

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**COMPRESSED DOMAIN VIDEO UNDERSTANDING METHODS
FOR
TRAFFIC SURVEILLANCE APPLICATIONS**

Ph.D. THESIS

Muhammet Sebul BERATOĞLU

Computer Science Department

Computer Science Program

MAY 2023

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**COMPRESSED DOMAIN VIDEO UNDERSTANDING METHODS
FOR
TRAFFIC SURVEILLANCE APPLICATIONS**

Ph.D. THESIS

**Muhammet Sebul BERATOĞLU
(704032001)**

Computer Science Department

Computer Science Program

Thesis Advisor: Prof. Dr. Behçet Uğur TÖREYİN

MAY 2023

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**TRAFİK İZLEME UYGULAMALARI İÇİN
SİKİŞTİRİLMİŞ ALANDA
VİDEO ANLAMLANDIRMA YÖNTEMLERİ**

DOKTORA TEZİ

**Muhammet Sebul BERATOĞLU
(704032001)**

Bilgisayar Bilimleri Anabilim Dalı

Bilgisayar Bilimleri Programı

Tez Danışmanı: Prof. Dr. Behçet Uğur TÖREYİN

MAYIS 2023

Muhammet Sebul BERATOĞLU, a Ph.D. student of ITU Graduate School student ID 704032001 successfully defended the thesis entitled “COMPRESSED DOMAIN VIDEO UNDERSTANDING METHODS FOR TRAFFIC SURVEILLANCE APPLICATIONS”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Behçet Uğur TÖREYİN**
Istanbul Technical University

Jury Members : **Prof. Dr. Behçet Uğur Töreyn**
Istanbul Technical University

Prof. Dr. Lütfiye Durak ATA
Istanbul Technical University

Prof. Dr. Muhittin GÖKMEN
MEF University

Asst. Prof. Tankut AKGÜL
Istanbul Technical University

Asst. Prof. Nurullah ÇALIK
Istanbul Medeniyet University

Date of Submission : **30 December 2022**

Date of Defense : **30 May 2023**

To my spouse and children,

FOREWORD

First and foremost, I would like to express my profound gratitude to my advisor, Professor Behçet Uğur TÖREYİN, for his invaluable guidance and support throughout the duration of this project. His expertise and wisdom have been pivotal in shaping my research and maintaining my focus and motivation.

My heartfelt thanks go to Professor Lütfiye DURAK ATA, whose faith in me and unwavering support throughout this process have been invaluable. She has been an incredible source of inspiration and a role model with her steadfast dedication and pursuit of excellence in all her responsibilities.

I am also deeply grateful to Professor Muhittin GÖKMEN for his constant support throughout my academic and professional career. His invaluable recommendations and ideas during this thesis have significantly contributed to the success of this work. I am thankful for his presence in all the significant events of my life.

I would like to extend my thanks to Dr. Kemal UĞUR for his assistance in understanding the intricacies of HEVC. I am also thankful to Asst. Prof. Tankut AKGÜL for answering my questions and providing valuable insights. Moreover, my sincere gratitude goes to Asst. Prof. Nurullah ÇALIK, who has consistently been supportive and helpful. Additionally, I would like to extend my appreciation to my friends, especially Dr. Abdulkerim ÇAPAR, for his belief, encouragement, and support. I am also grateful to my company, Divit Technology, and my colleagues for their support, including providing the necessary data for this research. My sincere thanks also go to the Informatics Institute and all its staff for their kindness and support.

Lastly, I would like to express my deepest gratitude to my family. Though my mom and dad are no longer with me, their support has been with me throughout my life. I am especially grateful to my wonderful wife, Filiz, for her unwavering support and encouragement, and to my daughters, Zeynep and Fatma Nur, for their understanding and patience during the long hours spent on this project. Without their love and support, this work would not have been possible.

This study was jointly supported by the ITÜ BAP project (agreement no. MGA-2017-40964), and the Scientific and Technical Research Council of Turkey (TUBITAK) under grants 121E378 and TEYDEB 3190538.

May 2023

Muhammet Sebul BERATOĞLU
MSc., Computer Engineer

TABLE OF CONTENTS

	<u>Page</u>
FOREWORD	ix
TABLE OF CONTENTS	xi
ABBREVIATIONS	xiii
SYMBOLS	xv
LIST OF TABLES	xvii
LIST OF FIGURES	xix
SUMMARY	xxi
ÖZET	xxiii
1. INTRODUCTION	1
1.1 Motivation and Problem Definition	1
1.2 Scope of The Thesis	3
1.3 Contributions	3
1.4 Outline	5
2. RELATED WORKS	7
2.1 Related Works in Compressed Domain	8
2.2 Related Works in License Plate Detection	11
2.3 Related Works in Vehicle Detection and Classification	12
3. BACKGROUND	15
3.1 High Efficiency Video Coding (HEVC)	15
3.1.1 Basic concepts	16
3.1.1.1 Color coding	16
3.1.1.2 Picture partitioning	17
3.1.1.3 Prediction	18
3.1.1.4 Transform and quantization	19
3.1.1.5 Entropy coding	20
3.1.1.6 Deblocking filter	20
3.1.1.7 Sample adaptive offset filter	20
3.1.1.8 Profiles	20
3.1.2 HEVC intra prediction	21
3.1.2.1 Reference samples	22
3.1.2.2 Angular intra prediction	22
3.1.2.3 Planar prediction	22
3.1.2.4 DC intra prediction	23
3.2 Object Detection	23
3.2.1 Detection algorithms	24
3.2.2 Convolutional neural networks	24
3.2.3 YOLO	27
4. PARTIAL INTRA DECODING METHODS FOR COMPRESSED VIDEO UNDERSTANDING	31
4.1 Introduction	31
4.2 Block Partition Based Method	35
4.3 Prediction Unit Based Method	37
4.4 Random Perturbation Based Method	39
4.4.1 Standard reconstruction	41
4.4.2 Impact of ignoring residuals	41
4.4.3 Impact of constant residuals	41
4.4.4 Random perturbations as a substitute for residuals	42
4.5 Luma Based Method	45
4.6 Visual Comparison of Reconstructed Images	46
4.7 Experimental Results	49
4.7.1 Experiment setup	49
4.7.2 Measurement of reconstruction steps	50

4.7.3 Comparison of image reconstruction time	51
4.7.4 Comparison of data size	53
4.8 Conclusion	56
5. LICENSE PLATE DETECTION IN COMPRESSED DOMAIN	59
5.1 Methodology	59
5.2 Experimental Results	61
5.2.1 Experiment setup	61
5.2.2 Datasets	62
5.2.2.1 CD-LP dataset	63
5.2.2.2 EnglishLP dataset	63
5.2.3 Metrics	64
5.2.4 Accuracy	64
5.2.4.1 CD-LP dataset accuracy	65
5.2.4.2 EnglishLP dataset accuracy	65
5.2.5 Time comparison	67
5.2.5.1 Image reconstruction	67
5.2.5.2 LP detection	68
5.2.5.3 Overall process for LP detection	68
5.3 Conclusion	69
6. VEHICLE DETECTION AND CLASSIFICATION IN COMPRESSED DOMAIN	71
6.1 Methodology	71
6.2 Experimental Results	72
6.2.1 Experiment setup	72
6.2.2 BIT vehicle dataset	75
6.2.3 Metrics	76
6.2.4 Vehicle detection accuracy	78
6.2.5 Vehicle classification accuracy	78
6.2.6 Time comparison	82
6.3 Conclusion	84
7. CONCLUSIONS	85
7.1 Discussion and Future Work	85
7.2 Closing Remarks	87
REFERENCES	89
CURRICULUM VITAE	99

ABBREVIATIONS

ALPR	: Automatic License Plate Recognition
AP	: Average Precision
BP	: HEVC Block Partition
CABAC	: Context Adaptive Binary Arithmetic Coding
CB	: Coding Block
CNN	: Convolutional Neural Network
CTB	: Coding Tree Block
CTU	: Coding Tree Unit
CU	: Coding Unit
DCT	: Discrete Cosine Transform
DPM	: Deformable Part-Based Model
DST	: Discrete Sine Transform
ED	: Entropy Decoding
FP	: False Positive
HEVC	: High Efficiency Video Coding
HEVC	: High Efficiency Video Coding
HOG	: Histograms of Oriented Gradients
IP	: Intra Prediction
LP	: License Plate Detection
LPR	: License Plate Recognition
LP	: Licence Plate
mAP	: Mean Average Precision
MPEG	: Moving Picture Experts Group
NMS	: Non-Maximum Suppression
PB	: Predicted Block
PU	: Predicted Unit
RD	: Residual Decompression
ReLU	: Rectified Linear Unit
ROI	: Region of Interest
SAO	: Sample Adaptive Offset
SGD	: Stochastic Gradient Descent
SIFT	: Scale-Invariant Feature Transform
SPP	: Spatial Pyramid Pooling
SVM	: Support Vector Machine
TB	: Transfer Block
TP	: True Positive
TU	: Transfer Unit
VDC	: Vehicle Detection and Classification

SYMBOLS

I_{bp}	: Block Partition Based Image
I_{pu}	: Prediction Unit Based Image
I_{rp}	: Random Perturbation Based Image
I_y	: Luma Based Image
I_{px}	: Pixel Domain Image
I_{org}	: The source image before any transformations
M_{bp}	: Block Partition Based Method
M_{pu}	: Prediction Unit Based Method
M_{rp}	: Random Perturbation Based Method
M_y	: Luma Based Method
M_{px}	: Standard HEVC Decoding Method
LPD_{bp}	: License Plate Detection using Block Partition Based Images
LPD_{pu}	: License Plate Detection using Prediction Unit Based Images
LPD_{px}	: License Plate Detection using Pixel Domain Images
V_{bp}	: Vehicle Classification using Block Partition Based Images
V_{pu}	: Vehicle Classification using Prediction Unit Based Images
V_{px}	: Vehicle Classification using Pixel Domain Images
V_{rp}	: Vehicle Classification using Random Perturbation Based Images
V_y	: Vehicle Classification using Luma Based Images

LIST OF TABLES

	<u>Page</u>
Table 4.1 : Quantitative analysis of image reconstruction steps.....	51
Table 4.2 : A comparative analysis of image reconstruction durations across different methods.....	53
Table 4.3 : Comparison of image sizes and depths across different reconstruction methods.....	53
Table 5.1 : Distribution of the CD-LP dataset across different image types and resolutions.....	63
Table 5.2 : Distribution of the EnglishLP dataset across different image types and resolutions.....	64
Table 5.3 : Evaluation of accuracy metrics across different methods on the CD-LP dataset.....	65
Table 5.4 : Evaluation of accuracy metrics across different methods on the EnglishLP dataset.....	67
Table 5.5 : Comparison of average image reconstruction durations across different methods.....	67
Table 5.6 : Processing time for LP detection using YOLOv3 Tiny at various image resolutions.....	68
Table 5.7 : Comparison of the duration of the entire LP detection process.....	68
Table 6.1 : Vehicle detection accuracy for different methods.....	78
Table 6.2 : Vehicle classification accuracy comparison of random perturbation based images generated with various standard deviations.....	79
Table 6.3 : Vehicle Classification Accuracy for Different Methods.....	80
Table 6.4 : Vehicle Detection & Classification Accuracy vs Time.....	83

LIST OF FIGURES

	<u>Page</u>
Figure 1.1 : The general framework for object detection in the pixel and compressed domains.	2
Figure 3.1 : The basic block diagram of an HEVC encoder and decoder.	16
Figure 3.2 : The HEVC block partitioning structure. a) The division of an image into CTUs of equal size. b) Subdivision of a CTU into CUs..	18
Figure 3.3 : Intra prediction directions. a) The nine intra prediction modes of H.264. b) The thirty five intra prediction modes of H.265 (HEVC)..	21
Figure 3.4 : Reference samples used in prediction to obtain predicted samples. .	22
Figure 3.5 : An example of CNN architecture for image classification.	26
Figure 4.1 : Five methods for HEVC decoding of intra-coded frames. M_{bp} : Block partition based method. M_{pu} : Prediction unit based method. M_{rp} : Random perturbation based method. M_y : Luma based method. M_{px} : Standard HEVC decoding method.	34
Figure 4.2 : A sample I_{bp} image. (a) Pixel domain image of a vehicle, (b) Corresponding Block Partition Based Image (I_{bp}) of the vehicle. ...	36
Figure 4.3 : A sample I_{pu} image. (a) Pixel domain image of a vehicle, (b) Corresponding Prediction Unit-Based Image (I_{pu}) of the vehicle. ...	38
Figure 4.4 : Reconstructing CTUs with HEVC standard decoding process and with random perturbation based residual substitution.	40
Figure 4.5 : Comparison of different approaches to substitute residuals.(a) Standard reconstruction using HEVC. (b) Impact of ignoring residuals, $R(x,y) = 0$. (c) Impact of constant residual, $R(x,y) = 1$. (d) Random perturbations as a substitute for residuals, I_{rp} with $\mu = 0$ and $\sigma = 7$	43
Figure 4.6 : Impact of standard deviation variations on image reconstruction in the Random Perturbation Based Method.	44
Figure 4.7 : A sample I_y image.	47
Figure 4.8 : Comparison of four different reconstructed images from various HEVC bitstreams. a) Pixel Domain Images (I_{px}). b) Block Partition Based Images (I_{bp}). c) Prediction Unit Based Images (I_{pu}). d) Random Perturbation Based Image (I_{rp}). e) Luma Based Images (I_y).	48
Figure 4.9 : Diagram showing the measurement of steps in the image reconstruction process.	51
Figure 4.10 : Comparison of image generation times for various methods. Image generation in the compressed domain is 28% to 79% faster than in the pixel domain.	54

Figure 4.11 : Comparison of the size of the pixel domain image and the compressed domain image. a) A 1024×768 pixel domain image. b) A compressed domain image corresponding to the pixel domain image with a resolution of 128×96 pixels, where each CU is represented by a single pixel.	55
Figure 5.1 : The generation of High Efficiency Video Coding (HEVC) images and detection of License Plates (LPs) using three separate methods.	60
Figure 5.2 : Results for three different methods of LP detection from the test database. Plates that have been found are indicated with a pink rectangle. a) Detection results of LPD_{px} . b) Detection results of LPD_{bp} . c) Detection results of LPD_{pu}	60
Figure 5.3 : Creating HEVC domain images from JPEG Images.	62
Figure 5.4 : The LP detection results for the CD-LP dataset. The performance of a compressed domain strategy based on prediction units is comparable to that of the pixel domain.	66
Figure 5.5 : The LP detection results for the publicly accessible English-LP dataset.	66
Figure 6.1 : Vehicle detection and classification using compressed domain images.	74
Figure 6.2 : Creating HEVC domain images from JPEG images.	76
Figure 6.3 : The vehicle type distribution of the BIT Dataset.	77

COMPRESSED DOMAIN VIDEO UNDERSTANDING METHODS FOR TRAFFIC SURVEILLANCE APPLICATIONS

SUMMARY

Video surveillance has become an integral component of modern traffic monitoring and management systems. The extensive network of cameras deployed worldwide captures a considerable amount of road and highway footage, leading to a daunting task in terms of efficient analysis and comprehension. Traditional methods for video stream analysis are resource-intensive, requiring significant computational power and time. To mitigate these challenges, this thesis introduces a novel set of methods specifically designed for traffic surveillance applications that operate in the compressed domain.

In this thesis, we introduce four new methods for reconstructing images from High Efficiency Video Coding (HEVC) Intra bitstreams and compare them with standard decoding in terms of speed and size. These methods - the Block Partition Based Method (M_{bp}), the Prediction Unit Based Method (M_{pu}), the Random Perturbation Based Method (M_{rp}), and the Luma based method (M_y) - aim to provide a condensed representation of the original image while retaining information pertinent to video understanding tasks.

Key highlights of the conducted study within this thesis include:

- Compression algorithms used in video transmission primarily aim to minimize data transfer. Notably, for Intra coded frames, residual information accounts for up to 90% of the transmitted data. Our proposed methods, including M_{bp} , M_{pu} , and M_{rp} , facilitate object detection without the need for residual data, leading to a substantial reduction in data transmission requirements.
- The compressed domain images created via the M_{bp} and M_{pu} methods occupy significantly less memory compared to the pixel domain image. Specifically, the image created with M_{bp} occupies 1/1,536 of the memory required by the pixel domain image, while the image created with M_{pu} requires 1/192 of the memory. This substantial reduction presents an appealing alternative, especially for applications where memory limitations matter.
- The proposed methods result in a computational speedup between 1.25 to 4 times relative to the pixel area. Comprehensive performance and speed comparisons allow us to identify the most suitable method for specific video analysis requirements.
- In a publicly available dataset, vehicle license plate locations were detected with 93.33% accuracy using M_{bp} , and with 99.02% accuracy using M_{pu} , matching the performance level in the pixel area. Furthermore, the M_{bp} and M_{pu} methods sped up the license plate location finding process nearly four and approximately 3.6 times, respectively.

- M_{rp} , the first of its kind to reconstruct images without using residual data in video transmission, was introduced and its performance evaluated for vehicle detection and classification processes. M_{rp} offers potential for adaptation in solving various classification problems.
- In the publicly available dataset, vehicle detection was achieved with 98.99% accuracy using M_{pu} and with an impressive 99.99% accuracy using M_{rp} . The performance of M_{rp} matched the pixel area's level, with the Luma based method (M_y) also showing comparable performance. Moreover, the vehicle detection process saw speed enhancements of over 1.5 times with M_{rp} and more than 1.25 times with M_y .
- In the compressed domain, vehicles were classified into six different categories. To our knowledge, this is the first such study relating to vehicle classification in the compressed domain. Performance rates achieved were 95.35% with M_{pu} , 96.84% with M_{rp} , and 97.48% with M_y . These results surpassed many studies previously reported in the pixel domain. In the pixel area, a performance of 97.82% was achieved with the classification made using the YOLOV7 Tiny deep learning model, surpassing all previously reported performances for this dataset.

Several promising directions for future work emerge from this study. These include extending the proposed methods to full video decoding by incorporating both Intra and Inter frames into the analysis process, exploring the inclusion of additional attributes from the compressed domain to enhance accuracy and performance, and investigating the potential of combining compressed and pixel domain information to achieve even better results.

In conclusion, the results of this study highlight the remarkable potential of using compressed domain image understanding methods for video analysis tasks in traffic surveillance applications. The innovative methods proposed herein constitute a significant stride towards harnessing the attributes produced during the HEVC encoding process, combined with deep learning techniques, to efficiently detect and classify vehicles. This approach substantially reduces the computational cost of the analysis process, enabling the timely and cost-effective analysis of large volumes of video data, thus paving the way towards more efficient traffic monitoring and management.

TRAFİK İZLEME UYGULAMALARI İÇİN SIKIŞTIRILMIŞ ALANDA VIDEO ANLAMLANDIRMA YÖNTEMLERİ

ÖZET

Günümüzde, güvenlik kameraları, trafik izleme ve yönetim sistemlerinin olmazsa olmaz birer bileşeni olarak, akıllı şehirlerin temel taşlarından birini oluşturur. Bu kameralar, otoyollardan şehirlerin giriş ve çıkış noktalarına, caddelerden sokaklara kadar her yerde araçların hareketlerini izlemekte ve kaydetmektedir. Çeşitli kameraların oluşturduğu bu geniş ağ, büyük veri yığınları oluşturur ve bu verilerin etkin bir şekilde analiz edilebilmesi, ciddi bir işlem gücü gerektirir. Yapılan çalışmada, video verilerinin daha etkin bir şekilde analiz edilmesini sağlayabilecek, sıkıştırılmış alanda video anlama yöntemleri önerilmiştir.

HEVC (High Efficiency Video Coding) ya da diğer adıyla H.265 video kodlama standardı, video dosyalarını, görüntü kalitesinden önemli ölçüde fedakarlık etmeden, ciddi bir biçimde sıkıştırabilir. Ancak, bu sıkıştırma işleminin kendisi de oldukça yoğun bir hesaplama kapasitesi talep eder. Geleneksel yöntemler genellikle video analizi sırasında, ilk olarak görüntüyü tamamen açmayı ve daha sonra analiz etmeyi gerektirir. Ancak sıkıştırılmış verinin içinde yer alan bilgileri, video anlamlı bir biçimde açılabilmesi ve analiz edilebilmesi için kullanmak, görüntüyü yeniden oluşturma işlemini önemli ölçüde hafifletebilir. Bu yaklaşım, büyük miktarda video verisinin daha hızlı ve verimli bir şekilde analiz edilmesine imkan tanır.

Bu tezin amacı, sıkıştırılmış video verilerini tümüyle açmadan (full decoding) anlamlandırılmasını sağlayacak yöntemler geliştirmektir. Bu kapsamda, sıkıştırılmış HEVC verisini kısmı geri çözmeye yarayan 4 yöntem önerilmiştir:

1. Blok Bölütleme Tabanlı Görüntü Geri Çatma Yöntemi (M_{bp}): Bu yöntem, HEVC Blok bölütleme öz nitelikleri kullanılarak elde edilen görüntülere dayanır. Görüntüde yüksek bilgi içeren bölgeler daha küçük bloklarla temsil edilmektedir. Görüntüde, plaka alanı gibi, yoğun gri-seviye değişimi gösteren bölgeler daha küçük bloklarla kodlanmaktadır. Önerilen yöntem, sıkıştırmanın bu karakteristiğini nesne algılama için kullanmıştır.
2. Tahmin Bilgisi Tabanlı Görüntü Geri Çatılması Yöntemi (M_{pu}). HEVC bölümlenme yaptığı her bir blok için bir de komşu bloklara benzerliğini ifade eden bir tahmin değeri oluşturur. Bu tahmin değeri, bir görüntü olarak ifade edildiğinde, nesnelerin daha iyi tespit edilmesine yardımcı olabilir. Önerilen yöntem, tahmin değerlerine karşı bu değerlerle orantılı gri seviye atamaları yaparak görüntü oluşturur. Bu görüntüler ile nesnelere başarılı bir şekilde algılama yapılabilmektedir.
3. Rastgele Pertürbasyon Tabanlı Görüntü Geri Çatılması Yöntemi (M_{rp}). HEVC standardı bloklar ve tahmin yönlerini kullanarak bir tahmin görüntüsü oluşturur.

Orjinal görüntü ile tahmin görüntüsü arasındaki farkı temsil eden artık bilgi, kodlanarak iletilir. Görüntü geri çatılırken, artık bilgi çözülerek oluşturulan tahmin bilgisine eklenir. Bu süreçte, artık bilginin geri çözülmesi önemli bir zaman alır. Kodlamanın doğası gereği sıkıştırma işlemi blok tabanlı yapılıdır. Her bir blok için, tahmin ve artık bilgi birlikte kullanılır ve bir sonraki blok için gerekli veri oluşturulur. Bu durum, artık bir bilgi olmadan görüntüyü elde edebilmeyi zorlaştırır. Önerilen yöntem, artık bilginin yerini alabilecek tutabilecek rastgele pertürbasyon tabanlı değerler ile görüntünün geri çatılmasını sağlamaktadır. Bu yöntem, nesne tanıma başarısını, piksel bazında elde edilebilecek başarı seviyesine yaklaştırmıştır.

4. Gri-Seviye Görüntü Geri Çatılması Yöntemi (M_y). Bu yöntem, renkli olarak sıkıştırılmış bir videoyu gri seviye olarak açma fikrine dayanır. HEVC standardı gri seviye kodlama izin vermektedir. Ancak renkli olarak sıkıştırılmış video, geri çatma işlemi sırasında renkli olarak açılmaktadır. Önerilen yöntemle, gri seviye geri çözme işlemi, nesne algılamadaki başarı kaybını minimuma indirirken, görüntüyü daha hızlı bir şekilde geri çözülebilmektedir.

Önerilen yöntemlerle, piksel alanında alınabilecek nesne tanıma başarımlarına yakın sonuçlar elde edilmiştir. Nesne tanıma erken, işlem hem daha hızlı gerçekleştirmekte, hem de transfer edilen veri gereksinimi ile kullanılan veri miktarını önemli ölçüde düşürmektedir.

Tez kapsamında yapılan çalışmanın öne çıkan katkıları aşağıda sıralanmıştır:

- Video iletiminde sıkıştırma algoritmaları aktarılan veri miktarını azaltmayı hedeflemektedir. İletilen verinin %90'a varan bölümünü artık bilgi (residual data) oluşturmaktadır. M_{bp} , M_{pu} ve M_{rp} yöntemleri artık bilgiye gereksinim duymadan nesne tanıma yapabilmektedir. Önerilen yöntemlerin kullanımı, iletilen veri miktarının önemli ölçüde düşürülmesini sağlar.
- M_{bp} ve M_{pu} yöntemleri ile oluşturulan sıkıştırılmış alan görüntüsü, piksel alanı görüntüsüne kıyasla çok daha düşük bir bellek yer kaplar. Spesifik olarak, M_{bp} ile oluşturulan görüntü, piksel alanı görüntüsünün sadece 1/1,536'sı kadar, M_{pu} ile oluşturulan görüntü ise 1/192'si kadar bellekte yer tutar. Bu önemli düşüş, özellikle bellek kısıtlamalarının olduğu uygulamalar için çekici bir alternatif sunar.
- Önerilen yöntemler piksel alanına göre 1,25 ila 4 kat arasında hızlanma sağlamaktadır. Başarım ve hız mukayesesi yapılarak, geliştirilmek istenen video analiz uygulamasının hedeflerine uygun yöntemin belirlenmesi sağlanmıştır.
- Erişime açık veri kümesinde, taşıt plaka yerinin; M_{bp} yöntemi kullanılarak %93,33 başarımla, M_{pu} yöntemi kullanılarak %99.02 başarımla (piksel alanında elde eşdeğer düzeyde) bulunabildiği gösterilmiştir. Plaka yer bulma süreci M_{bp} yöntemi ile 4 kattan, M_{pu} yöntemi ile 3,6 kattan daha fazla hızlandırılmıştır.
- M_{rp} , video aktarımında yer alan artık bilgi kullanılmadan görüntünün yeniden çatılabilmesi için, bildiğimiz kadarıyla, önerilen ilk yöntemdir. Başarımı, tez kapsamında, araç algılama ve sınıflandırma işlemi için raporlanmıştır. İlgili yöntem, pek çok farklı sınıflandırma probleminin çözümü için uyarlanabilir.

- Erişime açık veri kümesinde, taşıt algılamanın; M_{pu} yöntemi kullanılarak %98.99 başarımla, M_{rp} yöntemi kullanılarak %99,99 başarımla tespit edilebildiği gösterilmiştir. M_{rp} yönteminin başarımı piksel alanında elde edilebilen başarıma eşdeğerdir. M_y yöntemi de aynı başarımları göstermiştir. Araç algılama süreci M_{rp} yöntemiyle 1,5 kattan, M_y yöntemi ile 1,25 kattan daha fazla hızlandırılmıştır.
- Taşıtların, sıkıştırılmış alanda, 6 farklı kategoride sınıflandırılabilirdiği gösterilmiştir. Yapılan çalışma, sıkıştırılmış alanda taşıt sınıflandırma ile ilgili, bildiğimiz kadarıyla, ilk olmuştur. M_{pu} yöntemi kullanımıyla %95.35 başarımla, M_{rp} yöntemi kullanımıyla %96.84 başarımla, M_y yöntemi kullanımıyla %97,48 başarımla elde edilmiştir. Sıkıştırılmış alanda elde edilen başarımlar daha önce piksel alanında literatürde raporlanmış pek çok sayıda çalışmayı geçmiştir. Piksel alanında ise, YOLOv7 Tiny derin öğrenme modeli kullanılarak yapılan sınıflandırma ile elde edilen başarımlar %97,82 olmuştur. Bu başarımlar, literatürde ilgili veri kümesi için raporlanan diğer bütün başarımların üzerindedir.

Bu tez kapsamında sunulan yöntemler, sıkıştırılmış alan verisinin derin öğrenme teknikleriyle birleştirilmesiyle büyük miktarda video verilerini etkili bir şekilde analizi mümkün kılmaktadır. Yapılan çalışmaları aşağıda belirtilen yönlerde daha da ilerletilebilir.

- Önerilen yöntemlerde kare-içi (Intra Frame) çerçeveler üzerinde yoğunlaşmıştır. Kareler-arası (Inter Frame) çerçeveler ile kare-içi çerçeveler birleştirilerek yöntemin bir video bütününe uyarlanmasa sağlanabilir.
- Önerilen yöntemler, sadece taşıt ve plaka tanıma işlevleri için değil, nesne tespitinin yapıldığı diğer alanlara da uyarlanabilir genelliktir.
- Piksel alanında başarımın kritik olduğu uygulamalar için iki farklı alandaki veriler birleştirilebilir. Sıkıştırılmış alanda elde edilen öznelikler piksel alanında elde edilen öznelikler ile birleştirilerek daha yüksek yüksek başarımlara ulaşmak araştırılabilir.
- Sıkıştırılmış alanda yapılan çalışmaların özellikle veri mahremiyetinin önemli olduğu uygulama için önemini gösteren çalışmalar yapılabilir.

Sonuç olarak bu tez; video analizi için sıkıştırılmış veri alanını kullanarak veri boyutunu azaltmayı, analiz sürecini hızlandırmayı ve performansı artırmayı hedefleyen yenilikçi teknikler sunmaktadır. Gelecekteki çalışmalar, bu yöntemlerin daha geniş uygulama alanlarına genelleştirilmesini ve video analizi alanında daha büyük bir etki yaratmayı amaçlamaktadır. Bu tez, bu hedeflere doğru yapılan yolculukta önemli bir adımı temsil etmektedir.

1. INTRODUCTION

This thesis lays the foundation for novel techniques and methods that leverage the encoded stream data produced by the High Efficiency Video Coding (HEVC) standard for object detection in the realm of traffic surveillance applications. In the age of Big Data, the evolution of such innovative techniques is imperative to efficiently manage and analyze the deluge of video data generated by traffic surveillance systems across the globe. The proposed techniques are designed to not only reduce the computational demands but also to enhance data transmission efficiency, object detection accuracy, and compliance with data privacy regulations. Furthermore, the thesis represents a pioneering attempt to implement vehicle classification in the compressed domain, a significant stride towards efficient, insightful, and privacy-compliant video surveillance.

1.1 Motivation and Problem Definition

The modern world relies heavily on video surveillance systems, especially for traffic monitoring and management. Across the globe, surveillance cameras capture massive amounts of footage every day, which poses a significant challenge in terms of data analysis and understanding. Traditional techniques for analyzing this video data are resource-intensive, requiring substantial computational power and time. Moreover, these methods often overlook the potential usefulness of the data already calculated during the encoding phase.

With an increasing emphasis on data privacy, the necessity to comply with regulations such as the EU General Data Protection Regulation (GDPR) and the California Privacy Rights Act (CPRA) has made data minimization a crucial consideration in video analytics. These constraints further underline the need for innovative solutions that can efficiently handle data without compromising its utility for essential tasks.

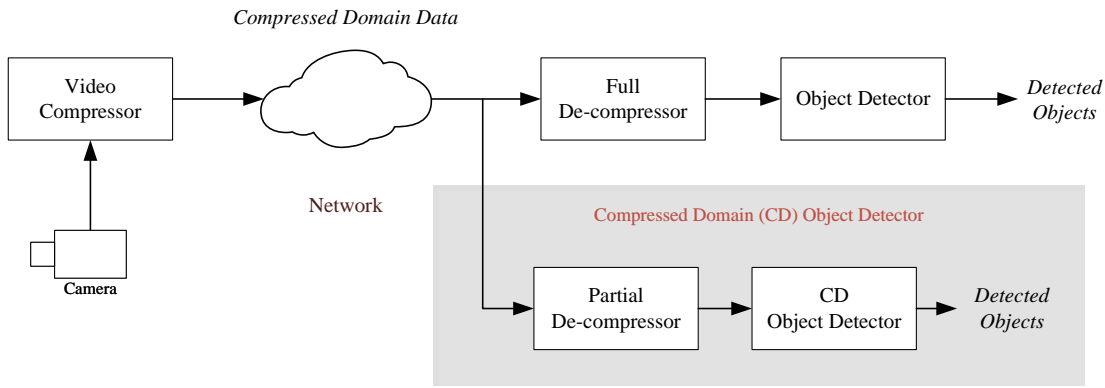


Figure 1.1 : The general framework for object detection in the pixel and compressed domains.

Compression algorithms, like the High Efficiency Video Coding (HEVC) standard, aim to reduce the amount of data transferred, thereby mitigating some of these challenges. However, these algorithms are computationally intensive, demanding significant resources for data compression. Additionally, for certain types of frames, known as Intra coded frames, residual information can make up to 90% of the transmitted data. This residual data, often vital for video understanding tasks, adds to the complexity of data analysis.

To address these issues, we need novel methods that can efficiently compress and analyze video data, leverage the information computed during the encoding process, and comply with data minimization requirements. Such methods could bring about improved data transmission efficiency, reduced computational demands, better performance in video understanding tasks, and enhanced compliance with privacy regulations.

The objective of this thesis, illustrated in Figure 1.1, is to explore such methods. It depicts the general framework for object detection in the pixel and compressed domains, where partial decompression of the encoded stream is used for object detection tasks. This approach could represent a significant stride forward, inaugurating a new age of efficient, insightful, and privacy-compliant video surveillance.

1.2 Scope of The Thesis

The scope of this thesis is centered on the conception, development, and evaluation of methods aimed at exploiting the encoded stream data generated by the High Efficiency Video Coding (HEVC) standard within the realm of traffic surveillance applications. The primary tasks focused upon within this ambit are license plate detection and vehicle classification.

The methods introduced in this thesis, namely the Block Partition Based Method (M_{bp}), the Prediction Unit Based Method (M_{pu}), the Random Perturbation Based Method (M_{rp}), and the Luma based method (M_y), are specifically tailored for intra frame encoded bitstreams. Their potential effectiveness will be assessed on the basis of their accuracy and computational speed for the aforementioned tasks, using established metrics such as Precision, Recall, F1-score, and mean Average Precision (mAP).

The performance evaluation will be carried out using specific video streams and datasets that are appropriate for traffic surveillance applications. This includes publicly available datasets suitable for the tasks of license plate detection and vehicle classification, with a focus on datasets that contain a wide variety of vehicles and license plates under different lighting conditions and viewpoints.

Despite the potential adaptability of the proposed methods for other object detection tasks, the primary attention of this work is riveted on the mentioned applications. The performance of the proposed methods will be gauged using reference software and will be communicated in terms of their performance on a CPU. It's important to note that this thesis will not incorporate data related to GPU or FPGA acceleration.

1.3 Contributions

The primary contributions of this thesis are as follows:

- Introduction of four novel methods: One of the major contributions of this work is the development and introduction of four new methods for efficient video analysis - the Block Partition Based Method (M_{bp}), the Prediction Unit Based Method

(M_{pu}), the Random Perturbation Based Method (M_{rp}), and the Luma based method (M_y). These methods provide a condensed representation of the original image while retaining information pertinent for video understanding tasks and require substantially less memory for storage compared to the pixel domain image.

- Efficient data transmission: Compression algorithms, especially in video transmission, aim primarily to minimize data transfer. Specifically, our proposed methods M_{bp} , M_{pu} , and M_{rp} facilitate object detection without the need for residual data that often comprises up to 90% of the transmitted data in Intra coded frames. This development leads to a significant reduction in data transmission requirements.
- Development of memory efficient reconstruction techniques: The proposed M_{bp} and M_{pu} methods generate compressed spatial images that occupy significantly less memory compared to the pixel domain image. Specifically, the image created with M_{bp} occupies 1/1,536 of the memory required by the pixel domain image, while the image created with M_{pu} requires 1/192 of the memory. This considerable reduction presents a desirable alternative for applications where memory limitations matter.
- Achievement of computational speedup: The proposed methods result in a computational speedup ranging from 1.25 to 4 times relative to the pixel area. This improvement provides a basis for identifying the most suitable method for specific video analysis requirements, making a significant contribution in terms of computational efficiency.
- Improvements in license plate detection: In a publicly available dataset, M_{bp} and M_{pu} successfully detected vehicle license plate locations with 93.33% and 99.02% accuracy respectively, matching the performance level in the pixel area. Importantly, these methods also sped up the license plate location finding process by nearly four and approximately 3.6 times, respectively.
- Introduction and evaluation of M_{rp} : A further significant contribution is the introduction of M_{rp} , the first method of its kind to reconstruct images without using residual data in video transmission. Its performance was evaluated for

vehicle detection and classification processes, indicating that M_{rp} offers potential for adaptation in various classification problems.

- Achievements in vehicle detection: This work demonstrates the efficacy of the proposed methods in vehicle detection tasks using a publicly available dataset. Vehicle detection was achieved with an impressive 98.99% accuracy using M_{pu} and 99.99% accuracy using M_{rp} . The performance of M_{rp} matched the pixel area's level, with the Luma based method (M_y) also showing comparable performance. Furthermore, these methods increased the speed of vehicle detection by over 1.5 times with M_{rp} and more than 1.25 times with M_y .
- Pioneering work in vehicle classification in the compressed domain: For the first time, vehicles were classified into six different categories within the compressed domain. The results obtained were impressive, with performance rates of 95.35% using M_{pu} , 96.84% using M_{rp} , and 97.48% using M_y . These results surpass many studies previously reported in the pixel domain.

1.4 Outline

The structure of this thesis is organized as follows: Chapter 2 summarizes related work in this field. Chapter 3 provides background information on the theories and technologies deployed in this research, including an overview of the High Efficiency Video Coding (HEVC) Intra Coding mechanism and an introduction to object detection methodologies. In Chapter 4, the methods developed for reconstructing images from HEVC intra bitstreams are introduced, with an evaluation of their performance concerning construction time and object detection accuracy compared to pixel domain images. The effectiveness of these methods for license plate detection is assessed in Chapter 5, while Chapter 6 explores their application for vehicle detection and classification in the compressed domain. Finally, Chapter 7 presents the discussion, future work, and conclusion of this research.

2. RELATED WORKS

The field of vehicle detection and classification, particularly license plate detection and recognition, is central to many applications in intelligent transportation systems, autonomous driving, traffic rule enforcement, and surveillance systems. Over the years, an extensive body of research has been developed in these areas, leading to numerous advancements and new techniques [1]–[4]. This chapter provides a comprehensive overview of the current state-of-the-art methods in vehicle detection and classification, license plate detection, and object detection in the compressed domain.

In *Related Works in Compressed Domain*, we review recent developments in object detection methods that operate directly in the compressed domain. This growing area of research leverages the potential benefits of reduced computational demands and real-time processing capabilities. It presents a fascinating new direction in the field, one that our work further explores and contributes to. This section reviews relevant works in this domain, focusing on those that use High Efficiency Video Coding (HEVC), and highlights the distinguishing features of our approach.

In *Related Works in License Plate Detection*, we delve into the evolution of license plate detection and recognition techniques. From traditional image processing methods and hand-crafted feature extractors to the advent of deep learning and convolutional neural networks, this section charts the progression of methods that have made remarkable strides in improving detection accuracy. Our research brings a novel contribution to this field by proposing a method to perform license plate detection directly in the compressed domain.

Finally, in *Related Works in Vehicle Detection and Classification*, we explore current techniques in vehicle detection and classification, with an emphasis on deep learning-based methods. These have proven to be highly effective in the task, particularly due to the ability of such models to process large volumes of data and

extract high-level features from raw pixels. This section also discusses our unique contribution to this domain: a method for vehicle detection and classification in the compressed domain that offers a balance between computational efficiency and accuracy.

This chapter sets the stage for the methodologies and techniques that our work builds upon, providing context and justification for our unique approach and demonstrating how it fits into the larger research landscape.

2.1 Related Works in Compressed Domain

Recent years have witnessed a burgeoning interest in the development of object detection and classification methods directly in the compressed domain [5]–[7]. In this section, we explore a diverse range of these research works and highlight the nuances that set our approach apart.

In a seminal work, Babu et al. [8] compiled an exhaustive survey on compressed domain video analysis techniques, discussing a variety of computer vision applications such as moving object segmentation, human action recognition, indexing, retrieval, face detection, video classification, and object tracking. Their research underscores the scope and limitations of compressed domain features, forming a foundational reference for subsequent studies in this domain.

Zhai et al. [9] provided an all-inclusive overview of object detection methods in the compressed domain, spanning various video compression standards, including MPEG-2, H.264, and High Efficiency Video Coding (HEVC). They accentuated different approaches to harness motion vector information for object detection across multiple compression standards.

Javed et al. [10] extended the realm of compressed domain analysis techniques to document image analysis. Though their work is primarily centered around document images, the methodologies they proposed for processing compressed data directly, without decompression, carry significant relevance to our work on vehicle detection and classification.

Meanwhile, Alvar et al. [11] pushed the envelope in this field by examining the feasibility of detecting faces without fully reconstructing the image from the HEVC bitstream. They proposed an innovative approach that utilized a Convolutional Neural Network (CNN), trained on the output of the HEVC entropy decoder, challenging the necessity of full image reconstruction.

In the same vein, Zhao et al. [12] devised a real-time moving object segmentation and classification method for surveillance videos using HEVC compressed domain features. Their technique, however, only classifies objects into broad categories such as persons or vehicles, in contrast to our method, which offers a detailed classification of vehicles into six specific types.

Chen et al. [13] presented a swift object detection method in the HEVC intra compressed domain. Unlike our approach, which strategically omits residuals to minimize computational demands, their technique relies on partitioning depths, prediction modes, and residuals for object detection.

Feng et al. [14] proposed TapLab, a rapid framework for semantic video segmentation. The technique, utilizing motion vectors and residuals from compressed videos, differs from our focus on intra features and not relying on motion vectors.

Transitioning towards the application of compressed domain analysis for moving region detection, Töreyn et al. [15] unveiled the potential of compressed video. They innovatively employed the Discrete Cosine Transform (DCT) or the Wavelet Transform (WT) to compress the video, demonstrating the computational efficiency of processing directly in the compressed domain.

In parallel, Choi and Bajic [16] proposed a human detection method relying on HEVC intra coding syntax elements, such as block size, intra prediction modes, and transform coefficient levels. However, their focus was primarily on human detection, diverging from our aim of vehicle classification.

Further broadening the horizons of image understanding tasks in the compressed domain, Torfason et al. [17] unveiled a technique that works directly on compressed representations produced by deep neural network (DNN)-based image compression methods. This approach not only reduces computational cost by avoiding the need for

decoding the compressed representation into RGB space but also performs better at aggressive compression rates compared to inference from compressed RGB images.

Dovetailing with the topic of vehicle detection, Wang et al. [18] created a method for highway vehicle counting in the compressed domain using coding-related metadata. Despite demonstrating competitive computational costs with pixel-domain approaches, their focus on counting vehicles differs from our goal of detailed vehicle type classification.

Concurrently, Savcı et al. [19], Benazza-Benyahia et al. [20], and Töreyn [21] have ventured into compressed domain analysis for fire detection, with Töreyn focusing specifically on smoke detection in MJPEG2000 compressed video. These works underscore the adaptability of compressed domain analysis for various applications, even in the realm of environmental monitoring like early smoke detection in forest areas.

Continuing the exploration of complex analysis in the compressed domain, Çavaş et al. [22] proposed an anomaly detection technique for H.265 compressed videos. They successfully used the motion vectors and their region information in the compressed video as input to an autoencoder model—an unsupervised artificial neural network method—for anomaly detection in video data.

Finally, Bombardelli et al. [23] presented a practical application of compressed video analysis for object tracking in H.264/AVC compressed videos. They achieved a balance between high accuracy tracking and low-complexity processing by leveraging codec motion vectors and block coding modes extracted from the H.264/AVC bitstream via cost-effective partial decoding.

While each of these works has made significant strides in their respective fields, our approach sets itself apart in several key aspects. We offer a more granular vehicle classification into six specific types, catering to advanced transportation applications. To our knowledge, we are the first to propose the use of random perturbation for frame reconstruction without the need for residual data, reducing computational demands significantly. While several of the discussed works utilize motion vectors, our method explores the untapped potential of intra features. By leveraging the state-of-the-art

YOLOv7, we demonstrate comparable accuracy to pixel-domain methods, attesting to the effectiveness of our approach. Considering the blend of intra and inter frames in a video, future works could further enhance our method by incorporating motion vectors.

2.2 Related Works in License Plate Detection

License Plate Detection (LPD) is a vital element of vehicle detection and classification systems, playing an integral role in numerous practical applications such as traffic surveillance, traffic rule enforcement, parking management, and automated toll collection. Given its importance and wide applicability, it has been the focus of extensive research over several decades, leading to the development of numerous techniques and strategies to tackle the challenges inherent in this task.

The early techniques for LPD relied heavily on traditional image processing methods and hand-crafted feature extractors [24]. Edge detection methods, such as those discussed in [25], [26] and [27], were frequently employed for detecting the boundaries of the license plate in an image. These methods often required careful tuning and were sensitive to lighting conditions and plate orientation. Gabor filters [28], another popular technique, were used to identify the specific patterns in the plate region, and the Scale-Invariant Feature Transform (SIFT) [29] was employed to identify key points that could be used for plate recognition, even under changes in scale and orientation. Connected Component Analysis (CCA) [30] was also widely used for segmenting the license plate characters for recognition.

The advent of deep learning and Convolutional Neural Networks (CNNs) brought significant advancements to the field of LPD. CNNs, with their capability to learn hierarchical feature representations automatically, have been proven highly successful in achieving remarkable accuracy rates in LPD tasks [31,32]. For instance, Delmar et al. [33] designed a CNN-based method that calculated a score for each image sub-region to detect the license plate. Hedry et al. [34] adopted the YOLO architecture [35] and reported an accuracy of 98.22% on Taiwanese license plates. Wanwei et al. [36] trained a multi-task CNN (MTCNN) specifically for Chinese license plates,

while Laroca et al. [37] achieved nearly perfect recall and high end-to-end recognition rates using a YOLO-based Automatic License Plate Recognition (ALPR) system.

In another study, Min et al. [38] leveraged YOLOv2 for detection and utilized k-means++ clustering to select the best number and size of candidate boxes for plates, adjusting the model's structure accordingly. Tao et al. [39] conducted a comparative study between YOLO and SSD [40] for LP detection and reported that YOLO achieved better accuracy.

These works highlight the evolution of LPD from traditional image processing methods to deep learning approaches, with the latter yielding significant improvements in accuracy and robustness. While these studies have primarily focused on pixel-domain techniques, our research explores the novel idea of performing LPD directly in the compressed domain. This presents an exciting new direction for LPD and offers potential advantages in terms of computational efficiency and real-time processing capability.

2.3 Related Works in Vehicle Detection and Classification

The task of Vehicle Detection and Classification (VDC) plays a significant role in a variety of applications, including traffic surveillance, autonomous driving, and intelligent transportation systems. Over the years, this task has garnered substantial attention from the research community due to its complexities and wide-ranging impacts on society.

In the early stages of VDC research, traditional image processing methods were largely utilized. These techniques often involved hand-crafted feature extractors and classifiers such as Support Vector Machines (SVMs) or Decision Trees. However, these methods often required meticulous tuning and were prone to underperforming in complex and dynamic environments.

The advent of deep learning has dramatically changed the landscape of VDC research. With its inherent ability to learn hierarchical feature representations, deep learning has brought significant improvements in detection and classification accuracy. The ability of deep learning models, particularly Convolutional Neural Networks (CNNs),

to process large amounts of data and extract high-level features from raw pixels has proven to be highly effective in VDC tasks.

Numerous research papers have focused on developing deep learning-based methods for VDC. In this section, we focus on methods that utilize the BIT Vehicle dataset, as this dataset allows us to directly compare our results with these works. We provide an overview of five representative vehicle classification methods based on the BIT Vehicle dataset.

Dong et al. [41] proposed a vehicle type classification method using a semi-supervised convolutional neural network from vehicle frontal-view images. They introduced sparse Laplacian filter learning to obtain the filters of the network with large amounts of unlabeled data and trained the network on the challenging BIT-Vehicle dataset. The method demonstrated the effectiveness of using deep learning for vehicle classification in complex scenes.

Roecker et al. [42] proposed a convolutional neural network model for vehicle type classification using low-resolution images from a frontal perspective. They trained the model on a subset of the BIT-Vehicle dataset and achieved an accuracy of 93.90%, proving the model to be discriminative and capable of generalizing the patterns of the vehicle type classification task.

Sang et al. [43] proposed a new vehicle detection model called YOLOv2_Vehicle based on YOLOv2. They used the k-means++ clustering algorithm to cluster vehicle bounding boxes on the training dataset, improved the loss calculation method for bounding box dimensions, and adopted a multi-layer feature fusion strategy. The model achieved a mean Average Precision (mAP) of 94.78% on the BIT-Vehicle validation dataset.

Wu et al. [44] proposed a multi-scale vehicle detection method by improving YOLOv2 to address the foreground-background class imbalance and varying vehicle sizes in a scene. They introduced a new anchor box generation method called Rk-means++ and incorporated Focal Loss into YOLOv2 for vehicle detection. The method demonstrated better performance on vehicle localization and recognition on the BIT-Vehicle public dataset compared to other existing methods.

Taheri Tajar et al. [45] developed a lightweight real-time vehicle detection model based on the Tiny-YOLOv3 network. They pruned and simplified the network and trained it on the BIT Vehicle dataset, achieving an mAP of 95.05% and a detection speed of 17 fps, which is about two times faster than the original Tiny-YOLOv3 network.

Despite the remarkable successes achieved in pixel-domain VDC, there is a growing interest in performing these tasks directly in the compressed domain. The motivation behind this interest lies in the potential benefits of reduced computational requirements and the feasibility of real-time processing. This novel direction in VDC research forms the basis of our work, where we aim to develop an effective method for VDC in the compressed domain, providing an interesting balance between computational efficiency and accuracy. We adopt the YOLOv7 framework [46] as the basis for our vehicle classification method. We focus on achieving comparable accuracy to pixel domain methods while operating in the compressed domain. By utilizing the strengths of YOLOv7 and adapting it to work with HEVC intra features, we propose a computationally efficient vehicle classification method that maintains high accuracy.

3. BACKGROUND

The goal of this chapter is to provide an overview of the theories and technologies used in the development of this thesis. Specifically, the first section covers the basics of High Efficiency Video Coding, with a focus on the Intra Coding mechanism. The second section discusses convolutional neural networks in general, and the YOLO model in particular. The purpose of this chapter is to provide context and background information for the reader.

3.1 High Efficiency Video Coding (HEVC)

Video coding standards are a set of technical specifications and guidelines that define how digital video data is encoded, compressed, and transmitted. These standards have evolved over time to address the increasing demand for high-quality video services, and to support the growing number of applications and devices that use video.

One of the earliest video coding standards was the Motion Picture Experts Group (MPEG) standard, which was developed in the early 1990s. This standard defined several different codecs, including MPEG-1 and MPEG-2, which were widely used for video compression and transmission.

In the late 1990s, the Moving Picture Experts Group (MPEG) developed the H.264/MPEG-4 AVC standard, which introduced new technologies such as block-based motion compensation and hierarchical prediction. This standard was designed to provide significantly higher compression ratios than previous standards, and is still widely used today for a variety of applications.

More recently, the High Efficiency Video Coding (HEVC) standard was developed to further improve the efficiency of video coding. High efficiency video coding (HEVC), also known as H.265, is a video compression format which is designed as a successor to the previous H.264 video compression format. H.264 is the most commonly used video coding standard worldwide [47]. HEVC, compared to H.264, can achieve from

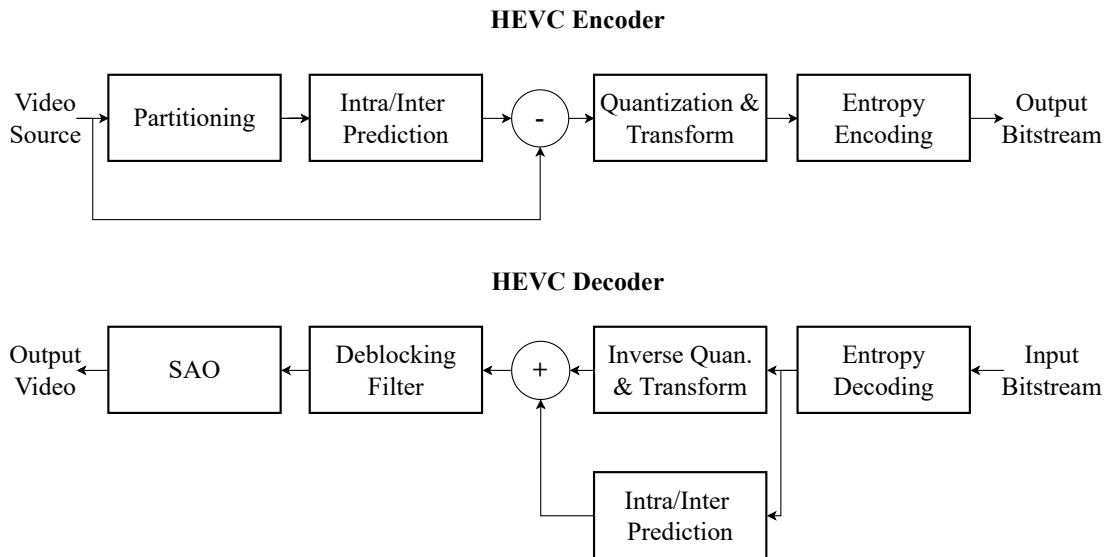


Figure 3.1 : The basic block diagram of an HEVC encoder and decoder.

25% to 50% better data compression at the same level of video quality [48]. HEVC is becoming one of the new video standards. Numerous IP camera manufacturers now include HEVC support by default. In addition to its use in IP cameras, the HEVC standard is also commonly used in other video applications, such as streaming video, video conferencing, and video on demand. Its high efficiency and ability to deliver high-quality video at low bitrates make it an attractive choice for these and other video applications.

3.1.1 Basic concepts

The block diagram of an HEVC encoder and decoder is shown in Figure 3.1, with the responsibilities of each block and other key components described as follows:

3.1.1.1 Color coding

The specific color format used in HEVC is specified in the bitstream, and can be chosen by the encoder based on the requirements of the specific application or use case.

The 4:2:0 color format is the most commonly used in HEVC, and is based on the standard YCbCr color space. It separates the luminance (Y) and chrominance (Cb and Cr) components of the video, and uses subsampling to reduce the resolution of the

chrominance information. This allows the encoder to more efficiently compress the video data while maintaining good visual quality.

Other color formats supported by HEVC include 4:2:2, which uses full resolution chrominance information, and 4:4:4, which uses full resolution for both luminance and chrominance. Monochrome is a special case of 4:2:0 that represents the video data using a single channel for the luminance information, without any chrominance information.

3.1.1.2 Picture partitioning

The HEVC standard uses a hierarchical block-based coding structure, in which an image or video frame is divided into a grid of blocks known as coding tree units (CTUs). Each CTU is further divided into smaller blocks known as coding units (CUs), which are the basic unit of prediction in the HEVC standard.

Figure 3.2(a) shows how an image is segmented into CTUs of equal size. The typical CTU size is 64×64 pixels. CTUs are divided into different sized Coding Units (CUs) depending on the complexity of the encoded region. CUs can be 8×8 , 16×16 , 32×32 or 64×64 . CUs of various sizes are portrayed in Figure 3.2(b).

A CU is typically composed of one or more coding blocks (CBs), which are the smallest unit of image data that can be independently encoded and decoded in the HEVC standard. Each color component of a CU is represented by a separate CB. In the case of an image using the YCbCr color space, each CU would be represented by three CBs: one for the luminance (Y) component, and one each for the chrominance (Cb and Cr) components. Coding blocks (CBs) are the smallest unit of image data that can be independently encoded and decoded in the HEVC standard. The size of the coding blocks can range from 8×8 to 64×64 pixels.

Prediction units (PUs) are blocks of image data that are used for prediction, in which the prediction of a block is based on the information in other blocks within the same frame (intra prediction) or in previous frames (inter prediction). The size of the prediction blocks used for intra prediction can range from 4×4 to 64×64 pixels.

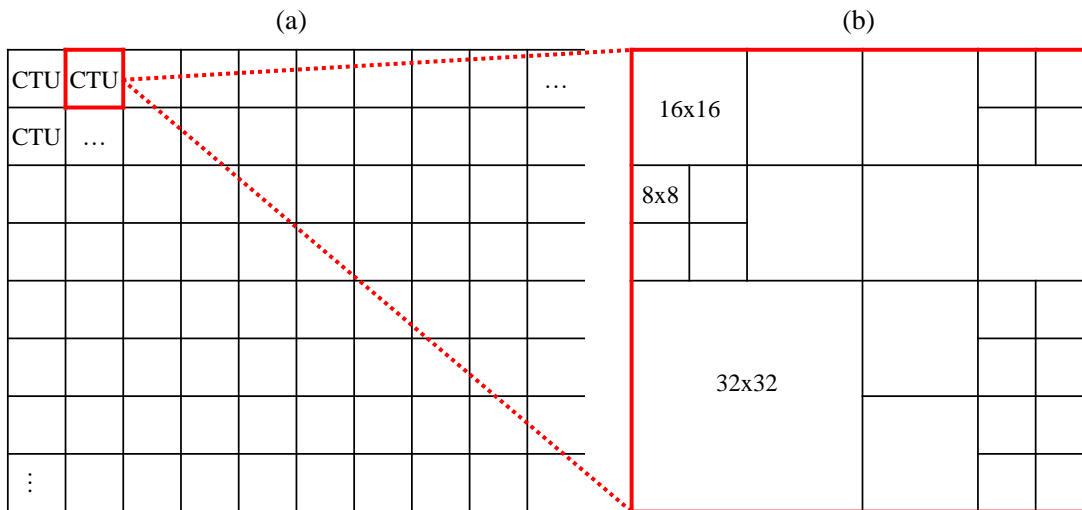


Figure 3.2 : The HEVC block partitioning structure. a) The division of an image into CTUs of equal size. b) Subdivision of a CTU into CUs.

Transform units (TUs) are blocks of image data, contains the residual information, which is the difference between the CU and the PU. TUs are transformed from the spatial domain into the frequency domain for the purposes of encoding and decoding. The size of the TUs can range from 4x4 to 32x32 pixels.

3.1.1.3 Prediction

In video encoding, temporal and spatial correlation refer to the relationships between the video data over time and in space. Temporal correlation refers to the relationship between the video data at different points in time, while spatial correlation refers to the relationship between the video data at different spatial locations within a frame.

Encoding is possible using temporal and spatial correlation by exploiting these relationships in order to reduce the amount of data that needs to be encoded and transmitted.

In HEVC (High Efficiency Video Coding), intra prediction and inter prediction are two different techniques that are used to reduce the amount of data required to represent a video.

Intra prediction, also known as intra-frame prediction, is a technique to remove spatial correlation. It uses information from previously coded blocks within the same frame to

predict the content of the current block. This is done by dividing the current frame into blocks, and using various prediction modes to find the best match for the current block from previously coded blocks in the same frame. Intra prediction allows for efficient coding of the current frame by exploiting the spatial correlations between neighboring pixels in an image.

Inter prediction, on the other hand, is a technique to remove temporal correlation. It uses information from previously coded frames to predict the content of the current frame. This is done by dividing the current frame into blocks and using various motion estimation algorithms to find the best match for each block from previously coded frames. Inter prediction allows for efficient coding of the current frame by exploiting the temporal correlations between consecutive frames in a video [49].

In HEVC, both intra and inter prediction are performed in the prediction step of the coding process. The encoder selects the best prediction mode for each block based on various factors, such as the amount of data saved, the quality of the prediction, and the computational complexity of the mode. The selected prediction mode is then used to generate the predicted block, and the difference between the predicted block and the actual block is calculated and encoded.

3.1.1.4 Transform and quantization

Transform coding involves transforming the image data from the spatial domain (where it is represented as an array of pixels) into the frequency domain (where it is represented as an array of transform coefficients). The transform coefficients represent the intensity of various frequency components in the original image data, and can be more efficiently encoded and compressed than the original pixel data. The HEVC standard employs the discrete cosine transform (DCT) and the discrete sine transform (DST).

Quantization is a technique that involves reducing the precision of the transform coefficients in order to achieve a higher level of compression. This is typically done by dividing the transform coefficients by a fixed value and rounding the result to the nearest integer. The fixed value, known as the quantization step size, controls the

amount of precision that is retained in the transformed data. A smaller step size results in higher precision and lower compression, while a larger step size results in lower precision and higher compression.

3.1.1.5 Entropy coding

Entropy coding is used to efficiently encode the video data after it has been transformed and quantized by the codec. HEVC uses a technique called context-adaptive binary arithmetic coding (CABAC) for entropy coding. This algorithm uses a set of probability estimates to predict the likelihood of each bit in the encoded data, and then encodes the data using the minimum number of bits required to represent the probabilities. This allows HEVC to achieve high levels of compression while maintaining good visual quality.

3.1.1.6 Deblocking filter

A deblocking filter is used to reduce the blocking artifacts that can occur in compressed video, which can degrade the visual quality of the video. The deblocking filter works by smoothing out the block boundaries in the video, making the transitions between blocks less noticeable to the human eye. This can improve the overall visual quality of the video and make it more pleasant to watch.

3.1.1.7 Sample adaptive offset filter

The Sample Adaptive Offset (SAO) filter works by analyzing the characteristics of each block in the encoded video and applying appropriate offsets to the sample values within the block. This can reduce the amount of ringing and other artifacts in the reconstructed video, resulting in improved visual quality.

3.1.1.8 Profiles

In the High Efficiency Video Coding (HEVC) standard, a profile is a set of constraints and limitations that define a particular subset of the HEVC specification. Profiles are used to target specific applications or use cases, and to ensure that encoded video streams are compatible with a particular set of decoders or playback devices. The

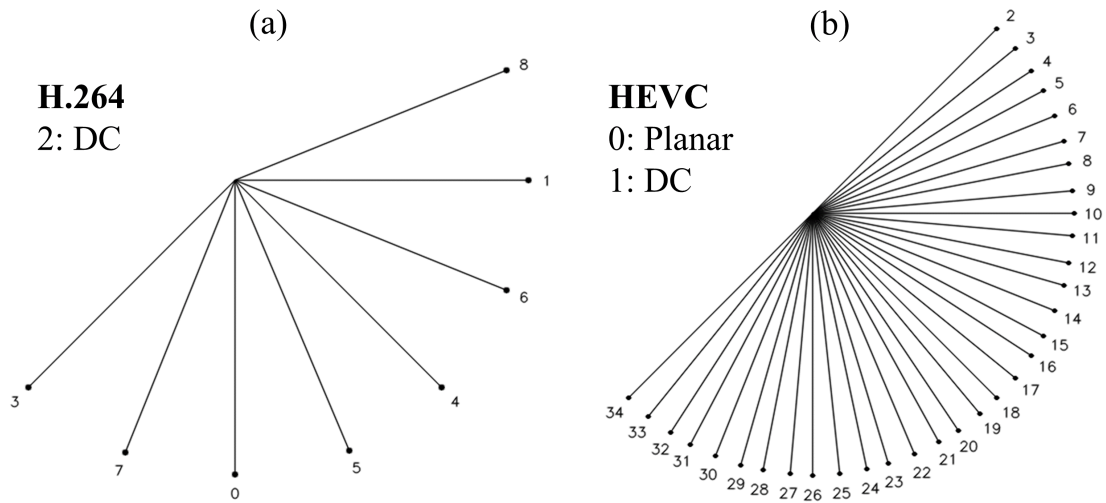


Figure 3.3 : Intra prediction directions. a) The nine intra prediction modes of H.264. b) The thirty five intra prediction modes of H.265 (HEVC).

HEVC standard defines several different profiles, including the Main, Main 10, and Main Still Picture profiles. Each profile specifies a different set of capabilities and constraints, such as the maximum resolution, maximum bit rate, and color format supported by the profile. By using a specific profile, video encoders and decoders can ensure that the encoded video streams are compatible and can be decoded correctly on the intended playback devices.

3.1.2 HEVC intra prediction

Intra-prediction uses previously decoded boundary samples from neighboring blocks in a frame to predict the content of a new prediction block (PB). It is used for the first picture of a video sequence and the first picture at each clean random access point in the video.

In an intra-predicted frame, each Prediction Unit (PU) draws estimates from adjacent image data contained within the same frame. These estimates employ three primary prediction methods: DC prediction, planar prediction, and directional prediction. As seen in Figure 3.3(a), the H.264 standard defines 8 directions to be used in prediction. Conversely, the HEVC expands upon this by incorporating 33 distinct directions, in addition to the DC and planar predictions, as demonstrated in Figure 3.3(b).

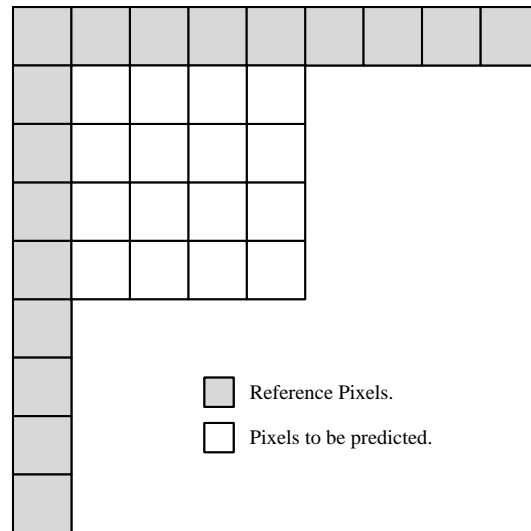


Figure 3.4 : Reference samples used in prediction to obtain predicted samples.

3.1.2.1 Reference samples

Reference samples are blocks of image or video data that are used as a reference for predicting the values of other blocks.

All the intra prediction modes use the same set of reference samples, which are extracted at the boundary from the upper and left blocks adjacent to the current PU. When reference samples are not available, they can be generated by copying samples from the closest available references. If no reference samples are available, a nominal average sample value (typically 128) is used in their place [50].

3.1.2.2 Angular intra prediction

Angular intra prediction is based on the idea of using a set of angled lines to predict the values of the pixels within the block. The angles of these lines are chosen based on the specific characteristics of the image or video data being encoded, and the prediction is made by interpolating the values of the pixels along the lines [51].

3.1.2.3 Planar prediction

Planar prediction mode is designed to preserve continuities along block boundaries. To do this, it uses a two-dimensional linear interpolation to predict the values of the pixels within a block based on the values of the surrounding pixels.

3.1.2.4 DC intra prediction

The DC prediction mode is based on the information in surrounding pixels within the same block. It works by predicting the value of each pixel within the block based on the average value of the surrounding pixels.

The DC prediction mode is typically used in cases where the image or video data exhibits smooth variations in pixel values, as it allows the encoder to efficiently compress the data. However, it can give a coarser prediction of the content in areas with more complex or detailed patterns, and may not be as efficient at predicting finely textured areas of an image.

3.2 Object Detection

Object detection is a crucial task in the field of computer vision, as it allows machines to identify and locate objects within images and video streams. This capability has numerous practical applications, including security and surveillance, self-driving cars, robotics, and augmented reality.

There are many approaches to object detection, but most methods can be broadly categorized into three main stages: Object localization, feature extraction, and classification.

1. Object localization involves identifying regions of the image or video that are likely to contain objects of interest. This can be done through techniques such as sliding windows, selective search, or anchor boxes.
2. Feature extraction involves extracting relevant features from the selected region of interests (ROIs) that can be used to distinguish the objects of interest from the background. This can be done through techniques such as edge detection, color histograms, or texture analysis. Scale-invariant feature transform [19], histograms of oriented gradients (HOG) [20], and Haar-like [21] features.
3. Classification involves using the extracted features to classify the objects into predefined categories. This can be done through techniques such as support vector

machines (SVM) [22], AdaBoost [23], deformable part-based model (DPM) [24] or neural networks.

3.2.1 Detection algorithms

Before the advent of convolutional neural networks (CNNs), object detection was typically performed using methods based on hand-crafted features, such as scale-invariant feature transform (SIFT) or histogram of oriented gradients (HOG). These methods involved extracting features from an image using specific algorithms, then applying a classifier to determine whether a particular object was present in the image. These approaches were effective, but they required a lot of hand-tuning and were not always able to achieve the same level of accuracy as CNN-based methods.

One of the earliest developments in object detection was the Viola-Jones object detection framework, which was published in 2001 [52]. This framework used a combination of Haar-like features and boosting algorithms to rapidly and accurately detect objects in images.

In the years since the Viola-Jones framework was introduced, there have been many advances in object detection, including the development of deep learning algorithms that have significantly improved the accuracy and speed of object detection.

3.2.2 Convolutional neural networks

Convolutional Neural Networks (CNNs) were first introduced in the 1980s by Yann LeCun. LeCun developed a multi-layer neural network called LeNet, which combined a CNN with backpropagation algorithms to enable a computer to classify handwritten numbers [53].

The use of multi-layered networks rather than single-layer networks resulted in significant performance improvements. In 2012, the CNN called AlexNet won the ILSVRC-2012 competition using the ImageNet dataset, which contained 1.2 million images from 1000 different classes [54]. AlexNet achieved an error rate of 15.3%, compared to the runner-up's error rate of 26.2%. This significant achievement in image

classification revolutionized the field of computer vision for processing human visual input, such as objects, handwriting, and facial recognition.

Since then, CNNs have been the state-of-the-art in machine learning and artificial intelligence for processing human visual input and are widely used in various applications.

The architecture of a Convolutional Neural Network (CNN) typically consists of a combination of convolutional layers, non-linear activation function, pooling layers, and fully connected layers [55].

1. Convolutional layers are the primary building blocks of a Convolutional Neural Network (CNN). These layers apply a series of filters to the input image, producing a set of transformed images called feature maps. The filter is applied to the input image by sliding it across the image, with a stride size that determines how far the filter moves each time. The stride size can be adjusted to control the size and resolution of the output feature map. Additionally, the edges of the input image can be padded with extra pixels to ensure that the filter covers the entire image.
2. After the convolutional operation, the resulting feature maps are typically passed through a non-linear activation function, such as the Rectified Linear Unit (ReLU). ReLU is a simple function that replaces all negative values in the feature map with zero, allowing the CNN to learn more complex relationships between the input features and the desired output.
3. Pooling layers are used to downsample the feature maps produced by the convolutional layers. There are several types of pooling, but the most common is max pooling, which selects the maximum value from each pooling window. Pooling layers reduce the dimensionality of the feature maps, making the CNN more efficient and robust to small translations and deformations in the input data.
4. Fully connected layers are used to classify the features learned by the CNN. These layers take the flattened feature maps from the convolutional and pooling layers as input and apply a series of linear transformations and non-linear activations to produce the final output of the CNN. The weights and biases of the fully

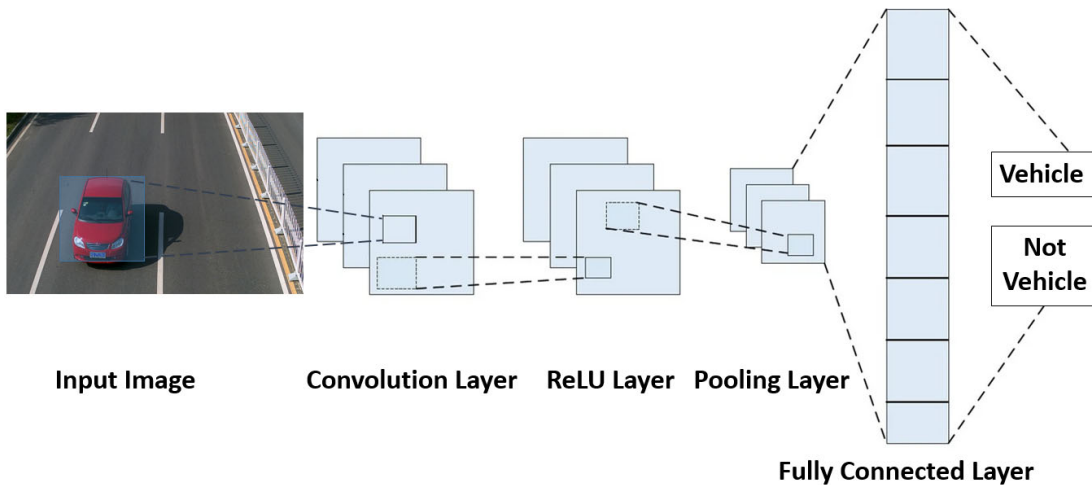


Figure 3.5 : An example of CNN architecture for image classification.

connected layers are learned through training, allowing the CNN to learn complex relationships between the input features and the desired output.

Figure 3.5 illustrates an example of a CNN architecture for image classification. The region of an input image containing a vehicle is fed into the first convolutional layer. This layer applies a series of filters to the input image, producing a set of transformed images called feature maps. The feature maps are then passed through the Rectified Linear Unit (ReLU) function, which introduces non-linearity that allows the CNN to learn more complex relationships between the input features and the desired output.

The feature maps are then downsampled by the pooling layer, which reduces the dimensionality of the feature maps and makes the CNN more efficient and robust to small translations and deformations in the input data. After the pooling layer, the feature maps are flattened and passed through the fully connected layer, where the class of the input image is determined. In this example, the CNN is trained to classify whether an input image contains a vehicle or not. The weights and biases of the fully connected layer are learned through training, allowing the CNN to learn complex relationships between the input features and the desired output.

It is important to identify the region of an input image containing a vehicle in order to determine its class. To do this efficiently, it would be beneficial to use a CNN that is

able to both locate the region and classify the vehicle simultaneously. This is where the You Only Look Once (YOLO) object detection algorithm comes into play.

3.2.3 YOLO

You Only Look Once (YOLO) is a real-time object detection algorithm developed by Joseph Redmon and Ali Farhadi in 2015 [35]. YOLO is a single-stage object detection algorithm that is able to detect and classify objects in an image or video in a single pass. It is a convolutional neural network (CNN) that uses a fully convolutional architecture, which means that it is able to process images of any size and maintain a constant output resolution.

The YOLO architecture consists of several convolutional layers followed by a few fully connected layers. The input to the network is an image of fixed size, which is passed through the convolutional layers to extract features. The features are then passed through a few fully connected layers to predict the bounding boxes and class probabilities for each object in the image.

YOLO works by dividing the input image into a grid of cells, where each cell is responsible for predicting a set of bounding boxes and class probabilities. If an object falls within a cell, that cell is responsible for predicting the bounding box and class probability for that object.

The bounding box prediction is made using 4 continuous values: the coordinates of the center of the box (x , y) and the width and height of the box (w , h). The class probabilities are predicted using a softmax classifier, which outputs the probability for each class.

To improve the accuracy of the predictions, YOLO uses anchor boxes, which are predefined bounding boxes with known aspect ratios. The anchor boxes are used to help the network predict the location and size of the objects in the image.

During training, YOLO uses an optimization algorithm such as stochastic gradient descent (SGD) to minimize the sum of squared errors between the predicted bounding boxes and the ground truth bounding boxes. The network is then fine-tuned using

non-maximum suppression (NMS) to eliminate overlapping bounding boxes and select the best prediction for each object.

At test time, YOLO is able to process images in real-time, making it useful for applications where real-time object detection is required. It has achieved state-of-the-art results on various object detection benchmarks and has become a popular choice for object detection tasks.

The YOLO (You Only Look Once) object detection algorithm has undergone several significant updates since its introduction in 2015.

- YOLOv2 was introduced in 2016 as an improvement over the original YOLO algorithm. It made several changes to the original YOLO architecture, including using anchor boxes to improve the accuracy of bounding box predictions and adding batch normalization to the convolutional layers to improve the convergence of the model [56].
- YOLOv3 was introduced in 2018 and further improved upon the YOLOv2 architecture. It made several changes, including using a new network architecture with a series of residual blocks, using anchor boxes with three different scales and ratios, and using a new loss function that combines classification and localization errors [57].
- YOLOv4 was introduced in 2021. Some of the key changes in YOLOv4 include:
Improved network architecture: YOLOv4 uses a new network architecture called SPP-YOLO, which combines the strengths of both YOLOv3 and the Spatial Pyramid Pooling (SPP) network. This allows YOLOv4 to better handle objects of different sizes and aspect ratios. In addition to that, YOLOv4 includes several techniques to improve the detection of small objects, including using a new loss function that down-weights large objects and using a feature pyramid network to better handle small objects at multiple scales [58].
- YOLOv7 is the latest version of the YOLO (You Only Look Once) object detection algorithm. It was introduced in July 2022 and is currently the state-of-the-art real-time object detector, according to its paper [59]. Some of the key features

of YOLOv7 include: E-ELAN (Extended Efficient Layer Aggregation Network), model scaling for concatenation-based models, trainable BoF (Bag of Freebies), and planned re-parameterized convolution.

In this study, tiny versions of YOLOv3 and YOLOv7 are used for LP detection and vehicle classification, respectively. Tiny model is a variant of full YOLO models that has been optimized for faster and more efficient through the use of fewer convolutional layers, fewer filters, and a smaller overall network size. While tiny models are generally less accurate than full YOLO models, they offer a trade-off between accuracy and speed that makes them well-suited for use on devices with limited resources or in real-time applications.

4. PARTIAL INTRA DECODING METHODS FOR COMPRESSED VIDEO UNDERSTANDING

The dawn of video compression standards such as High Efficiency Video Coding (HEVC) has spurred new opportunities in the realm of compressed video understanding. As we aim to achieve efficient video understanding techniques, it becomes increasingly essential to develop methods for effectively deciphering the encoded information inherent in compressed videos. The objective of this chapter is to introduce novel methods aimed at reconstructing images from the HEVC intra bitstream, which forms the foundation for our proposed object detection techniques in compressed domain.

4.1 Introduction

In this section, the methods developed for reconstructing images from HEVC intra bitstream are introduced. These techniques involve the extraction of specific attributes from the HEVC encoded bitstream and the early termination of decoding, which are then used to create images containing sufficient visual information for object detection. The resulting reconstructed images, which are obtained by partial decoding of the bitstream, are referred to as compressed domain images, while the original video frames are known as pixel domain images.

The main aim of these methods is to create compressed domain images that can be obtained faster than pixel domain images. By creating compressed domain images that can be obtained faster than pixel domain images, it is possible to speed up the process of object detection and other applications. The performance of these methods is evaluated by comparing them to that of pixel domain images, in order to determine their potential benefits for various applications in terms of construction time and object detection accuracy. The evaluation for construction time is given in the last section of this chapter, and object detection performance is presented in the chapters.

Four new methods for partial decoding intra-coded frames in the High Efficiency Video Coding (HEVC) standard are presented. These methods are denoted as M_{bp} , M_{pu} , M_{rp} , and M_y , respectively, and are described as follows:

- The Block Partition Based Method (M_{bp}) method relies on the partition information of the frame to reconstruct the image. The partition information indicates how the frame is divided into smaller blocks. The M_{bp} method uses this partition information to reconstruct the frame without using any additional information, such as prediction values or residual data.
- The Prediction Unit Based Method (M_{pu}) method also uses the partition information of the frame to reconstruct the image, but in addition to this, it also uses prediction values. These prediction values are derived from neighboring blocks and are used to predict the value of each pixel in the current block. Using these two attributes a gray level image is generated by this method.
- The Random Perturbation Based Method (M_{rp}) method uses the Standard HEVC Intra Decoding method recommended by the specification to predict the image. However, it differs from the standard method in that it does not use residual data in the reconstruction process. Residual data is the difference between the predicted value of a pixel and its actual value, and it is typically used to improve the accuracy of the prediction. The M_{rp} method does not use residual data and instead injects random perturbations as a substitute. These random perturbations, drawn from a defined distribution, act as a replacement for the actual residual values.
- The Luma Based Method (M_y) uses only the luma (gray-scale) information of the frame to reconstruct the image. The chroma information, which contributes to the color attributes of the image, is not considered in the process. This method is based on the premise that object detection and classification tasks can often be effectively carried out using grayscale information alone, as key identifying features like shapes and edges are typically preserved in the grayscale representation. The exclusion of chroma information allows a reduction in the computational complexity and data size of the reconstruction process.

On the other hand, the standard HEVC decoding process, denoted as M_{px} , follows the full utilization of both residual data and chroma information in the reconstruction process, providing the most accurate representation of the original image. However, this method involves significant computational overhead and data size, making it less efficient for real-time video understanding tasks. The images produced by M_{px} serve as a benchmark against which we compare the images reconstructed by our proposed compressed domain methods.

In Figure 4.1, the required data and the modules involved in the reconstruction of a Coding Tree Unit (CTU) in the High Efficiency Video Coding (HEVC) standard are shown in the yellow rectangle area. The HEVC bitstream is first decoded by the entropy decoding block, which extracts the syntax elements such as the partition structure, prediction modes, and residual data. Following this, the prediction modes and reference pixels, derived from previously decoded Coding Tree Blocks (CTBs), are utilized to create the predicted CTB. Subsequently, the residual data is integrated with the predicted CTB to form the final CTB. This sequence is also executed for chroma CTBs. Eventually, all the CTBs are combined to create the final CTU. This comprehensive process is repeated iteratively, enabling the reconstruction of the entire video frame from the HEVC bitstream.

The purple circles in Figure 4.1 represent the suggested 4 different methods for reconstructing the CTU and the standard decoding method. The data requirements for each method are outlined in the figure. The M_{bp} method uses only the block partition information, which is available after entropy decoding. The M_{pu} method uses both the block partition information and the Prediction Units, which are also available after entropy decoding. The M_{rp} method reconstructs the CTU without involving the residual data, and the M_y method uses only the luma CTB to reconstruct the image. Note that, the process of reconstructing CTUs with HEVC standard decoding process and estimated residuals is actually done in Coding Unit (CU) level. However, for simplicity and ease of understanding, the process is presented in the context of CTUs.

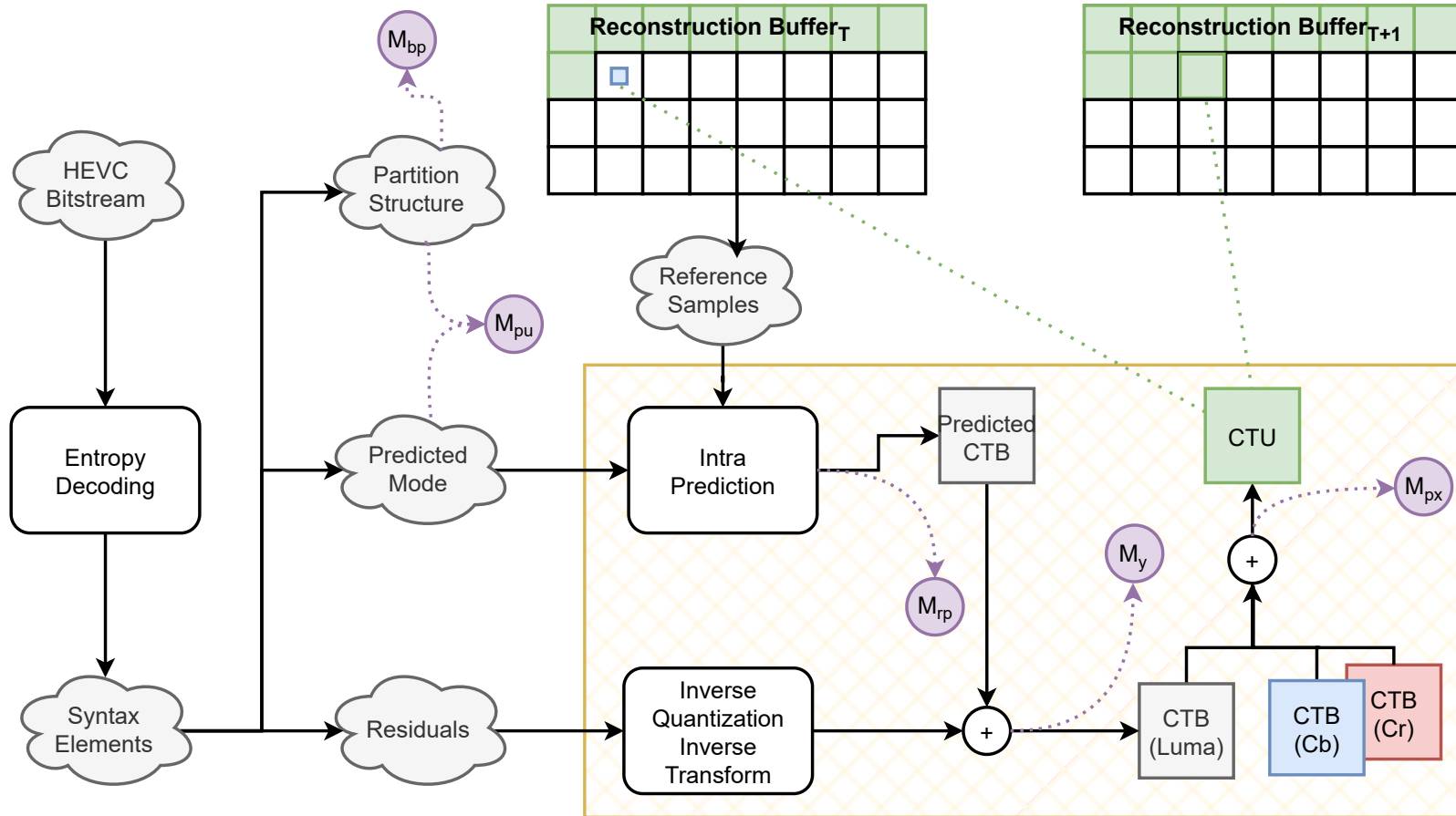


Figure 4.1 : Five methods for HEVC decoding of intra-coded frames. M_{bp} : Block partition based method. M_{pu} : Prediction unit based method. M_{rp} : Random perturbation based method. M_y : Luma based method. M_{px} : Standard HEVC decoding method.

4.2 Block Partition Based Method

The Block Partition Based Method (M_{pu}) generates an image using HEVC block partition (BP) structure.

Let $I(x,y)$ be the intensity value of an image I at the spatial coordinates (x,y) . Let A be the set of boundary pixel locations of coding units (CUs) corresponding to the image I . An image I_{bp} is generated using (4.1).

$$I_{bp}(x,y) = \begin{cases} 1 & \text{if } (x,y) \in A \\ 0 & \text{else} \end{cases} \quad (4.1)$$

In this method, an intra-coded image is converted into a binary image using the set of boundary pixel locations of coding units (CUs) corresponding to the image. Pixels that cross CU boundaries are converted to white pixels, and the rest are converted to black pixels. The output image generated by this method is referred to as the Block Partition Image, or I_{bp} .

The HEVC partitioning structure is designed to use small-sized CUs when encoding complex texture in the image. This is because high-band spatial content in the pixel domain requires the use of smaller CUs. The Figure 4.2 is an example of a I_{bp} . In this figure, there is an image of a vehicle and the corresponding block partitioning image. High-band spatial frequency areas, such as the licence plate (LP) zone, can be seen to be partitioned using smaller CU blocks. For LP detection in the compressed domain, this characteristic partitioning of LP regions is exploited ¹.

The decision to represent HEVC attributes using an image format and generate images from the encoded stream has two main advantages. First, object detection techniques can be applied directly to pixel images, which allows for efficient and accurate detection of objects in the image. Second, the resulting images facilitate data visualization and analysis, making it easier to understand the characteristics of the encoded image and the underlying attributes.

¹An earlier study of the I_{bp} based method was presented in [60].

(a)



(b)

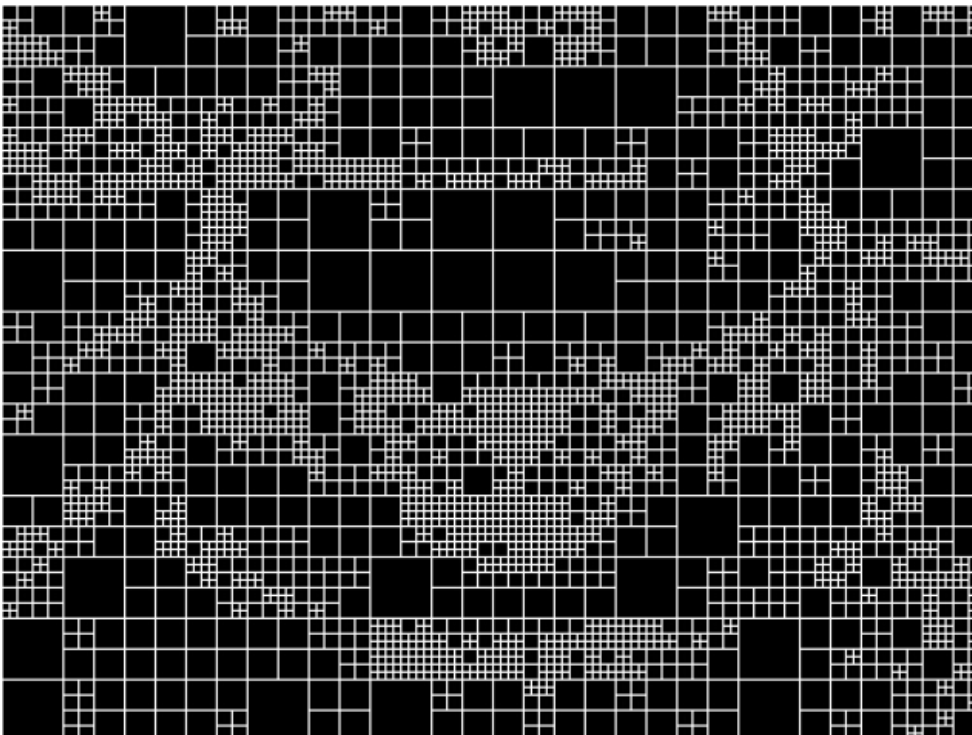


Figure 4.2 : A sample I_{bp} image. (a) Pixel domain image of a vehicle, (b) Corresponding Block Partition Based Image (I_{bp}) of the vehicle.

Overall, the Block Partition Based Method (M_{bp}) is a fast and efficient method for reconstructing an image from the encoded stream, as it relies only on the partition information, which is available after entropy decoding. However, the accuracy of the reconstructed image is lower compared to other methods that use additional information, such as prediction values or residual data.

4.3 Prediction Unit Based Method

Prediction Unit Based Method (M_{pu}) generates a gray-level image based on Prediction Units, which are used to predict the value of each pixel in the current block based on neighboring blocks. The Prediction Units in the intra-coded HEVC stream are calculated from adjacent image data, and therefore hold the correlation information between neighboring pixels. This can be useful for object detection, as the correlation between pixels can be used to identify distinctive features of the object of interest.

Let $I(x,y)$ be the intensity value of an image I at the spatial coordinates (x,y) . An image I_{pu} is generated using (4.2).

$$I_{pu}(x,y) = \begin{cases} \alpha PU(x,y) + \beta & \text{if } |CB(x,y)| = 8 \times 8 \\ \gamma & \text{else if } (x,y) \in A \\ 0 & \text{else} \end{cases} \quad (4.2)$$

where $CB(x,y)$ is the Coding Block corresponding to the image I at location (x,y) and $PU(x,y)$ denotes the Prediction Unit value for the Coding Block of an image I at location (x,y) . In the HEVC standard, there are 35 different Prediction Units that are described in Chapter 2. These modes are used to determine the values that can be assigned to $PU(x,y)$, which range from 0 to 34 (Figure 3.3). A linear equation is used to convert these Prediction Unit values to the 0-255 range for pixel intensities, with the constants α and β determined empirically. β is used to distinguish the Prediction Unit values from the black background of the image; a value of 45 for β is sufficient to create a noticeable difference from the background. α is used to create distinct gray-level areas for different Prediction Unit values; a value of 6 for α generates clear distinctions between these areas. This conversion is applied only for the Coding Blocks (CBs) of size 8×8 .

(a)



(b)



Figure 4.3 : A sample I_{pu} image. (a) Pixel domain image of a vehicle, (b) Corresponding Prediction Unit-Based Image (I_{pu}) of the vehicle.

For the remaining of the generated image, the block partition structure is preserved with an intensity value of γ . γ is the constant average value between the black background and the minimum intensity value for 8×8 blocks. It is found using the equation $\gamma = \frac{\beta}{2} = 24$. The output image generated by this method is referred to as the Prediction Unit Image, or I_{pu} .

An I_{pu} is illustrated in Figure 4.3. In this figure, there is an image of a vehicle and the corresponding Prediction Unit image. The pixels are summed around the LP region, which makes that region clearly visible. The data needed to generate an I_{pu} image is immediately available after the entropy decoding phase of the High Efficiency Video Coding (HEVC) standard. The M_{pu} method bypasses the remaining decoding phases of HEVC, including Intra Prediction and residual calculation.²

Overall, the Prediction Unit Based Method (M_{pu}) is more accurate than the Block Partition Based Method (M_{bp}), as it utilizes additional information in the form of Prediction Units. The performance of LP detection using the M_{pu} method is further investigated in the Chapter 5.

4.4 Random Perturbation Based Method

In this section, an in-depth exploration of the role of residual data in image reconstruction and its impact on the process is presented. In the High Efficiency Video Coding (HEVC) standard, residuals, which represent the difference between the predicted and actual pixel values in a video frame, are integral to the video decoding process. These residuals significantly enhance the accuracy of the reconstructed image. However, they also form a substantial portion of the total data in the encoded bitstream, thus increasing the computational overhead and storage requirements.

To mitigate these challenges, various methods for managing residuals in the image reconstruction process are explored. The objective is to identify an efficient and effective method that can maintain the accuracy-boosting benefits of residuals while reducing their associated drawbacks, particularly in terms of size and computational overhead.

²An earlier study of the I_{pu} is studied in [61] with abbreviation H_{pu}

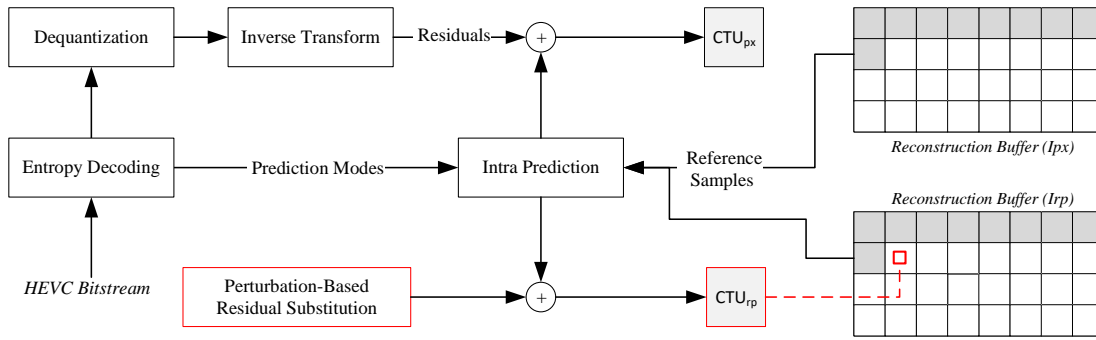


Figure 4.4 : Reconstructing CTUs with HEVC standard decoding process and with random perturbation based residual substitution.

Initially, the impact of completely ignoring the residuals during the image reconstruction process is assessed. This serves as a basic examination of the essential role residuals play in the quality of the reconstructed image. Subsequently, the approach of using constant residuals is investigated, a technique that could potentially simplify the reconstruction process.

However, as is discussed in the ensuing sections, neither the approach of ignoring residuals nor the technique of employing constant residuals produces satisfactory results. This leads to the proposal of a novel method - the random perturbation method. In this method, residuals are replaced with random perturbations drawn from a predefined distribution, thereby retaining the benefits of residuals while minimizing the computational overhead and data size. Detailed discussions on these methods are presented in the following subsections.

Figure 4.4 illustrates the process of reconstructing a coding tree unit (CTU) within the context of High Efficiency Video Coding(HEVC). The HEVC bitstream is first decoded by the entropy decoding block, which extracts syntax elements such as partition structure, prediction modes, and residual data. The decoder generates the CTU prediction by employing prediction modes and utilizing reference samples. The reference samples consist of neighboring pixels found within the image. The residual data is then added to the predicted CTU to generate the final CTU_{px} .

4.4.1 Standard reconstruction

Let $CU(x,y)$ be the intensity value of a reconstructed Coding Unit of an image I at the spatial coordinates (x,y) . The intensity value of the CU can be expressed as the sum of its prediction value, $P(x,y)$, and its residual value, $R(x,y)$, as shown in the following equation:

$$CU(x,y) = P(x,y) + R(x,y) \quad (4.3)$$

Decoding is an iterative process where the reconstructed CU serves as the input for subsequent CUs [48]. The prediction information's close relationship with residual information makes bypassing residuals a significant challenge [16]. To address this, we explored potential signals that could replace residuals while maintaining the overall integrity of the reconstructed image.

4.4.2 Impact of ignoring residuals

In the case where $R(x,y)$ is assumed to be equal to 0, let's examine the consequences for the first CU to be decoded, which is situated at the top-left corner of the image. Due to the absence of reference pixels and in accordance with the HEVC standard, the reference value is assigned as 128 [50]. This value corresponds to the mean of the pixel range, equating to 128 for an 8-bit image. Consequently, the first reconstructed CU will be entirely gray. The first CU contains the reference pixels for subsequent CUs to be decoded. This means that for the second CU, the reference pixels will also be 128. As this process continues, all reference pixels will have a value of 128, leading to a fully gray predicted image, as depicted in (2):

$$I(x,y) = 128 \quad \text{if} \quad R(x,y) = 0 \quad (4.4)$$

4.4.3 Impact of constant residuals

In the case where $R(x,y)$ is assumed to be equal to a constant value, this constant would be added to all decoded CUs throughout the decoding process. As the process

progresses, this addition would accumulate, causing saturation in the reconstructed image. The constant value of $R(x,y)$ would directly impact the resulting saturation level and might lead to a loss of detail or visual information in the reconstructed image. This illustrates the importance of properly handling the residual values to achieve accurate and high-quality image reconstructions.

4.4.4 Random perturbations as a substitute for residuals

Taking these cases into account, we propose using a series of integers Rp , as a replacement for the residuals. These integers are based on a Gaussian distribution and can take both negative and positive values. The choice of using a Gaussian distribution to generate the estimated perturbations (Rp) is motivated by the fact that the Gaussian distribution is a commonly used model for representing the distribution of errors and noise in natural images. By modeling the estimated perturbations using a Gaussian distribution, we can generate a more realistic approximation of the actual residual values, which in turn results in a more accurate reconstruction of the CTUs.

The Gaussian distribution is given by the following formula:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (4.5)$$

To generate the series Rp , we sample n discrete integer random numbers from this distribution with varying standard deviations, such as $std = 1, 2, 3, 4, 5$:

$$Rp_n = rp_1, rp_2, \dots, rp_n \quad (4.6)$$

Each rp_i in the series Rp is obtained by iteratively sampling from the Gaussian distribution until the sample set meets the desired mean ($\mu = 0$) and standard deviation ($\sigma = std$):

$$rp_i = \text{round}(f(x_i)) \quad (4.7)$$

where x_i is a random sample from the Gaussian distribution, and the round function is used to obtain integer values.

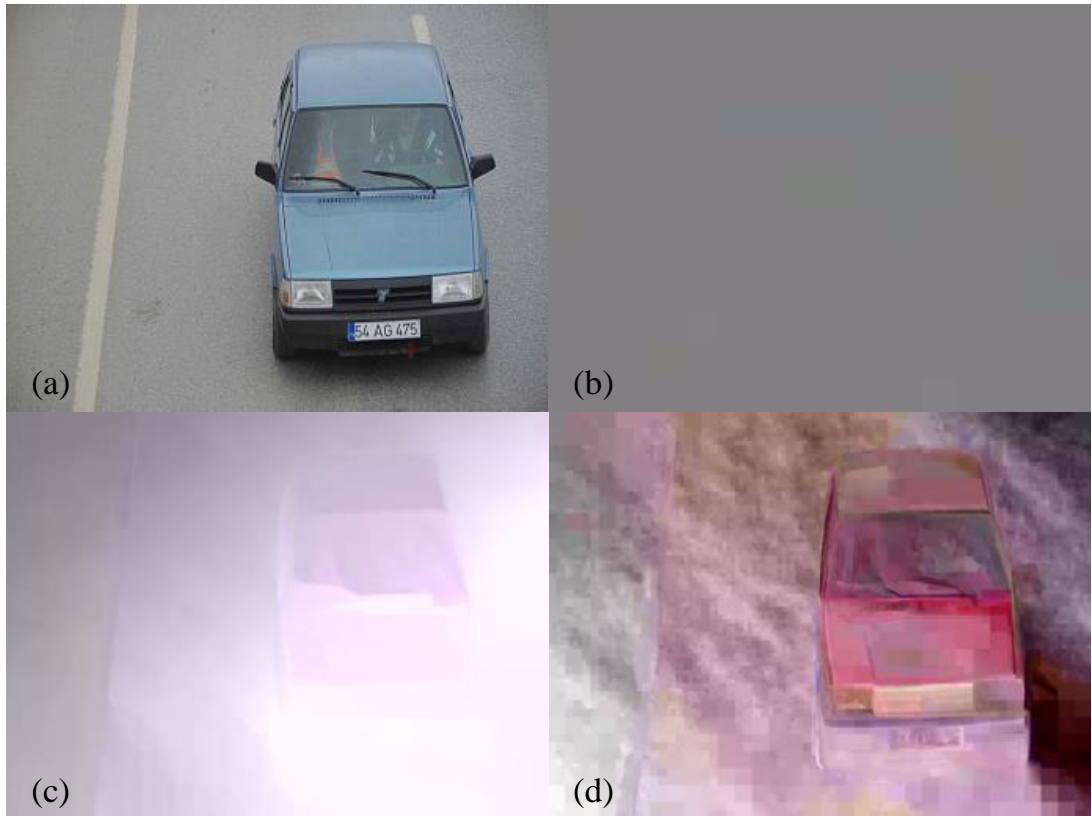


Figure 4.5 : Comparison of different approaches to substitute residuals.(a) Standard reconstruction using HEVC. (b) Impact of ignoring residuals, $R(x,y) = 0$. (c) Impact of constant residual, $R(x,y) = 1$. (d) Random perturbations as a substitute for residuals, I_{rp} with $\mu = 0$ and $\sigma = 7$.

Given the series Rp , we can now construct the predicted CTUs without residual data. For each pixel in a CTU, we replace the residual value with the corresponding value from the series Rp :

$$CU_{rp}(x,y) = P(x,y) + rp_i \quad (4.8)$$

As we process the pixels in the CU, we use the next value from the series Rp , rp_i , as a replacement for the residual. The length of Rp is determined by the maximum size of a CU, which is $64 \times 64 = 4096$. The series Rp_n , with a mean of zero and varying standard deviations, is generated once and used for all predicted image generations. This ensures that the predicted images are consistent and reproducible. The resulting reconstructed image using the random perturbation method will be referred to as I_{rp} .

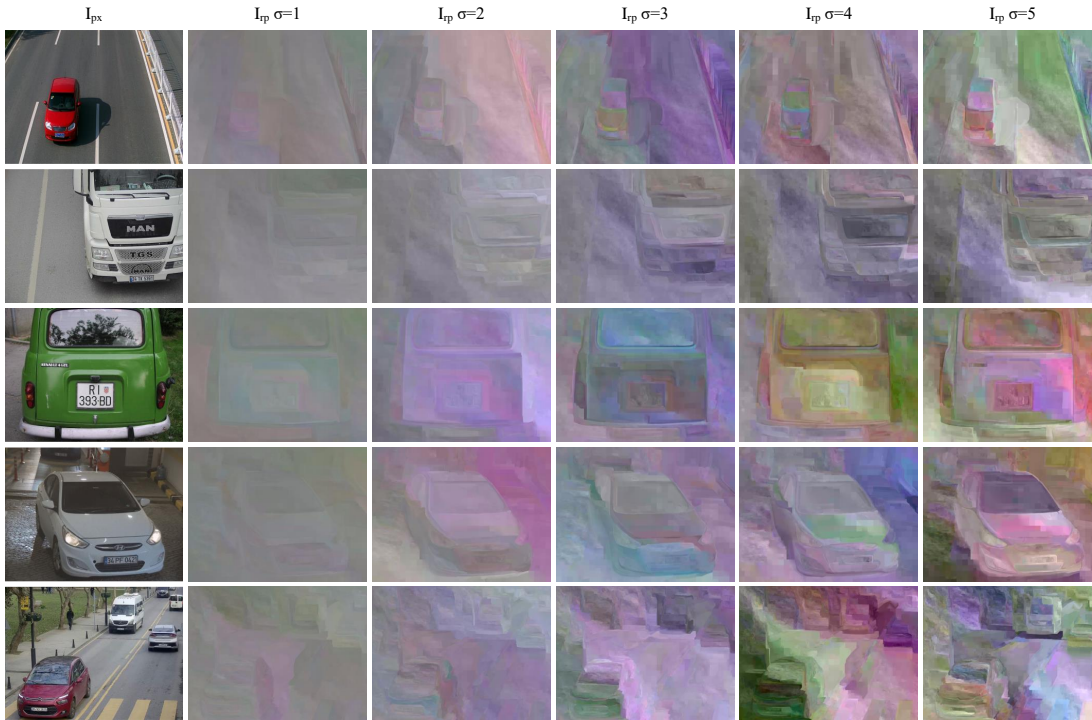


Figure 4.6 : Impact of standard deviation variations on image reconstruction in the Random Perturbation Based Method.

An example of the different approaches to substitute residuals is shown in Figure 4.5. This figure provides a visual comparison of the original pixel domain image reconstructed using the standard HEVC method (a) and the effects of various approaches to substitute residuals on the image quality, such as ignoring residuals (b), using constant residuals (c), and employing random perturbations (d). The random perturbations approach, with a mean (μ) of 0 and standard deviation (σ) of 7, clearly demonstrates a better representation of the original image compared to other approaches.

In Figure 4.6, we present a matrix of images showcasing the effectiveness of our Gaussian-based Random Perturbations method in reconstructing images of vehicles from various scenes while measuring the effect of different standard deviations. The first column displays pixel domain images, while the other columns represent reconstructed images using our method with different standard deviations. As illustrated, our method successfully constructs a close silhouette of the pixel domain image, retaining the general boundaries of objects in the frame and the necessary

information for image classification. As the standard deviation increases, the image becomes more apparent to the human eye, yet classification can still be done successfully for all different standard deviations. Our approach bypasses the calculation of residual data, yet still enables the creation of an image with general boundaries retained, which serves as valuable input for object detection tasks.³.

The effectiveness of this proposed method is measured by conducting accuracy tests in Chapter 6, while the construction time is compared with other methods in the last section.

4.5 Luma Based Method

A monochrome profile is provided in the High Efficiency Video Coding (HEVC) standard for use with grayscale or monochrome video content, and specifies that only a single color component (luma, or Y) should be used in the encoding and decoding process [63]. However, no method for reconstructing a color video bitstream as a monochrome video is specified in the standard. In order to address this issue and take advantage of the robust representation of image content provided by monochrome video in object detection, a new Luma Based Method has been proposed. This method involves the decoding of only the luma component and the bypassing of the color components, allowing for the creation of a monochrome version of the original video that accurately represents the intensity of each pixel while preserving the overall structure and content of the video.

In HEVC, a CU is consist of three CBs namely luma (Y) and two chroma samples (C_r and C_b) and hence the (4.3) can be expressed as follow, where i represents one of the color component of $\{Y, C_r, C_b\}$.

$$CB_i(x, y) = P_i(x, y) + R_i(x, y) \quad (4.9)$$

The gray-scale image I_y is constructed by (4.9).

$$I_y(x, y) = P_Y(x, y) + R_Y(x, y) \quad (4.10)$$

³A study of the M_{rp} method was presented in [62].

A sample of a gray-scale reconstructed image is shown in Figure 4.7. The original image was encoded in color format, but it is reconstructed as a gray image in the I_y image. The I_y image is now identical to the original image, except for the color information. In many object detection tasks, the use of a gray image can be very effective. For example, the lack of color information may not be important for the task at hand, and the use of a gray image can simplify the processing and analysis of the image. Alternatively, the removal of color information may allow for better performance in certain object detection tasks, as it can remove distractions and focus on the shape and texture of the objects in the image.

The advantage of this proposed method lies in the image construction phase, as the process of decoding the color components is bypassed, leading to faster decoding of the video. The effectiveness of the Gray-Scale Image Reconstruction method in terms of both image construction time and object detection performance is evaluated in the last section of this chapter and at Chapter 6.

4.6 Visual Comparison of Reconstructed Images

In order to provide a visual comparison our proposed compressed domain methods, a diverse selection of four different vehicle images, characterized by varying zoom levels, color palettes, and environmental settings, is employed. As depicted in Figure 4.8, these images reflect a broad range of scenarios likely to be encountered in real-world traffic surveillance applications. Factors such as indoor and outdoor environments, different lighting conditions, and variable backgrounds, all contribute to the complexity of the images, impacting the visibility and clarity of the vehicles.

In the process of full decoding of an intra bitstream, the end result is an image in the pixel domain, denoted as I_{px} in this study. For each I_{px} image, four distinct reconstructed images, obtained using the different methods proposed in this study, are presented for comparison: Block Partition Based Image (I_{bp}), Prediction Unit Based Image (I_{pu}), Random Perturbation Based Image (I_{rp}), and Luma Based Image (I_y). These images provide a visual demonstration of the diverse outputs possible with each method, illuminating their respective strengths and potential limitations.

(a)



(b)



Figure 4.7 : (a) Pixel domain image of a vehicle, (b) Corresponding gray scale image (I_y) of the vehicle.

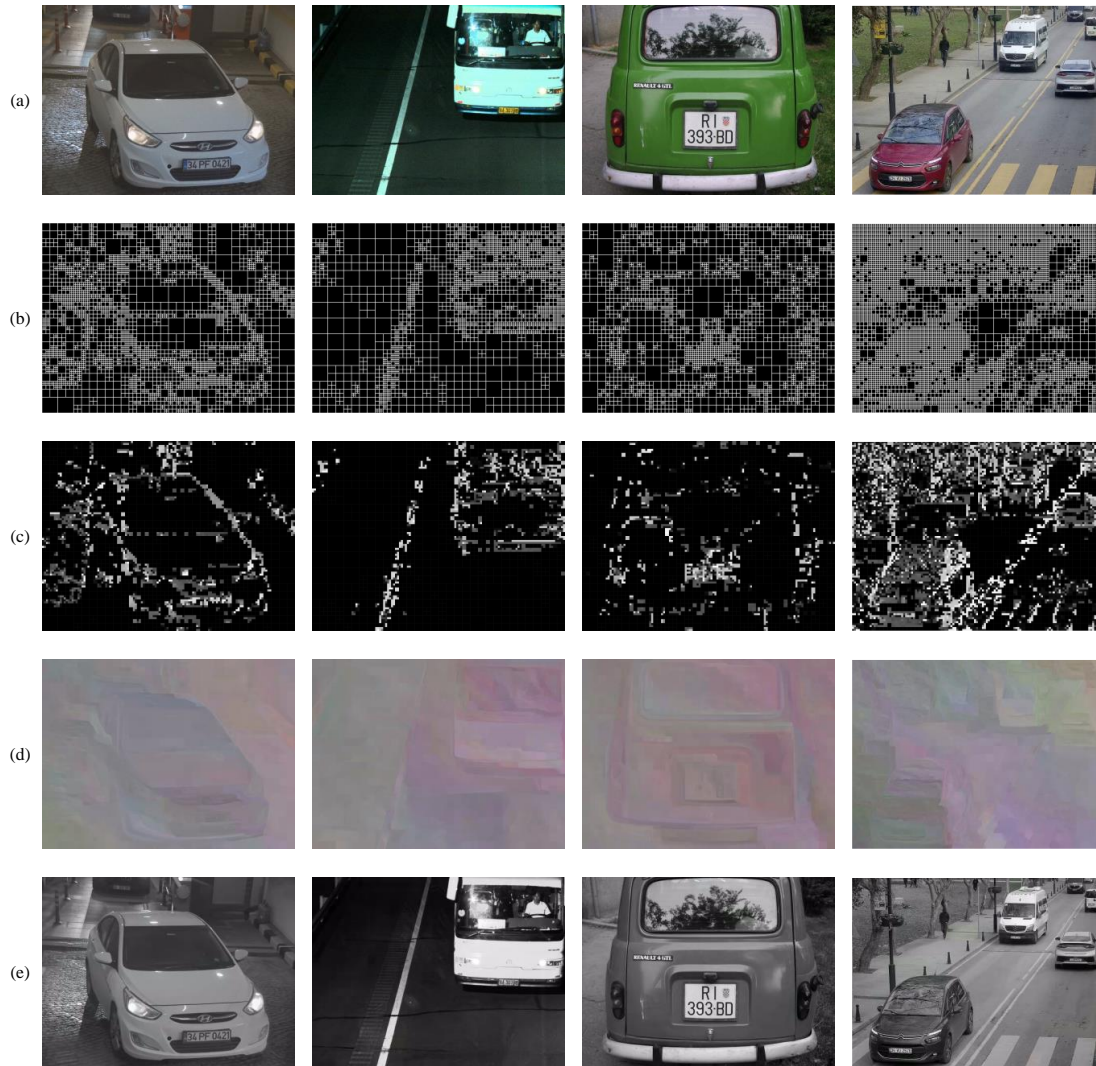


Figure 4.8 : Comparison of four different reconstructed images from various HEVC bitstreams. a) Pixel Domain Images (I_{px}). b) Block Partition Based Images (I_{bp}). c) Prediction Unit Based Images (I_{pu}). d) Random Perturbation Based Image (I_{rp}). e) Luma Based Images (I_y).

It can be seen that as we progress from using only the block partition information to including prediction units, random perturbations, and finally luma data, the clarity of the reconstructed images significantly improves. However, this enhancement in image quality comes at the expense of increased computational complexity, presenting a trade-off that must be carefully considered in the design of efficient and effective video surveillance systems. This comprehensive evaluation allows us to ascertain the suitability of each method for different applications, guiding the development of more refined and efficient solutions for video analytics in the compressed domain.

4.7 Experimental Results

This section delves into a detailed experimental assessment of the proposed compressed domain methods, focusing primarily on the time efficiency and data size implications of each approach. The setup and procedure of the conducted experiments are elaborated, followed by an examination of the computational complexity in terms of reconstruction steps, time taken for image reconstruction, and the data size requirements for storage and transmission. These comprehensive evaluations offer a clear understanding of the practical trade-offs involved when employing these methods in real-world video analytics applications. It is important to note that the current examination is centered on the time and data aspects of the proposed methods. The accuracy of these methods, a crucial element for their practical application, will be the subject of in-depth analysis in the subsequent chapters.

4.7.1 Experiment setup

The following setup is used for the experiments:

- **Computer:** The experiments were conducted on a computer with an Intel(R) Core(TM) i9-9900X CPU, NVIDIA GeForce GTX 1080 Ti GPU, and 48 GB RAM running a Windows 11 64-bit operating system.
- **Software:** The reference software for the H.265/HEVC coding standard, known as HM (version 16.20), was used to implement the proposed method. This software, developed by the Fraunhofer HH Institute, is written in C++ and is provided as

source code. Researchers can modify the software as needed and then compile it to generate the executable encoder, decoder, and other modules.

- **Compiler:** The Microsoft Visual Studio 2019 (v142) platform tool-set is used to compile the reference software.
- **Image Database:** The experiments are conducted on 1000 distinct images, all of which are encoded using the reference software. The results are averaged to provide a representative value. The images are 1024×768 pixels in size and in color.
- **Encoding Parameters:** The main profile is used for encoding, with 4:2:0 color encoding and a quantization parameter of 32. All images are encoded using these parameters.

4.7.2 Measurement of reconstruction steps

The following processes are measured to determine the reconstruction time of each method:

- **Entropy Decoding (*ED*):** This is a common step for all methods.
- **Intra Prediction (*IP*):** Intra prediction is utilized for I_{rp} , I_y , and I_{px} images. In I_y , this step takes approximately 60% of the time required for I_{rp} and I_{px} .
- **Residual Decompression (*RD*):** While this stage is omitted for I_{rp} , it is essential for color and grayscale images in I_{px} and I_y , respectively. Notably, residual decompression in I_y demands about 60% of the time required for I_{px} .
- **Loop Filters (*LF*):** Loop filters are used for I_{pr} , I_y , and I_{px} according to the standard.
- **Block Partition Based Image Generation (*IG_{bp}*):** This process involves generating I_{bp} after entropy decoding.
- **Prediction Unit Based Image Generation (*IG_{pu}*):** This process involves generating I_{pu} after entropy decoding.

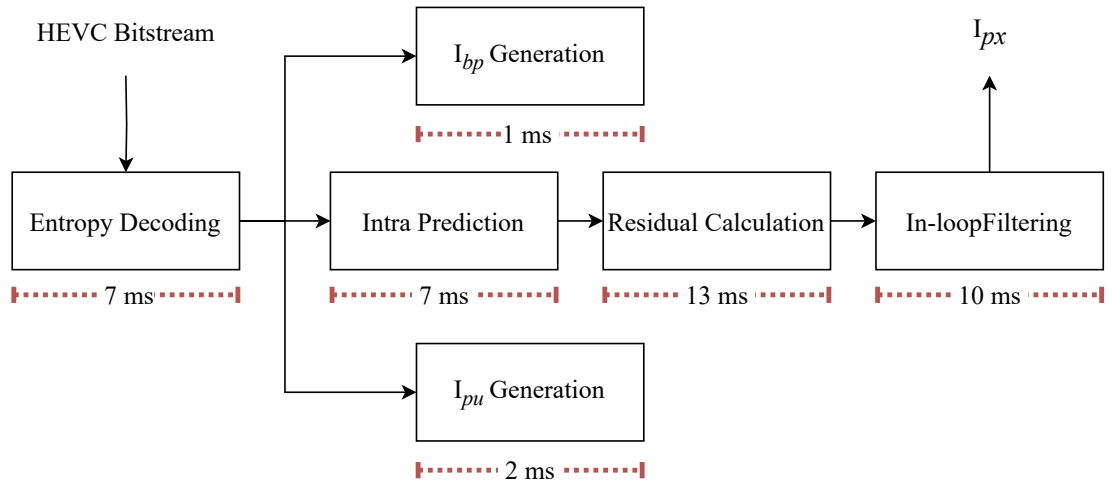


Figure 4.9 : Diagram showing the measurement of steps in the image reconstruction process.

Figure 4.9 illustrates the steps involved in reconstructing images using the different methods described in this chapter and the details of time measurement is given in Table 4.1.

Table 4.1 : Quantitative analysis of image reconstruction steps.

Time (ms)	Average	Min.	Max.
Entropy Decoding	6.59	6.00	8.00
Loop Filters	9.70	8.00	16.00
Intra Prediction	7.04	4.00	10.00
Residual Construction	12.92	8.00	17.00
I_{bp} Generation	0.96	1.00	2.00
I_{pu} Generation	1.96	1.00	4.00

The experiments are done on an Intel(R) Core(TM) i9-9900X CPU @ 3.50GHz CPU, NVIDIA GeForce GTX 1080 Ti GPU, 48 GB RAM and Windows 11 64-bit operating system using JCT-VC HEVC reference software version HM 16.9.

4.7.3 Comparison of image reconstruction time

The reconstruction time for each image type varies depending on the operations required to generate it. To quantify this, we calculate the elapsed time for different images as follows:

For I_{px} , the pixel domain image, the reconstruction time $T(I_{px})$ is determined by summing the time taken by Entropy Decoding (ED), Intra Prediction (IP), Residual Decompression (RD), and Loop Filters (LF):

$$T(I_{px}) = T(ED) + T(IP) + T(RD) + T(LF) \quad (4.11)$$

For I_y , the Luma-based image, the reconstruction time $T(I_y)$ accounts for the same steps as I_{px} , but the time taken by Intra Prediction and Residual Decompression is reduced by 40%, yielding:

$$T(I_y) = T(ED) + 0.6 \cdot T(IP) + 0.6 \cdot T(RD) + T(LF) \quad (4.12)$$

For I_{rp} , the Random Perturbation-based image, the reconstruction time $T(I_{rp})$ excludes the time for Residual Decompression, and is computed as:

$$T(I_{rp}) = T(ED) + T(IP) + T(LF) \quad (4.13)$$

For I_{pu} , the Prediction Unit-based image, the reconstruction time $T(I_{pu})$ includes the time for Entropy Decoding and Image Generation based on Prediction Units (IG_{pu}):

$$T(I_{pu}) = T(ED) + T(IG_{pu}) \quad (4.14)$$

Lastly, for I_{bp} , the Block Partition-based image, the reconstruction time $T(I_{bp})$ is determined by the time taken by Entropy Decoding and Image Generation based on Block Partition (IG_{bp}):

$$T(I_{bp}) = T(ED) + T(IG_{bp}) \quad (4.15)$$

Table 4.2 demonstrate that the proposed compressed domain methods significantly expedite the image reconstruction process compared to the traditional full decoding method. The compressed domain methods, specifically the Block Partition Based (M_{bp}) and Prediction Unit Based (M_{pu}) methods, achieved average reconstruction

Table 4.2 : A comparative analysis of image reconstruction durations across different methods.

Time(ms)	I_{bp}	I_{pu}	I_{rp}	I_y	I_{px}
Average	7.55	8.54	23.33	28.27	36.25
Minimum	7.00	7.00	18.00	21.20	26.00
Maximum	10.00	12.00	34.00	40.20	51.00

times of 7.55 ms and 8.54 ms respectively. These figures represent reductions in image generation time by approximately 79% and 76% when compared to the pixel domain method (M_{px}) generation time of 36.25 ms.

While the M_{rp} and M_y methods, which incorporate more sophisticated processing techniques, naturally require more time to reconstruct, they still demonstrate significant time efficiencies. The Random Perturbation Based (M_{rp}) method took an average of 23.33 ms and the Luma Based (M_y) method took 28.27 ms, both still considerably faster than the full decoding process.

The marked reduction in reconstruction time, particularly with the M_{bp} and M_{pu} methods, underlines the efficacy of these approaches in scenarios where speed is paramount. The associated trade-off between reconstruction time and image detail is a key consideration, and the choice of method will depend on the specific requirements of the application. These results are further illustrated in the graph shown in Figure 4.10, underscoring the significant time efficiencies achieved by the proposed compressed domain methods.

4.7.4 Comparison of data size

Table 4.3 : Comparison of image sizes and depths across different reconstruction methods.

Image	Dimension	Depth	Size in bit	Ratio
I_{px}	1,024×768	24 (color)	18,874,368	1/1
I_{pu}	128×96	8 (gray)	98,304	1/192
I_{bp}	128×96	1 (B&W)	12,288	1/1,536
I_{rp}	1,024×768	24 (color)	18,874,368	1/1
I_y	1,024×768	8 (gray)	6,291,456	1/3

The comparison of data sizes for different image reconstruction methods is crucial, especially when considering the volume of data that must be managed in memory after

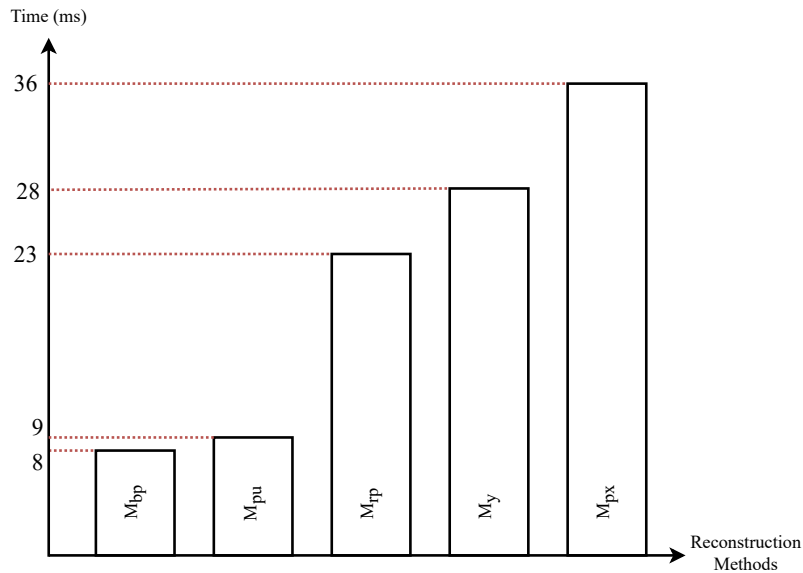


Figure 4.10 : Comparison of image generation times for various methods. Image generation in the compressed domain is 28% to 79% faster than in the pixel domain.

image reconstruction. As shown in Table 4.3, the different reconstruction methods result in significantly different data sizes. It's important to note that this table is created under the assumption of conversion in 24-bit RGB or 8-bit grayscale format or 1-bit format when possible, to hold the data in memory.

The pixel domain image I_{px} is a full-color image with three channels and a resolution of $1,024 \times 768$, resulting in the largest data size. Conversely, the Prediction Unit Based Image I_{pu} , being a grayscale image with a drastically reduced resolution of 128×96 , requires 192 times less data to be managed in memory. The Block Partition Based Image I_{bp} , despite sharing the resolution of I_{pu} , is a binary image and thus demands even less data, 1,536 times less than the pixel image. The Random Perturbation Based Image I_{rp} carries an equal amount of data to the pixel image due to its identical resolution and color depth, while the Gray-level Image I_y , as a grayscale rendition of the pixel image, requires only one-third of the data.

It's important to note the significant reduction in data size when using the I_{pu} and I_{bp} reconstruction methods. This decrease directly impacts the volume of data that needs to be handled in memory, offering a tangible advantage for applications where resource usage is a critical factor.



Figure 4.11 : Comparison of the size of the pixel domain image and the compressed domain image. a) A 1024×768 pixel domain image. b) A compressed domain image corresponding to the pixel domain image with a resolution of 128×96 pixels, where each CU is represented by a single pixel.

During the creation of Block Partition Based (I_{bp}) and Prediction Unit Based (I_{pu}) images, each Coding Unit (CU) is symbolized by a single pixel, resulting in an image that is eight times smaller in both dimensions than the original size. The grayscale reconstructed images (I_y) also lead to less data, further optimizing the data size. The significant disparity in data size between the pixel image and the images constructed from HEVC attributes is graphically presented in Figure 4.11.

In video compression standards like HEVC, the encoded bitstream holds a substantial amount of data related to prediction error for image samples, known as residuals. For I-frames, these residuals make up about 85-90% of the total data in the bitstream [64]. Furthermore, residual coding accounts for an average of 77% and 84% of the total bits for dynamic continuous and discrete video textures, respectively [65].

Our proposed methods, I_{bp} , I_{pu} , and I_{rp} , greatly reduce the amount of data required for image reconstruction by avoiding the use of residual data. They offer substantial data reduction, contributing to more efficient storage and transmission capabilities. This

is particularly valuable in surveillance applications, where efficient handling of vast amounts of video data is of utmost importance.

4.8 Conclusion

This chapter has provided an in-depth exploration of four novel methods for partial intra decoding aimed at compressed video understanding. Each of these methods harnesses unique properties of the video compression standard HEVC, offering fresh perspectives and strategies for image reconstruction.

The Block Partition Based Method (M_{bp}) employs the block partitioning attribute available in the bitstream, generating a binary image that offers a rough outline of the structure within the frame. This method has demonstrated remarkable speed in the image reconstruction process, making it a valuable tool for applications that require quick response times.

In contrast, the Prediction Unit Based Method (M_{pu}) uses prediction unit attributes to create grayscale images. Although it requires slightly more time than I_{bp} , it offers a more detailed representation of the image, providing a balance between speed and detail that could be beneficial in certain applications.

The Random Perturbation Based Method (M_{rp}) constructs a near silhouette of the pixel domain image while preserving object boundaries in the frame, all without requiring residual data. This approach maintains significant reductions in transferred data size and reconstruction time, providing crucial information for image classification and object detection tasks.

The Luma Based Method (M_y) creates grayscale images using luma data from the bitstream. While it necessitates more data and time than the attribute-based methods, it provides a high-quality grayscale image that can be beneficial in applications where image detail is crucial.

Experimental results confirmed the effectiveness of these methods, especially in terms of reconstruction time and data size.

In conclusion, the partial intra decoding methods introduced in this chapter demonstrate a compelling potential for efficient and effective compressed video

understanding. By capitalizing on the unique attributes of the HEVC bitstream, these methods offer viable alternatives for image reconstruction, striking a balance between speed, data efficiency, and image detail according to the requirements of the application.

5. LICENSE PLATE DETECTION IN COMPRESSED DOMAIN

The effectiveness of the partial intra decoding methods in terms of license plate (LP) detection performance is evaluated in this chapter. License plate recognition (LPR) is a widely used task in various applications, including vehicle identification, traffic control, and security surveillance. The task of license plate detection (LPD) involves detecting the presence and location of license plates (LPs) in images or video. The detection of LPs is a crucial step in LPD, as it serves as the foundation for subsequent tasks such as license plate recognition (LPR). In this study, the YOLO v3 Tiny object detection algorithm is employed to detect LPs in both pixel domain and compressed domain I_{bp} and I_{pu} images. To evaluate the performance of the proposed methods, publicly available datasets are utilized. The results show that while the performance of the proposed methods is comparable to that of the pixel domain image, they offer the advantage of faster processing times. In the following sections, the experimental setup and results of applying the proposed methods to LP detection are presented in detail.

5.1 Methodology

The YOLO object detection method [35], specifically the smaller version known as YOLOv3-tiny, is used in this study to train a convolutional neural network (CNN) for the detection of license plates (LPs). Good precision and recall rates and fast execution times (around 70 FPS) are reported for YOLO, which is also used in many recent works for real-time LP detection [34,38,39,66].

To compare the compressed domain with the pixel domain, the developed method is shown in Figure 5.1. The primary input is a HEVC bit-stream, which is decoded into three distinct image types. The first is a pixel domain image I_{px} that is decoded according to the HEVC standard. The other two are HEVC domain images, introduced as the Block Partition Based Image I_{bp} and the Prediction Unit Based Image I_{pu} . Three YOLO Tiny networks are trained, one for each image type, to detect LPs. Once the

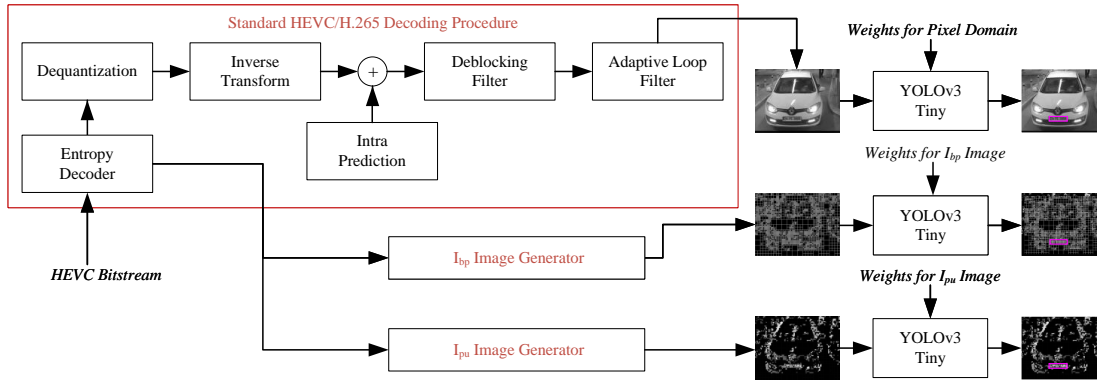


Figure 5.1 : The generation of High Efficiency Video Coding (HEVC) images and detection of License Plates (LPs) using three separate methods.

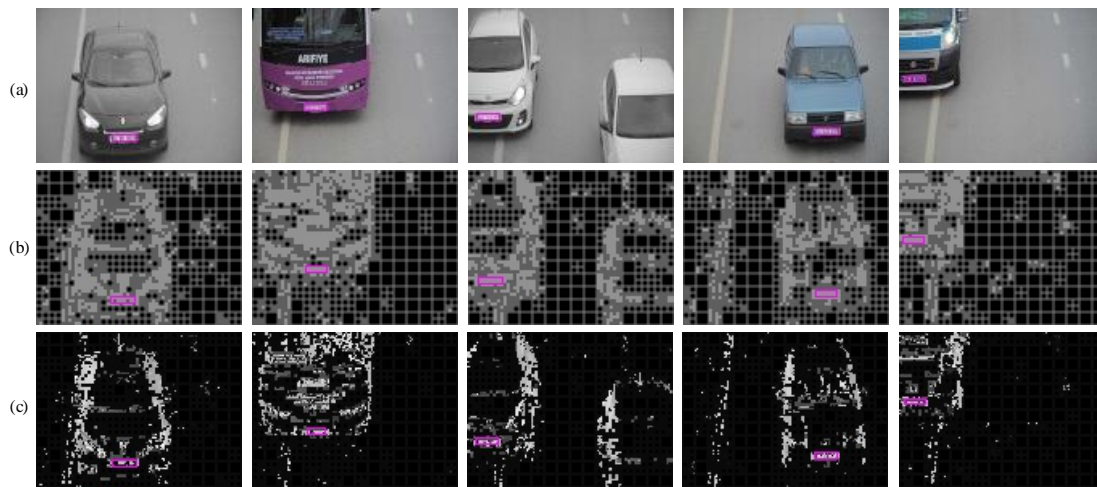


Figure 5.2 : Results for three different methods of LP detection from the test database. Plates that have been found are indicated with a pink rectangle. a) Detection results of LPD_{px} . b) Detection results of LPD_{bp} . c) Detection results of LPD_{pu} .

training is completed, the images are fed into the YOLO network with the appropriate weights for the detection of LP regions. The LP detection method using pixel domain images is referred to as LPD_{px} , and the methods using Block Partition Based Image and Prediction Unit Based Image are referred to as LPD_{bp} and LPD_{pu} , respectively.

In Figure 5.2, the LP detection results for various vehicles are shown, with the detected LP area marked with a purple rectangle. The LPs are successfully detected by all methods, as demonstrated in the results. The ability to detect LPs in compressed domain images, which allows for faster processing times compared to using pixel domain images, is an important achievement. This is particularly relevant in real-time

applications where speed is a critical factor. In the following section, the accuracy of the different LP detection methods is compared.

5.2 Experimental Results

In this section, we delve into the outcomes of the experimental procedure we conducted to evaluate our model. The detailed explanation of how we established the experiment, the nature of datasets used, the metrics adopted for measurement, the accuracy attained, and the time taken by our model in comparison with other models is included in this section.

5.2.1 Experiment setup

The following setup is used for the experiments:

- **Computer:** A computer with an Intel(R) Core(TM) i9-9900X CPU, NVIDIA GeForce GTX 1080 Ti GPU, and 48 GB RAM running a Windows 11 64-bit operating system is used.
- **Software:** The reference software for the H.265/HEVC coding standard, known as HM (version 16.20), is used for both encoding and decoding purposes. The "Main profile" is used for encoding, with 4:2:0 color encoding and a quantization parameter of 32.
- **Compiler:** The Microsoft Visual Studio 2019 (v142) platform tool-set is used to compile the reference software.
- **Obtaining HEVC bitstream:** All images in the used database in JPEG format are first intra encoded using HEVC encoders, resulting in corresponding bitstreams. This process is illustrated in Figure 5.3. It is important to note that the Ipx images are first encoded and then decoded again. This is done to ensure that a fair comparison can be made, as each source image is forced to face the same HEVC encoding distortion.
- **Training:** To train separate YOLO networks for each method, the following conditions are used:

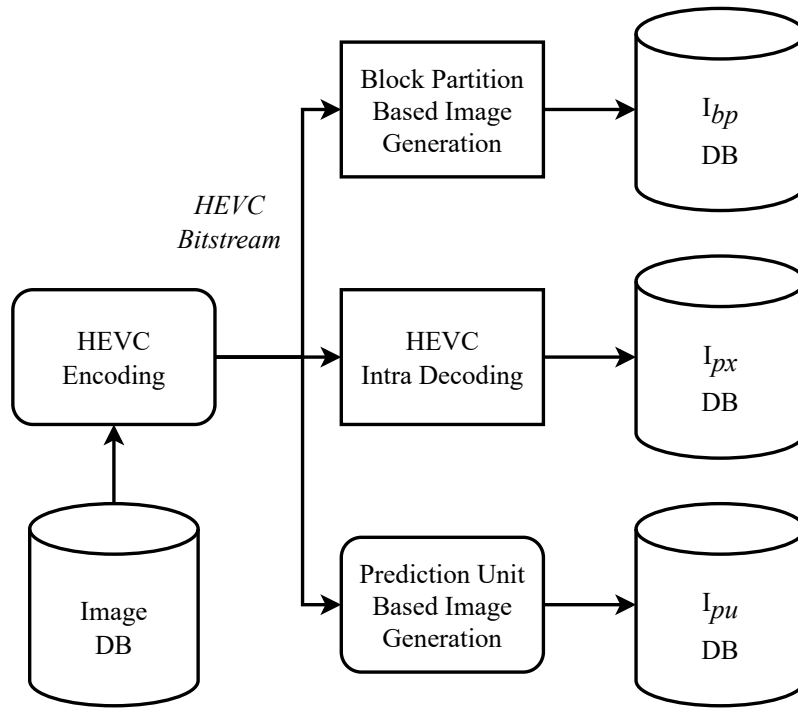


Figure 5.3 : Creating HEVC domain images from JPEG Images.

1. The same number of images with the same resolution are used for training and testing.
2. The same deep learning network structure is utilized using the same hyper parameters, including "learning rate", "batch size", "number of epochs to train for", and "number of nodes in the given layer".
3. A total of 500,000 batches are conducted to ensure that the average loss no longer decreases, and the weight with the best Mean Average Precision (mAP) is chosen from among the generated weights.

5.2.2 Datasets

Despite the widespread use of license plate recognition (LPR) systems globally, there is a notable scarcity of publicly available datasets. To address this, this thesis introduces and shares a new public-domain dataset, the Compressed Domain LP Dataset (CP-LP Dataset) [67]. Additionally, a second dataset, EnglishLP [68], is employed to compare the results with previous studies in this field.

5.2.2.1 CD-LP dataset

The CD-LP dataset contains images from commercial cameras that are currently operational and located on a highway and at a shopping mall’s entrance [69]. Generally, the images in this dataset depict the front view of a vehicle. They typically contain a single vehicle, but can occasionally contain two or more. The dataset contains $3 \times 2,400$ images in three formats: 2,400 I_{px} , 2,400 I_{bp} and 2,400 I_{pu} .

The original images had a resolution of $1,024 \times 768$ pixels in size. I_{bp} and I_{pu} represent each CU with a single pixel, resulting in images with a resolution of 128×96 . The I_{px} images are also resized to 128×96 pixels for the following reasons:

1. To make a fair comparison with HEVC images.
2. To make the database publicly accessible without raising privacy concerns.
3. It has been demonstrated that a resolution of 128×96 pixels is sufficient for achieving high accuracy in LP detection.

Table 5.1 : Distribution of the CD-LP dataset across different image types and resolutions.

Image	Resolution	Train Set	Test Set	Total
I_{px}	1024×768	1,800	600	2,400
I_{bp}	128×96	1,800	600	2,400
I_{pu}	128×96	1,800	600	2,400

The train set includes 1,800 images for each of the three formats, while the test set contains the remaining 600. Our dataset is summarized in Table 5.1. Each image in the database has a companion file containing plate annotation information in YOLO format.

5.2.2.2 EnglishLP dataset

A digital camera with a resolution of (640×480) pixels was used to capture images of the EnglishLP dataset. Over 500 images of the rear views of various vehicles (trucks,

cars, buses) were included in the database, taken under various lighting conditions (cloudy, sunny, rainy).

This dataset is divided in the same way as in [36] and [70], with 80% of the images being used for training. 20% of the images are used for testing as given in Table 5.2.

Table 5.2 : Distribution of the EnglishLP dataset across different image types and resolutions.

Image	Resolution	Train Set	Test Set	Total
I_{px}	640×480	407	102	509
I_{bp}	128×96	407	102	509
I_{pu}	128×96	407	102	509

5.2.3 Metrics

The results are evaluated using the F1-score, precision, recall, average intersection of union (Avg. IoU) and mean average precision (mAP). The Precision (I_{px}), Recall(R) and F1-score ($F1$) values are calculated based on True Positive (TP), False Positive (FP) and False Negative (FN) as shown in (5.1), (5.2) and (5.3), respectively. The F1-score is a metric that evaluates the sensitivity and accuracy criteria together.

$$R = \frac{TP}{TP + FN} \quad (5.1)$$

$$P = \frac{TP}{TP + FP} \quad (5.2)$$

$$F1 = \frac{2 \times P \times R}{P + R} \quad (5.3)$$

5.2.4 Accuracy

Accuracy is measured under two separate datasets, which are named 'CD-LP Dataset' and 'EnglishLP Dataset'. The distinction between these datasets and the accuracy achieved on each of them is explained in detail in the following subsections.

5.2.4.1 CD-LP dataset accuracy

Table 5.3 : Evaluation of accuracy metrics across different methods on the CD-LP dataset.

Method	TP	FP	P	R	F1	mAP
LPD_{px}	598	4	0.99	0.99	0.99	99.11
LPD_{bp}	533	70	0.88	0.89	0.88	82.78
LPD_{pu}	588	11	0.98	0.98	0.98	97.38

Detection results, where "TP" means true positive, "FP" means false positive, "P" means precision and "R" means recall.

The license plate (LP) detection results for the three different methods, namely, LPD_{px} , LPD_{bp} , and LPD_{pu} , are shown in Table 5.3. The pixel domain method (LPD_{px}) achieves the highest mAP overall.

On the other hand, the block partition based method in the HEVC domain (LPD_{bp}) produces an mAP result of 82.78%. Despite the substantial simplification of the image representation, this result shows that license plate detection is feasibly achievable to a considerable extent with this method.

Lastly, the prediction unit-based method in the HEVC domain (LPD_{pu}) demonstrates impressive performance. The mAP accuracy closely rivals that of the pixel domain method, with a mere difference of 1.73%. This implies that the LPD_{pu} method can effectively detect license plates with almost the same precision as the traditional pixel-based method, validating its potential for practical application.

5.2.4.2 EnglishLP dataset accuracy

The results of the comparison between methods developed in the compressed domain and those developed in the pixel domain using the publicly available EnglishLP dataset are summarized in Table 5.4. The block partitioning method achieved a recall rate of 0.94. The partition unit-based method achieved 1.00 recall and precision rates by correctly detecting all LPs in the test set, which is the same rate as the pixel domain approach and the research published in [37]. Although the suggested methods are in the compressed domain, they outperformed some of the studies in accuracy, including [37].

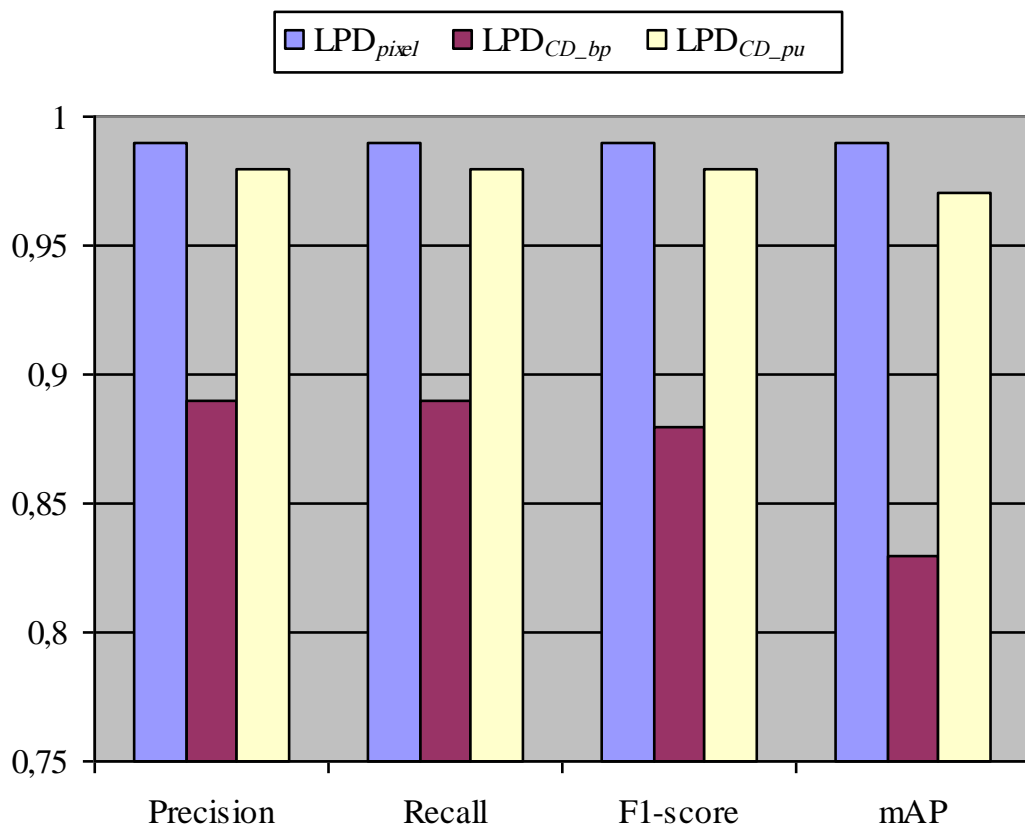


Figure 5.4 : The LP detection results for the CD-LP dataset. The performance of a compressed domain strategy based on prediction units is comparable to that of the pixel domain.

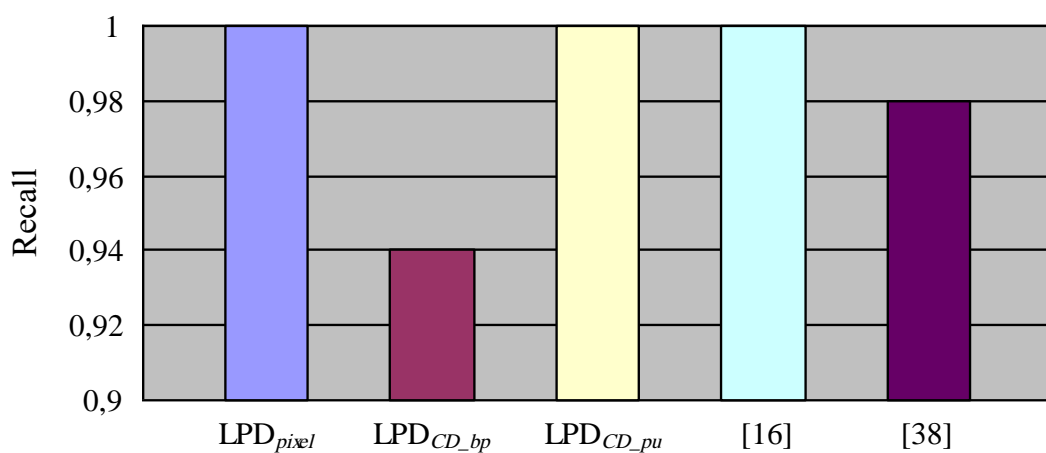


Figure 5.5 : The LP detection results for the publicly accessible English-LP dataset.

Table 5.4 : Evaluation of accuracy metrics across different methods on the EnglishLP dataset.

Method	TP	FP	P	R	F1	mAP
LPD_{px}	102	0	1.00	1.00	1.00	99.02%
LPD_{bp}	96	6	0.94	0.94	0.94	93.33%
LPD_{pu}	102	0	1.00	1.00	1.00	99.02%
[70]	-	-	-	0.98	-	-
[36]	-	-	-	1.00	-	-

Detection results, where "TP" means true positive, "FP" means false positive, "P" means precision and "R" means recall. The dash indicates that no data is shared for the related measurement.

5.2.5 Time comparison

In this section, the time performance of our LPD is put under scrutiny. The whole process is consists of the Image Reconstruction phase and the detection phase. These crucial elements of the LPD process are analyzed in the subsequent subsections.

5.2.5.1 Image reconstruction

The average reconstruction times for the three methods are given in Table 5.5. The compressed domain methods are about 4.5 times faster than the pixel domain approach.

Table 5.5 : Comparison of average image reconstruction durations across different methods.

Time (ms)	I_{px}	I_{bp}	I_{pu}
Average	36,24	7,55	8.54

5.2.5.2 LP detection

The chosen input image resolution of 128×96 significantly reduces the inference time of the DNN to approximately 2ms, corresponding to a frame rate of 500 frames per second, which is sufficient for real-time processing. The inference time for different input image size is given in 5.6. For an input size of 416×416 , the processing time is 3.6 ms on average. This LP Detection duration is same for both pixel and compressed domain images.

Table 5.6 : Processing time for LP detection using YOLOv3 Tiny at various image resolutions.

Resolution	Time (ms)	FPS
128×96	2	500
416×416	3.6	280

The experiments are done on an Intel(R) Core(TM) i9-9900X CPU @ 3.50GHz CPU, NVIDIA GeForce GTX 1080 Ti GPU, 48 GB RAM and Windows10 64-bit operating system.

A comparison of Table 5.5 with Table 5.6 reveals that image generation is the most time-consuming process, taking roughly 15 times as much time as LP detection. The proposed compressed domain methods provide a significant improvement in this time-consuming phase.

5.2.5.3 Overall process for LP detection

The entire LP Detection Process is consist of image reconstruction and LP Detection. The total time spent for LP Detection is given in Table 5.7. When the entire process is taken into account, LPDbp is 4 times faster and LPDpu is 3.6 times faster than pixel domain LP Detection.

Table 5.7 : Comparison of the duration of the entire LP detection process.

Time (ms)	LPD_{px}	LPD_{bp}	LPD_{pu}
Average	38.24	9.55	10.54

All methods are compared at a 128×96 image resolution.

Here's the modified conclusion without a mention of data size:

5.3 Conclusion

The experimental results obtained in this study demonstrate the effectiveness of the proposed license plate (LP) detection methods in the compressed domain. In terms of overall processing time, both LDP_{bp} and LDP_{pu} outperform the traditional pixel domain LPD, with LDP_{bp} achieving up to a 4 times speed-up, and LDP_{pu} up to 3.6 times.

The proposed LDP_{pu} method is particularly promising, successfully detecting all license plates with nearly the same precision as the pixel domain approach. These findings highlight the potential of utilizing compressed domain images for LP detection, particularly in real-time applications where computational speed is a critical factor. Given these promising results, the applicability and potential impact of the proposed methods in video analysis tasks appear significant.

6. VEHICLE DETECTION AND CLASSIFICATION IN COMPRESSED DOMAIN

Vehicle detection and classification stands as a pivotal application of computer vision technology, providing indispensable statistical data for managing and optimizing traffic in densely populated urban areas [71]. As these techniques continue to evolve, more efficient and innovative approaches are being explored. This study seeks to contribute to this progress by investigating the performance of vehicle detection in the compressed domain.

Notably, this marks one of the first attempts to leverage compressed domain images not just for object detection, but for vehicle classification as well. The efficacy of these compressed domain methods will be assessed in comparison with traditional pixel domain methods, offering insights into the strengths and potential benefits of this approach.

In this chapter, we will introduce our unique method for vehicle detection and classification in the compressed domain. Subsequent sections will then delve into the details of this method, followed by a thorough presentation and discussion of our experimental results.

By exploring these novel approaches to vehicle detection and classification, we aim to uncover opportunities for advancing traffic analysis techniques, ultimately contributing to the development of smarter and more sustainable urban environments.

6.1 Methodology

Our proposed methodology for vehicle detection and classification within the compressed domain involves two primary stages: Image Reconstruction, and Vehicle Detection and Classification.

The detailed workings of Image Reconstruction have been discussed thoroughly in Chapter 4.

For the Vehicle Detection and Classification stage, we utilize the cutting-edge YOLO V7 object detection model. YOLOv7, introduced recently, is a single-stage, real-time detector that stands as the fastest and most precise real-time object detector as of its conception, as per its foundational paper [46]. By outperforming its predecessors, this model has marked a significant milestone in real-time object detection. Among its variants, the YOLOv7-Tiny model, endowed with just over 6 million parameters, strikes an impressive balance between speed and accuracy, with its validation Average Precision (AP) of 35.2% outperforming those of earlier YOLO-Tiny models.

In our proposed approach, the reconstructed images from the compressed domain serve as input to the Vehicle Detection and Classification stage. Each type of compressed domain image is used to train a separate model within the Darknet framework [72]. Upon completion of training, these models are then utilized for vehicle detection and classification within the HEVC compressed domain.

By combining the power of the YOLOv7 object detector with our reconstructed images, our proposed method can effectively and accurately detect and classify vehicles in the compressed domain.

6.2 Experimental Results

In this section, we delve into the outcomes of the experimental procedure we conducted to evaluate our model. The detailed explanation of how we established the experiment, the nature of datasets used, the metrics adopted for measurement, the accuracy attained, and the time taken by our model in comparison with other models is included in this section.

6.2.1 Experiment setup

The following setup is used for the experiments:

- Computer: A computer with an Intel(R) Core(TM) i9-9900X CPU, NVIDIA GeForce GTX 1080 Ti GPU, and 48 GB RAM running a Windows 11 64-bit operating system is used.

- Software: The reference software for the H.265/HEVC coding standard, known as HM (version 16.20), is used for both encoding and decoding purposes. The "Main profile" is used for encoding, with 4:2:0 color encoding and a quantization parameter of 32 [73].
- Compiler: The Microsoft Visual Studio 2019 (v142) platform tool-set is used to compile the reference software.

Your text is already clear and precise, but here's a slightly refined version:

- Obtaining HEVC bitstream: Our experimental procedure commences with the acquisition of HEVC bitstreams, which serve as the principal input for our methodology. These bitstreams are derived from JPEG format images, I_{org} , from the BIT database, which are then transformed through intra-encoding utilizing the HEVC encoder [74].

Using these bitstreams, we generate five distinct image types. The first, a pixel domain image I_{px} , is decoded in line with the HEVC standard. The subsequent four image types originate from the HEVC domain and include the Block Partition Based Image I_{bp} , the Prediction Unit Based Image I_{pu} , the Random Perturbation Based Image I_{rp} , and the Luma Based Image I_y . The procedure for deriving different image types from I_{org} is delineated in Figure 6.2.

To ensure an equitable comparison, it's important to note that I_{px} images are first encoded and subsequently decoded from I_{org} . This approach ensures that all source images are subjected to the same level of HEVC encoding distortion.

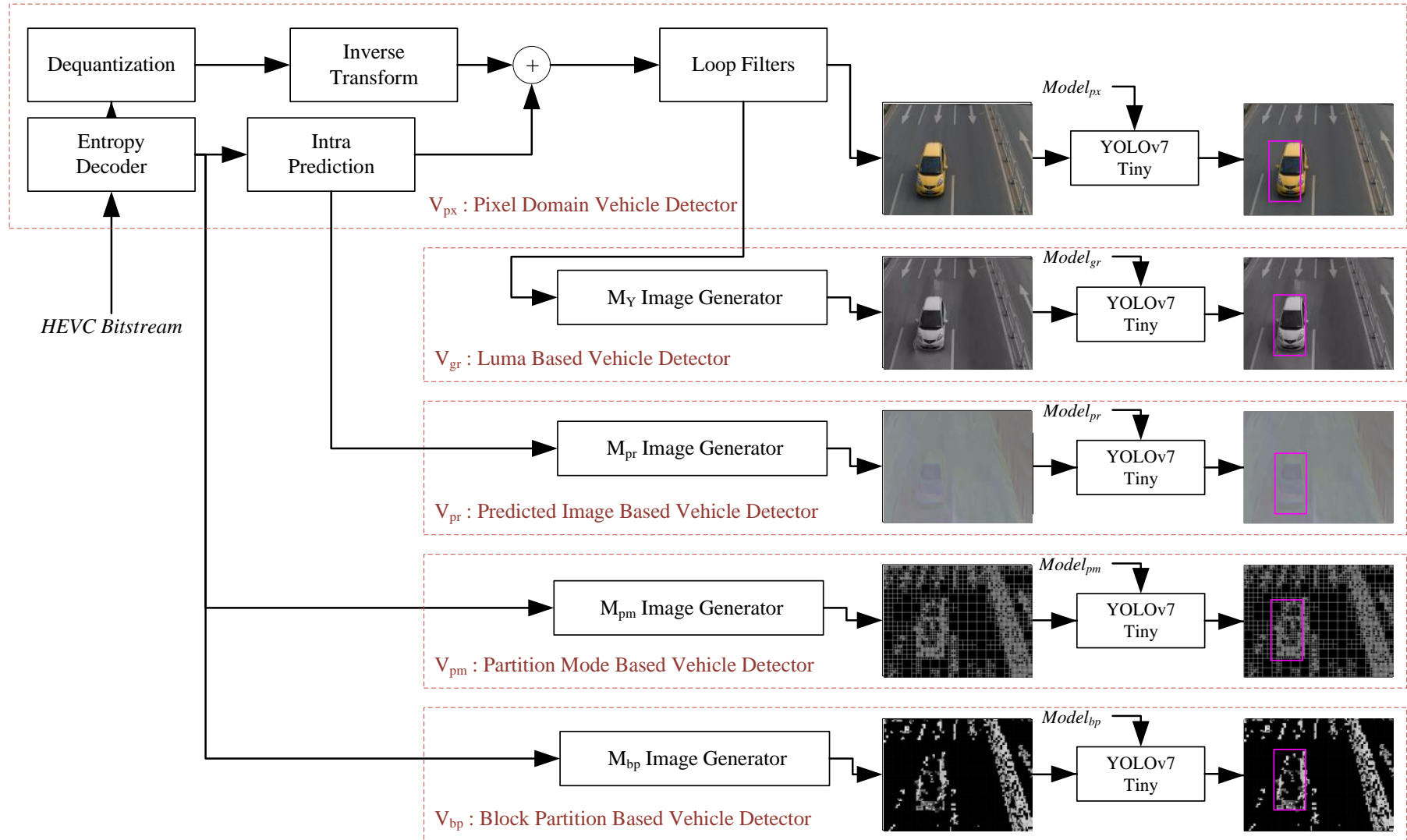


Figure 6.1 : Vehicle detection and classification using compressed domain images.

- Training: To train separate YOLO v7 Tiny networks for each method, the following conditions are used:

1. The same number of images with the same resolution are used for training and testing.
2. The same deep learning network structure is utilized using the same hyper parameters, including "learning rate", "batch size", "number of epochs to train for", and "number of nodes in the given layer".
3. A total of 250,000 batches are conducted to ensure that the average loss no longer decreases, and the weight with the best Mean Average Precision (mAP) is chosen from among the generated weights.

Our method, developed to compare the performance of compressed domain images against pixel domain images, is depicted in Figure 6.1. Separate YOLOv7-Tiny models networks are trained, each corresponding to one of these image types. Once the training is completed, the images are fed into the YOLO network with the appropriate weights for the detection of vehicles. The abbreviations V_{bp} , V_{pu} , V_{rp} , V_y , and V_{px} denote the vehicle detection and classification methods in combination with the image type used in each respective method.

In addition to these models, another model is trained for the original JPEG images, I_{org} , for comparison purposes. Furthermore, to observe the effect of standard deviation variation for random perturbation based method, a separate model is trained for each different standard deviation. In total, 15 different models are trained for the vehicle classification task and 5 different models for vehicle detection task.

6.2.2 BIT vehicle dataset

We selected the BIT dataset [41] for our experiments due to its widespread use in previous research and the diverse set of images it offers for vehicle classification. The BIT-Vehicle dataset, provided by the Beijing Institute of Technology, comprises 9580 vehicle images featuring six types of vehicles: sedans, sport-utility vehicles (SUVs),

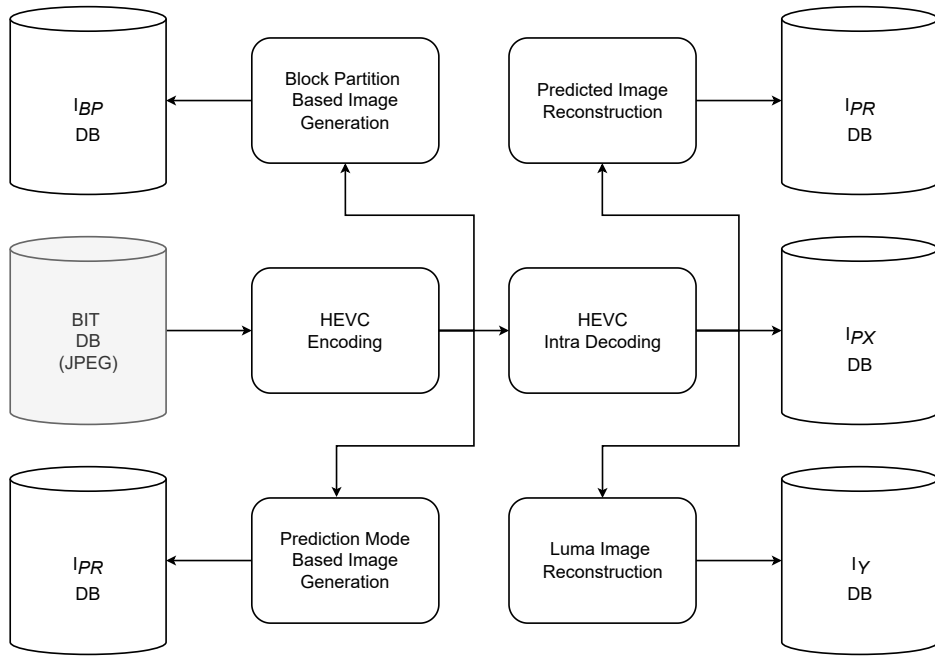


Figure 6.2 : Creating HEVC domain images from JPEG images.

microbuses, trucks, buses, and minivans. The dataset exhibits varying frequencies of vehicle types. Specifically, the number of vehicles per class is as follows: 558 buses, 883 microbuses, 476 minivans, 5922 sedans, 1392 SUVs, and 822 trucks. These images were captured by road surveillance cameras and include both day and night scenes, as well as sunny days with no background noise, rain, snow, people, or other vehicle types.

To ensure a fair comparison with previous works, the dataset was divided into a training set and a validation set with a ratio of 8:2, containing 7880 and 1970 images, respectively. This ratio was also maintained for each vehicle type to ensure a balanced representation across classes. Among these images, approximately 1000 and 250 were nighttime images for training and validation, respectively.

6.2.3 Metrics

The results are evaluated using the F1-score, precision (the proportion of correct detections among all positive predictions), recall (the proportion of correct detections among all instances of the class in the dataset), average intersection of union (Avg. IoU), and mean average precision (mAP) at the IoU threshold of 0.50 (mAP@0.50).

BIT Dataset

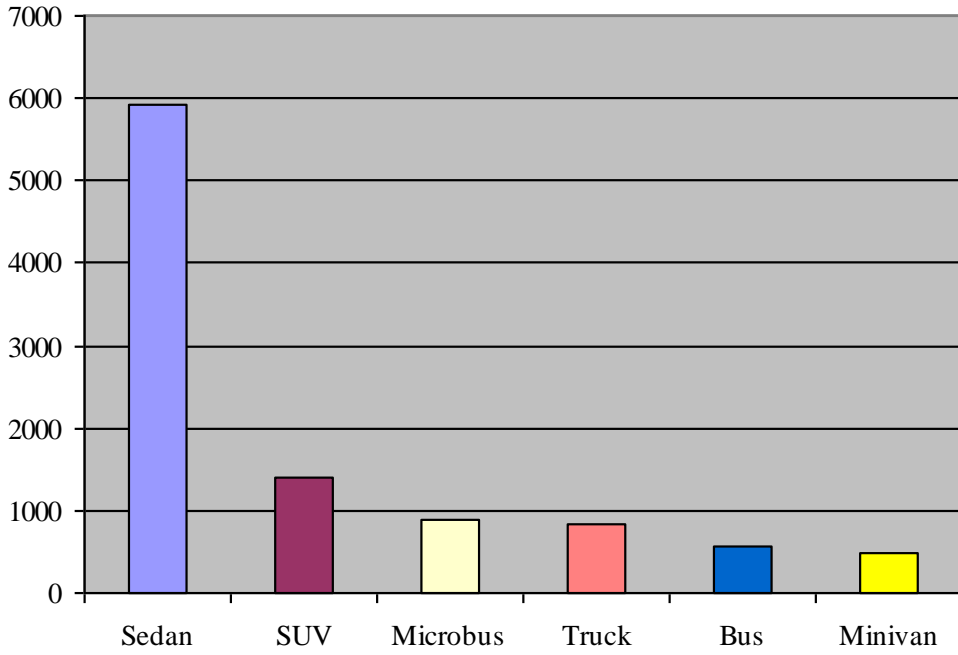


Figure 6.3 : The vehicle type distribution of the BIT Dataset.

mAP@0.50 is a metric commonly used to evaluate the performance of object detection algorithms. It is calculated as the mean of the Average Precision (AP) for each class in a dataset (6.1).

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i \quad (6.1)$$

where n is the number of classes in the dataset and AP_i is the Average Precision for class i .

To calculate AP, the algorithm's predictions are first sorted by their confidence scores. Then, AP is calculated as the area under the precision-recall curve with an IoU threshold of 0.50, as shown in (6.2).

$$AP = \frac{\sum_{k=1}^n P(k) \cdot rel(k)}{\sum_{k=1}^n rel(k)} \quad (6.2)$$

where $P(k)$ is the precision at cut-off k , and $rel(k)$ is a binary indicator of whether the prediction at cut-off k is a true positive or not, considering an IoU threshold of 0.50.

6.2.4 Vehicle detection accuracy

The BIT Vehicle dataset is originally annotated for six different vehicle types. To measure the vehicle detection performance, the dataset has been re-annotated, combining all vehicle types into a single "vehicle" class. The models are retrained using these four different image types. The obtained vehicle detection performance is presented in Table 6.1.

Table 6.1 : Vehicle detection accuracy for different methods.

Method	Mean Average Precision(mAP)
V_{bp}	98.34%
V_{pu}	99.89%
V_{rp}	99.99%
V_y	99.99%
V_{px}	99.99%

Looking at the results, the method V_{bp} , which uses Block Partition Based Image, achieved a mAP of 98.34%. The method V_{pu} , utilizing Prediction Unit Based Image, showed an improvement with a mAP of 99.89%.

The methods V_{rp} , V_y , and V_{px} all achieved the highest accuracy, with a mAP of 99.99%. These methods employ Random Perturbation Based Image, Luma Based Image, and Pixel Domain Image respectively.

It can be inferred from the results that the methods utilizing compressed domain images (V_{bp} , V_{pu} , V_{rp} , and V_y) show comparable, and in some cases equivalent, performance to the method using pixel domain images (V_{px}). This underscores the potential of compressed domain images for efficient vehicle detection tasks.

6.2.5 Vehicle classification accuracy

Table 6.2 presents the vehicle classification accuracy comparison for random perturbation images generated with various standard deviations. Each row corresponds to the classification accuracy obtained using a different standard deviation value for the random perturbations. The columns represent the Average Precision (AP) for each vehicle type, as well as the mean average precision (mAP) at an IoU threshold of 0.50.

Table 6.2 : Vehicle classification accuracy comparison of random perturbation based images generated with various standard deviations.

$V_{rp} \sigma$	The Class AP (%)						mAP@0.50 (%)
	Bus	Microbus	Minivan	Sedan	SUV	Truck	
1	99.46%	93.40%	93.12%	99.54%	93.98%	97.25%	96.13%
2	99.95%	95.71%	91.04%	99.61%	94.18%	97.03%	96.25%
3	99.97%	96.38%	92.59%	99.40%	93.94%	95.11%	96.23%
4	99.92%	94.14%	91.34%	99.72%	95.54%	94.56%	95.87%
5	99.91%	94.19%	92.43%	99.64%	94.40%	97.00%	96.26%
6	99.92%	95.33%	91.57%	99.65%	95.73%	96.47%	96.45%
7	99.99%	95.75%	93.08%	99.48%	95.04%	97.69%	96.84%
8	99.28%	95.06%	92.52%	99.67%	93.79%	96.62%	96.16%
9	99.91%	94.34%	91.09%	99.57%	95.47%	97.73%	96.35%
10	99.84%	94.62%	90.09%	99.52%	93.11%	98.16%	95.89%

From the results, it can be observed that for most vehicle types, the classification accuracy remains relatively high across different standard deviation values. The highest mAP is obtained when the standard deviation is set to 7, with a value of 96.84%. This indicates that the Random Perturbation Based Method performs well in the classification task for various standard deviations.

It is also worth noting that certain vehicle types, such as buses and sedans, consistently have higher classification accuracies than others, such as micro buses and minivans. This might be due to the distinctive features of these vehicle types, which make them easier to classify, as well as the higher frequency of sedans in the dataset. Overall, the table demonstrates the effectiveness of the Random Perturbation Based Method for vehicle classification across a range of standard deviation values.

Table 6.3 : Vehicle Classification Accuracy for Different Methods.

Implementation	Method	Domain	The Class AP (%)						mAP@0.50 (%)
			Bus	Microbus	Minivan	Sedan	SUV	Truck	
[45]	YOLOv3.Tiny	Pixel	-	-	-	-	-	-	95.05%
[42]	CNN	Pixel	-	-	-	-	-	-	93.90%
[43]	YOLOv2_Vehicle	Pixel	98.42%	97.04%	95.02%	97.37%	93.73%	97.80%	96.56%
[44]	Improved YOLOv2	Pixel	98.86%	96.63%	95.90%	98.23%	94.86%	99.30%	97.30%
[44]	YOLOv2 [56]	Pixel	98.34%	95.03%	91.11%	97.42%	93.62%	98.41%	95.65%
[44]	YOLOv3 [57]	Pixel	98.65%	96.98%	94.04%	97.65%	94.36%	98.17%	96.64%
[44]	Faster R-CNN VGG16 [75]	Pixel	99.05%	93.75%	91.38%	98.14%	94.75%	98.17%	95.87%
[44]	SSD300 VGG16 [40]	Pixel	97.97%	97.98%	90.28%	97.15%	91.25%	97.75%	93.75%
Ours	YOLOv7.Tiny I_{org}	Pixel	100.00%	97.70%	96.40%	99.76%	96.49%	99.23%	98.26%
Ours	YOLOv7.Tiny I_{px}	Pixel	100.00%	96.10%	96.36%	99.76%	95.95%	98.74%	97.82%
Ours	YOLOv7.Tiny I_y	Pixel	100.00%	96.54%	96.39%	99.09%	95.58%	97.28%	97.48%
Ours	YOLOv7.Tiny I_{bp} [60]	Compressed	83.52%	65.40%	63.51%	87.36%	66.61%	84.29%	75.11%
Ours	YOLOv7.Tiny I_{pu} [76]	Compressed	99.83%	92.22%	89.16%	99.51%	93.02%	98.37%	95.35%
Ours	YOLOv7.Tiny I_{rp} $\sigma = 7$	Compressed	99.99%	95.75%	93.08%	99.48%	95.04%	97.69%	96.84%

Table 6.3 presents the vehicle classification accuracy for different methods in both pixel and compressed domains, comparing our proposed compressed domain methods with results from the literature and pixel domain methods. The columns represent the implementation, method, domain, the Average Precision (AP) for each vehicle type, and the mean average precision (mAP) at an IoU threshold of 0.50.

In the given table, there is a distinction between the I_{org} and I_{px} results. The I_{org} refers to the original JPEG files in the BIT dataset, while I_{px} represents the encoded and re-decoded versions of the I_{org} images. The purpose of this comparison is to measure the potential performance loss caused by lossy compression. As a result, the performance of I_{org} is higher than that of I_{px} , indicating that lossy compression may have a negative impact on the classification accuracy. It is important to note that this effect is also applicable to the compressed domain experiments, such as I_y , I_{rp} , I_{bp} , and I_{pu} .

For our proposed compressed domain method, the YOLOv7.Tiny V_{rp} with $\sigma = 7$ achieves an impressive mAP of 96.84%, indicating that the method performs well even in the compressed domain. Although this performance is slightly lower than the pixel domain performance (97.82%, YOLOv7.Tiny I_{px}), it still demonstrates the potential of compressed domain approaches for traffic surveillance applications. In comparison to the literature, the proposed compressed domain method outperforms some of the pixel domain methods, such as YOLOv3.Tiny, CNN, YOLOv2_Vehicle, YOLOv2, YOLOv3, SSD300 VGG16, and Faster R-CNN VGG16. This indicates that the V_{rp} can provide a viable alternative to pixel domain methods for certain applications, especially when considering the significant speedup in the image reconstruction process.

Our compressed domain method, YOLOv7.Tiny V_{rp} , offers better accuracy than the other compressed domain methods YOLOv7.Tiny V_{pu} and YOLOv7.Tiny V_{bp} , with mAP values of 95.35% and 75.11%, respectively. This demonstrates the effectiveness of the proposed random perturbation-based approach.

When comparing the pixel domain methods, it is evident that our YOLOv7-based models excel in this domain. Although the main focus of our research is on compressed domain methods, the YOLOv7-based models have demonstrated their superiority

over other pixel domain approaches such as YOLOv3.Tiny, CNN, YOLOv2_Vehicle, Improved YOLOv2, YOLOv2, YOLOv3, SSD300 VGG16, and Faster R-CNN VGG16.

Regarding the performance of the I_y compressed domain variant, the YOLOv7.Tiny I_y method yields an mAP of 97.48%. While this is slightly lower than the pixel domain performance, it proves that the Luma Based Image approach remains competitive and achieves commendable results. The results underscore the possibility of achieving high accuracy levels in the compressed domain, contributing to the advancement of compressed domain methods in the realm of vehicle classification.

In the context of Block Partition Based (I_{bp}) and Prediction Unit Based (I_{pu}) images, it is noteworthy that although the I_{bp} method yields a lower mAP of 75.11%, it provides crucial insights about the impact of different types of compressed representations on the classification performance. On the other hand, I_{pu} , with an mAP of 95.35%, further corroborates the practicality of leveraging compressed domain information for the task at hand. It also substantiates the impact of more refined and complex structures such as Prediction Units on the classification performance.

In conclusion, our proposed compressed domain methods, particularly the random perturbation and luma based ones, are promising alternatives to the pixel domain methods, striking a balance between performance and computational efficiency. The success of these methods paves the way for further research and exploration into the realm of compressed domain image processing for vehicle classification.

6.2.6 Time comparison

The process of vehicle detection and classification consists of image construction and vehicle classification. The vehicle classification phase takes approximately 2ms for both the pixel domain and compressed domain methods, thanks to the use of the YOLO Convolutional Neural Network (CNN) which can carry out both detection and classification in one stage.

The main distinction between the pixel domain and compressed domain methods lies in the average reconstruction time, with the compressed domain methods taking 2 to

4 times less time than the pixel domain decoding method. The total time spent for the reconstruction and classification phases for each method, as well as a comparison of the vehicle detection and classification accuracy summarized in Table 6.4.

Table 6.4 : Vehicle Detection & Classification Accuracy vs Time.

Method	Vehicle Detection mAP	Vehicle Classification mAP	Time (ms)
V_{bp}	98.34%	75.11%	9.55
V_{pu}	98.9%	95.35%	10.54
V_{rp}	99.9%	96.84%	25.33
V_y	99.9%	97.48%	30.37
V_{px}	99.9%	97.82%	38.25

Table 6.4 provides a comparison of the vehicle detection and classification accuracy, as well as the processing time for several methods. It illustrates the inherent trade-off between accuracy and computational efficiency, highlighting the potential of compressed domain methods for applications that prioritize speed without significantly compromising accuracy.

The methods V_{bp} (Block Partition based), V_{pu} (Prediction Unit based), and V_{rp} (Random Perturbation based) represent our proposed compressed domain methods, while V_y (Luma Based Image) and V_{px} (pixel-based approach) represent the pixel domain methods.

In the compressed domain, V_{bp} has the quickest processing time at 9.55 milliseconds, albeit with a relatively lower classification accuracy at 75.11%. The V_{pu} method shows a considerable increase in accuracy (95.35%) with a slightly increased processing time of 10.54 milliseconds, demonstrating an impressive balance between accuracy and efficiency.

V_{rp} presents a further improvement in accuracy (96.84%), albeit at the cost of significantly higher processing time, 25.33 milliseconds. Despite this increase, it is still faster than pixel domain methods (V_y and V_{px}), which require 30.37 and 38.25 milliseconds respectively, while achieving comparable accuracies.

In terms of processing time, our proposed compressed domain methods demonstrate shorter reconstruction times compared to the standard HEVC decoding method (V_{px}). This, coupled with their respectable classification accuracies, showcases the potential

of compressed domain methods for real-time or near real-time applications where quick processing is paramount.

This notable trade-off between accuracy and reconstruction time underscores the viability of compressed domain methods. For applications that prioritize speed without significantly compromising on accuracy, these methods present a more attractive and efficient alternative to traditional pixel domain methods.

6.3 Conclusion

This chapter presented an innovative approach to vehicle detection and classification in the compressed domain, demonstrating that it is possible to achieve high accuracy without the need for exhaustive decoding of HEVC compressed images. The utilization of compressed domain information provided substantial benefits in terms of efficiency and speed, opening new avenues for real-time traffic surveillance applications.

Our experimental results underscored the effectiveness of this compressed domain approach. Despite the inherent lossy compression distortion, the classification accuracy remained comparable to the results from pixel domain methods. The proposed compressed domain methods, namely the Block Partition Based Method (V_{bp}), Prediction Unit Based Method (V_{pu}), Random Perturbation Based Method (V_{rp}), and Luma Based Method (V_y), demonstrated promising results, reinforcing the feasibility of the compressed domain approach.

It is noteworthy that the trade-off between accuracy and reconstruction time, particularly in compressed domain methods, offers significant implications for practical applications. These methods substantially reduced the reconstruction time compared to the standard HEVC decoding process, thus demonstrating their suitability for real-time or near-real-time applications where speed is a critical factor.

Furthermore, our research contributes to the literature by offering a new perspective on the potential of compressed domain information in traffic surveillance. The findings underscore the viability of such approaches, providing a robust foundation for further exploration and optimization of compressed domain methods.

7. CONCLUSIONS

Embarking on an exploration of video analytics with a particular focus on traffic surveillance applications, this thesis navigated the complexities of data transmission and video understanding tasks using intra frame encoded bitstreams. This journey has provided us with a myriad of insights and findings, brought to light in the preceding chapters. In this final chapter, we will recapitulate these discoveries, reflect on their potential future implications, and discuss prospective areas of research.

7.1 Discussion and Future Work

The body of work presented here signifies a leap forward in video analytics and traffic surveillance. By introducing and evaluating innovative methods for efficient data transmission and video understanding tasks using intra frame encoded bitstreams, we have demonstrated notable results in license plate detection and vehicle classification. These achievements in terms of accuracy, speed, and data minimization compliance underscore the value and potential of our methods.

However, this only marks the beginning of a larger research journey. Our current solutions are focused on intra frames. To further elevate the efficiency and accuracy of object detection, future work can venture into the realm of inter frames, aiming for full video decoding that encompasses both intra and inter encoded bitstreams.

In light of our exploration, we also propose further research to fine-tune the standard deviation used in the M_{rp} method. Our examination of the effects of different standard deviations on the database suggests that finding an optimal value can enhance the method's effectiveness. This optimal standard deviation could be seamlessly integrated into the encoding stream. Alternatively, analyzing residual data without reconstruction may provide insights into the appropriate standard deviation. This presents an intriguing direction for future research that could improve the efficiency and accuracy of the M_{rp} method.

Exploring the application of diffusion models [77] could also be a promising direction for future research. These models, which simulate a diffusion process to model data distributions, could potentially enhance the visualization quality of the M_{rp} method. Better visualization could lead to an improvement in the overall effectiveness of this method, and provide a more detailed understanding of the encoded information within the bitstream.

Modifications to existing compression standards may also be a topic of investigation in the future. Enabling transmission without residual data, for instance, could bolster the efficiency of the proposed methods. This may require the development of appropriate software and hardware solutions that can effectively implement these modifications in practical applications.

As we look ahead, we see an opportunity in amalgamating unique features obtained in the compressed domain with information in the pixel domain. This hybrid approach may result in a robust feature set surpassing the performance of solutions relying solely on the pixel domain. This endeavor holds promise for enhancing object detection accuracy and efficiency.

In the era of digital privacy, the proposed methods carry significant implications for data privacy regulations such as the EU General Data Protection Regulation (GDPR) and the California Privacy Rights Act (CPRA). Operating directly in the compressed domain with significantly reduced data requirements for image reconstruction, these methods are closely aligned with the data minimization principles of these regulations. Detailed investigation into this domain can potentially give rise to privacy-preserving solutions in video analytics.

Furthermore, the flexibility of our methods suggests their potential adaptability to other object detection applications, such as pedestrian detection. This adaptability enhances their value and applicability across a broad range of scenarios.

In sum, the impact of this research, coupled with the potential for future advancements, underscores the significance of this field of study. As we move forward, we anticipate

continuous growth and development, leading to meaningful advancements in the realm of video analytics.

7.2 Closing Remarks

This thesis presented novel methods for reconstructing images from High Efficiency Video Coding (HEVC) intra bitstreams, underlining a promising alternative to traditional video understanding tasks in the domain of traffic surveillance. Four distinct methods were introduced and evaluated for license plate detection and vehicle classification, with a new dataset curated specifically for the former task.

The results illustrated that all proposed methods were successful in achieving respectable accuracy levels. Some of them even matched the performance of the standard decoding method while demonstrating considerable speed advantages. This enhanced speed makes the methods a promising tool for applications where timely processing is crucial.

Our specific findings include:

- For intra coded frames, residual information accounts for up to 90% of the transmitted data. Our proposed methods, including M_{bp} , M_{pu} , and M_{rp} , allow for object detection without the need for residual data, reducing data transmission requirements.
- The compressed domain images created via the M_{bp} and M_{pu} methods occupy significantly less memory compared to the pixel domain image. Specifically, the image created with M_{bp} occupies 1/1,536 of the memory required by the pixel domain image, while the image created with M_{pu} requires 1/192 of the memory.
- The proposed methods result in a computational speedup between 1.25 to 4 times relative to the pixel area.
- Vehicle license plate locations were detected with 93.33% accuracy using M_{bp} , and with 99.02% accuracy using M_{pu} , comparable to the performance level in the pixel area. Additionally, the M_{bp} and M_{pu} methods sped up the license plate location finding process by nearly four and approximately 3.6 times, respectively.

- M_{rp} , the first method to reconstruct images without using residual data in video transmission, was introduced and evaluated for vehicle detection and classification processes.
- Vehicle detection was achieved with 98.99% accuracy using M_{pu} and with an impressive 99.99% accuracy using M_{rp} . Moreover, the vehicle detection process saw speed enhancements of over 1.5 times with M_{rp} and more than 1.25 times with M_y .
- Vehicles were classified into six different categories in the compressed domain. Performance rates achieved were 95.35% with M_{pu} , 96.84% with M_{rp} , and 97.48% with M_y .

By capitalizing on the encoded information and reducing the data required for image reconstruction, the methods we proposed improve processing efficiency. They demonstrate a significant stride towards a more effective, efficient, and privacy-compliant traffic monitoring system.

In conclusion, the research encapsulated in this thesis makes a significant contribution to the field of video analytics. It offers a new perspective and methods for efficiently handling video data for traffic surveillance applications while respecting privacy norms. As we move forward, it is hoped that this work will serve as a foundation for further exploration and innovation in this domain, inching us closer to a future where video surveillance is not only more efficient but also respects the importance of privacy.

REFERENCES

- [1] **Yang, Z. and Pun-Cheng, L.S.** (2018). Vehicle detection in intelligent transportation systems and its applications under varying environments: A review, *Image and Vision Computing*, 69, 143–154.
- [2] **Wang, Z., Zhan, J., Duan, C., Guan, X., Lu, P. and Yang, K.** (2022). A Review of Vehicle Detection Techniques for Intelligent Vehicles, *IEEE Transactions on Neural Networks and Learning Systems*, 1–21.
- [3] **Zou, Z., Chen, K., Shi, Z., Guo, Y. and Ye, J.** (2023). Object Detection in 20 Years: A Survey, *Proceedings of the IEEE*, 111(3), 257–276.
- [4] **Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X.** (2019). Object Detection With Deep Learning: A Review, *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212–3232.
- [5] **Xu, K. and Yao, A.** (2022). Accelerating video object segmentation with compressed video, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.1342–1351.
- [6] **Liu, Q., Liu, B., Wu, Y., Li, W. and Yu, N.** (2022). *Real-time Online Multi-Object Tracking in Compressed Domain*, 2204.02081.
- [7] **Wang, Z., Qin, M. and Chen, Y.K.** (2022). Learning from the cnn-based compressed domain, *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp.3582–3590.
- [8] **Babu, R.V., Tom, M. and Wadekar, P.** (2016). A survey on compressed domain video analysis techniques, *Multimedia Tools and Applications*, 75(2), 1043–1078.
- [9] **Zhai, D., Zhang, X., Li, X., Xing, X., Zhou, Y. and Ma, C.** (2023). Object detection methods on compressed domain videos: An overview, comparative analysis, and new directions, *Measurement*, 207, 112371.
- [10] **Javed, M., Nagabhushan, P. and Chaudhuri, B.B.** (2017). A review on document image analysis techniques directly in the compressed domain, *Artificial Intelligence Review*, 1–30.
- [11] **Alvar, S.R., Choi, H. and Bajic, I.V.** (2018). Can you find a face in a HEVC bitstream?, *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp.1288–1292.

- [12] **Zhao, L., He, Z., Cao, W. and Zhao, D.** (2018). Real-Time Moving Object Segmentation and Classification From HEVC Compressed Surveillance Video, *IEEE Transactions on Circuits and Systems for Video Technology*, 28(6), 1346–1357.
- [13] **Chen, L., Sun, H., Katto, J., Zeng, X. and Fan, Y.** (2021). Fast Object Detection in HEVC Intra Compressed Domain, *2021 29th European Signal Processing Conference (EUSIPCO)*, Dublin, Ireland, pp.756–760.
- [14] **Feng, J., Li, S., Li, X., Wu, F., Tian, Q., Yang, M. and Ling, H.** (2022). TapLab: A Fast Framework for Semantic Video Segmentation Tapping Into Compressed-Domain Knowledge, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 44(03), 1591–1603.
- [15] **Toreyin, B.U., Cetin, A.E., Aksay, A. and Akhan, M.B.** (2004). Moving region detection in compressed video, *International Symposium on Computer and Information Sciences*, Springer, pp.381–390.
- [16] **Choi, H. and Bajic, I.V.** (2017). HEVC intra features for human detection, *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Montreal, QC, Canada, pp.393–397.
- [17] **Torfason, R., Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R. and Gool, L.V.** (2018-01-01). *Towards Image Understanding from Deep Compression without Decoding*.
- [18] **Wang, Z., Liu, X., Feng, J., Yang, J. and Xi, H.** (2019). Compressed-Domain Highway Vehicle Counting by Spatial and Temporal Regression, *IEEE Transactions on Circuits and Systems for Video Technology*, 29(1), 263–274.
- [19] **Savcı, M.M., Yıldırım, Y., Saygılı, G. and Toreyin, B.U.** (2019). Fire detection in H. 264 compressed video, *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp.8310–8314.
- [20] **Benazza-Benyahia, A., Hamouda, N., Tlili, F. and Ouerghi, S.** (2012). Early smoke detection in forest areas from DCT based compressed video, *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, IEEE, pp.2752–2756.
- [21] **Toreyin, B.U.** (2018). Smoke detection in compressed video, *Applications of Digital Image Processing XLI*, volume10752, SPIE, pp.896–900.
- [22] **Çavaş, S., Beratoğlu, M.S. and Töreyn, B.U.** (2021). Anomaly Detection In Compressed Video, *2021 SIU*, pp.1–4.
- [23] **Bombardelli, F., Gül, S., Becker, D., Schmidt, M. and Hellge, C.** (2018). Efficient object tracking in compressed video streams with graph cuts, *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, IEEE, pp.1–6.

- [24] **Farajian, N. and Rahimi, M.** (2014). Algorithms for licenseplate detection: A survey, *2014 International Congress on Technology, Communication and Knowledge (ICTCK)*, IEEE, pp.1–8.
- [25] **Zheng, D., Zhao, Y. and Wang, J.** (2005). An efficient method of license plate location, *Pattern recognition letters*, 26(15), 2431–2438.
- [26] **Yousif, B.B., Ata, M.M., Fawzy, N. and Obaya, M.** (2020). Toward an optimized neutrosophic K-means with genetic algorithm for automatic vehicle license plate recognition (ONKM-AVLPR), *IEEE Access*, 8, 49285–49312.
- [27] **Beratoğlu, M.S.** (2003). *Araç Plaka Yerinin Saptanması*, Istanbul Technical University.
- [28] **Kahraman, F., Kurt, B. and Gökmen, M.** (2003). License plate character segmentation based on the gabor transform and vector quantization, *International Symposium on Computer and Information Sciences*, Springer, pp.381–388.
- [29] **Zahedi, M. and Salehi, S.M.** (2011). License plate recognition system based on SIFT features, *Procedia Computer Science*, 3, 998–1002.
- [30] **Pustokhina, I.V., Pustokhin, D.A., Rodrigues, J.J., Gupta, D., Khanna, A., Shankar, K., Seo, C. and Joshi, G.P.** (2020). Automatic vehicle license plate recognition using optimal K-means with convolutional neural network for intelligent transportation systems, *Ieee Access*, 8, 92907–92917.
- [31] **Bhujbal, A. and Mane, D.** (2020). A survey on deep learning approaches for vehicle and number plate detection, *Int. J. Sci. Technol. Res.*, 8, 1378–1383.
- [32] **Weihong, W. and Jiaoyang, T.** (2020). Research on license plate recognition algorithms based on deep learning in complex environment, *IEEE Access*, 8, 91661–91675.
- [33] **Kurpiel, F.D., Minetto, R. and Nassu, B.T.** (2017). Convolutional neural networks for license plate detection in images, *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, pp.3395–3399.
- [34] **Chen, R.C. et al.** (2019). Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning, *Image and Vision Computing*, 87, 47–56.
- [35] **Redmon, J., Divvala, S., Girshick, R. and Farhadi, A.** (2015). *You Only Look Once: Unified, Real-Time Object Detection*, <http://arxiv.org/abs/1506.02640>.
- [36] **Wang, W., Yang, J., Chen, M. and Wang, P.** (2019). A light CNN for end-to-end car license plates detection and recognition, *IEEE Access*, 7, 173875–173883.

- [37] **Laroca, R., Zanlorensi, L.A., Gonçalves, G.R., Todt, E., Schwartz, W.R. and Menotti, D.** (2021). An efficient and layout-independent automatic license plate recognition system based on the YOLO detector, *IET Intelligent Transport Systems*, 15(4), 483–503.
- [38] **Min, W., Li, X., Wang, Q., Zeng, Q. and Liao, Y.** (2019). New approach to vehicle license plate location based on new model YOLO-L and plate pre-identification, *IET Image Processing*, 13(7), 1041–1049.
- [39] **Tao, J., Hu, W. and Ouyang, J.** (2019). Research and implementation of license plate location based on improved Yolo algorithm, *Proceedings of the 2nd International Conference on Information Technologies and Electrical Engineering*, pp.1–7.
- [40] **Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C.** (2016). Ssd: Single shot multibox detector, *European conference on computer vision*, Springer, pp.21–37.
- [41] **Dong, Z., Wu, Y., Pei, M. and Jia, Y.** (2015). Vehicle Type Classification Using a Semisupervised Convolutional Neural Network, *IEEE Transactions on Intelligent Transportation Systems*, 16(4), 2247–2256.
- [42] **Roecker, M.N., Costa, Y.M.G., Almeida, J.L.R. and Matsushita, G.H.G.** (2018). Automatic Vehicle type Classification with Convolutional Neural Networks, *2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Maribor, Slovenia, pp.1–5.
- [43] **Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q. and Cai, B.** (2018). An Improved YOLOv2 for Vehicle Detection, *Sensors*, 18(12), 4272.
- [44] **Wu, Z., Sang, J., Zhang, Q., Xiang, H., Cai, B. and Xia, X.** (2019). Multi-Scale Vehicle Detection for Foreground-Background Class Imbalance with Improved YOLOv2, *Sensors*, 19(15), 3336.
- [45] **Taheri Tajar, A., Ramazani, A. and Mansoorizadeh, M.** (2021). A lightweight Tiny-YOLOv3 vehicle detection approach, *Journal of Real-Time Image Processing*, 18, 2389–2401.
- [46] **Wang, C.Y., Bochkovskiy, A. and Liao, H.Y.M.** (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, *arXiv preprint*, 2202.10882.
- [47] **Wiegand, T., Sullivan, G.J., Bjontegaard, G. and Luthra, A.** (2003). Overview of the H. 264/AVC video coding standard, *IEEE Transactions on circuits and systems for video technology*, 13(7), 560–576.
- [48] **Sullivan, G.J., Ohm, J.R., Han, W.J. and Wiegand, T.** (2012). Overview of the high efficiency video coding (HEVC) standard, *IEEE Transactions on circuits and systems for video technology*, 22(12), 1649–1668.

- [49] **Bross, B., Helle, P., Lakshman, H. and Ugur, K.**, (2014). Inter-Picture Prediction in HEVC., **V. Sze, M. Budagavi and G.J. Sullivan**, editors, High Efficiency Video Coding, Integrated Circuits and Systems, Springer, pp.113–140, <http://dblp.uni-trier.de/db/series/icas/SBS2014.html#BrossHLU14>.
- [50] **Lainema, J., Bossen, F., Han, W.J., Min, J. and Ugur, K.** (2012). Intra Coding of the HEVC Standard, *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12), 1792–1801.
- [51] **Lainema, J. and Ugur, K.** (2011). Angular intra prediction in High Efficiency Video Coding (HEVC), *2011 IEEE 13th International Workshop on Multimedia Signal Processing*, pp.1–5.
- [52] **Viola, P. and Jones, M.** (2001). Rapid object detection using a boosted cascade of simple features, *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, Ieee, pp.I–I.
- [53] **LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P.** (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 86(11), 2278–2324.
- [54] **Krizhevsky, A., Sutskever, I. and Hinton, G.E.** (2017). ImageNet Classification with Deep Convolutional Neural Networks, *Commun. ACM*, 60(6), 84–90, <https://doi.org/10.1145/3065386>.
- [55] **Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L.** (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions, *Journal of Big Data*, 8(1), 53, <https://doi.org/10.1186/s40537-021-00444-8>.
- [56] **Redmon, J. and Farhadi, A.** (2016). *YOLO9000: Better, Faster, Stronger*, <http://arxiv.org/abs/1612.08242>.
- [57] **Redmon, J. and Farhadi, A.** (2018). *YOLOv3: An Incremental Improvement*, <http://arxiv.org/abs/1804.02767>.
- [58] **Wang, C.Y., Bochkovskiy, A. and Liao, H.Y.M.** (2020). Scaled-YOLOv4: Scaling Cross Stage Partial Network., *CoRR*, *abs/2011.08036*, <http://dblp.uni-trier.de/db/journals/corr/corr2011.html#abs-2011-08036>.
- [59] **Wang, C.Y., Bochkovskiy, A. and Liao, H.Y.M.** (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, *arXiv preprint arXiv:2207.02696*.
- [60] **Beratoglu, M.S. and Töreyn, B.U.** (2019). Vehicle License Plate Detection Using Only Block Partitioning Structure of the High Efficiency Video Coding (HEVC)., *SIU, IEEE*, pp.1–4.

- [61] **Beratoğlu, M.S. and Töreyn, B.U.** (2021). Vehicle License Plate Detector in Compressed Domain, *IEEE Access*, 9, 95087–95096.
- [62] **Beratoğlu, M.S. and Töreyn, B.U.** (2023). Vehicle Detection and Classification without Residual Calculation: Accelerating HEVC Image Decoding with Random Perturbation Injection, *arXiv preprint arXiv:2305.08265*.
- [63] **Flynn, D., Marpe, D., Naccari, M., Nguyen, T., Rosewarne, C., Sharman, K., Sole, J. and Xu, J.** (2016). Overview of the Range Extensions for the HEVC Standard: Tools, Profiles, and Performance, *IEEE Transactions on Circuits and Systems for Video Technology*, 26(1), 4–19.
- [64] **Stankowski, J. et al.** (2015). Analysis of Compressed Data Stream Content in HEVC Video Encoder, *International Journal of Electronics and Telecommunications*, 61, 121–127.
- [65] **Katsenou, A.V., Afonso, M. and Bull, D.R.** (2022). Study of compression statistics and prediction of rate-distortion curves for video texture, *Signal Processing: Image Communication*, 101.
- [66] **Silva, S.M. and Jung, C.R.** (2018). License plate detection and recognition in unconstrained scenarios, *Proceedings of the European conference on computer vision (ECCV)*, pp.580–596.
- [67] **Beratoğlu, M.S. and Töreyn, B.U.** (2020). *CD-LP: Compressed Domain License Plate Detection Database*, <https://dx.doi.org/10.21227/2511-ym76>.
- [68] **Srebric, V.** (2003). *EnglishLP Database*, http://www.zemris.fer.hr/projects/LicensePlates/english/baza_slika.zip.
- [69] **Url-1** (2023). *Divit Technology*, <http://www.divit.com.tr>, date retrieved: 26.05.2023.
- [70] **Azam, S. and Islam, M.M.** (2016). Automatic license plate detection in hazardous condition, *Journal of Visual Communication and Image Representation*, 36, 172–186.
- [71] **Zhuang, P., Shang, Y. and Hua, B.** (2009). Statistical methods to estimate vehicle count using traffic cameras, *Multidimensional Systems and Signal Processing*, 20, 121–133.
- [72] **Url-2** (2022). *Darknet: Open Source Neural Networks in C.*, <https://pjreddie.com/darknet/>, date retrieved: 30.12.2022.
- [73] **Bossen, F. et al.** (2013). Common test conditions and software reference configurations, *JCTVC-L1100*, 12(7).
- [74] **Url-3** (2022). *JCT-VC HEVC reference software version HM 16.9*, https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.9/.

- [75] **Ren, S., He, K., Girshick, R. and Sun, J.** (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137–1149.
- [76] **Beratoğlu, M.S. and Töreyn, B.U.** (2021). Vehicle License Plate Detector in Compressed Domain, *IEEE Access*, 9, 95087–95096.
- [77] **Sohl-Dickstein, J., Weiss, E.A., Maheswaranathan, N. and Ganguli, S.** (2015). Deep Unsupervised Learning using Nonequilibrium Thermodynamics, *arXiv preprint*, 1503.03585.

CURRICULUM VITAE

Name SURNAME: Muhammet Sebul BERATOĞLU

EDUCATION:

- **B.Sc.:** 2000, Istanbul Technical University, Electric Electronic Faculty, Control and Computer Engineering
- **M.Sc.:** 2003, Istanbul Technical University, Electric Electronic Faculty, Computer Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- Research Assistant, Istanbul Technical University, Informatics Institute, 2002-.
- President, Divit Teknoloji A.Ş., Istanbul, Turkey, 2016-.
- Principal Investigator, Vehicle Dimension Measurement System Based On Image Processing, TUBITAK TEYDEB 1501, 2022-.
- Principal Investigator, Real-Time Traffic Analysis System For Highways And Intersections, TUBITAK TEYDEB 1501, 2021.
- Principal Investigator, Design And Development of a Resource-Optimized System For High-Performance License Plate Recognition In Embedded Linux Environment, TUBITAK TEYDEB 1501, 2019-2020.
- 1st Prize at the European Business Angels Investment 2015 for the project "Digital Shopping Windows" under the startup Kuax.
- 2nd Prize at İTÜ Çekirdek Big Bang 2014 for the project "Digital Shopping Windows" under the startup Kuax.
- Researcher, Automated Metaphase Scanner for Chromosome Analysis System, TUBITAK TEYDEB 1507, 2012-2014.
- Researcher, Vehicle Identification and Video Analyses TUBITAK TEYDEB 1507, 2011-2013.
- Researcher, Image Processing Based Genetic Diagnosis and Analyses System TUBITAK TEYDEB 1507, 2010-2012.
- Researcher, Image Processing Based Chromosome Analyses System, funded by Ministry of Science and Technology, 2009-2010.

- Founder, Argenit Akıllı Bilgi Teknolojileri, Istanbul, Turkey, 2009-2014.
- Researcher, Vehicle Identification System, TUBITAK TEYDEB 1507, 2008-2009.
- Founder, Divit Dijital Video ve İmge Teknolojileri Ltd. Şti., Istanbul, Turkey, 2004-2012.
- Researcher, License Plate Recognition System, TUBITAK GUMSIS, 2002-2004.
- Computer Engineer, Adam Elektronik Ltd., Istanbul, Turkey, 1998-2002.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Beratoğlu M. S.**, Töreyn B. U. (2019). Vehicle License Plate Detection Using Only Block Partitioning Structure Of The High Efficiency Video Coding (HEVC), *27th Signal Processing and Communications Applications Conference (SIU)*. *IEEE*, April 24-26, 2019 Sivas, Turkey.
- **Beratoğlu M. S.**, Töreyn B. U. (2021). Vehicle License Plate Detector in Compressed Domain, *IEEE Access*, 9, 95087-95096.
- **Beratoğlu, M. S.**, Töreyn, B. U. (2023). Vehicle Detection and Classification without Residual Calculation: Accelerating HEVC Image Decoding with Random Perturbation Injection, *arXiv preprint*, arXiv:2305.08265.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- Çavaş, Sümeyye, Muhammet Sebul Beratoğlu, and Behçet Uğur Toreyin. "Anomaly Detection In Compressed Video." In 2021 29th Signal Processing and Communications Applications Conference (SIU), pp. 1-4. IEEE, 2021.
- Dule, Erida, Muhittin Gökmen, and M. S. Beratoğlu. "A convenient feature vector construction for vehicle color recognition." Proceedings of the 11th WSEAS international conference on neural networks and 11th WSEAS international conference on evolutionary computing and 11th WSEAS international conference on Fuzzy systems. 2010.
- Asta, Shahriar, Muhammet S. Beratoğlu, and Abdulkirim Capar. "A boundary based feature extraction method for G-banded chromosome classification." 2012 20th Signal Processing and Communications Applications Conference (SIU). IEEE, 2012.
- Çapar, A., Beratoğlu, M. S., Taşdemir, K., Kılıç, Ö., & Gökmen, M. (2003). İTÜ Araç Plaka Tanıma Sistemi. IEEE 11. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, 371-374.