically focusing on the presence of furniture and human subjects within the surveillance footage. Out of the annotated frames, 4500 frames were identified as relevant, containing instances of furniture or human subjects, while the remaining 2100 frames were deemed irrelevant. This labeled dataset acts as an outline for developing and testing DL models for human recognition tasks in ATM surveillance videos. Figure 3.17 illustrates the annotation process using the MakeSense AI tool, showcasing its user-friendly interface and annotation

[1]FFmpeg: A Complete, Cross-Platform Solution to Record, Convert and Stream Audio and Video. Available at: `https://ffmpeg.org/`

capabilities. MakeSense AI provides a user-friendly interface that allows users to upload images or videos and annotate them with bounding boxes, polygons, keypoints, and other annotations required for training machine learning models, such as object detection, segmentation, and pose estimation.

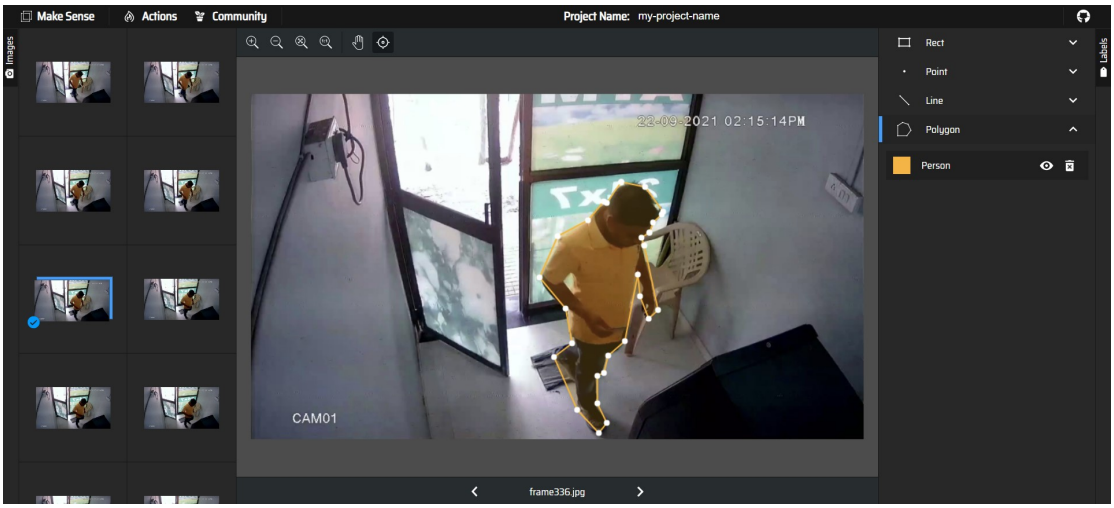


Figure 3.17: Person Annotation Process in Make Sense AI Tool for Dataset Preparation.

3.3.2 Methodology

Phase I: Dataset Preparation

i. *Surveillance data collection:* ATM surveillance footage serves as the use case for our research, and the dataset is not publicly accessible. The videos are collected and divided into smaller segments of duration 15 minutes (min) to 25 min using FFmpeg software.

ii. *Video to frame conversion:* Following the segmentation of the video into smaller segments, a total of 35 videos were obtained. Subsequently, 28 of these videos were employed for the training phase, while the remaining 7 videos were allocated for testing purposes. Initially, these videos underwent frame division utilizing the OpenCV library.

iii. *Data Annotation:* Afterward, a random selection process was utilized, for the retrieval of 6600 frames from the previously obtained frame pool. Among

these 6600 frames, 4500 frames were annotated as relevant frames and 2100 frames as irrelevant frames for the detection network using the Make Sense AI tool.

Phase II: Relevance Frame Classification

To identify the relevant and irrelevant frames of surveillance video, YOLOv9 and Mask R-CNN models are used in relevance frame classification stage. The details about the of both the models are given as follows:

i. *YOLOv9:* YOLOv9 [32], is an advanced real-time object detection model that builds upon the strengths of its predecessors which offer improved accuracy, speed, and efficiency. It introduces new features, such as Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN), to mitigate information loss during feature extraction. YOLOv9 utilizes standard convolutional layers instead of depth-wise convolutions for better parameter efficiency and gradient stability which makes it a significant upgrade from YOLOv8 [31]. The architecture of YOLOv9 is divided into three stages which are explained as follows:

   a. *Backbone:* The backbone in YOLOv9 extracts essential features from the input image. YOLOv9 employs a CNN architecture based on GELAN. This design optimizes parameter efficiency using:(i) Conventional convolutional layers, which improve representation learning and gradient stability and (ii) Cross-Stage Partial (CSP) blocks, which split and merge feature maps to enhance gradient flow. The feature map F is obtained using Eq. (3.16):

$$\mathbf{F} = f_{\text{backbone}}(\mathbf{X}), \tag{3.16}$$

   where X represents the input image, and $f_{\text{backbone}}(\cdot)$ is the transformation applied by the backbone. GELAN's efficient aggregation ensures maximum feature reuse and reduces redundant computations.

   b. *Neck:* The neck refines features for multi-scale detection by combining outputs from different levels of the backbone. YOLOv9 integrates PGI,

which includes: a main branch for efficient inference, an auxiliary reversible branch to preserve gradient precision and multi-level auxiliary information that enable hierarchical supervision. The combined feature map from multiple scales is given by Eq. (3.17):

$$F_{\text{neck}} = \text{Combine}(F_1, F_2, F_3), \tag{3.17}$$

where F_1, F_2, and F_3 are feature maps at different resolutions, and $Combine(\cdot)$ denotes feature aggregation using upsampling, downsampling, and lateral connections.

c. *Head:* The head generates predictions for bounding boxes, class probabilities, and confidence scores. Like other YOLO modules, the loss function is determined using Eq. (3.7). Figure 3.18 shows the output of YOLOv9 on ATM surveillance video to detect two people with a confidence score of 0.90 and 0.77 respectively.

Figure 3.18: Output of YOLOv9 Trained on ASV Dataset (Relevant Frame).

ii. *Mask R-CNN:* He *et al.* [33], introduced a novel, intuitive, flexible, and simple network to perform instance segmentation named as Mask R-CNN. The Mask R-CNN architecture embodies the essence of faster R-CNN while extending its

capabilities to do instance-level segmentation. Hence, Mask R-CNN performs object categorization, bounding box regression, and pixel-level segmentation simultaneously. Figure 3.19 gives architecture details of Mask R-CNN module. The architecture of Mask R-CNN is divided into four stages which are explained as follows:

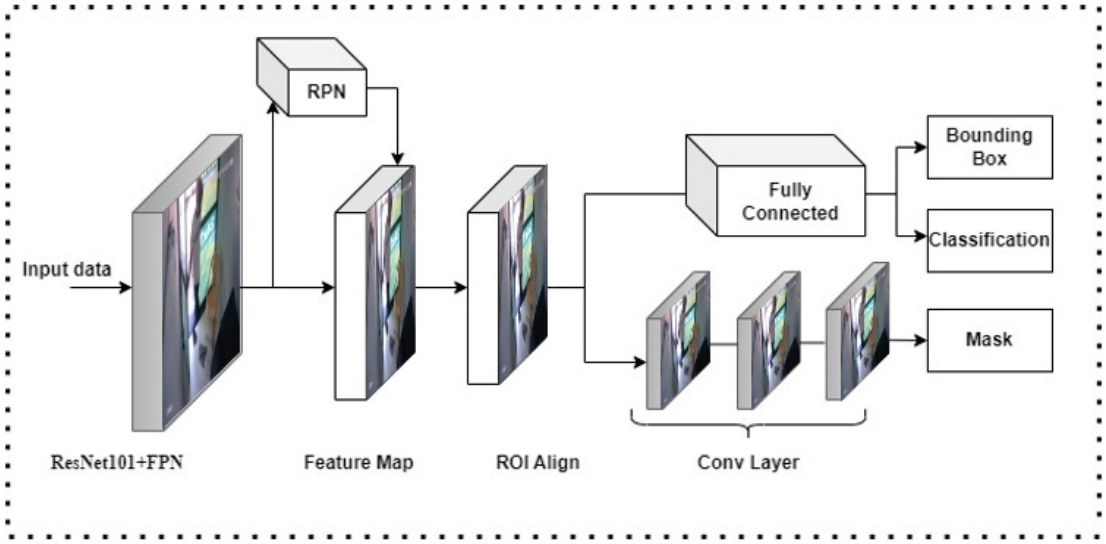


Figure 3.19: Architecture of Mask R-CNN Module [33].

a. *Feature Extraction :* Mask R-CNN used ResNet-50 / ResNet-101 architecture as a baseline to retrieve good characteristics from images [33]. This baseline is also responsible for capturing hierarchical features from the input image. The input image feds to ResNet-50/ ResNet-101 architecture where a series of convolutional layers extract poor and good quality features, which represent various aspects of the image, from edges to abstract object features. These feature maps provide the foundation for subsequent processing, including OD and instance segmentation. To address the issue of vanishing gradients, ResNet uses residual block connection which includes a "skip connection" to bypass one or more layers. This residual block contains CNN, BN, skip connection, and ReLU activation function. Similar to YOLOv5, Mask R-CNN uses FPN with the ResNet model to extract fined and coarse features at various scales. The following mathe-

matical steps represent the process of feature extraction:

Lets assume an input image $I \in \mathbb{R}^{H \times W \times 3}$, where H, W, and 3 are the height, width, and number of color channels, respectively, is fed into the backbone network. The backbone network applies a series of convolutional layers to the input image which is represented by Eq. (3.1). The FPN enhances the backbone features by combining low-resolution, semantically strong features with high-resolution as mentioned in Eq. (3.18):

$$P_i = g(F_i, P_{i+1}), \quad \forall i \in \{1, 2, \ldots, K\} \tag{3.18}$$

where:

- P_i: Pyramid feature map at level i,
- F_i: Backbone feature map at level i,
- $g(\cdot)$: Feature fusion function, usually element-wise addition or concatenation of upsampled higher-level features (P_{i+1}) with the backbone features.

b. *Region Proposal Network (RPN) :* Like Faster RCNN, Mask R-CNN generates RPN that examines the features generated from the backbone and proposes the potential regions of images that might contain objects. The RPN works using the feature maps created by the ResNet-50 network. For these regions, the RPN produces objectness scores and precise bounding box coordinates using anchor-box and non-max suppression approaches. As RPN generates region proposals based on the extracted features. The RPN applies a sliding window over the feature map P using Eq. (3.19)

$$R = \text{RPN}(P), \tag{3.19}$$

where R contains object proposals as bounding boxes and their objectness scores.

c. *Region-of-Interest Align :* After obtaining region proposals from the RPN,

the next step is Region-of-interest (RoI) Align. RoI Align is crucial for ensuring that the features within each region proposal are consistently aligned and resized to a fixed spatial dimension. For each proposed region, features are extracted using RoI Align Eq. (3.20):

$$F_{\text{RoI}} = \text{RoIAlign}(P, R), \tag{3.20}$$

where F_{RoI} represents the extracted region-specific feature map.

Faster R-CNN use RoI pooling for extracting fixed-size parameters from proposals. However, in Mask R-CNN, the RoI-align operation retrieves accurate and aligned features from the proposals, which addresses the issue of spatial misalignment among the feature map and the original image. RoI-align uses bilinear interpolation to retrieve feature values from the original feature map at non-integer locations rather than quantizing the RoI boundaries to the grid cells. Interpolation guarantees that the features are precisely aligned with the actual positions of the RoI's boundaries and spatial features are also preserved inside the RoI. Hence, using RoI alignment the loss of spatial information that occurs in RoI pooling is recoverable in Mask R-CNN and generates a binary mask for objects. Figure 3.20 shows the output of the Mask R-CNN model for relevant frames while Figure 3.21 denotes irrelevant frames. Using object detection modules, relevant and irrelevant frames of surveillance videos are separated [49, 145].

d. *Head :* The Head component is responsible for making predictions and is divided into two parallel branches: object classification and mask prediction. Hence the extracted F_{RoI} are fed into three branches for classification, mask prediction and bounding box regression as shown in Eq. (3.21).

$$B = f_{\text{bbox}}(F_{\text{RoI}}), \quad C = f_{\text{cls}}(F_{\text{RoI}}), \quad M = f_{\text{mask}}(F_{\text{RoI}}) \tag{3.21}$$

where B is the refined bounding box coordinates, C is the class label probabilities and M is the binary mask for the detected object. Object Classifi-

cation is responsible for classifying the objects within the region proposals. It utilizes dense layers and a softmax activation to assign a class label to each object, indicating what type of object is present. In mask Prediction, pixel-wise masks are predicted for the objects within the region proposals. A convolutional subnetwork produces binary masks that highlight the exact shape and location of each object. The final output for each detected object includes the predicted bounding box coordinates, the assigned class label, and an instance segmentation mask, enabling both object detection and detailed instance segmentation.

Figure 3.20: Output of Mask R-CNN Module Trained on ASV Dataset (Relevant Frame).

Phase III: Compression Module

In the video compression module of FRVC framework, processing is performed on the relevant frames of ATM surveillance which are identified using the object detection module. While the irrelevant frames are discarded from the network. On relevant frames, we determine the similarity index between frames across various threshold values and further categorize these frames into primary frames and similar frames. Primary frames are defined as frames that exhibit similarity when compared with their consecutive frames. If consecutive frames are identical, they

Figure 3.21: Output of Mask R-CNN Module Trained on ASV Dataset (Irrelevant Frame).

are categorized as similar frames. The first dissimilar frame is marked as the next primary frame. For instance, if the initial frame is similar to the second, third, fourth, and fifth frames but dissimilar to the sixth frame, the first and sixth frames are considered primary frames. The second, third, fourth, and fifth frames are designated as similar frames. Given that the sixth frame is dissimilar to the first frame, it is subsequently identified as the next primary frame. We determine the similarity between frames at various threshold values and save the count of the similar frames corresponding to the primary frame in a data structure. Lastly, the video is constructed using relevant primary frames, at the original FPS.

3.3.3 Workflow and Algorithm

The sequence flow diagram of the FRVC model is shown in Figure 3.22. Here, the framework is divided into the following steps:-

i. In the initial phase, namely "Dataset Preparation", the process begins with the collection of surveillance videos, and this video is further split into frames. Subsequently, annotations are applied to these frames.

ii. In the second phase named "Relevance Frame Classification", we trained the

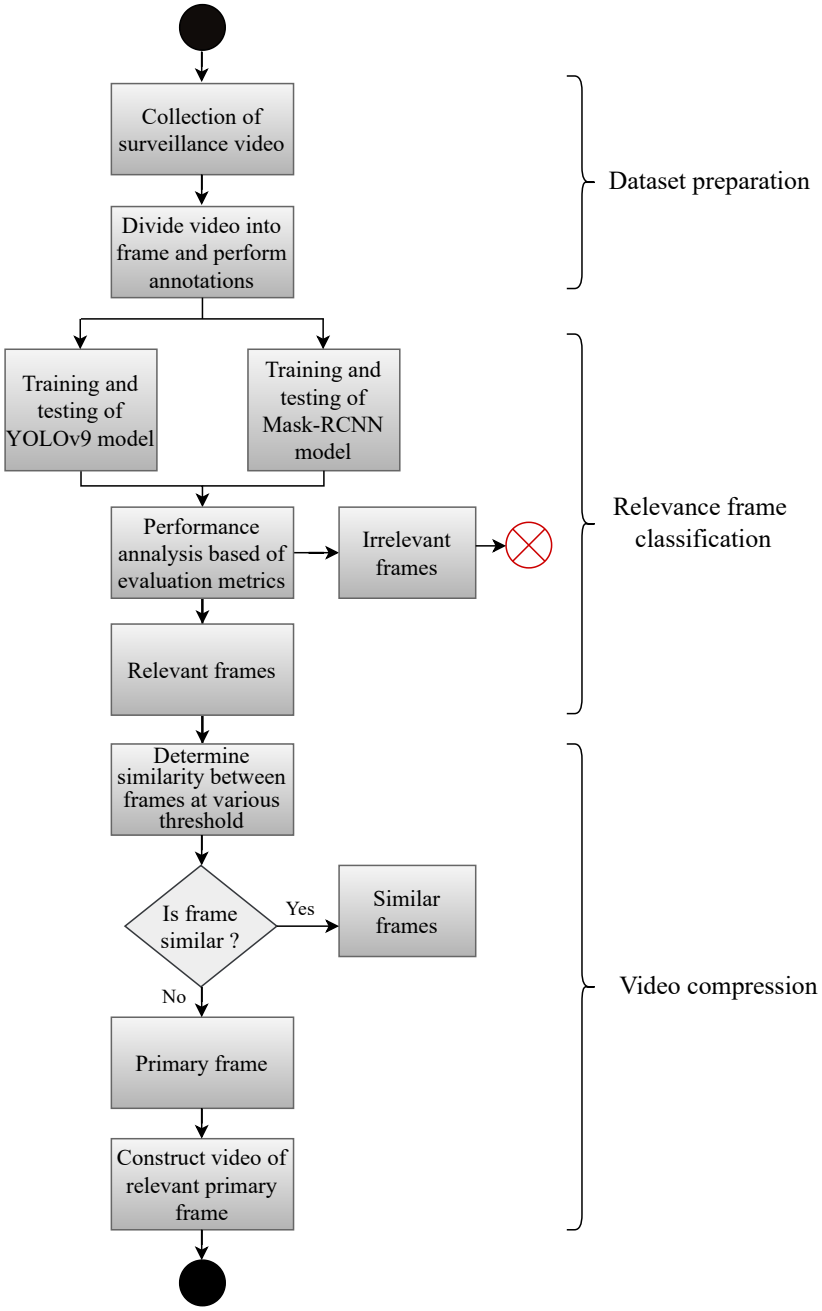


Figure 3.22: Sequence Flow Diagram of FRVC Model.

one-phase YOLOv9 and the two-phase Mask-RCNN on the ASV dataset. Subsequently, we employed these trained models to analyze the remaining ATM surveillance video as part of our testing process and predict the relevant and irrelevant frames of surveillance video. Depending upon the evaluation metrics and time required to identify the relevant and irrelevant frames, the best model for OD is determined.

iii. In the final stage, known as "Video Compression", relevant frames are taken as input, and depending upon similarity at various threshold values primary and similar frames are categorized. In the end, the compressed video is constructed using relevant primary frames at the original video's FPS value.

Algorithm 3 demonstrates the step-by-step execution of the proposed FRVC model:-

i. The algorithm begins by determining the FPS of the input video. Later, the frames are extracted from the video.

ii. Then various variable lists are initialized to store the data. For e.g., Relevant_Frames is an empty list created to store frames that are considered relevant. Primary_Frames is an empty list to save primary frames, and Similar_Frames is an empty list to store similar frames. Next_Primary is initialized with a value of 1 and will be used to track the next primary frame. Similarity_Threshold is an initial value used as a reference for measuring the similarity of frames at various threshold values (100%,98%,96%,94%,92%,90%).

iii. The algorithm goes through all the frames of the video. and apply two object detection algorithms: Mask-RCNN and YOLOv9, to identify objects within the frame. It checks if the frame is relevant based on the detected objects. If a frame contains relevant objects (*e.g.*, ., person), it adds the frame to the relevant frames list. If not, it discards the frame as it is not considered relevant.

iv. For each relevant frame in the relevant frames list, the algorithm compares the frame's similarity with the next frame in the list. If the frame's similarity with the next frame is above the similarity threshold, it is classified as a similar frame and added to the similar frames list. If the similarity is below the threshold, the frame is classified as a primary frame and added to the primary frames list. Additionally, the next primary variable is updated to keep track of the current primary frame. The count of the similar frame corresponding to the primary frame is maintained in an Excel file.

Algorithm 3: FRVC Algorithm

Input: ATM surveillance video
Output: Compressed surveillance video

1. Determine the FPS of the video and extract frames from the input video.
2. Initialize an empty list for relevant frames: Relevant_Frames = []
3. Initialize an empty list for primary and similar frames :
   Primary_Frames = [] and Similar_Frames = []
4. Initialize a variable $Next\_Primary$ to 1
5. Initialize a variable $Similarity\_Threshold$ to any value between 90-100
6. Initialize an empty dictionary $Frame\_Counts$
7. **for** each frame in the video **do**
   Apply Mask-RCNN and YOLOv9 object detection to identify objects
   if frame is relevant **then**
   Add the frame to the relevant frames list:
   Relevant_Frames.append(frame)
   else
   Discard the frame
   end if
8. **end for**
9. **for** each relevant frame in the relevant frames list **do**
   if frame similarity with the next frame is above $Similarity\_Threshold$ **then**
   Add the frame to the similar frames list:
   Similar_Frames.append(frame)
   else
   Add the frame to the primary frames list:
   Primary_Frames.append(frame)
   $Next\_Primary \leftarrow$ current frame
   end if
   Save $Frame\_Counts$ in an excel file
10. **end for**
11. Construct the compressed video using relevant primary frames at the original FPS

v. Finally, the algorithm constructs the compressed surveillance video using the relevant primary frames at the original FPS. This step creates a new video

that retains important frames while removing less significant ones, effectively compressing the video. After the construction of the compressed video, the data structure is stored on the disk.

3.4 Summary

This chapter introduces three surveillance video compression techniques that are designed to optimize storage. The first technique, ODSC Model, uses YOLOv5, YOLOv7, and YOLOv8 frameworks to identify relevant frames containing humans or animals. These frames are retained and compressed, while irrelevant frames are discarded. The second approach, D&C Model, incorporates YOLOv5, YOLOv7 YOLOv8 and YOLOv9 to classify the relevant frame. Later, the similarity between frames at various thresholds is determined and depending upon similarity relevant frames are further divided into primary and similar frames. Lastly, primary frames are used to construct an efficient compressed video. In the end, the third approach i.e. FRVC model is trained on the domain-specific ATM Surveillance Video dataset and employs YOLOv9 as well as Mask R-CNN for frame categorization. The compression module in FRVC is similar to that of D&C model. The key differences among the three models are: In the ODSC and D&C models, the COCO dataset was used for training and the ASV dataset was employed for testing. In contrast, the FRVC model utilized the ASV dataset for both training and testing. Secondly, the object detection techniques differ across the models. The ODSC model employed YOLOv5, YOLOv7, and YOLOv8, whereas the D&C model extended this by incorporating YOLOv9. However, the FRVC model uses Mask R-CNN and YOLOv9 module to detect the relevant and irrelevant frames. Lastly, the approach to handle video compression varies. In the ODSC model, relevant frames were simply merged to form the compressed video. On the other hand, the D&C and FRVC models introduced the concept of a similarity index to further refine the compression process.

Chapter 4

RESULT ANALYSIS AND COMPARISON

The essence of video compression lies in its ability to reduce the redundancy inherent in video signals. Unlike still images, which can be compressed through spatial redundancy reduction techniques such as JPEG, videos exhibit temporal redundancy—redundancy between consecutive frames—as well as spatial redundancy within each frame. By exploiting these redundancies, video compression algorithms can achieve significant reductions in file size and preserve essential visual information. This chapter presents the result analysis of three proposed approaches for surveillance video compression. It begins by introducing the evaluation parameters used to assess the performance of various object detection and compression modules. Subsequently, the performance of these modules is compared based on these parameters. The most effective object detection module, identified through this comparison, is then utilized in the compression module. Additionally, the three proposed models are evaluated against each other, and finally, the best-performing model is compared with state-of-the-art approaches.

4.1 Evaluation Metrics

Evaluation metrics are quantitative measures used to assess the performance of models. They provide insights into accuracy, efficiency, and reliability, guiding improvements and comparisons. In research and applications, evaluation metrics are crucial for validating results and aligning outcomes with defined goals.

4.1.1 Evaluation Metrics in Object Detection

In the context of object detection, evaluation metrics determine how well the models detect objects a classify frames as relevant or irrelevant. The various evaluation metrics used in OD are elaborate as follows:

i. *Confusion Matrix:* Confusion matrices are used to summarize the results of predictions made for several classes to evaluate the performance of OD modules. The confusion matrix has four prediction values that are False Positive (FP), True Positive (TP), False Negative (FN), and True Negative (TN) which are shown in Table 4.1.

Table 4.1: Confusion Matrix

	Determined Positive	Determined Negative
Real Positive	TP	TN
Real Negative	FP	FN

ii. *Precision:* Precision measures the ratio of accurately detected objects to all objects detected as mentioned in Eq. (4.1).

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

iii. *Recall:* The proportion of accurately identified objects to all ground truth objects as shown in Eq. (4.2).

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

iv. *F1-Score:* The harmonic mean of recall and precision is known as the F1-Score. It offers a balance between recall and precision, which can be very helpful in situations where there are class imbalances as mentioned in Eq. (4.3).

$$\text{F1-Score} = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{4.3}$$

v. *mAP:* mAP is the Average Precision (AP) values across different classes as shown in Eq. (4.4). It provides an overall assessment of the algorithm performance across various object categories.

$$\text{mAP} = \frac{1}{N} \sum_{c=1}^{N} \text{AP}_c \tag{4.4}$$

where N is the number of classes, and AP_c is the Average Precision for class c.

vi. *Accuracy:* Accuracy measures correctly predicted objects and balances true positives, false positives, and false negatives as noted in Eq. (4.5). In other terms, it is a ratio of correct predictions (true positives and true negatives) to the total number of predictions.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Predictions}} \tag{4.5}$$

4.1.2 Evaluation Metrics in Compression

Evaluation metrics for surveillance video compression assess the effectiveness of reduced video size while maintaining critical visual and informational quality. The evaluation metrics include:

i. *Compression Ratio (CR):* The compression ratio is a measure of the video size reduction achieved through compression. It is defined as the ratio of the original file size to the compressed file size mentioned in Eq. (4.6). A higher compression ratio indicates greater compression efficiency.

$$\text{Compression Ratio (CR)} = \frac{\text{Original Size}}{\text{Compressed Size}} \tag{4.6}$$

ii. *Percentage of Compression Achieved (%):* The percentage of compression achieved quantifies the reduction in data size as a percentage. It is calculated using Eq. (4.7). This metric indicates how much of the original data size has been eliminated during compression.

$$Compression(\%) = \frac{\text{Original Size} - \text{Compressed Size}}{\text{Original Size}} \times 100 \tag{4.7}$$

iii. *Average Compression Achieved for Videos:* The average compression achieved is the mean percentage of compression across multiple videos and calculated using Eq (4.8).

$$\text{Average \% of Compression} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\text{Original Size}_i - \text{Compressed Size}_i}{\text{Original Size}_i}\right)\times 100 \tag{4.8}$$

where

- i**:** is the index of a specific video, ranging from 1 to n. For example, $i = 1$ refers to the first video, $i = 2$ to the second video, and so on.
- n**:** represents the total number of videos being analyzed. It is the total number of videos included in the computation.

4.2 Object Detection Based Surveillance Video Compression Model

4.2.1 Comparison of Object Detection Modules

In the Object Detection based Surveillance Video Compression (ODSC) model, ATM footage is given as input to YOLO modules to differentiate relevant and irrelevant frames of video. Table 4.2 represents the characteristics of seven different tested ATM surveillance videos in terms of FPS, size, resolution of frames and total frames in the video.

Table 4.2: Original Characteristic of Seven Tested Videos

Video No.	**FPS**	**Resolution**	**Size**	**Total Frames**
Video1.mp4	15	1280 × 720	527 MB	269996
Video2.mp4	15	1280 × 720	700 MB	35990
Video3.mp4	15	1280 × 720	260 MB	13477
Video4.mp4	15	1280 × 720	142 MB	7290
Video5.mp4	15	1280 × 720	250 MB	12769
Video6.mp4	15	1280 × 720	264 MB	13275
Video7.mp4	15	1280 × 720	265 MB	13492

To implement YOLOv5, YOLOv7, and YOLOv8, Google Collab Pro is used since these models demand GPU resources and are computationally demanding [148]. In Table 4.3, the parameter specification needed to implement YOLO modules is listed. A redesigned darknet architecture known as CSP-DarkNET serves as the backbone network for YOLOv5. Cross-Stage Partial connections are among their characteristics, which enhance efficiency and performance. For training, YOLOv5 has a learning rate of 0.001, over 80 epochs. During training, a batch size of 16 is frequently used. A multi-task loss function is used by YOLOv5, which combines localization, class, and objectness losses. YOLOv5 uses the swish activation function which is shown in Eq. (4.9). The function $f(x)$ is defined as:

$$f(x) = x \cdot \text{Sigmoid}(\beta \cdot x) \tag{4.9}$$

Where x is input to function, β is hyperparameter, and sigmoid is another activation function represented in Eq. (4.10). YOLOv8 module also used the Eq. (4.10). while YOLOv7 uses the LeakyRelu function shown in Eq. (4.11).

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{4.10}$$

$$\text{LeakyReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{otherwise} \end{cases} \tag{4.11}$$

YOLOv7 employs EELAN as its backbone network to enhance the performance of OD. It has a lower learning rate of 0.0002, which helps fine-tune the

Table 4.3: Parameter Specification of ODSC Model

Parameter	**YOLOv5**	**YOLOv7**	**YOLOv8**
Backbone Network	CSP-DarkNET	EELAN	C2F
Learning Rate	0.01	0.02	0.01
Epoch	60	60	60
Batch Size	16	16	16
Loss Function	Multi-task loss	Multi-task loss	Multi-task loss
GPU	Yes	Yes	Yes
Activation function	Swish	Leaky ReLU	Sigmoid

model with more precision during training with 60 epochs and 16 batch sizes. The specific loss functions in YOLOv7 may include Localization Loss, Objectness Loss, Classification Loss, Coarseness Loss, and Fineness Loss. For YOLOv8 C2F is used as the backbone with 0.01 learning rate having 16 batch size and 60 epochs. Using the ATM surveillance dataset, this thesis evaluated the performance of YOLOv5, YOLOv7, and YOLOv8. The results are shown in Table 4.4.

Table 4.4: Comparison of YOLOv5, YOLOv7 and YOLOv8 Modules

Parameter (%)	**YOLOv5**	**YOLOv7**	**YOLOv8**
Precision	97.6	98.6	**98.5**
Recall	97.9	98.2	**97.2**
F1-Score	97.1	97.3	**97.3**
Accuracy	98.3	98.5	**98.7**

With a high precision of 98.7%, the YOLOv8s module outperforms YOLOv7 and YOLOv5, indicating that a significant number of its positive predicts are correct. YOLOv7 and YOLOv8 have recall rates of 98.2% and 97.9%, respectively, demonstrating their efficacy in recording the majority of real positive outcomes. For YOLOv7 and YOLOv8, the precision-recall-balanced F1-score is 98.1%, indicating a well-balanced trade-off between accuracy and completeness. YOLOv8s has an accuracy of 99.7% when it comes to overall correctness, meaning that most of its predictions—both positive and negative—are accurate. However, YOLOv7-tiny shows 99.5% for YOLOv5 and 98.3% for YOLOv7. YOLOv7 and YOLOv8 models give similar promising performance but the accuracy of YOLOv8 is 0.2% greater than YOLOv7. Figure 4.1 shows the graph of time taken by YOLOv5, YOLOv7 and YOLOv8 modules to detect relevant and irrelevant frames. Though

both models exhibit better accuracy, the time taken to detect relevant frames and irrelevant frames using YOLOv8 is far better as compared to YOLOv7. YOLOv5 module takes the maximum time to detect the relevant frames while YOLOv8 takes the least time for the same task. Hence, for further processing, the output of the YOLOv8 module is used.

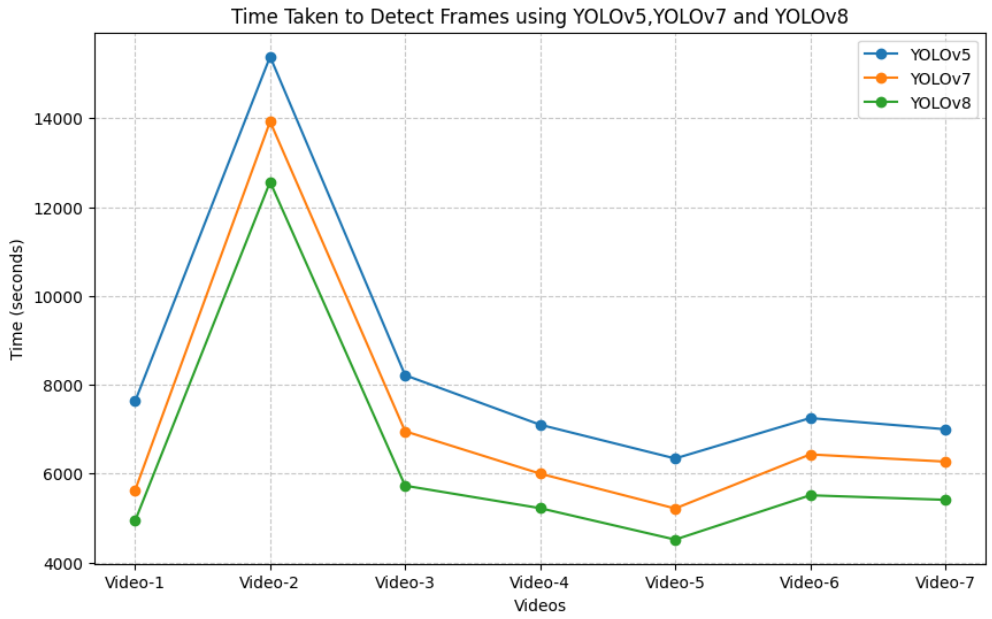


Figure 4.1: Time Taken to Detect Relevant and Irrelevant Frames of Surveillance Video using YOLOv5, YOLOv7 and YOLOv8 Module in ODSC Model.

Table 4.5: Frame Count Using YOLOv8 Model

Clip Name	**Total Frames**	**Relevant Frame**	**Irrelevant Frame**	**Time Taken to Detect Frames**
Video1.mp4	26996	2748	24248	4938 Sec
Video2.mp4	35990	5712	30278	12566 Sec
Video3.mp4	13477	9563	3914	5728 Sec
Video4.mp4	7290	4760	4530	5222 Sec
Video5.mp4	12769	75	12694	4515 Sec
Video6.mp4	13275	2824	10451	5512 Sec
Video7.mp4	13492	2734	10758	5410 Sec

Table 4.5 denotes the output of YOLOv8 on seven ATM surveillance videos where the first video1 contains total 26996 frames, out of which 2748 frames are recognized as relevant and 24248 recognised as irrelevant frames in 4938 Seconds

(Sec). The second video has 35990 total frames where 5712 frames are detected as relevant and 30278 are irrelevant 12566 Sec. As video2 has the highest number of frames, it takes the longest time to detect relevant frames. Similarly, the complete categorization of all videos regarding relevant and irrelevant frames is mentioned in the above Table 4.5.

4.2.2 Result Analysis of Compression Module

In this module of ODSC framework, the output of the YOLOv8 module is used to perform compression. Compression is achieved by selectively preserving the relevant frames in surveillance videos and irrelevant frames are deleted. Hence, the relevant frames of surveillance video are merged at the FPS value of the original video using a proper video codec. Table 4.6 presents an analysis of the compression module where compression ratio denote the reduction in the size of video. For

Table 4.6: Output of ODSC Model with Resolution 1280 × 720 on YOLOv8

Video No.	**Original Video Size (MB)**	**Compressed Video Size (MB)**	**% of Compression Achieved**	**Compression Ratio**
Video1.mp4	527	104.2	80.23%	5.06
Video2.mp4	700	100.1	85.7%	7.00
Video3.mp4	260	183.3	29.5%	1.42
Video4.mp4	142	117.1	17.53%	1.21
Video5.mp4	250	2.4	99.04%	125
Video6.mp4	264	148.4	43.78%	1.78
Video7.mp4	265	157.6	40.52%	1.68
Average % Compression	–	–	66.23%	–

instance, the size of video1 is 527 MB and after passing through the ODSC model the size of the video is reduced to 104.2 MB which achieved 80.23% compression by maintaining the same resolution of frames as that of the original video. Hence, using the ODSC Model size of the compressed video is 5 times smaller than the original size which is mentioned in the last column of the table. Similarly, the video2 achieved 85.70% compression and the size of compressed video is seven

time smaller than the original size. However, the third video gained to 29.50% compression, and video4 achieved 17.53% of compression. The video5 achieved the highest % of compression as the video contained only 75 relevant frames among 12769 frames. The video6 gained 43.78% of compression while the video7 achieved 40.52% compression. The overall average % of compression achieved using the ODSC model is 66.23%. Figure 4.2 shows a graphical representation of Original and Compressed Video Sizes. The proposed ODSC model facilitate efficient storage and transmission of video content without loss in quality.

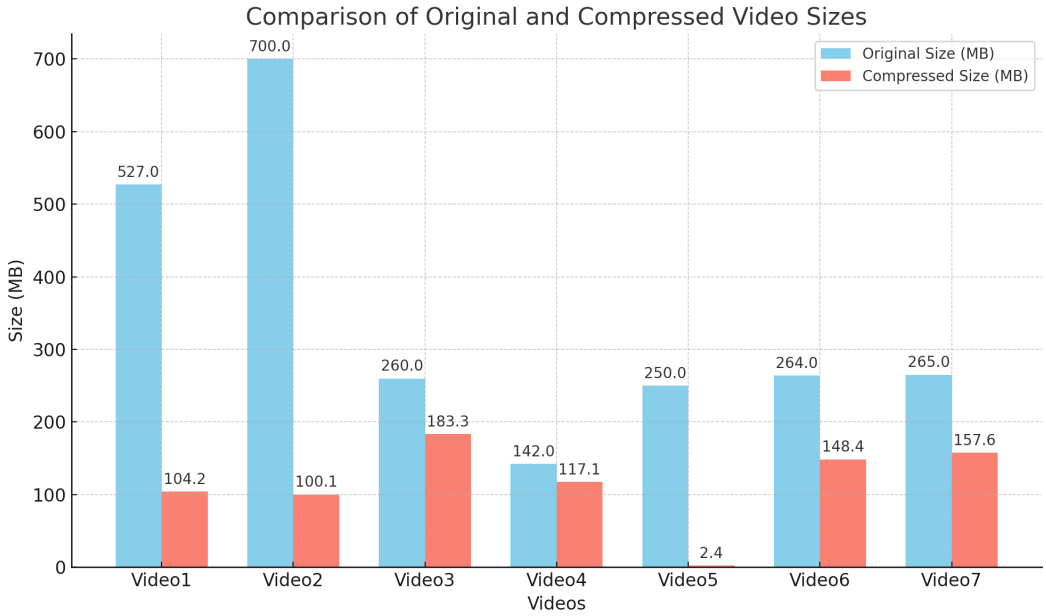


Figure 4.2: Graphical Representation Original and Compressed Video Sizes Across Seven Videos of ODSC Model.

4.3 Deep Learning-Based Relevant Video Frame Detection and Compression Model

4.3.1 Comparison of Object Detection Modules

This section entails the details of training and testing of OD modules with parameter specifications and comparisons among them. To train and test YOLO OD modules of D&C Model, we used two datasets *(i)* COCO dataset [142] and *(ii)* ATM Surveillance Video. The COCO dataset consists of 80 classes, is used to train OD modules while ATM surveillance video is used to test the modules. To detect the relevant frames in surveillance video we used only one class (1: person) of the COCO dataset. Through the training, neural network learns to extract informative features from the input images that are relevant for detecting persons. During inference, it applies the learned features to new video frames to perform OD. It splits the source image into a grid of cells and assigns BB, confidence scores, and class probabilities to each cell. By applying a confidence score threshold, typically set to a predefined value (e.g., 0.5), OD modules filter out low-confidence detections, retaining only those with confidence scores above the threshold.

Table 4.7: Original Characteristic of Six Tested Videos.

Video No.	**Scenario**	**FPS**	**Resolution**	**Size**	**Total Frames**
Video-1.mp4	Scenario-I	24	1920×1080	607 MB	48945
Video-2.mp4	Scenario-I	24	1920×1080	317 MB	28896
Video-3.mp4	Scenario-I	24	1920×1080	924 MB	359553
Video-4.mp4	Scenario-II	24	1920×1080	1.44 GB	39001
Video-5.mp4	Scenario-II	24	1920×1080	910 MB	23049
Video-6.mp4	Scenario-II	24	1920×1080	1.10 GB	28799

Table 4.7 represents the original characteristic of surveillance video. All videos contain High Definition (HD) standard resolution 1920×1080 and 24 FPS. The last column shows the total number of frames present in the video, to detect the relevant and irrelevant frames of surveillance video. For experimental purposes, we consider six videos for testing of YOLO modules. These six videos are considered under two different scenarios: *(i)* scenario-I:- urban areas and *(ii)* scenario-II:-

rural areas. Video-1.mp4, video-2.mp4 and video3.mp4 represent urban scenarios while video-4.mp4, video-5.mp4, and video-6.mp4 denote rural scenarios. The use of ATM in urban areas is maximum as compared to rural areas. Urban areas typically experience higher traffic flow, dense crowds, and greater activity. In case of ATM usage or road traffic, the frequency of events captured by surveillance cameras in urban settings is much higher. This results in a larger number of relevant frames that need to be processed and stored. While rural areas have less dynamic activity, the count relevant frames that are important for analysis are fewer in number. So, we consider this two scenarios to analyze the count of relevant frames in ATM surveillance video. YOLO modules are trained on the COCO dataset with a learning rate of 0.0001, using Adam optimizer for 60 epochs having batch size 16.

Table 4.8: Comparison of YOLO Modules

Module	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
YOLOv5	98.3	97.6	97.3	97.4
YOLOv7	98.5	98.3	98.2	98.3
YOLOv8	98.7	98.5	97.2	96.9
YOLOv9	**99.1**	**98.7**	**98.5**	**98.3**

Table 4.8 compares the performance of YOLO modules to detect the relevant and irrelevant frames of video using various evaluation metrics. YOLOv9 surpasses the other YOLO modules with respect to accuracy and other evaluation metrics like time taken to detect frames. YOLOv9 exhibit the highest 99.1% accuracy, 98.7% precision, 98.5% recall, and 98.3% F1-score. Figure 4.3 shows the graph of time taken by YOLOv5, YOLOv7, YOLOv8 and YOLOv9 modules to detect relevant and irrelevant frames. As YOLOv9 achieves the highest accuracy and the least time to detect the frames, the output of YOLOv9 is used in the next phase to find the similarity between relevant frames.Table 4.9 provides the count and time taken to detect the relevant and irrelevant frames of surveillance videos using YOLOv9 module. video1 takes a maximum time of 6761 Sec as it contains the highest number of 48945 frames, out of which 10845 frames are detected as relevant and 38100 as irrelevant and video7 takes the least time, i.e. 3150 Sec as it

contains least 23049 frames, where 7649 frames are detected as relevant and 15400 as irrelevant frames.

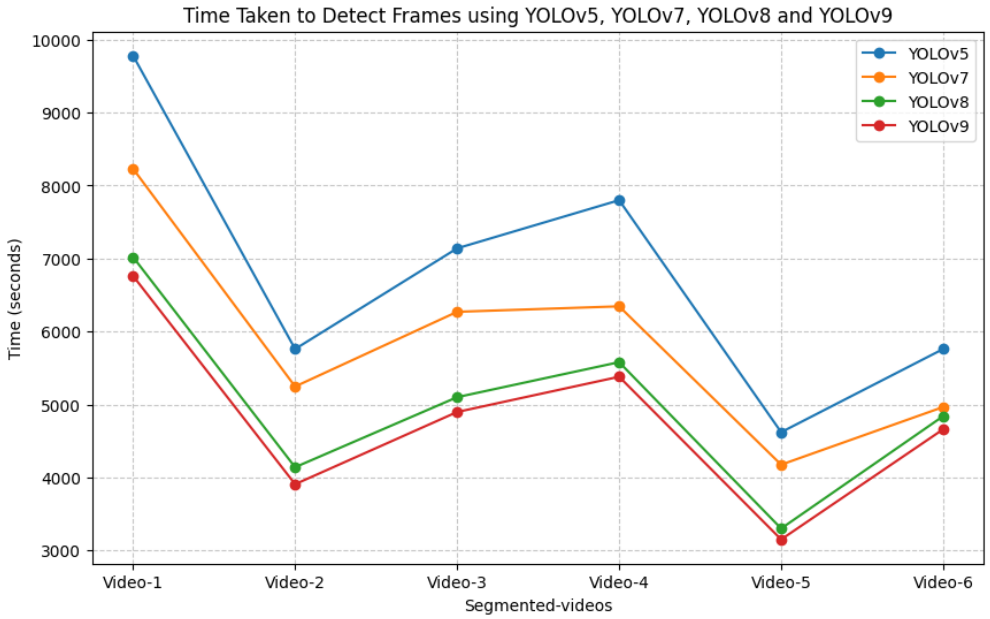


Figure 4.3: Time Taken to Detect Relevant and Irrelevant Frames of Surveillance Video using YOLOv5, YOLOv7, YOLOv8 and YOLOv9 Module in D&C Model.

Table 4.9: Frames Count Using YOLOv9 Module

Video No.	**Total frames**	**Relevant Frames**	**Irrelevant Frames**	**Time taken to detect frames**
Video-1.mp4	48945	10845	38100	6761 Sec
Video-2.mp4	28896	16819	12077	3907 Sec
Video-3.mp4	35953	27350	8603	4897 Sec
Video-4.mp4	39001	7913	31088	5380 Sec
Video-5.mp4	23049	7649	15400	3150 Sec
Video-6.mp4	28799	1289	27510	4660 Sec

4.3.2 Result Analysis of Compression Module

For the proposed D&C model, six ATM surveillance videos differ from the ODSC model are considered under two different scenarios for testing purposes. In this phase, the relevant frames detected through the YOLOv9 module are used to find the similarity between consecutive frames at various Threshold (TH) values ranging from 100% to 90%. The first frame is by default considered as a primary frame

Table 4.10: Count of Primary and Similar Frames at Various Threshold Values for Tested Six Videos of D&C Model

Threshold (%)	**Metric**	**Video-1.mp4**	**Video-2.mp4**	**Video-3.mp4**	**Video-4.mp4**	**Video-5.mp4**	**Video-6.mp4**
100	Primary Frame	6993	14350	18910	7010	6635	875
	Similar Frame	3852	2470	8440	2903	3019	414
98	Primary Frame	4061	10050	14405	6345	6129	651
	Similar Frame	6784	6769	12945	3568	4720	638
96	Primary Frame	2815	6432	10940	4784	4839	413
	Similar Frame	7178	10387	16410	5065	5078	876
94	Primary Frame	1669	3625	7950	2850	3721	325
	Similar Frame	8416	13067	19400	5063	5098	964
92	Primary Frame	1018	2210	6825	2321	2854	210
	Similar Frame	9027	14609	20525	5592	5399	1079
90	Primary Frame	690	1367	5430	1720	1957	159
	Similar Frame	9155	15152	21920	6193	5692	1130

and at different TH values corresponding similar frames are detected. Table 4.10 shows the count of primary and similar frames at different thresholds and Figure 4.4 graphically represents the frame count of each video. For instance, the original size of video1.mp4 is 607 MB, and it has a total of 10845 relevant frames, at Th=100%, 6993 frames are recognized as primary frames while 3852 are similar frames and the size of compressed video is 252.8 MB. For TH=98%, 4061 frames are primary frames, 6784 are similar frames and the size of compressed video is 162.9 MB. At TH=96% 2815 frames are primary frames, and 7178 frames are similar. For this threshold, suddenly the count of similar frames increases, and the count of primary frames reduces as a result the size of the video is compressed to 124 MB. At TH=94% 1669 frames are primary frames, 8416 frames are similar and the compressed video size is 97.6 MB. At TH=92% 1018 frames are primary frames, 9027 frames are similar and compressed video has a size of 64.9 MB. We achieved the highest compression of 92.68% for video1.mp4 at TH=90% where only 690 frames are primary frames and 9155 are similar. The size of the compressed video at this threshold value is 44.4 MB which is 13.67 times smaller than the original video as mentioned in Table 4.11. It is observed that, videos with fewer overall frames, such as Video-6, show a proportionally smaller reduction in primary frames, dropping from 875 at 100% to 159 at 90%, while similar frames rise from 414 to 1,130. These results illustrate the model's adaptability to different video sizes and content complexities. Additionally, videos with more dynamic content, like Video-3, exhibit larger reductions in primary frames compared to more static videos like Video-6, further emphasizing the role of content variability in compression efficiency. After analyzing six videos, a similar frame count increases after TH=96% and the primary frame count reduces. Hence, the size and duration of the video are greatly reduced after TH=96% using the proposed D&C model. This data also indicates a consistent trend where a lower threshold results in a significant reduction in primary frames and a corresponding increase in similar frames, demonstrating the trade-off between precision and compression efficiency.

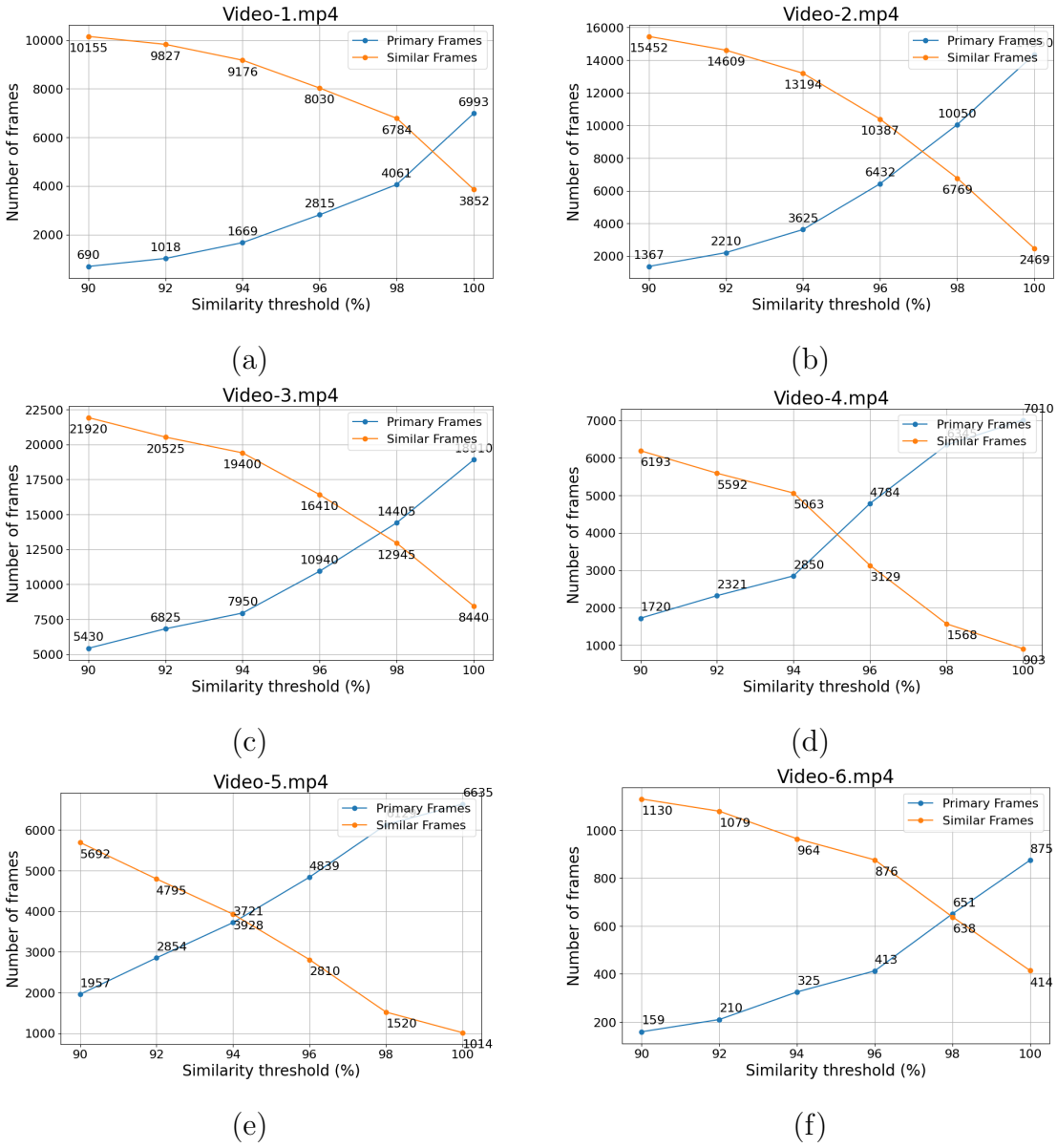


Figure 4.4: Count of Frames vs. Similarity Threshold in D&C Model.

Table 4.11 shows the % of compression achieved at threshold value (TH) = 90, FPS = 15 and of resolution = 1920×1080 along with CR which are determined using Eq. (4.6). For all six videos, the highest % of Compression is achieved at the threshold value = 90. For scenario I, which represents urban areas with high ATM usage, video1, video2, and video3 exhibit less % of compression compared to rural areas. Video1 captures intermittent transaction activities by 7-8 individuals over time. As a result, it achieves a compression rate of 92.68%. Video2 contains sporadic entries of one or two individuals into the ATM room seven times.

Table 4.11: Compressed Video Characteristics at Threshold Value = 90, FPS = 15, and Resolution = 1920×1080 for D&C Model

Video No.	**Scenario**	**Original Video Size**	**Compressed Video Size**	**% of Compression**	**Compression Ratio (CR)**
Video-1.mp4	Scenario-I	607 MB	44.4 MB	92.68%	13.64
Video-2.mp4	Scenario-I	317 MB	99.4 MB	68.64%	3.18
Video-3.mp4	Scenario-I	924 MB	206.1 MB	77.70%	4.48
Video-4.mp4	Scenario-II	1.44 GB (1440 MB)	23.4 MB	98.48%	61.54
Video-5.mp4	Scenario-II	910 MB	45.9 MB	94.95%	19.82
Video-6.mp4	Scenario-II	1.10 GB (1100 MB)	11.7 MB	98.96%	94.02

As a result, frames are less similar and the quantity of primary frames also reduces. Hence, it achieves the lowest compression of 68.64% among the urban scenario videos. While video3, spanning 25 minutes, features continuous transaction activities of 10-12 individuals. Consequently, it achieves a compression rate of 77.70%. In contrast, rural areas, denoted by video4, video5, and video6, exhibit lower ATM usage, leading to higher compression rates. Notably, video6 attains the highest compression rate of 98.96% due to minimal ATM activity in rural areas. Among the videos of scenario I, video1 received the highest CR, which means that compressed video is 13.64 times smaller than the original and in the case of scenario II, video6 gained the maximum compression ratio. Here, the size of the compressed video is 94 times smaller than the original. From this analysis, it is observed that Scenario II achieves significantly higher compression ratios and file size reductions, which could be beneficial for applications where storage space is a priority. Scenario I offers a more moderate approach, making it suitable for cases where a balance between compression and quality is necessary. The variation in compression ratios across videos suggests that the content type and complexity of the video play a crucial role in determining how effectively it can be compressed.

Table 4.12: Compression Achieved in MB at Various Threshold (TH) Values

Video No.	**Original size**	**TH = 100**	**TH = 98**	**TH = 96**	**TH = 94**	**TH = 92**	**TH = 90**
Video-1.mp4	607 MB	252.8 MB	162.9 MB	124.0 MB	97.6 MB	64.9 MB	44.4 MB
Video-2.mp4	317 MB	304.7 MB	275.1 MB	236.8 MB	202.5 MB	145.0 MB	99.4 MB
Video-3.mp4	924 MB	759.2 MB	732.2 MB	588.1 MB	401.1 MB	308.4 MB	206.1 MB
Video-4.mp4	1.44 GB	204.8 MB	119.3 MB	84.5 MB	57.7 MB	37.6 MB	23.4 MB
Video-5.mp4	910 MB	257.7 MB	186.0 MB	131.5 MB	95.8 MB	70.2 MB	45.9 MB
Video-6.mp4	1.10 GB	39.5 MB	29.3 MB	23.9 MB	19.2 MB	14.4 MB	11.7 MB
Average % of C	——	**54.33%**	**60.68%**	**70.53%**	**76.87%**	**82.83%**	**85.92%**

Table 4.12 shows the size of compressed video at different TH values ranging from 100% to 90% along with the average compression percentage (C) for each threshold value using Eq. (4.8). For example, Video1 has an initial size of 607 MB, which becomes 252.8 MB at TH = 100% and progressively shrinks to 44.4 MB at TH = 90%. Similarly, video2 starts at 317 MB and reduces to 99.4 MB at TH = 90%. Larger videos, like Video4 with an original size of 1.44 GB, exhibit a significant reduction down to 23.4 MB at the lowest threshold. It is noted that, at TH = 100%, the average compression is 54.33%, and it rises as the threshold decreases. At TH = 90% proposed D&C achieves 85.92% average compression. Figure 4.5 shows a fall in the size of the videos at various threshold values. From this, it is clear that as the threshold decreases, the compression and average compression performance improves at lower thresholds, with the most significant reductions occurring at TH = 90%. This pattern suggests that the D&C model is more effective in compressing videos with larger file sizes, as these tend to have more redundant frames that can be eliminated without significant loss of content.

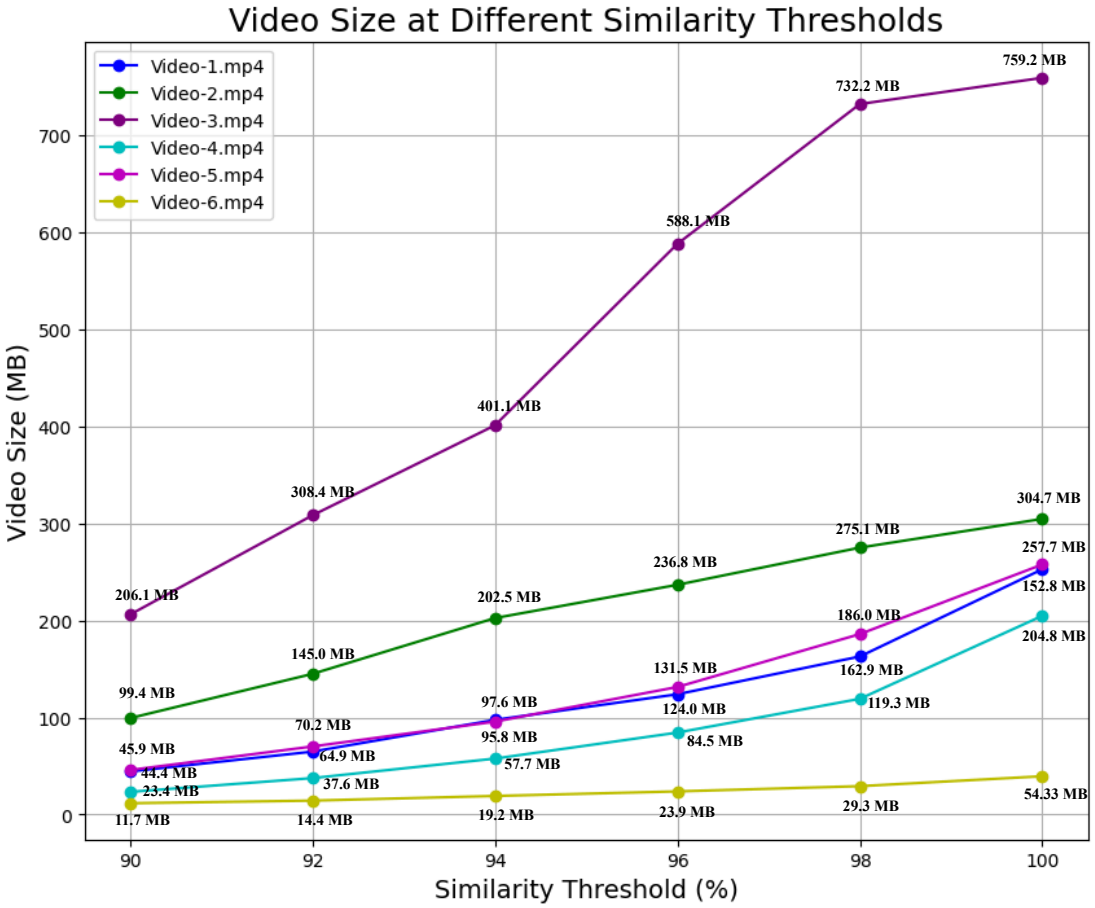


Figure 4.5: Size of Compressed Videos at Different Threshold of D&C Model.

4.4 Frame Relevance Based Video Compression Model

4.4.1 Comparison of Object Detection Modules

The proposed FRVC algorithm is applied to seven ATM surveillance videos, which are also utilized in the ODSC model. Table 4.13 summarizes the key characteristics of surveillance video files. The table systematically lists pertinent details for each video, which includes the video number, time of capture termed as period (day or night), FPS, resolution, size and total frames in the respective video. The initial five videos comprise day-time surveillance video, whereas the last two videos feature night-time surveillance.

Table 4.13: Original Characteristic of Seven Tested Videos

Video No.	**Period**	**FPS**	**Resolution**	**Total Frames**	**Size**
Video-1.mp4	Day	15	1920×1080	26996	527 MB
Video-2.mp4	Day	15	1920×1080	35990	700 MB
Video-3.mp4	Day	15	1920×1080	13477	260 MB
Video-4.mp4	Day	15	1920×1080	7290	142 MB
Video-5.mp4	Day	15	1920×1080	12769	250 MB
Video-6.mp4	Night	15	1920×1080	13275	264 MB
Video-7.mp4	Night	15	1920×1080	13492	265 MB

To train YOLOv9 and Mask-RCNN module in the FRVC model, four different possible combinations of ASV dataset in a ratio 90-10%, 85-15%, 80-20%, and 75-25% are considered for training-testing purposes. The performance of these modules is evaluated based on accuracy, speed, recall, and precision. Table 4.14 gives an idea about the parameter specification required to train YOLOv9 and the Mask-RCNN module. As the detection accuracy of the two-phase detector is higher than the one-phase detector, we used 40 epochs with batch size 16 to train YOLOv9 while only 30 epochs with batch size 2 to train the Mask-RCNN network. During training, Mask-RCNN uses various loss functions with Adam optimizer and a learning rate of 0.01. The loss function in Mask R-CNN is an integration of RPN loss, classification loss, regression loss, and mask loss. While the loss function in YOLOv9 is also a combination of objectness loss, regression loss, classification loss, balanced loss, and weighted loss. The gradients of this loss are used to update the network's parameters during backpropagation. Figure 4.6 denotes the graphical performance of YOLOv9 module.

Table 4.14: Parameter Specification

Parameters	**YOLOv9**	**Mask-RCNN**
Backbone	PGI+GELAN	ResNet-101
Feature aggregation (Neck)	PaNet/SPP	RoI-Align
Batch size	16	2
Epoch	40	30
Activation function	Sigmoid Linear Unit(SiLU)	ReLU
Learning rate	0.01	0.02
Momentum	0.9	0.9
Weight decay	0.00005	0.00001

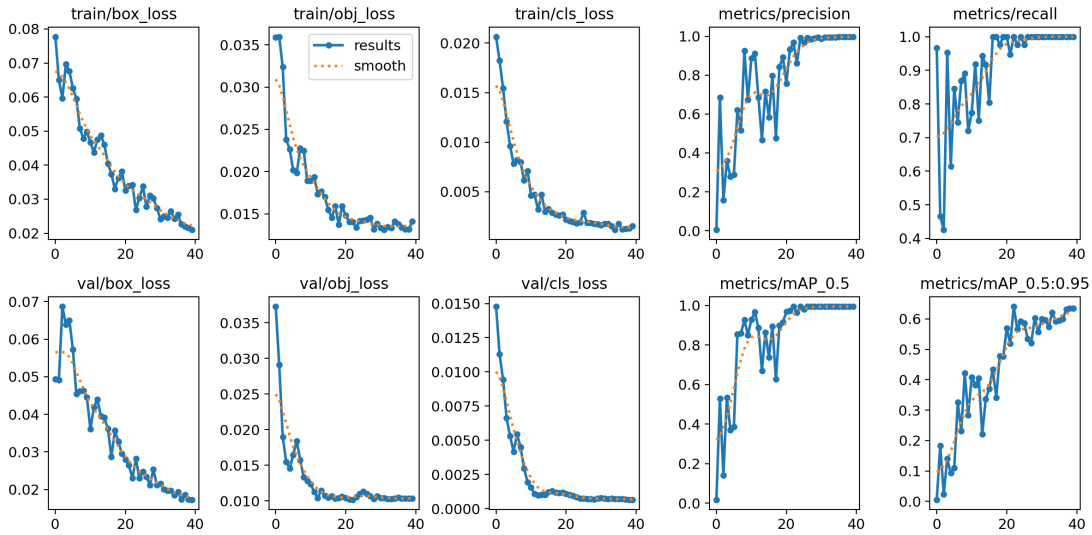


Figure 4.6: Graphical Performance of YOLOv9 Module in FRVC Model.

After an extensive experiment, it is observed that 80-20% ratio gives promising performance and achieves the highest accuracy. Table 4.15 offers a nuanced understanding of the performance characteristics of the YOLOv9 and Mask R-CNN network, considering key metrics and different data distribution scenarios. The

Table 4.15: Comparison of YOLOv9 and Mask-RCNN Module

Dataset Split ratio (%)	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
YOLOv9 Model				
90-10	97.2	93.9	98.5	96.1
85-15	98.3	94.9	98.6	96.7
80-20	**99.6**	**95.6**	**99.3**	**97.4**
75-25	98.6	94.7	97.9	96.2
Mask-RCNN Model				
90-10	97.5	93.5	98.2	95.9
85-15	98.6	92.2	98.3	95.1
80-20	**99.2**	**93.8**	**99.5**	**96.4**
75-25	98.8	93.1	99.2	96.0

YOLOv9 module, achieved the highest accuracy of 99.6% using a split ratio of 80-20% while for Mask-RCNN it is 99.2%. while the highest precision of 95.6% is achieved using YOLov9 and 93.8% for mask-RCNN using 80-20% ratio. Moving to Recall, a measure of a module capability to identify all relevant instances,

YOLOv9 exhibits consistent improvements, reaching 99.3% and Mask R-CNN showcases varying recall values, with the highest 99.5% achieved at 80-20 ratio. Derived from these results, an examination reveals that customized YOLOv9 exhibits promising results for the FRVC dataset as compared to Mask R-CNN. Notably, in the context of the FRVC dataset, the detection accuracy of the one-phase YOLOv9 model surpasses the two-phase Mask-RCNN. Furthermore, it is observed that Mask-RCNN requires a maximum amount of time (probably twice or thrice) for the detection of relevant and irrelevant frames within the video as compared to YOLOv9. Figure 4.7 shows the graphical comparison of OD modules on seven different videos. Hence, the output generated through the YOLOv9 module is employed for subsequent compression tasks.

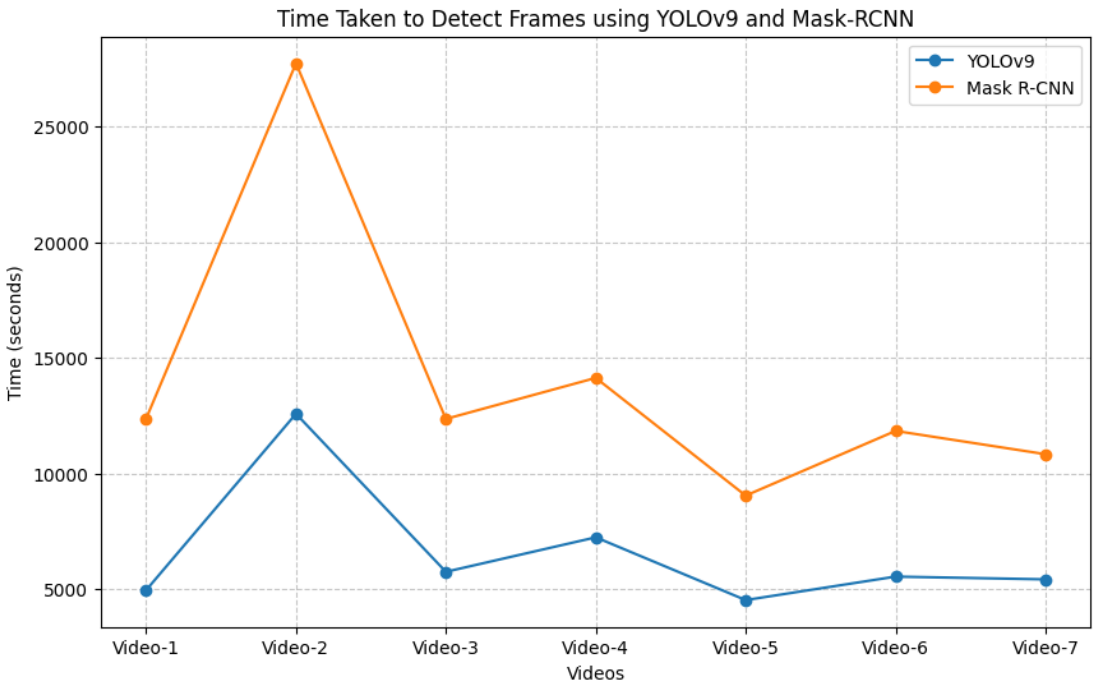


Figure 4.7: Time Taken to Detect Relevant and Irrelevant Frames of Surveillance Video Using YOLOv9 and Mask R-CNN in FRVC Model.

Table 4.16: Frames Count Using YOLOv9 Module

Video No.	**Total frames**	**Relevant Frames**	**Irrelevant Frames**	**Time taken to detect frames**
Video-1.mp4	26996	2631	24365	4327 Sec
Video-2.mp4	35990	5620	30370	11678 Sec
Video-3.mp4	13477	9472	4005	5152 Sec
Video-4.mp4	7290	4667	2623	4834 Sec
Video-5.mp4	12769	55	12714	4121 Sec
Video-6.mp4	13275	2711	10564	5145 Sec
Video-7.mp4	13492	2622	10870	5110 sec

Table 4.16 represents the count of relevant and irrelevant frames using the YOLOv9 network on seven different tested videos and the time taken by each video to predict the frames. The video2 has the highest number of total frames i.e. 35,990, of which only 5,620 are relevant, while 30,370 are irrelevant and require 11,678 seconds for detection. Video3 has a notable 13,477 frames, with the highest relevance frames, as 9,472 are identified as relevant. Conversely, Video5 has 12,769 total frames, but only 55 are relevant, which suggests minimal activity. The analysis reveals that longer processing times for videos with high frame counts are required. Hence, the Table 4.16 underscores the importance of YOLOv9 in automated video frame classification and its relevance for applications like surveillance and storage compression.

4.4.2 Result Analysis of Compression Module

For the proposed FRVC model, we considered seven ATM surveillance videos under three different scenarios. Here, scenario I indicates the situation when a single person enters an ATM room and performs a transaction activity. Scenario II denotes the interval when no individuals enter in ATM while scenario III shows the timelapse when two people are present in the surveillance room simultaneously. Figure 4.8 represents a sample frame of Scenario I, Figure 4.9 indicates Scenario II and Figure 4.10 shows sample frames of scenario II. Among seven tested videos, video1, video2 and video3 belong to the scenario I, video5 belongs to scenario II and the remaining video4, video6, and video7 are associated with Scenario III.

Figure 4.8: Sample Surveillance video Frame for Scenario I.

Figure 4.9: Sample Surveillance Video Frame for Scenario II.

In this compression module, the output of the YOLOv9 is used to conduct further compression processes. In this process, the similarity index between the relevant frames of surveillance video is determined at various threshold values starting from 100% to 90%. and splits the frames into further primary frames and similar frames. It is observed that, at 100% threshold value all relevant frames are primary frames, hence there is no further division of frames, while this splitting

Figure 4.10: Sample Surveillance Video Frame for Scenario III.

starts from 98% threshold value. Table 4.17 illustrates the trade-off between the number of primary and similar frames as the threshold decreases. For instance, at the highest threshold of 100%, all frames of video1 are unique and has 2631 primary frames with no similar frames. As the threshold decreases to 98%, the count of primary frames reduces to 2491 and 140 similar frames are obtained. At a threshold of 96%, the number of primary frames decreases further to 1709, and the count of similar frames increases to 922. This pattern continues as the threshold is reduced to 94%, with the primary frames dropping to 1157 and similar frames increasing to 1474. When the threshold is set to 92%, the primary frames are reduced further to 780, with 1851 similar frames, which demonstrate a significant compression effect. Finally, at the lowest threshold of 90%, the primary frames drop to just 539, while the count of similar frames reaches its peak at 2092. This progression shows that the FRVC model effectively reduces the number of frames in surveillance video as the similarity threshold decreases and achieves significant compression. Similarly, video2 contains a total of 35990 frames out of which 5620 frames are recognized as relevant frames using the YOLOv9 module. At 100% threshold all relevant frames are recognized as primary frames. However, from the 98% threshold onward, the number of primary frames begins to decline. At this

Table 4.17: Count of Primary and Similar Frames at Various Threshold Values for Tested Seven Videos of FRVC Model

Threshold (%)	**Metric**	**Video-1.mp4**	**Video-2.mp4**	**Video-3.mp4**	**Video-4.mp4**	**Video-5.mp4**	**Video-6.mp4**	**Video-7.mp4**
100	Primary Frames	2631	5620	9472	4667	55	2711	2622
	Similar Frames	0	0	0	0	0	0	0
98	Primary Frames	2491	3806	5699	4667	35	2711	2622
	Similar Frames	140	1814	3773	0	20	0	0
96	Primary Frames	1709	2086	3056	4667	25	2711	2622
	Similar Frames	922	3534	6416	0	30	0	0
94	Primary Frames	1157	1515	1702	4389	15	2560	2344
	Similar Frames	1474	4105	7770	278	40	151	278
92	Primary Frames	780	1126	984	3891	2	2284	1968
	Similar Frames	1851	4494	8488	776	53	427	654
90	Primary Frames	539	916	537	2841	1	1517	1443
	Similar Frames	2092	4704	8935	1826	54	1194	1179

threshold, 3806 frames are identified as primary, while 1814 frames are categorized as similar. At 96% threshold, 2086 frames are detected as primary frames and 3534 as similar frames resulting in a video size of 94.2 MB. Similarly, at a 94% threshold, there are 1515 primary frames, 4105 similar frames, and a reduced video size of 78.2 MB. The count of primary frames further decreases to 1126, with 4494 similar frames, yielding a compressed video size of 62.9 MB at a 92% threshold. Finally, at a 90% threshold, there are 916 primary frames, 4704 similar frames, and a video size of 48.5 MB. It is evident that as the similarity value between frames decreases, the compression size increases. The highest compression is achieved at a 90% threshold, while the minimum compression is observed at a 100% threshold. Figure 4.11 provides a visual representation of the frame count across different threshold values for each video.

Table 4.18: Compressed Video Characteristics at Threshold Value = 90%, FPS = 15, and Resolution = 1920×1080

Video No.	**Scenario**	**Original Size**	**Compressed Video Size**	**% Compression**	**CR**
Video-1.mp4	Scenario-I	527 MB	19.1 MB	96.3%	27.57
Video-2.mp4	Scenario-I	700 MB	47.9 MB	93.1%	14.61
Video-3.mp4	Scenario-I	260 MB	19.0 MB	92.6%	13.68
Video-4.mp4	Scenario-III	142 MB	71.1 MB	49.9%	2.00
Video-5.mp4	Scenario-II	250 MB	187 KB	99.9%	1336.90
Video-6.mp4	Scenario-III	264 MB	101.1 MB	61.7%	2.61
Video-7.mp4	Scenario-III	265 MB	96.5 MB	63.5%	2.75

Table 4.18 provides a detailed analysis of compressed video characteristics at a 90% threshold value, FPS of 15, and a resolution of 1920×1080. The table presents key metrics for seven video samples which are evaluated under different

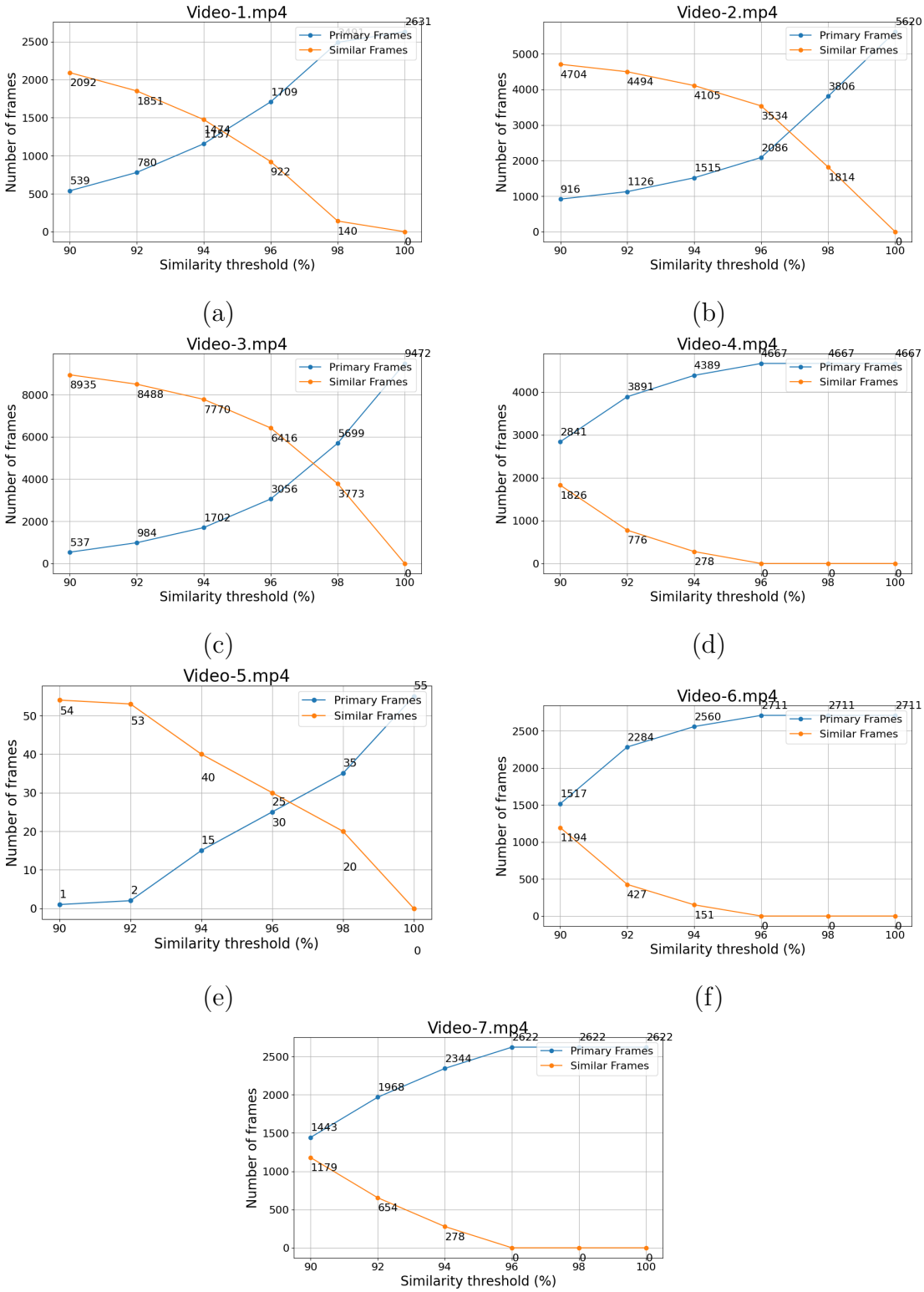


Figure 4.11: Count of Frames vs. Similarity Threshold in FRVC Model.

scenarios and provide insights into the efficiency of the compression process. Here, each video captures transactional activities that focus on the presence and actions of individuals within the surveillance video. In Scenario I, videos are captured during the daytime feature where an individual is present in an ATM room and engaged in transactional activities. Specifically, video1, video2, and video3 belong to scenario I and scenario I demonstrate the highest compressions among all scenarios. The "% Compression" column represents the percentage reduction in file size achieved during compression. For example, video1 achieves a 96.37% reduction in size. While video2 and video3 achieve 93.16%, and 92.69%, respectively, when individuals are detected. However, the Compression Ratio (CR) is the ratio of the original size of video to the compressed video size which is determined using Eq. (4.6) and mentioned in the last column. The CR of video1 is 27.17 which means that compressed video is 27.17 times smaller than the original video.

In Scenario II, the video depicts a time frame where no human presence is detected. Specifically, video5 corresponds to Scenario II. For video5, out of the 12,769 frames, only 55 frames are identified as relevant, achieving an impressive compression rate of 99.93% at a threshold value of 90%. This implies that the entire video is effectively recognized based on a small subset of frames. In fact, at a 100% threshold value, the compression rate is still substantial, reaching 98.9%. Scenario III entails video4, video6 and video7, where two individuals jointly enter the surveillance room and engage in transactional activities. In this scenario, the frames exhibit lower similarity, resulting in a larger proportion being considered as primary frames. Consequently, these videos achieve lower compressions of 49.9%, 61.7%, and 63.5%, respectively, compared to other scenarios. The percentage of video compression is determined using the Eq. (4.7). It is essential to note that, while conducting video compression across various threshold values, there is a deliberate effort taken to uphold the quality of the video. This entails maintaining the values of FPS and frame resolution consistently, aligning them with the original specifications. This strategic approach is designed to prioritize the preservation of essential visual characteristics throughout the compression process, ensuring that

the perceptual quality of the video remains uncompromised.

Table 4.19: Compression Achieved in Megabytes (MB) at Various Threshold (TH) Values in FRVC Model

Video No.	**Original Size (MB)**	**TH = 100 (MB)**	**TH = 98 (MB)**	**TH = 96 (MB)**	**TH = 94 (MB)**	**TH = 92 (MB)**	**TH = 90 (MB)**
Video-1.mp4	527	104.2	60.3	46.4	35.1	26.4	19.1
Video-2.mp4	700	100.1	100.1	94.2	78.2	62.9	47.9
Video-3.mp4	260	183.3	132.3	85.1	53.5	33.3	19.0
Video-4.mp4	142	117.1	117.1	109.3	101.1	95.5	71.1
Video-5.mp4	250	2.4	1.8	1.6	1.3	0.0	0.0
Video-6.mp4	264	148.4	148.4	148.4	141.7	132.1	101.1
Video-7.mp4	265	157.6	157.6	157.6	132.3	116.1	96.5
Average % Compression	—-	**56.6%**	**60.6%**	**64.5%**	**69.5%**	**73.1%**	**79.6%**

The Average % of Compression is calculated using Eq. (4.8). Table 4.19 also serves as a quantitative representation of the impact of similarity threshold values on video compression. It allows for a detailed examination of how variations in these thresholds influence the compression ratios for different videos. At a 100% threshold, a minimum 56.6% average compression is achieved. For the 98% threshold, the average compression is 60.6%, while at 96%, it is recorded as 64.5%. The average compression for a 92% threshold is 73.1%, and the highest compression of 79.6% is achieved at a 90% threshold. Notably, the average compression increases as the similarity threshold decreases, the observed trends in compression sizes provide valuable insights into the relationship between similarity threshold values and the efficiency of the compression algorithm, which can be crucial for optimizing video storage and transmission in surveillance systems. Figure 4.12 shows a graphical representation of the size of videos at different thresholds.

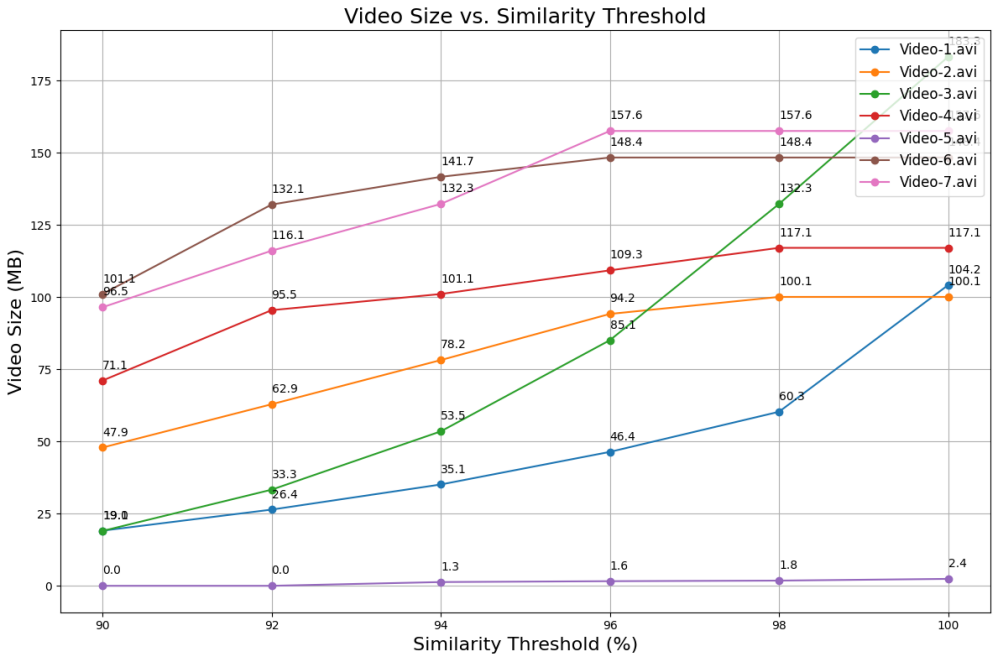


Figure 4.12: Size of Compressed Videos at Different Threshold of FRVC Model.

Based on the above analysis, we can succinctly summarize that the most significant compression is observed in scenario I, where a lone individual engages in transactional activities. Conversely, scenario II, characterized by the absence of relevant parts, is instrumental in optimizing storage space. Notably, the highest compression achieved is 96.3%, with an average compression of 79.6% at a 90% threshold value. The observed trend underscores that as the similarity between frames diminishes, compression increases. Figure 4.13 visually encapsulates this relationship through a Pareto chart, providing a comprehensive overview of the outcomes across the seven tested videos. The Pareto chart discernibly demonstrates the inherent trade-off between prioritizing similarity and compression. It demonstrates that while increasing similarity thresholds reduces individual compression rates, the cumulative compression remains significant. Decision-makers should carefully select similarity thresholds based on the desired trade-off between data accuracy and compression efficiency. For applications where data accuracy is critical, higher thresholds should be preferred despite lower compression. However, for storage-intensive applications where efficiency is prioritized, lower thresholds may be more suitable.

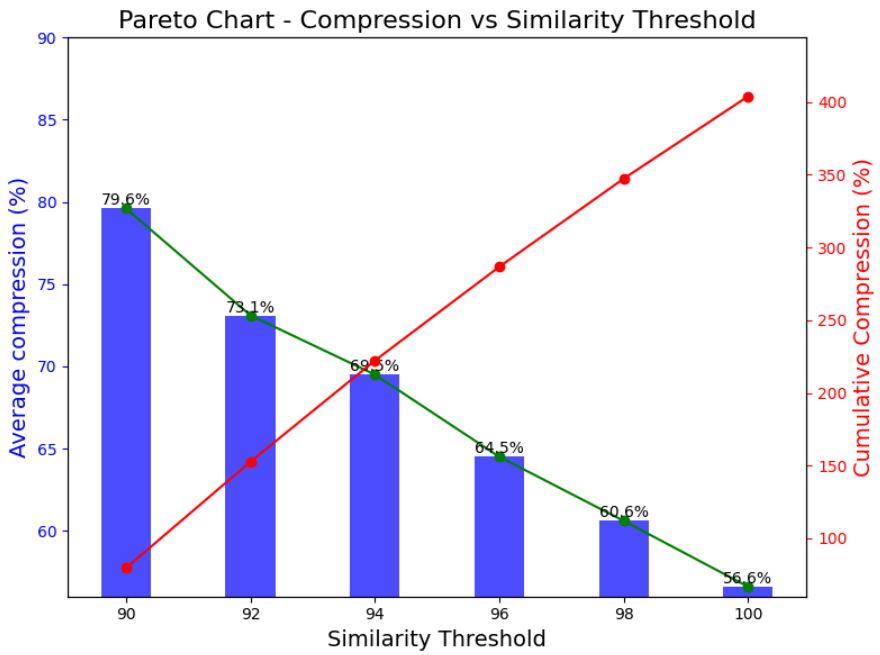


Figure 4.13: Pareto Chart for Average % of Compression on Seven Tested Videos.

Comprehensively, these findings contribute to the optimization of video storage and transmission within surveillance systems, that underscore the delicate balance between similarity thresholds, compression efficiency, and visual fidelity. The scientific approach employed ensures that key video attributes, such as FPS and resolution, are preserved throughout the compression process, which prioritizes the retention of essential visual information.

4.5 Comparison of ODSC, D&C and FRVC Model

The proposed three approaches: (i) ODSC, (ii) D&C and (iii) FRVC were analyzed based on their performance across metrics such as accuracy, precision, recall, and F1-score. The training and testing datasets differ among these approaches. ODSC and D&C models were trained on the COCO dataset and tested on the ASV dataset, while FRVC model were trained and tested on the ASV dataset using different split ratios. The following analysis highlights their relative strengths .

In the ODSC approach, YOLOv5, YOLOv7, and YOLOv8 were evaluated. Among these, YOLOv8 demonstrated the highest accuracy of 98.7% when tested

on the ASV dataset. However, its F1-score 97.3% was matched by YOLOv7, which also exhibited balanced recall 98.2%) and precision 98.3%. The D&C approach introduced YOLOv9, which outperformed all previous YOLO versions tested in ODSC. With an accuracy of 99.1%, YOLOv9 achieved superior precision of 98.7%) and recall (of 98.5%,that reflect its enhanced capability to detect and classify objects effectively. This improvement underscores the advancements in YOLOv9 architecture and its adaptability to the ASV dataset, despite being trained on COCO. The FRVC approach focused on reducing redundant video frames using primary frames identified through object detection. Both YOLOv9 and Mask-RCNN were evaluated on ASV with different dataset split ratios. YOLOv9 consistently outperformed Mask-RCNN across all metrics, particularly in the 80-20 split configuration, where it achieved the highest accuracy of 99.6% and F1-score of 97.4%. Mask-RCNN, while performing well with an accuracy of 99.2% and F1-score of 96.4%, lagged behind YOLOv9 in precision of 93.8% and overall robustness. These results highlight YOLOv9 ability to maintain superior performance even on domain-specific datasets like ASV. Table 4.20 elaborates the performance of all three approaches using various evaluation metrics. From the analysis, YOLOv9 in the FRVC approach stands out as the top-performing model. Although the ODSC and D&C approaches demonstrated strong results, their dependence on COCO for training restricts their effectiveness on the ASV dataset which emphasized the significance of leveraging domain-specific datasets for optimal performance.

Table 4.20: Comparison of Approaches on Evaluation Metrics

Approach	**Model**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
ODSC	YOLOv8	98.7	98.5	97.2	97.3
D&C	YOLOv9	99.1	98.7	98.5	98.3
FRVC (80-20 split)	**YOLOv9**	**99.6**	**95.6**	**99.3**	**97.4**
FRVC (80-20 split)	Mask R-CNN	99.2	93.8	99.5	96.4

4.6 Comparison of Proposed Model with State-of-the-Art Models

This Section compares the proposed FRVC with existing approaches as the YOLOv9 of FRVC model achieved better results among all other modules on ASV dataset. Comparing the proposed work with existing approaches highlights its unique aspects and contributions. This comparison underscores the research's novelty and illustrates how it enhances the current body of knowledge in the field of surveillance compression. The proposed FRVC model is compared with four other existing approaches which are summarized in Table 4.21.

Table 4.21: Proposed Model Comparison with Existing Approach on FRVC Dataset

Parameter	**Panneerselvam et. al.**[135]	**Ghamsarian et.al.**[133]	**Lu Zonglei et. al.**[132]	**FRVC framework**
Approach	Used CNN+GAN to perform pixel level compression	From ophthalmologist, view determine the types of frame using neural network	Object separation based deep compression technique for apron surveillance	Adopts relevance frame classification
OD approach	No	Yes (Mask-RCNN)	Yes (Faster-RCNN)	Yes (YOLOv9)
compression achieve	up to 50.71%	up to 63%	up to 88%	up to 96.3% (for scenario-I) and 99.9% (for scenarios-II)
Type of compression	lossy compression	lossy compression	lossy compression	lossless compression

i. Panneerselvam et. al. [135] introduced a video compression method that divides layers into different groups for data processing. It employs CNN to identical frames, identifies and detects minute changes through GAN, and records the altered frames using Long Short-Term Memory (LSTM). Substituting small changes generated by GAN instead of the complete image facilitates frame-level compression. The process involves pixel-wise comparison

using K-nearest Neighbours across the frame, clustering with K-means, and applying singular value decomposition to each frame in the video for all three color channels.

ii. Negin et.al.[133]:- present cataract surgery video compression, where researchers used Mask-RCNN to employ advanced NN for the semantic segmentation and classification of videos. Through this process, the technique aims to discern and identify spatio-temporal information embedded in the video content. It also allocates a reduced bitrate to content identified as less crucial. This approach achieved 63% of compression.

iii. Lu Zonglei et.al.[132]:- introduces an advanced compression approach utilizing OD techniques to distinguish between moving and steady objects in an apron surveillance video and achieves compression upto 88%. This approach stores the objects in an image with its coordinates data in a data structure called a linked list. During compression, data in a linked list is retrieved and objects are placed at their location. AS the objects are cut and pasted back to the image, when the similarity is checked among frames it exhibits an average of 98.97%.

iv. FRVC framework:- Performs object detection using deep learning's approach on ATM surveillance dataset and achieved highest compression ratio up to 96.3%. Our approach performs on par with existing methods.

4.7 Summary

This chapter presents a detailed analysis of the results obtained from the proposed object detection and compression models. The proposed ODSC model, used YOLOv5, YOLOv7 and YOLOv8 frameworks in the object detection module where YOLOv8 outperforms other approaches in terms of accuracy, precision, recall and F1-score. Consequently, the relevant frames detected by YOLOv8 were utilized in the compression module. The performance of the ODSC model was

tested on seven ATM surveillance videos, achieving an average compression rate of 66.23%. For the D&C model, six ATM surveillance videos were analyzed using YOLOv5, YOLOv7, YOLOv8, and YOLOv9 to identify relevant and irrelevant frames. Among these, YOLOv9 demonstrated superior performance across evaluation metrics. As a result, the relevant frames detected by YOLOv9 were used in the compression module. The six videos in the D&C model were categorized into two scenarios: (i) urban areas and (ii) rural areas. Experimental results revealed that videos from urban areas achieved a higher percentage of compression and a better compression ratio compared to rural areas. The seven videos from the ODSC model were reanalyzed in the FRVC model under three distinct scenarios, depending on the number of individuals entering the ATM surveillance room simultaneously. Scenario II, which depicted intervals with no person entering the room, achieved the highest compression rate of 99.9%. Scenario I, representing times when a single person was present, achieved up to 96.3% compression, while Scenario III, involving intervals with more than two people in the room, resulted in the lowest compression rate. Under the FRVC model, an average compression rate of 79.6% was achieved at a 90% threshold. Lastly, the FRVC framework was compared with state-of-the-art approaches and demonstrated superior performance which established its effectiveness in optimizing surveillance video compression.

Chapter 5

CONCLUSION AND FUTURE SCOPE

The proliferation of CCTV systems offers significant benefits which include enhanced security, crime deterrence, and improved public safety. They assist in monitoring traffic, managing crowds, and providing critical evidence for investigations. As the demand increases, the quality of surveillance video need to be improved and to meet the growing demand, service providers have improved video quality by increasing spatial and temporal resolutions as well as frame rates. However, these enhancements lead to a significant rise in storage requirements. Solely increasing storage capacity is not a practical solution, as continuous 24/7 recording quickly exhausts storage resources on local devices such as microSD cards and hard drives. The compression of surveillance video is a crucial step for enhancing storage efficiency, which reduces the transmission bandwidth and improves overall system performance. Surveillance cameras generate a large volume of video data, especially in high-definition formats, which can quickly overwhelm storage devices and network infrastructure. By applying compression techniques, the data size is significantly reduced without compromising the essential visual quality needed for security and monitoring purposes. To address these challenges, a relevance-based, domain-specific video compression technique is essential. Such methods minimize storage demands, enabling longer retention periods without incurring high costs. This thesis introduces three innovative surveillance video compression techniques to tackle this issue effectively. The researcher performed a methodolog-

ical literature review, proposed solutions, developed the mechanism and simulated the techniques to escalate the current state-of-the-art in this area. In particular, Chapter 1 explained the details about compression and types of compression along with the evolution of compression. Later, it explains the significance of surveillance video compression and the motivation behind the proposed research work. It also describes the development of the deep learning framework and its mechanism. It also defines the thesis organization.

In Chapter 2, the methodological survey is carried for deep learning-based object detection, video compression, and surveillance video compression techniques. Here, DL-based OD surveys are classified into two major aspects, i.e., region-based OD and region-free OD. While the video compression review explained the integration of DL and the tools of H.265 codec. Lastly, various surveillance video compression approaches are described. The researcher also identified challenges in the compression domain list the objectives of the thesis at the end of chapter.

Chapter 3 elaborate the methodology of the proposed three approaches: (i) Object Detection Based Surveillance Video Compression (ODSC), (ii) Relevant Video Frame Detection and Compression (D&C) and (iii) Frame Relevance Based Video Compression (FRVC) model. This chapter also explains the required and prepared dataset used for training and testing of approaches. It gives detailed steps of approaches in the form of workflow and algorithm.

Chapter 4 explain the details performance of proposed methodologies. Initially, it lists out various evaluation metrics used to carry out a comparison of detection and compression modules. Later, it explained the detailed analysis of object detection and video compression modules of all three approaches using evaluation metrics. In the end, the FRVC framework was evaluated against state-of-the-art methods and showcased superior performance, that highlights its effectiveness in optimizing surveillance video compression. At last, Chapter 5 concludes our research work and explains the further future scope.

5.1 Future Work

The Future Scope of our research work, highlights the potential advancements and innovative applications that can build upon the proposed models and methodologies. Though the current study addresses challenges in surveillance video storage and frame relevance detection, there are some opportunities to extend its applicability. The performance of presented models can be improve by integrating emerging technologies and exploring new domains. This section outlines the potential future directions in terms of leveraging deep learning (DL), optimizing models for embedded systems, ensuring energy-efficient solutions, expanding to diverse applications, and enhancing real-time analytics. These directions promise to drive transformative changes in surveillance systems and beyond.

One promising area is the integration of DL with the Internet of Things (IoT), which offers the potential to create intelligent and decentralized surveillance solutions. By embedding DL models directly into IoT devices, such as smart cameras or low-power computing units like Raspberry Pi and NVIDIA Jetson Nano, real-time data analysis can be performed locally. This reduces the dependency on centralized servers, cutting down on bandwidth usage and latency. Lightweight frameworks such as TensorFlow Lite or PyTorch Mobile can be employed to adapt DL models for IoT environments, that enable seamless operations on resource-constrained devices. This fusion of DL and IoT can enable smarter, faster, and more scalable surveillance systems. To enable the deployment of these models on embedded systems, lightweight model optimization is essential. Techniques like quantization, pruning, and knowledge distillation can significantly reduce the size and complexity of DL models while maintaining accuracy. Quantization involves converting model weights into lower-precision formats, while pruning removes redundant parameters. These optimized models can then be converted into formats like TensorFlow Lite or ONNX for compatibility with embedded devices. Leveraging hardware accelerators, such as Edge TPU or NVIDIA's TensorRT, ensures efficient processing even with limited computational resources. Such optimizations pave the way for compact and efficient systems suitable for real-world deployments.

Another promising direction is incorporating machine learning or deep learning models for automated frame classification. Instead of manually setting threshold values, deep learning techniques could be trained to predict primary and similar frames. This would enhance adaptability and reduce manual intervention, allowing the system to generalize better across different surveillance scenarios. Another critical area is the development of energy-efficient surveillance solutions. Given the proliferation of IoT-enabled devices, it is crucial to design systems that consume minimal power while maintaining high performance. This can be achieved through event-driven architectures, where DL models are activated only upon detecting motion or anomalies, reducing idle power consumption. Techniques like dynamic resource allocation and power management further enhance energy efficiency. For instance, lightweight algorithms can handle simpler tasks, reserving more powerful models for complex scenarios. These approaches not only lower operational costs but also make surveillance systems sustainable for long-term use in remote or resource-constrained areas. Expanding the models to other domains and use cases presents another exciting opportunity. The techniques developed in this study can be adapted for applications in healthcare, wildlife monitoring, industrial safety, and beyond. For example, in healthcare, DL models can be trained to monitor patient behavior for early detection of anomalies, while in wildlife monitoring, they can track animal activity using thermal or infrared imaging. Transfer learning allows the adaptation of these models to domain-specific datasets with minimal retraining, reducing development time. Additionally, integrating multimodal data—such as combining visual inputs with sensor data—can enhance decision-making capabilities, enabling comprehensive solutions for complex challenges.

Bibliography

[1] Niels Bohr. "Neutron Capture and Angular Momentum". *Physical Review*, 37(8):780–781, 1931.

[2] Martin Gill and Angela Spriggs. *"Assessing The Impact of CCTV"*, volume 292. Home Office Research, Development and Statistics Directorate London, 2005.

[3] Steven Graham. "The Eyes Have It: CCTV as The Fifth Utility". *Environment and Planning B: Planning and Design*, 26(5):639–642, 1999.

[4] Niels Haering, Péter L Venetianer, and Alan Lipton. "The Evolution of Video Surveillance: An Overview". *Machine Vision and Applications*, 19(5-6):279–290, 2008.

[5] Liza Lin and Newley Purnell. "A World With A Billion Cameras Watching You Is Just Around The Corner". *The Wall Street Journal*, 2019.

[6] E Dhiravidachelvi, E Anna Devi, Ms E Jayanthi, and Ms IS Suganthi. "Advanced Video Surveillance System Using Computer Vision". *Semiconductor Optoelectronics*, 42(1):897–906, 2023.

[7] R Akash and Leandra Shania Anderson. "Efficient Storage and Analysis of Videos Through Motion-Based Frame Removal". *Research Square*, 2023.

[8] John Doe. "The Essence of Data Compression". *Journal of Information Science*, 25(3):150–165, 20XX.

[9] David Salomon. *"Data Compression"*. Springer, 2002.

[10] Khalid Sayood. *"Introduction to Data Compression"*. Morgan Kaufmann, 2017.

[11] P Gabriel Peterson, Sung K Pak, Binh Nguyen, Genevieve Jacobs, and Les Folio. "Extreme Compression for Extreme Conditions: Pilot Study to Identify Optimal Compression of CT Images using MPEG-4 Video Compression". *Journal of Digital Imaging*, 25:764–770, 2012.

[12] Abir Jaafar Hussain, Ali Al-Fayadh, and Naeem Radi. "Image Compression Techniques: A Survey in Lossless and Lossy Algorithms". *Neurocomputing*, 300:44–69, 2018.

[13] Samruddhi Y Kahu, Rajesh B Raut, and Kishor M Bhurchandi. "Review and Evaluation of Color Spaces for Image/Video Compression". *Color Research & Application*, 44(1):8–33, 2019.

[14] Nasir D Memon and Khalid Sayood. "Lossless Compression of Video Sequences". *IEEE Transactions on Communications*, 44(10):1340–1345, 1996.

[15] Ben Strasser, Adi Botea, and Daniel Harabor. "Compressing Optimal Paths with Run Length Encoding". *Journal of Artificial Intelligence Research*, 54:593–629, 2015.

[16] Apoorv Gupta, Aman Bansal, and Vidhi Khanduja. "Modern Lossless Compression Techniques: Review, Comparison and Analysis". In *Proceedings of 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–8. IEEE, 2017.

[17] Yu-Chen Hu and Chin-Chen Chang. "A New Lossless Compression Scheme Based on Huffman Coding Scheme for Image Compression". *Signal Processing: Image Communication*, 16(4):367–372, 2000.

[18] Michelle Effros, Karthik Visweswariah, Sanjeev R Kulkarni, and Sergio Verdú. "Universal Lossless Source Coding with The Burrows Wheeler Transform". *IEEE Transactions on Information Theory*, 48(5):1061–1081, 2002.

[19] Didier Le Gall. "MPEG: A Video Compression Standard for Multimedia Applications". *Communications of The ACM*, 34(4):46–58, 1991.

[20] Touradj Ebrahimi and Caspar Horne. "MPEG-4 Natural Video Coding–An Overview". *Signal Processing: Image Communication*, 15(4-5):365–385, 2000.

[21] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. "Overview of The H. 264/AVC Video Coding Standard". *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.

[22] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. "Overview of The High Efficiency Video Coding (HEVC) Standard". *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012.

[23] Debargha Mukherjee, Jingning Han, Jim Bankoski, Ronald Bultje, Adrian Grange, John Koleszar, Paul Wilkins, and Yaowu Xu. "A Technical Overview of vp9–The Latest Open-Source Video Codec". In *Proceedings of SMPTE 2013 Annual Technical Conference & Exhibition*, pages 1–17. SMPTE, 2013.

[24] Yue Chen, Debargha Murherjee, Jingning Han, Adrian Grange, Yaowu Xu, Zoe Liu, Sarah Parker, Cheng Chen, Hui Su, and Urvang Joshi. "An Overview of Core Coding Tools in The AV1 Video Codec". In *Proceedings of 2018 Picture Coding Symposium (PCS)*, pages 41–45. IEEE, 2018.

[25] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. "Overview of The Versatile Video Coding (VVC) Standard and Its Applications". *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021.

[26] I Goodfellow. *"Deep Learning"*. MIT Press, 2016.

[27] Rene Y Choi, Aaron S Coyner, Jayashree Kalpathy-Cramer, Michael F Chiang, and J Peter Campbell. "Introduction to Machine Learning, Neural

Networks, and Deep Learning". *Translational Vision Science & Technology*, 9(2):14–14, 2020.

[28] Amey Thakur and Archit Konde. "Fundamentals of Neural Networks". *International Journal for Research in Applied Science and Engineering Technology*, 9(VIII):407–426, 2021.

[29] Glenn Jocher. "YOLOv5 by Ultralytics". AGPL-3.0, 2020.

[30] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-The-Art for Real-Time Object Detectors". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

[31] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. "Ultralytics YOLOv7", January 2023.

[32] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. "Yolov9: Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv:2402.13616*, 2024.

[33] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask R-CNN". In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[34] Arthur C. Clarke. *"Profiles of The Future"*. Macmillan, 1984.

[35] Sarah Chen. "Advancements in Object Detection: Bridging the Gap Between Vision and Action". *Computer Vision Journal*, 15:300–315, 20ZZ.

[36] Xiongwei Wu, Doyen Sahoo, and Steven CH Hoi. "Recent Advances in Deep Learning for Object Detection". *Neurocomputing*, 396:39–64, 2020.

[37] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[38] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. "Selective Search for Object Recognition". *International Journal of Computer Vision*, 104:154–171, 2013.

[39] Hedi Harzallah, Frédéric Jurie, and Cordelia Schmid. "Combining Efficient Object Localization and Image Classification". In *Proceedings of 2009 IEEE 12th International Conference on Computer Vision*, pages 237–244. IEEE, 2009.

[40] Navneet Dalal and Bill Triggs. "Histograms of Oriented Gradients for Human Detection". In *Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[41] Rainer Lienhart and Jochen Maydt. "An Extended Set of Haar-Like Features for Rapid Object Detection". In *Proceedings of International Conference on Image Processing*, volume 1, pages I–I. IEEE, 2002.

[42] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded Up Robust Features". In *Proceedings of Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 404–417. Springer, 2006.

[43] Ingo Steinwart and Andreas Christmann. *"Support Vector Machines"*. Springer Science & Business Media, 2008.

[44] Derek Hoiem, Santosh K Divvala, and James H Hays. "PASCAL VOC 2008 Challenge". *World Literature Today*, 24(1):1–4, 2009.

[45] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.

[46] Ross Girshick. "Fast R-CNN". In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[47] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". *Advances in Neural Information Processing Systems*, 28, 2015.

[48] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. "Feature Pyramid Networks for Object Detection". In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[49] Nikita Mohod, Prateek Agrawal, and Vishu Madaan. "YOLOv4 Vs YOLOv5: Object Detection on Surveillance Videos". In *Proceedings of International Conference on Advanced Network Technologies and Intelligent Computing*, pages 654–665. Springer, 2022.

[50] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. "Masklab: Instance Segmentation by Refining Object Detection with Semantic and Direction Features". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2018.

[51] Zhaowei Cai and Nuno Vasconcelos. "Cascade R-CNN: High Quality Object Detection and Instance Segmentation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1483–1498, 2019.

[52] Jiale Cao, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. "D2det: Towards High Quality Object Detection and Instance Segmentation". In *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11485–11494, 2020.

[53] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. "Grid R-CNN". In *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7363–7372, 2019.

[54] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. "Mask Scoring R-CNN". In *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6409–6418, 2019.

[55] Tianheng Cheng, Xinggang Wang, Lichao Huang, and Wenyu Liu. "Boundary-Preserving Mask R-CNN". In *Proceedings of Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Part XIV 16*, pages 660–676. Springer, 2020.

[56] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You 0nly Look Once: Unified, Real-Time Object Detection". In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.

[57] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. "SSD: Single Shot Multibox Detector". In *Proceedings of Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Part I 14*, pages 21–37. Springer, 2016.

[58] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. "Focal Loss for Dense Object Detection". In *Proceedings of The IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[59] Joseph Redmon and Ali Farhadi. "YOLO9000: Better, Faster, Stronger". In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pages 7263–7271, 2017.

[60] Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement". *ArXiv Preprint arXiv:1804.02767*, 2018.

[61] Liquan Zhao and Shuaiyang Li. "Object Detection Algorithm Based on Improved YOLOv3". *Electronics*, 9(3):537, 2020.

[62] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection". *ArXiv Preprint arXiv:2004.10934*, 2020.

[63] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, , Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. "Ultralytics/YOLOv5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation", nov 2022.

[64] Jiaying Liu, Dong Liu, Wenhan Yang, Sifeng Xia, Xiaoshuai Zhang, and Yuanying Dai. "A Comprehensive Benchmark for Single Image Compression Artifact Reduction". *IEEE Transactions on Image Processing*, 29:7845–7860, 2020.

[65] Wenxue Cui, Tao Zhang, Shengping Zhang, Feng Jiang, Wangmeng Zuo, and Debin Zhao. "Convolutional Neural Networks Based Intra Prediction for HEVC". *ArXiv Preprint ArXiv:1808.05734*, 2018.

[66] Jiahao Li, Bin Li, Jizheng Xu, Ruiqin Xiong, and Wen Gao. "Fully Connected Network-Based Intra Prediction for Image Coding". *IEEE Transactions on Image Processing*, 27(7):3236–3247, 2018.

[67] Tak Wu Sam Kwong, Linwel Zhu, and Yun Zhang. "Generative Adversarial Network Based Intra Prediction for Video Coding", April 30 2020. US Patent App. 16/169,729.

[68] Yueyu Hu, Wenhan Yang, Sifeng Xia, and Jiaying Liu. "Optimized Spatial Recurrent Network for Intra Prediction in Video Coding". In *Proceedings of 2018 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2018.

[69] Yueyu Hu, Wenhan Yang, Mading Li, and Jiaying Liu. "Progressive Spatial Recurrent Neural Network for Intra Prediction". *IEEE Transactions on Multimedia*, 21(12):3024–3037, 2019.

[70] Yang Wang, Xiaopeng Fan, Shaohui Liu, Debin Zhao, and Wen Gao. "Multi-Scale Convolutional Neural Network-Based Intra Prediction for Video Coding". *IEEE Transactions on Circuits and Systems for Video Technology*, 30(7):1803–1815, 2019.

[71] Heming Sun, Zhengxue Cheng, Masaru Takeuchi, and Jiro Katto. "Enhanced Intra Prediction for Video Coding by Using Multiple Neural Networks. *IEEE Transactions on Multimedia*, 22(11):2764–2779, 2020.

[72] Marc Górriz Blanch, Saverio Blasi, Alan F Smeaton, Noel E O'Connor, and Marta Mrak. "Attention-Based Neural Networks for Chroma Intra Prediction in Video Coding". *IEEE Journal of Selected Topics in Signal Processing*, 15(2):366–377, 2020.

[73] Amna, Maraoui and Imen, Werda and Ezahra, Sayadi Fatma. "Deep Learning For Intra Frame Coding". In *Proceedings of 2021 International Conference on Engineering and Emerging Technologies (ICEET)*, pages 1–4, 2021.

[74] Ionut Schiopu, Hongyue Huang, and Adrian Munteanu. "CNN-Based Intra-Prediction for Lossless HEVC". *IEEE Transactions on Circuits and Systems for Video Technology*, 30(7):1816–1828, 2019.

[75] Chao Yao, Chenming Xu, and Meiqin Liu. "RDNet: Rate–Distortion-Based Coding Unit Partition Network for Intra-Prediction". *Electronics*, 11(6):916, 2022.

[76] Jing Zhang, Yonghong Hou, Zhe Zhang, Dengchao Jin, Peihan Zhang, and Ge Li. "Deep Region Segmentation-Based Intra Prediction for Depth Video Coding". *Multimedia Tools and Applications*, pages 1–12, 2022.

[77] Giyong Choi, PyeongGang Heo, and HyunWook Park. "Triple-Frame-Based Bi-Directional Motion Estimation for Motion-Compensated Frame Interpolation". *IEEE Transactions on Circuits and Systems for Video Technology*, 29(5):1251–1258, 2018.

[78] Yongbing Zhang, Lixin Chen, Chenggang Yan, Peiwu Qin, Xiangyang Ji, and Qionghai Dai. "Weighted Convolutional Motion-Compensated Frame Rate Up-Conversion Using Deep Residual Network". *IEEE Transactions on Circuits and Systems for Video Technology*, 30(1):11–22, 2018.

[79] Jue Mao and Lu Yu. "Convolutional Neural Network Based Bi-Prediction Utilizing Spatial and Temporal Information in Video Coding". *IEEE Transactions on Circuits and Systems for Video Technology*, 30(7):1856–1870, 2019.

[80] Xiandong Meng, Xuan Deng, Shuyuan Zhu, Xinfeng Zhang, and Bing Zeng. "A Robust Quality Enhancement Method Based on Joint Spatial-Temporal Priors for Video Coding". *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[81] Han Zhang, Li Song, Li Li, Zhu Li, and Xiaokang Yang. "Compression Priors Assisted Convolutional Neural Network for Fractional Interpolation". *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5):1953–1967, 2020.

[82] Liangwei Yu, Liquan Shen, Hao Yang, Xuhao Jiang, and Bo Yan. "A Distortion-Aware Multi-task Learning Framework for Fractional Interpolation in Video Coding". *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[83] Han Zhang, Li Li, Li Song, Xiaokang Yang, and Zhu Li. "Advanced CNN Based Motion Compensation Fractional Interpolation". In *Proceedings of 2019 IEEE International Conference on Image Processing (ICIP)*, pages 709–713. IEEE, 2019.

[84] Bohan Li, Jingning Han, Yaowu Xu, and Kenneth Rose. "Optical Flow Based Co-Located Reference Frame for Video Compression". *IEEE Transactions on Image Processing*, 29:8303–8315, 2020.

[85] Zhao Wang, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, and Siwei Ma. "Three-Zone Segmentation-Based Motion Compensation for Video Compression". *IEEE Transactions on Image Processing*, 28(10):5091–5104, 2019.

[86] Jiaying Liu, Sifeng Xia, and Wenhan Yang. "Deep Reference Generation with Multi-Domain Hierarchical Constraints for Inter-Prediction". *IEEE Transactions on Multimedia*, 22(10):2497–2510, 2019.

[87] Jianjun Lei, Zongqian Zhang, Dong Liu, Ying Chen, and Nam Ling. "Deep Virtual Reference Frame Generation For Multiview Video Coding". In *Proceedings of 2020 IEEE International Conference on Image Processing (ICIP)*, pages 1123–1127. IEEE, 2020.

[88] Yang Wang, Xiaopeng Fan, Ruiqin Xiong, Debin Zhao, and Wen Gao. "Neural Network-Based Enhancement to Inter Prediction for Video Coding". *IEEE Transactions on Circuits and Systems for Video Technology*, 32(2):826–838, 2022.

[89] Kang-Ho Lee and Sung-Ho Bae. "Compressing Neural Networks With Inter Prediction and Linear Transformation". *IEEE Access*, 9:69601–69608, 2021.

[90] Mingliang Zhou, Xuekai Wei, Sam Kwong, Weijia Jia, and Bin Fang. "Rate Control Method Based on Deep Reinforcement Learning for Dynamic Video Sequences in HEVC". *IEEE Transactions on Multimedia*, 23:1106–1121, 2020.

[91] Hongkui Wang, Shengju Yu, Ying Zhang, Zhuo Kuang, and Li Yu. "Hard-Decision Quantization Algorithm Based on Deep Learning in Intra Video Coding". In *Proceedings of 2019 Data Compression Conference (DCC)*, pages 607–607. IEEE, 2019.

[92] Binglin Li, Mohammad Akbari, Jie Liang, and Yang Wang. "Deep Learning-Based Image Compression with Trellis Coded Quantization". In *Proceedings of 2020 Data Compression Conference (DCC)*, pages 13–22. IEEE, 2020.

[93] Mu Li, Kede Ma, Jane You, David Zhang, and Wangmeng Zuo. "Efficient and Effective Context-Based Convolutional Entropy Modeling for Image Compression". *IEEE Transactions on Image Processing*, 29:5900–5911, 2020.

[94] Gergely Flamich, Marton Havasi, and José Miguel Hernández-Lobato. "Compressing Images by Encoding Their Latent Representations with Relative Entropy Coding". *ArXiv Preprint ArXiv:2010.01185*, 2020.

[95] Théo Ladune, Pierrick Philippe, Wassim Hamidouche, Lu Zhang, and Olivier Deforges. "Binary Probability Model for Learning Based Image Compression", 2020.

[96] Changyue Ma, Dong Liu, Li Li, Yao Wang, and Feng Wu. "Convolutional Neural Network-Based Coefficients Prediction for HEVC Intra-Predicted Residues". In *Proceedings of 2020 Data Compression Conference (DCC)*, pages 183–192. IEEE, 2020.

[97] David Minnen and Saurabh Singh. "Channel-Wise Autoregressive Entropy Models for Learned Image Compression". In *Proceedings of 2020 IEEE International Conference on Image Processing (ICIP)*, pages 3339–3343. IEEE, 2020.

[98] 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, and Yu-Chi Su. "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, 2019.

[99] Bolun Zheng, Yaowu Chen, Xiang Tian, Fan Zhou, and Xuesong Liu. "Implicit Dual-Domain Convolutional Network for Robust Color Image Compression Artifact Reduction". *IEEE Transactions on Circuits and Systems for Video Technology*, 30(11):3982–3994, 2019.

[100] Hongyue Huang, Ionut Schiopu, and Adrian Munteanu. "Frame-Wise CNN-Based Filtering for Intra-Frame Quality Enhancement of HEVC Videos". *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[101] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Dong Xu, Li Chen, and Zhiyong Gao. "Deep Non-Local Kalman Network for Video Compression Artifact Reduction". *IEEE Transactions on Image Processing*, 29:1725–1737, 2019.

[102] Zhaoqing Pan, Xiaokai Yi, Yun Zhang, Byeungwoo Jeon, and Sam Kwong. "Efficient In-Loop Filtering Based on Enhanced Deep Convolutional Neural Networks for HEVC". *IEEE Transactions on Image Processing*, 29:5352–5366, 2020.

[103] Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. "Deep Universal Generative Adversarial Compression Artifact Removal". *IEEE Transactions on Multimedia*, 21(8):2131–2145, 2019.

[104] Weiyao Lin, Xiaoyi He, Xintong Han, Dong Liu, John See, Junni Zou, Hongkai Xiong, and Feng Wu. "Partition-Aware Adaptive Switching Neural Networks for Post-Processing in HEVC". *IEEE Transactions on Multimedia*, 22(11):2749–2763, 2019.

[105] Jiangyue Xia and Jiangtao Wen. "Asymmetric Convolutional Residual Network for AV1 Intra In-Loop Filtering". In *Proceedings of 2020 IEEE International Conference on Image Processing (ICIP)*, pages 1291–1295. IEEE, 2020.

[106] Taeoh Kim, Hyeongmin Lee, Hanbin Son, and Sangyoun Lee. "Sf-CNN: A Fast Compression Artifacts Removal via Spatial-to-Frequency Convolutional

Neural Networks". In *Proceedings of 2019 IEEE International Conference on Image Processing (ICIP)*, pages 3606–3610. IEEE, 2019.

[107] Zhijie Huang, Yunchang Li, and Jun Sun. "Multi-Gradient Convolutional Neural Network Based In-Loop Filter For VVC". In *Proceedings of 2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020.

[108] Yibo Yang, Robert Bamler, and Stephan Mandt. "Improving Inference for Neural Image Compression". *ArXiv Preprint ArXiv:2006.04240*, 2020.

[109] Eirina Bourtsoulatze, Aaron Chadha, Ilya Fadeev, Vasileios Giotsas, and Yiannis Andreopoulos. "Deep Video Precoding". *IEEE Transactions on Circuits and Systems for Video Technology*, 30(12):4913–4928, 2019.

[110] Hongwei Lin, Xiaohai He, Linbo Qing, Qizhi Teng, and Songfan Yang. "Improved Low-Bitrate HEVC Video Coding Using Deep Learning Based Super-Resolution and Adaptive Block Patching". *IEEE Transactions on Multimedia*, 21(12):3010–3023, 2019.

[111] Shengwei Yu, Xun Tong, Yan Huang, Rong Xie, and Li Song. "Learning-Based Quality Enhancement for Scalable Coded Video Over Packet Lossy Networks". In *Proceedings of 2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020.

[112] Fan Zhang, Mariana Afonso, and David R Bull. "Enhanced Video Compression Based on Effective Bit Depth Adaptation". In *Proceedings of 2019 IEEE International Conference on Image Processing (ICIP)*, pages 1720–1724. IEEE, 2019.

[113] Longtao Feng, Xinfeng Zhang, Xiang Zhang, Shanshe Wang, Ronggang Wang, and Siwei Ma. "A Dual-Network Based Super-Resolution for Compressed High Definition Video". In *Proceedings of Pacific Rim Conference on Multimedia*, pages 600–610. Springer, 2018.

[114] Yue Li, Dong Liu, Houqiang Li, Li Li, Feng Wu, Hong Zhang, and Haitao Yang. "Convolutional Neural Network-Based Block Up-Sampling for Intra Frame Coding". *IEEE Transactions on Circuits and Systems for Video Technology*, 28(9):2316–2330, 2017.

[115] Zhenyu Liu, Xianyu Yu, Yuan Gao, Shaolin Chen, Xiangyang Ji, and Dongsheng Wang. "CU Partition Mode Decision for HEVC Hardwired Intra Encoder Using Convolution Neural Network". *IEEE Transactions on Image Processing*, 25(11):5088–5103, 2016.

[116] Nan Song, Zhenyu Liu, Xiangyang Ji, and Dongsheng Wang. "CNN Oriented Fast PU Mode Decision for HEVC Hardwired Intra Encoder". In *Proceedings of 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 239–243. IEEE, 2017.

[117] Jingyao Xu, Mai Xu, Yanan Wei, Zulin Wang, and Zhenyu Guan. "Fast H. 264 to HEVC Transcoding: A Deep Learning Method". *IEEE Transactions on Multimedia*, 21(7):1633–1645, 2018.

[118] Zhipeng Jin, Ping An, Liquan Shen, and Chao Yang. "CNN Oriented Fast QTBT Partition Algorithm for JVET Intra Coding". In *Proceedings of 2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2017.

[119] Mai Xu, Tianyi Li, Zulin Wang, Xin Deng, Ren Yang, and Zhenyu Guan. "Reducing Complexity of HEVC: A Deep Learning Approach". *IEEE Transactions on Image Processing*, 27(10):5044–5059, 2018.

[120] Jun-Hao Hu, Wen-Hsiao Peng, and Chia-Hua Chung. "Reinforcement Learning for HEVC/H. 265 Intra-Frame Rate Control". In *Proceedings of 2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2018.

[121] Ye Li, Bin Li, Dong Liu, and Zhibo Chen. "A Convolutional Neural Network-Based Approach to Rate Control in HEVC Intra Coding". In *Proceedings*

of 2017 IEEE Visual Communications and Image Processing (VCIP), pages 1–4. IEEE, 2017.

[122] Aaditya Prakash, Nick Moran, Solomon Garber, Antonella DiLillo, and James Storer. "Semantic Perceptual Image Compression Using Deep Convolution Networks". In *Proceedings of 2017 Data Compression Conference (DCC)*, pages 250–259. IEEE, 2017.

[123] Tong Chen, Haojie Liu, Qiu Shen, Tao Yue, Xun Cao, and Zhan Ma. "Deepcoder: A Deep Neural Network Based Video Compression". In *Proceedings of 2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2017.

[124] Feng Jiang, Wen Tao, Shaohui Liu, Jie Ren, Xun Guo, and Debin Zhao. "An End-to-End Compression Framework Based on Convolutional Neural Networks". *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):3007–3018, 2017.

[125] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. "Video Compression Through Image Interpolation". In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 416–431, 2018.

[126] Zhibo Chen, Tianyu He, Xin Jin, and Feng Wu. "Learning for Video Compression". *IEEE Transactions on Circuits and Systems for Video Technology*, 30(2):566–576, 2019.

[127] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. "Learning Image and Video Compression Through Spatial-Temporal Energy Compaction". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[128] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. "Dvc: An End-to-End Deep Video Compression Framework". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11006–11015, 2019.

[129] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Li Chen, Zhiyong Gao, and Dong Xu. "An End-to-End Learning Framework for Video Compression". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[130] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. "M-LVC: Multiple Frames Prediction for Learned Video Compression". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3546–3554, 2020.

[131] Abdelaziz Djelouah, Joaquim Campos, Simone Schaub-Meyer, and Christopher Schroers. "Neural Inter-Frame Compression for Video Coding". In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6421–6429, 2019.

[132] Lu Zonglei and Xu Xianhong. "Deep Compression: A Compression Technology for Apron Surveillance Video". *IEEE Access*, 7:129966–129974, 2019.

[133] Negin Ghamsarian, Hadi Amirpourazarian, Christian Timmerer, Mario Taschwer, and Klaus Schöffmann. "Relevance-Based Compression of Cataract Surgery Videos Using Convolutional Neural Networks". In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3577–3585, 2020.

[134] Lirong Wu, Kejie Huang, Haibin Shen, and Lianli Gao. "Foreground-Background Parallel Compression with Residual Encoding for Surveillance Video". *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[135] Karthick Panneerselvam, K Mahesh, VL Josephine, and A Ranjith Kumar. "Effective and Efficient Video Compression by The Deep Learning Techniques". *Computer Systems Science & Engineering*, 45(2), 2023.

[136] Andrew Hope. "CCTV, School Surveillance and Social Sontrol". *British Educational Research Journal*, 35(6):891–907, 2009.

[137] Eric L Piza, Brandon C Welsh, David P Farrington, and Amanda L Thomas. "CCTV Surveillance for Crime Prevention: A 40-Year Systematic Review with Meta-Analysis". *Criminology & public policy*, 18(1):135–159, 2019.

[138] Matthew PJ Ashby. "The Value of CCTV Surveillance Cameras as An Investigative Tool: An Empirical Analysis". *European Journal on Criminal Policy and Research*, 23(3):441–459, 2017.

[139] *"The Maximum Surveillance Society: The Rise of CCTV", author=Armstrong, Gary and Norris, Clive.* Routledge, 2020.

[140] Muhammad Adil, Saqib Mamoon, Ali Zakir, Muhammad Arslan Manzoor, and Zhichao Lian. "Multi Scale-Adaptive Super-Resolution Person Re-Identification Using GAN". *IEEE Access*, 8:177351–177362, 2020.

[141] "Oxford Dictionaries: Dataset Definition". `https://www.oxfordlearnersdictionaries.com/definition/english/data-set`. Accessed 10/15/2021.

[142] 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 *Proceedings of Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[143] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. "CSPNet: A New Backbone That can Enhance Learning Capability of CNN". In *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 390–391, 2020.

[144] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. "Path Aggregation Network for Instance Segmentation". In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018.

[145] Nikita Mohod, Prateek Agrawal, and Vishu Madan. "Human Detection in Surveillance Video using Deep Learning Approach". In *Proceedings of 2023 6th International Conference on Information Systems and Computer Networks (ISCON)*, pages 1–6. IEEE, 2023.

[146] Chien-Yao Wang, Hong-Yuan Mark Liao, and I-Hau Yeh. "Designing Network Design Strategies Through Gradient Path Analysis. *arXiv preprint arXiv:2211.04800*, 2022.

[147] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity". *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[148] Nikita Mohod and Vishu Madaan. "Deep Learning-Based Video Compression for Surveillance Footage". *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(9):787–795, October 2023.

Appendix A

Research Outcomes

1. **Journal Paper**

   i. Mohod Nikita, Prateek Agrawal, and Vishu Madan, "A Novel Approach for Surveillance Compression using Neural Network Technique." International Research Journal of Multidisciplinary Technovation 6(3), pp.77-89, 2024.

   ii. Mohod Nikita, Prateek Agrawal, and Vishu Madan, "Deep Learning-Based Video Compression for Surveillance Footage " International Journal on Recent and Innovation Trends in Computing and Communication, 11(9), 787–795, https://doi.org/10.17762/ijritcc.v11i9.8967, 2023.

   iii. Mohod Nikita, Prateek Agrawal, and Vishu Madan, "Deep Learning-Based Pertinent Video Frame Detection to Compressed Surveillance Video". Submitted Second Review to the SNComputerScience Journal.

   iv. Mohod Nikita, Prateek Agrawal, and Vishu Madan, Submitted first review to the peer journal on the topic "Frame Relevance Based Video Compression for Surveillance Videos using Deep Learning Methods".

2. **Conference Paper**

   i. Mohod Nikita, Prateek Agrawal, and Vishu Madan, "YOLOv4 vs YOLOv5: Object detection on surveillance videos," Proceeding of International Conference on Advanced Network Technologies and Intelligent Computing. Springer, pp. 654–665, 2022.

ii. Mohod Nikita, Prateek Agrawal, and Vishu Madan, "Human Detection in Surveillance Video using Deep Learning Approach." Proceeding of 2023 6th International Conference on Information Systems and Computer Networks (ISCON), pp. 1-6. IEEE, 2023.

iii. Mohod Nikita, Prateek Agrawal, and Vishu Madan, "Systematic Review on Various Deep Learning Models for Object Detection in Videos." Proceeding of 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), pp. 781-788. IEEE, 2023.

3. **Copyright**

i. Copyright on topic "Deep Learning-Driven Compression of Surveillance Videos: Emphasizing Relevance for Optimal Encoding".

4. **Patent**

i. Patent Published on the topic "Methods and System for Relevance-Based Video Compression of CCTV Surveillance Videos Using Deep Learning Techniques" with application no. 202211063759 A on date 18/11/2022.

ii. Patent Published on the topic "A System for Compressing Surveillance Video" with application no. 202411079566 A on date 01/11/2024.