

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

DOKTORA TEZİ

HEVC VİDEOLARI İÇİN YÜKSEK KAPASİTELİ VE YÜKSEK
VERİMLİ VERİ GİZLEME YÖNTEMLERİNİN
GELİŞTİRİLMESİ VE UYGULANMASI

MEHMET ZEKİ KONYAR

KOCAELİ 2020

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

DOKTORA TEZİ

HEVC VİDEOLARI İÇİN YÜKSEK KAPASİTELİ VE YÜKSEK
VERİMLİ VERİ GİZLEME YÖNTEMLERİNİN
GELİŞTİRİLMESİ VE UYGULANMASI

MEHMET ZEKİ KONYAR

Dr.Öğr.Üyesi Sıtkı ÖZTÜRK
Danışman, Kocaeli Üniversitesi
Doç.Dr. Aysun TAŞYAPI ÇELEBİ
Jüri Üyesi, Kocaeli Üniversitesi
Dr.Öğr.Üyesi Orhan AKBULUT
Jüri Üyesi, Kocaeli Üniversitesi
Doç.Dr. Sezgin KAÇAR
Jüri Üyesi, Sakarya Uygulamalı Bilimler Üniv.
Doç.Dr. Ünal ÇAVUŞOĞLU
Jüri Üyesi, Sakarya Üniversitesi

Tezin Savunulduğu Tarih: 26.11.2020

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında, HEVC videolarının artıklık bilgileri içerisine video özelliklerini bozmayan yüksek kapasiteli ve yüksek verimli veri gizleme çalışması gerçekleştirilmiştir. Yapılan çalışmaların bilgi güvenliği, veri gizleme gibi alanlarda çalışan kişilere faydalı olmasını ümit ederim.

Doktora eğitimim boyunca beni sürekli motive eden, özellikle tez çalışmalarındaki problemleri aşmamda, desteklerini esirgemeyen kıymetli danışmanım, Dr. Öğr. Üyesi Sıtkı Öztürk hocama çok teşekkür ederim. Doktora Tez izleme komitemde de yer alan Dr. Öğr. Üyesi Orhan Akbulut'a başta HEVC referans yazılımını çözmemdeki destekleri olmak üzere, bütün değerli katkıları için teşekkür ederim. Doktora Tez izleme komitemde de yer alan Doç. Dr. Aysun Çelebi'ye değerli katkıları için teşekkür ederim.

Bu günlere gelmemde en büyük destekçilerim olan başta Annem ve Babam olmak üzere tüm aileme minnettarım. Yoğun doktora çalışmalarım süresince bana gösterdiği sabır ve anlayış için sevgili eşim Ayşenur'a ve tek başına oynamak zorunda kalan canım oğlum Kerem Burak'a ayrıca teşekkür ederim.

Doktora çalışmalarımı 2211/A Genel Yurt İçi Doktora Bursuyla destekleyen TÜBİTAK Bilim İnsanı Destek Programları Başkanlığı'na (BİDEB) ve 2017/125 numaralı proje ile destekleyen Kocaeli Üniversitesi Bilimsel Araştırma Programları (BAP) Koordinatörlüğüne teşekkür ederim.

Ek: Tez yazımını bitirip ve bu önsözü yazmamdan 9 gün sonra ailemize katılan kızım Betül'e sevgilerimi sunuyorum.

Temmuz – 2020

Mehmet Zeki KONYAR

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iv
TABLOLAR DİZİNİ	vi
SİMGELER VE KISALTMALAR DİZİNİ	vii
ÖZET.....	viii
ABSTRACT.....	ix
GİRİŞ	1
1. VIDEO KODLAMA.....	5
1.1. Sayısal Video Temelleri.....	5
1.2. Video Kodlama Temelleri	7
1.3. Video Kodlama Standartları	10
1.3.1. H.261.....	13
1.3.2. MPEG-1	13
1.3.3. H.262/MPEG-2	13
1.3.4. H.263.....	14
1.3.5. MPEG-4.....	14
1.3.6. H.264/AVC	14
2. H.265/HEVC.....	17
2.1. HEVC’de Blok Bölme ve Dörtlü Ağaç Yapısı.....	19
2.2. Çerçeve-içi Kestirim ve Mod Seçimi	22
2.3. Dönüşüm ve Kuantalama.....	25
2.4. Test Konfigürasyonları ve Profiller	26
2.5. HEVC Başarım Değerlendirme Ölçütleri	28
3. VERİ GİZLEME.....	30
3.1. Veri Gizleme Türleri.....	31
3.2. Görüntülerde Veri Gizleme	33
3.3. Videolarda Veri Gizleme	35
3.3.1. Ham videolarda veri gizleme.....	35
3.3.2. Kodlanmış videolarda veri gizleme	37
3.4. Steganaliz.....	38
3.5. Veri Gizleme Başarım Değerlendirmesi Yöntemleri.....	39
4. HEVC VERİ GİZLEME ÇALIŞMALARI	43
4.1. Blok Bölümleme Yapısını Değiştirme.....	43
4.2. Çerçeve-içi Kestirim Modunu Değiştirme.....	46
4.3. Hareket Vektörlerine Veri Gizleme	48
4.4. Dönüşüm Katsayılarını Kullanma	50
4.5. Entropi Kodlama Aşamasında Veri Gizleme.....	56
4.6. Doğrudan Piksel Değerlerine Veri Gizleme	58
4.7. Literatür Çalışmalarının Karşılaştırılması	59
4.8. Steganaliz Çalışmaları	62
5. TEZ KAPSAMINDA YÜKSEK KAPASİTE VE YÜKSEK VERİM İÇİN GELİŞTİRİLEN YÖNTEMLER	64

5.1. Kullanılan Veri Gizleme Yöntemi	66
5.2. Kullanılan Hata Yayılımını Önleme Yöntemi	68
5.3. Yüksek Kapasite Temelli Veri Gizleme Yöntemi	72
5.3.1. Önerilen yöntem	72
5.3.2. Deneysel çalışmalar	75
5.3.3. Sonuçlar	80
5.4. Yüksek Verimlilik Temelli Veri Gizleme Yöntemi	81
5.4.1. CTU içindeki TU bloklarının analizi	81
5.4.2. Önerilen yöntem	83
5.4.3. Deneysel çalışmalar	87
5.4.4. Sonuçlar	92
5.5. Kapasite ve Verimlilik Temelli Veri Gizleme Yöntemi	92
5.5.1. Önerilen yöntem	93
5.5.2. Deneysel çalışmalar	96
5.5.3. Sonuçlar	100
5.6. Karşılaştırmalar	101
6. SONUÇLAR VE ÖNERİLER	105
KAYNAKLAR	108
EKLER	121
KİŞİSEL YAYIN VE ESERLER	127
ÖZGEÇMİŞ	129

ŞEKİLLER DİZİNİ

Şekil 1.1.	Video formatları için uzamsal çözünürlükler	6
Şekil 1.2.	Farklı kroma alt-örnekleme biçimleri	7
Şekil 1.3.	Video kodlama blok yapısı	8
Şekil 1.4.	Video kod çözme blok yapısı.....	9
Şekil 1.5.	Örnek GOP yapısı.....	10
Şekil 1.6.	VCEG ve MPEG tarafından geliştirilen video kodlama standartları.....	11
Şekil 1.7.	Full HD bir içerik için video kodlama standartlarının başarımları	12
Şekil 1.8.	H.264/AVC Çerçeve içi kestirim modları	15
Şekil 2.1.	HEVC video kodlayıcı yapısı.	17
Şekil 2.2.	64×64 bir CTU için CU boyutları ve dörtlü ağaç yapısı.....	20
Şekil 2.3.	Çerçeve-içi (intra) ve çerçeveler-arası (inter) PU Modları	21
Şekil 2.4.	64×64 bir CTU'nun CU, PU ve TU yapısı	22
Şekil 2.5.	N×N'lik bloktaki elemanların kestiriminde kullanılan referans örnekler	23
Şekil 2.6.	HEVC'de kullanılan çerçeve-içi kestirim modları	24
Şekil 3.1.	Temel veri gizleme yaklaşımı	31
Şekil 4.1.	HEVC videolarında veri gizleme yöntemleri	43
Şekil 4.2.	Gizlenecek mesaj (w) verisine göre CU boyutları	44
Şekil 4.3.	16×16, 32×32 ve 64×64 CU boyutları için PU bölümlleme modlarının gizli bitlere göre gruplanması	45
Şekil 5.1.	Tez çalışmasında kullanılan test videoların ilk çerçeveleri	65
Şekil 5.2.	ME'de gizleme oranı ve gizleme verimliliği arasındaki ilişki.....	67
Şekil 5.3.	Kestirim için mevcut bloğun referans değerlerini kullanan komşu PU bloklar	69
Şekil 5.4.	Önerilen yöntemin akış şeması	73
Şekil 5.5.	Önerilen yöntemin oran-bozunum başarımları	76
Şekil 5.6.	Önerilen yöntemin öznel değerlendirmesi. Orijinal kodlananlar (solda), önerilen veri gizleme ile kodlananlar (sağda). (a)-(b): Traffic QP=28, (c)-(d): Cactus QP=30, (e)-(f): Johnny QP=32, (g)-(h): RaceHorsesC QP=34 ve (i)-(j): Keiba QP=36	79
Şekil 5.7.	Örnek video dizilerinin 1. çerçevelerindeki CTU'larda bulunan 4x4 TU sayıları.....	81
Şekil 5.8.	Video dizilerinin tüm çerçevelerindeki CTU'larda bulunan ortalama 4x4 TU sayıları.....	82
Şekil 5.9.	Farklı QP'ler ile kodlanan BasketballPass dizisinin çerçevelerinde bulunan CTU'lardaki ortalama 4x4 TU sayıları	83
Şekil 5.10.	Önerilen yöntemin akış şeması	84
Şekil 5.11.	Önerilen yöntemin oran-bozunum başarımları	88
Şekil 5.12.	Önerilen yöntemin öznel değerlendirmesi. Orijinal kodlananlar (solda), önerilen veri gizleme ile kodlananlar (sağda). (a)-(b): PeopleOnStreet QP=28, (c)-(d): BQTerrace QP=30, (e)-(f): FourPeople QP=32, (g)-(h): BasketballDrill QP=34 ve (i)-(j): BasketballPass QP=36	91

Şekil 5.13. Önerilen yöntemin akış şeması	94
Şekil 5.14. Önerilen yöntemin oran-bozunum başarımı	96
Şekil 5.15. Önerilen yöntemin öznel değerlendirmesi. Orijinal kodlananlar (solda), önerilen veri gizleme ile kodlananlar (sağda). (a)-(b): BasketballDrill QP=28, (c)-(d): BQTerrace QP=30, (e)-(f): Cactus QP=32, (g)-(h): RaceHorsesC QP=34 ve (i)-(j): Traffic QP=36	99
Şekil 5.16. Tez çalışmasındaki tüm yöntemlerin oran-bozunum karşılaştırmaları.....	101
Şekil 5.17. Tez çalışmasındaki tüm yöntemlerin kapasite karşılaştırması.....	103



TABLÖLAR DİZİNİ

Tablo 4.1.	Literatürdeki HEVC veri gizleme yöntemlerinin karşılaştırılması.....	60
Tablo 5.1.	Tez çalışmasında kullanılan test videoları ve özellikleri.....	64
Tablo 5.2.	ME(1,3,2) için değişecek bloğun seçimi	75
Tablo 5.3.	BasketballPass test dizisinin farklı QP değerleri için veri gizleme başarımı	77
Tablo 5.4.	Önerilen yöntemin çeşitli test dizilerinde QP=32 için kapasite (bit) başarımı	78
Tablo 5.5.	Önerilen yöntemin çeşitli test dizilerinde QP=32 için Δ PSNR (dB) başarımı.....	78
Tablo 5.6.	Önerilen yöntemin çeşitli test dizilerinde QP=32 için BIR($\times 10^{-3}$) başarımı	80
Tablo 5.7.	ME(1,15,4) için değişecek bloğun seçimi	86
Tablo 5.8.	BasketballPass test dizisinin farklı QP değerleri için veri gizleme başarımı	87
Tablo 5.9.	Önerilen yöntemin çeşitli test dizilerinde QP=32 için BIR($\times 10^{-3}$) başarımı	89
Tablo 5.10.	Önerilen yöntemin çeşitli test dizilerinde QP=32 için kapasite (bit) başarımı	90
Tablo 5.11.	Önerilen yöntemin çeşitli test dizilerinde QP=32 için Δ PSNR (dB) başarımı.....	92
Tablo 5.12.	ME(1,7,3) için değişecek bloğun seçimi	95
Tablo 5.13.	BasketballPass test dizisinin farklı QP değerleri için veri gizleme başarımı	98
Tablo 5.14.	Önerilen yöntemin çeşitli test dizilerinde QP=32 için kapasite (bit) başarımı	98
Tablo 5.15.	Önerilen yöntemin çeşitli test dizilerinde QP=32 için Δ PSNR (dB) başarımı.....	98
Tablo 5.16.	Önerilen yöntemin çeşitli test dizilerinde QP=32 için BIR($\times 10^{-3}$) başarımı	100

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

AVC	: Advance Video Coding (İleri Video Kodlama)
BIR	: Bitrate Increase Ratio (Bit oranının artış oranı)
BPP	: Bit Per Pixel (Piksel Başına Düşen Bit Miktarı)
CABAC	: Context Adaptive Binary Arithmetic Coding (İçerik Uyarlamalı İkili Aritmetik Kodlama)
CAVLC	: Context Adaptive Variable Length Coding (İçerik Uyarlamalı Değişken Uzunluklu Kodlama)
CIF	: Common Intermediate Format (Ortak Ara format)
CTU	: Coding Tree Unit (Kodlama Ağaç Birimi)
CU	: Coding Unit (Kodlama Birimi)
DCT	: Discrete Cosine Transform (Ayrık Kosinüs Dönüşümü)
DE	: Difference Expansion (Fark Genişlemesi)
DST	: Discrete Sine Transform (Ayrık Sinüs Dönüşümü)
DWT	: Discrete Wavelet Transform (Ayrık Dalgacık Dönüşümü)
EE	: Embedding Efficiency (Gizleme Verimliliği)
EMD	: Exploiting modification direction (Değişim Yönü Kullanımı)
ER	: Embedding Rate (Gizleme Oranı)
FPS	: Frame per Second (Saniyedeki Çerçeve Sayısı)
GOP	: Group of Picture (Resim Grubu)
HD	: High Definition (Yüksek Çözünürlük)
HEVC	: High Efficiency Video Coding (Yüksek Verimli Video Kodlama)
HM	: HEVC Test Model
IPM	: Intra Prediction Mode (Çerçeve-içi Kestirim Modu)
JVET	: Joint Video Experts Team (Ortak Video Uzmanları Takımı)
LSB	: Least Significant Bit (En Düşük Değerlikli Bit)
MPEG	: Moving Picture Experts Group (Hareketli Resim Uzmanları Grubu)
MSE	: Mean Square Error (Ortalama Karesel Hata)
PSNR	: Peak Signal-to-Noise Ratio (Tepe Sinyal Gürültü Oranı)
PU	: Prediction Unit (Kestirim Birimi)
PVD	: Pixel value differencing (Piksel Değer Farkı)
QP	: Quantization Parameter (Kuantalama Parametresi)
QTC	: Quantized Transform Coefficient (Kuantalanmış Dönüşüm Katsayısı)
RGB	: Red Green Blue (Kırmızı Yeşil Mavi)
RQT	: Residual Quad Tree (Artıklık Dörtlü Ağacı)
SD	: Standard Definition (Standart Çözünürlük)
SSIM	: Structural Similarity Index (Yapısal Benzerlik İndeksi)
TU	: Transform Unit (Dönüşüm Birimi)
VCEG	: Video Coding Experts Group (Video Kodlama Uzmanları Grubu)
VVC	: Versatile Video Coding (Çok-yönlü Video Kodlama)

HEVC VİDEOLARI İÇİN YÜKSEK KAPASİTELİ VE YÜKSEK VERİMLİ VERİ GİZLEME YÖNTEMLERİNİN GELİŞTİRİLMESİ VE UYGULANMASI

ÖZET

Bilgi güvenliğinin bir parçası olan veri gizleme, sahiplik bilgisi kontrolü, içerik koruması, telif hakları ihlalinin tespiti, görünmez ve güvenli haberleşme, gibi çeşitli ihtiyaçlar için kullanılan önemli bir bilim dalıdır. Günümüzde, video kullanımının yaygınlaşmasıyla, en verimli video kodlama yöntemi olan HEVC videolarının veri gizlemede kullanımını arttırmıştır. HEVC videolarında farklı aşamalarda veri gizlenmekte olup, en sık kullanılan yerler çerçeve-içi kuantalanmış dönüşüm katsayıların bulunduğu bloklardır. Veri gizleme, taşıyıcı nesneye gizli mesaj olarak bilinen bir işaretinin eklenmesi işlemidir. Dolayısıyla, HEVC veri gizleme yöntemlerinde en önemli problemler, gizleme kapasitelerinin düşük kalması ve çerçeve-içi blokları kullanan yöntemlerde veri gizlenen bloğun komşu bloklara hata yaymasıdır.

Bu tez çalışmasında HEVC video çerçevelerindeki 4×4 artıklık bloklarına farklı yöntemlerle veri gizlenmektedir. Yapılan çalışmaların odak noktaları, hata yayılımını önleyen yöntemlerin basitleştirilmesi, veri gizleme kapasitesinin artırılması ve gizleme verimliliğinin artırılmasıdır. Gerçekleştirilen matris kodlama tabanlı veri gizleme yöntemleri, hem yüksek kapasite hem de yüksek verimlilik ihtiyaçlarına çözüm sunmaktadır. İyileştirilen hata yayılımını önleme yaklaşımında ise, veri gizleme öncesi videonun kodlanma ve analiz edilme ihtiyaçları ortadan kaldırılmaktadır. Ayrıca gerçekleştirilen bu yöntem sayesinde, veri gizleme aşamasında komşu blokların kontrol edilmesine de gerek kalmamaktadır.

DeneySEL çalışmalar, bu tez çalışmasında gerçekleştirilen yöntemlerin videolardaki görsel kalitenin korunması, yüksek kapasite sunması ve gizlenen veri miktarına göre bit artışını daha düşük tutması gibi özellikleriyle literatürdeki benzer çalışmalara üstünlük sağladığını göstermektedir.

Anahtar Kelimeler: Bilgi Güvenliği, Güvenli Haberleşme, HEVC, Veri Gizleme, Video Kodlama

DEVELOPMENT AND IMPLEMENTATION OF HIGH-CAPACITY AND HIGH-EFFICIENCY DATA HIDING METHODS FOR HEVC VIDEOS

ABSTRACT

Data hiding that is a part of information security is an important science used for various needs such as ownership information control, content protection, copyright infringement detection, invisible and secure communication. Nowadays, with the widespread use of video, the usage of HEVC videos which is the most efficient video coding method, has increased for data hiding. Secret data are hidden at different stages in HEVC videos, and the most frequently used locations are the quantized transform coefficients of the intra frames. Data hiding actually means adding an error signal that known as secret message. Therefore, the most important problems in HEVC data hiding methods are that their hiding capacities remain low and the data hiding block propagate errors to neighboring blocks in methods using intra-frame blocks.

In this thesis, it is proposed to hide secret data in 4×4 residual blocks of HEVC video frames. The focal points of the proposed methods are to simplify the methods that prevent error propagation, increase the data hiding capacity and increase hiding efficiency. The proposed matrix encoding-based data hiding methods provide solutions for both high capacity and high efficiency needs. The improved error propagation prevention approach eliminates the need to encode and analyze video before hiding data. In addition, thanks to this proposed method, it is not necessary to control neighboring blocks during the data hiding phase.

Experimental studies show that the proposed methods in this thesis provide superiority to similar studies in the literature, such as preserving the visual quality of the videos, offering high capacity and keeping the bit increase lower for the amount of hidden data.

Keywords: Information Security, Secure Communication, HEVC, Data Hiding, Video Coding.

GİRİŞ

Veri gizleme, bir taşıyıcı nesne içerisine çeşitli bilgilerin gizlenmesinden sonra, elde edilen yeni medya nesnesinin haberleşme kanalından alıcıya gönderilmesi ve alıcıda gizlenen bilgilerin çıkarılmasıyla ilgilenmektedir. Gizlenen verinin amacı gizli haberleşme olabileceği gibi, içerik koruması veya kimlik doğrulaması da olabilmektedir. Varsayılan senaryo türüne göre, yeterince kapasite sunmak, güvenli olmak ve üçüncü kişilerin saldırılarına karşı dayanıklı olmak veri gizlemenin en önemli gereksinimleri olmaktadır. Bu gereksinimler tüm yöntemler için önemli olmakla beraber, kapasite ve güvenlik gizli haberleşme için öne çıkarken, saldırılara karşı dayanıklı olmak içerik koruma ve kimlik doğrulama yöntemleri için öne çıkmaktadır [1].

Veri gizleme, steganografi ve damgalama olmak üzere ikiye ayrılmaktadır. Steganografi güvenli gizli haberleşme için, damgalama ise içerik koruması ve kimlik doğrulaması gibi bilgi güvenliği uygulamaları için kullanılmaktadır. Genellikle resmi birimler veya gayri resmi örgütler tarafından haberleşme amaçlı kullanılan steganografide, gizli bilgiyi içeren taşıyıcı medyadaki değişimler, haberleşme kanalındaki üçüncü kişiler tarafından fark edilmeyecek düzeyde olmalıdır. Aksi durumda şüpheli bulunan taşıyıcıya çeşitli müdahaleler yapılarak haberleşme kesilebilmektedir. Daha çok ticari kuruluşlar tarafından telif hakkı ihlali tespiti, kimlik doğrulaması ve içerik koruma amaçlarıyla kullanılan damgalamada ise biraz daha farklı olarak, korunmak isteyen medya içerisine hak sahibine ait, gözle görülür ve görülmez biçimde bir damga eklenmektedir. Eklenen damganın saldırılar karşısında dayanıklı olması ve taşıyıcı içerisinden geri elde dileyebilmesi damgalamanın en önemli önceliklerinden birisidir [2].

İlk zamanlar metinlerin içerisine veri gizlenirken, gelişen teknolojiyle birlikte taşıyıcı medya nesnesi olarak ses, görüntü ve video gibi ortamlar kullanılmaya başlanmıştır. Gerçek zamanlı görüntülü sohbet uygulamaları, bireysel içerik üreticileri, uzaktan ev ve işyeri gözetim sistemleri ve profesyonel yayıncılık uygulamaları sürekli olarak

video üretmektedirler. Gelişen teknolojiyle birlikte video içerik üretimi oldukça yaygınlaşmasıyla birlikte veri gizleme yöntemlerinde videoların kullanılması da artmaya başlamıştır. Videolar görüntülere göre daha yüksek veri gizleme kapasiteleri sunmaktadır. Ayrıca filmler ve diziler, televizyon ve internet yayıncılığı gibi video kullanılan yerlerin içeriklerini korumak istemeleri videoları popüler veri gizleme ortamları haline getirmiştir. Veri gizlemek için doğrudan sıkıştırılmamış video çerçevelerini kullanan çalışmalar bulunmaktadır. Sıkıştırılmamış video sinyalleri oldukça büyük veri miktarına sahip olduklarından dolayı iletişim kanalından alıcıya gönderilmeleri zorludur. Bu sebeple kodlanmış videolara veri gizleme daha ön plana çıkmıştır. Mevcut uygulamaların kodlanmış videolar kullanmasından dolayı veri gizleme için genellikle video sıkıştırma formatlarının süreçleri kullanılmaktadır. Ham videolardan farklı olarak veri gizleme işlemi video kodlama sürecinin bir parçası olarak yapılmaktadır.

Bir yandan kodlanmış videolara veri gizleme yapan yöntemler artarken, öte yandan video kodlama yöntemleri de sürekli olarak gelişmektedir. Bunun en önemli sebebi kullanıcılarda daha yüksek çözünürlük, daha yüksek kare hızları ve görsel kalitenin aslına uygunluğuna yönelik ciddi bir talebin bulunmasıdır. Kullanıcıların video içeriklerine sürekli erişme isteği iletişim ağları ve veri depolama üniteleri üzerindeki en büyük yüküdür [3]. Video kodlama standartlarının en önemli hedefi kodlama verimliliğini arttırmaktır. Kodlama verimliliği, video kalitesi belirli bir seviyede tutulduğunda, ihtiyaç duyulan bit oranını minimize etme yeteneğidir. Bir başka deyişle, belirli bit hızı, örneğin sabit internet bağlantı hızı, için video kalitesini en üst düzeye çıkarmaktır [4]. HEVC (High Efficiency Video Coding), bu ihtiyaçları karşılama anlamında önerilen en gelişmiş video kodlama yöntemidir. Önceki video kodlama standardına göre %50'ye yakın bit oranı tasarrufu sağlayan HEVC ile kodlanmış videoların popülerliğinin önümüzdeki yıllarda devam edeceği öngörülmektedir [5]. Bu sebeple, HEVC videolarıyla yapılacak veri gizleme çalışmaları oldukça değerli olacağı fikri bu tez çalışmasının ana motivasyonunu oluşturmaktadır.

HEVC'de videonun farklı kısımlarına veri gizlenebilmektedir. Bu tez çalışmasında HEVC videolarına veri gizlerken, diğer kısımlara göre sağladığı yüksek kapasite sebebiyle çerçeve-içi artıklık işaretinin kuantalanmış dönüşüm katsayıları

kullanılmaktadır. Veri gizleme mantığında taşıyıcıya eklenen gizli bilgiler hata işareti gibidir. Dolayısıyla dönüşüm katsayılarına eklenen bilgiler piksel değerlerinde hataya/değişime sebep olmaktadır. HEVC kodlayıcı yapısı sebebiyle çerçeve-içi her blokta piksel değerleri komşu bloklar kullanılarak kestirilmektedir. Bundan dolayı gizlenen verilerin komşu bloklar üzerinde hata yayılımına sebep olmaması önemlidir. Bu bilgiler ışığında bu tez çalışmasındaki yenilikler aşağıdaki gibidir:

Literatürdeki hata yayılımını önleyen veri gizleme yöntemlerinde kullanılan komşu blokların çerçeve-içi kestirim modlarına bağımlılık ortadan kaldırılmaktadır. Böylece her blok kendi içerisinde değerlendirilerek veri gizleme işlemi yapılmaktadır.

Gizleme aşamasında komşu blokların kontrol edilme işleminin ortadan kaldırılmasıyla, kodlama işlemlerindeki karmaşıklık giderilmektedir. Ayrıca komşu blok modlarının tespit edilmesi için videonun birden fazla kodlanma ihtiyacı ortadan kaldırılmaktadır.

Sadece bir katsayının değiştirilmesiyle birden fazla bitin gizlenmesini sağlayan matris kodlama yöntemi HEVC artıklık bloklarına uyarlanmaktadır.

Yüksek kapasite temelli, yüksek verimlilik temelli ve yüksek kapasite ile yüksek verimliliğin bir arada olduğu farklı veri gizleme yöntemleri geliştirilmektedir.

Veri gizleme için video çerçevelerindeki en küçük artıklık bloklarının kullanılmasıyla, veri gizleme sonrası kodlanan videoların, orijinal kodlanmış videolara görsel olarak benzerliği korunmaktadır.

Bu tez çalışmasının bundan sonraki bölümleri aşağıdaki biçimdedir:

Birinci bölümde, sayısal videonun temel kavramları ve video kodlama temelleri anlatıldıktan sonra geçmişten günümüze en çok kullanılan video kodlama standartları hakkında genel bilgiler verilmektedir.

İkinci bölümde, en güncel video kodlama yöntemi olan HEVC'nin öne çıkan özellikleri kısaca anlatıldıktan sonra, özellikle tez çalışmasında temel teşkil eden blok bölümlene yapısı, çerçeve-içi kestirim, artıklık işaretinin dönüşümü ve kuantalama süreçleri anlatılmaktadır. HEVC için geliştirilen referans yazılımının test

konfigürasyonları ve videoların başarımlarını değerlendirme yöntemleri hakkında genel bilgiler verilmektedir.

Üçüncü bölümde, veri gizleme, steganografi ve damgalama kavramları üzerinde durulmaktadır. Metin, ses ve internet protokol paketlerine yapılan veri gizlemeler literatür çalışmaları üzerinden örneklendirilirken, görüntü veri gizleme çalışmaları iki örnek yöntem ve tıbbi görüntülerle yapılan çalışmalar üzerinden sunulmaktadır. Videolarda veri gizleme yöntemleri kapsamlı olarak ele alındıktan sonra, gizli verilerin tespitini hedef alan steganaliz hakkında bilgiler verilip, veri gizleme yöntemlerinin başarımlarını değerlendirmesinde kullanılan yöntemler sunulmaktadır.

Dördüncü bölümde, tez çalışmasının yazıldığı tarih itibarıyla erişilen HEVC veri gizleme çalışmaları, gizlemenin yapıldığı aşamalara göre sınıflandırılarak verilmektedir. Veri gizleme çalışmalarının kapsamlı karşılaştırılması yapıldıktan sonra mevcut olan HEVC steganaliz çalışmalarına değinilmektedir.

Beşinci bölümde, tez kapsamında geliştirilen veri gizleme yöntemlerine ait bilgiler, deneysel sonuçlar ve karşılaştırmalı başarımların analizleri verilmektedir.

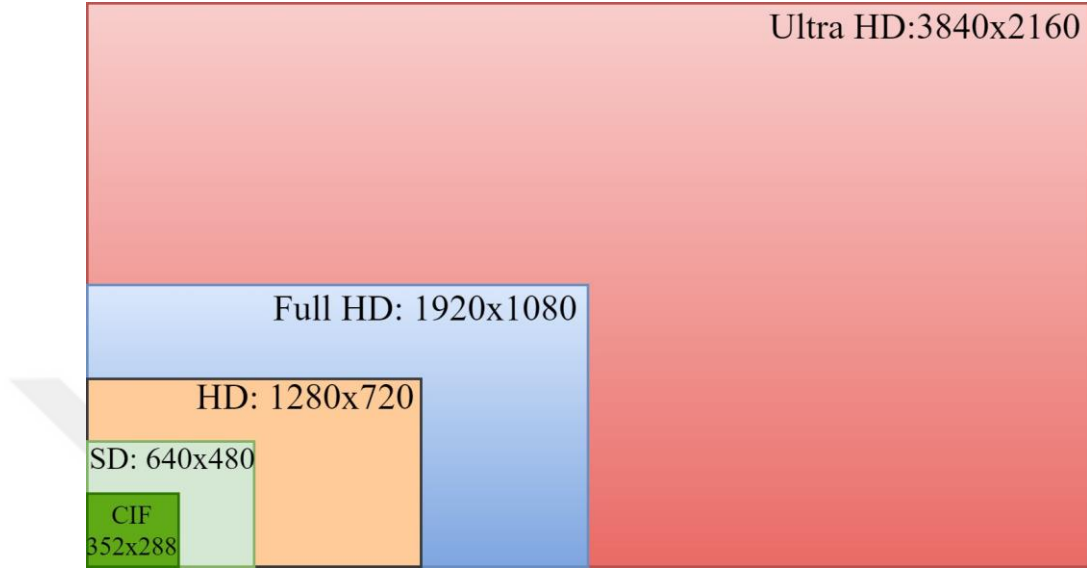
Tez kapsamında elde edilen bulgular, sonuçlar ve öneriler bölümünde verilmektedir.

1. VIDEO KODLAMA

1.1. Sayısal Video Temelleri

Video, birbiri arkasından gelen resim çerçevelerinden meydana gelmektedir. İnsan görsel sistemi tarafından kusursuz biçimde hareketli olarak algılanmaları ardışık olarak gelen çerçevelerin (frame) zamansal hızından kaynaklanmaktadır. Videonun hızı bir saniyede oynatılan resim sayısı ile ifade edilmektedir ve kare hızı veya saniyedeki çerçeve sayısı (frame per second, fps) ile gösterilmektedir. Videonun akıcılığı doğrudan videonun zamansal hızı olan fps kavramıyla ilişkilidir. Videolarda, yüksek fps akıcılığa, düşük fps durağanlığa karşılık gelmektedir. Videodaki her bir bağımsız resim çerçevesi kırmızı, yeşil ve mavi (Red, Green, Blue, RGB) bileşenlerden meydana gelmektedir ve her bir bileşen 8 bit olmak üzere toplam 24 bit ile temsil edilmektedir. Uzamsal çözünürlük olarak da adlandırılan çerçeve boyutları ise video çerçevesinin yatay ve düşeydeki piksel sayılarını ifade etmektedir [6]. Sinema, televizyon ve video paylaşım uygulamalarının her birinin kullandıkları videoların uzamsal çözünürlükleri farklı olabilmektedir. Şekil 1.1’de ortak ara format (common intermediate format, CIF), standart çözünürlüklü (standard definition, SD) ve yüksek çözünürlüklü (high definition, HD) video çerçeve örnekleri için piksel boyutları görselleştirilmiştir. Uzamsal çözünürlük video kalitesiyle doğrudan ilişkilidir. Çözünürlük ne kadar yüksek olursa, videodaki piksel sayısı o kadar fazla olur ve videodaki ayrıntıların algılanması o kadar yüksek olmaktadır. Öte yandan yüksek çözünürlüklü video için yüksek depolama kapasitesi gerekmektedir. Bu ise teknik olarak başka problemleri ortaya çıkarmaktadır. Örneğin cep telefonu ile 24 bitlik bir RGB videonun Full HD (1920×1080) çözünürlük ve 24 fps ile ham olarak kaydedileceğini varsayalım. Bu durumda 1 saniyelik videonun kapasitesi 149,3 MB, 60 dakikalık bir videonun kapasitesi yaklaşık 530 GB olacaktır. Bu videoyu cep telefonu veya kişisel depolama birimlerinde saklamak oldukça maliyetli olacaktır. Böyle bir boyuttaki ham videoyu internet iletim hattından paylaşmak için uzun süreler gerekmektedir. Çünkü bu videonun bit hızı 1194 Mbit/s olacaktır ki böyle bir hızı sağlayacak bir yaygın iletişim kanalı bulunmamaktadır. Videoların hem depolama

biriminde hem de iletişim hattında kullanışlı olması için kapasitelerinin ve bit hızının sıkıştırma işlemleriyle düşürülmesi gerekmektedir. Bu işlemlerin ilk adımlarından birisi ham videonun alt-örnekleme işlemidir.

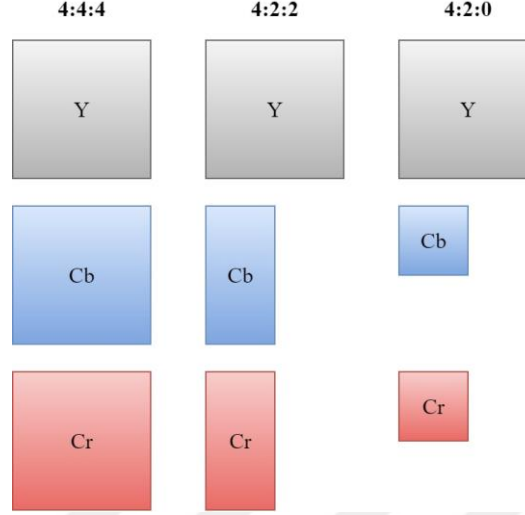


Şekil 1.1. Video formatları için uzamsal çözünürlükler

İnsan görsel sisteminin parlaklık değerlerine daha duyarlı olması sebebiyle sayısal video sistemleri çoğunlukla RGB renk uzayı yerine YCbCr (YUV) renk uzayında alt-örnekleme kullanmaktadırlar. YCbCr sayısal videolar için YUV ise analog videolar için kullanılan renk uzayı ifadeleri olmakla birlikte, kolaylık için birbirinin yerine kullanılabilir [7]. Herhangi bir video çerçevesi için Y parlaklık (luma) bileşenini ifade etmektedir. Cb ve Cr ise sırasıyla mavi ve kırmızı renk farkı (kroma) bileşenlerini ifade etmektedirler. RGB renk uzayında bütün bileşenler eşit öneme sahipken, YCbCr uzayında renk farkı bileşenleri daha düşük bilgi miktarıyla gösterilebilmektedir. İnsan görsel sisteminin yapısından dolayı kroma alt-örneklemesi yapıldığında renk farkı bilgileri azaltılmasına rağmen, görsel kalitede önemli bir kayıp olmamaktadır. Genellikle RGB renk uzayında elde edilen videolar, YCbCr uzayına dönüşüm işlemi yapıldıktan sonra kodlanmaktadır. Kodu çözölen video tekrar RGB uzayına dönüştürülerek gösterilmektedir. Şekil 1.2’de en çok kullanılan kroma alt-örnekleme biçimleri görselleştirilmiştir.

Tüm piksel noktalarında tam çözünürlüğe sahip parlaklık ve renk farkı bileşenleri 4:4:4 örnekleme oranıyla gösterilmektedir. Renk farkı bileşenlerinin yatayda yarı oranda azaltıldığı 4:2:2 örnekleme stüdyolarda ve profesyonel kamera çıkışlarında

elde edilen videolarda kullanılmaktadır. 4:2:0 örnekleme yaygın iletimi, internet akışı ve tüketici cihazlarında en çok kullanılan formattır.



Şekil 1.2. Farklı kroma alt-örnekleme biçimleri

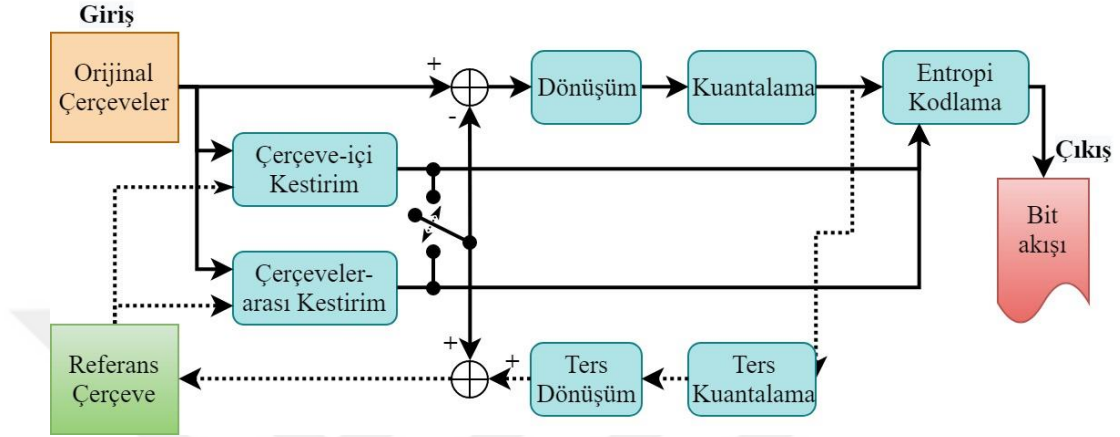
4:2:0 örnekleme hem yatayda hem dikeyde renk farkı bileşenlerini yarı yarıya alt-örnekleyerek algısal renk fazlalığı en üst düzeye azaltmaktadır. Kroma alt-örnekleme en önemli avantajı, daha kodlama başlamadan bit hızını %50'ye varan miktarda azaltmasıdır [8].

1.2. Video Kodlama Temelleri

Son dönemde görüntüleme sistemlerinin kalitesinin artması internet teknolojisinin hızlı gelişmesi ve internette kullanıcı sayısının sürekli artması resim ve video gibi medyaların insanlar arasında paylaşılma miktarını da arttırmıştır. Yüksek boyutlara sahip videolar için sistemsel altyapıların ve bağlantı imkânlarının yeterli olmamasından dolayı çeşitli sıkıştırma teknikleri üzerine çalışılmış ve birçok standart geliştirilmiştir. Video sıkıştırmada en yüksek sıkıştırma oranı yakalamanın yanında, veriyi en düşük miktarda bit ile temsil etmekte amaçlanır. Sıkıştırma miktarı arttıkça videodaki kayıplar da daha fazla olurken alıcısındaki videonun kalitesi ise daha düşük olmaktadır.

Video kodlama süreçleri genel olarak benzer mantıktaki hibrit sıkıştırma yaklaşımlarını kullanmaktadır. Hibrit video sıkıştırma süreci genellikle çerçeve-içi ve çerçeveler arası kestirimler, dönüşüm, nicemleme ve entropi kodlamadan

oluşmaktadır. Kestirim süreçlerinden sonra, kestirim artıklarının dönüşümü ve kuantalanması yapılmaktadır. En son aşamada ise entropi kodlama işlemi yapılmaktadır [6]. Şekil 1.3'te video sıkıştırma sürecinin kodlama aşaması verilmektedir.



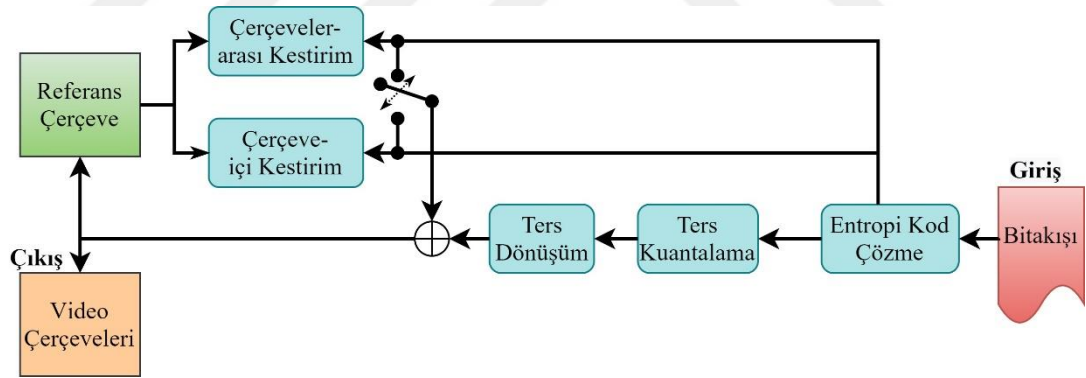
Şekil 1.3. Video kodlama blok yapısı

Video çerçeveleri, değişen özellikler haricinde, birbirine oldukça fazla benzeyen bilgiler barındırmaktadır. Video sıkıştırmada amaç bu artıklıkları (redundancy) mümkün olan en iyi biçimde azaltmaktır. Daha önce kroma alt-örnekleme ile algısal artıklık bilgileri azaltılan videoda, kodlama ile uzamsal, zamansal ve istatistiksel artıklık bilgileri de azaltılmaktadır. Video çerçevesinde komşu piksel değerleri arasındaki benzerlikler bulunmaktadır. Bir çerçevenin içindeki piksel benzerliklerinden yararlanabilmek için bu çerçeve küçük bloklara bölünmektedir. Video kodlama sürecinde çerçeve-içi kestirim işlemiyle uzamsal artıklık bilgileri azaltılmaktadır. Zamansal artıklık videodaki çerçeveler arasındaki kestirim ile ilgilidir. Videonun fps değeri arttığı zaman ardışık çerçevelerin benzerlik oranları oldukça artmaktadır. Video sıkıştırma işleminde ardışık çerçevelerdeki herhangi bir hareketin olmadığı bölgelerin referans bir çerçeve ile ifade edilmesi yeterli olmaktadır. Videoda zamanla değişen yerlerin bulunması için çerçeveler arası hareket kestirim işlemi yapılmaktadır. Bölünmüş blok temelli yapılan hareket kestiriminde komşu çerçevedeki bloklar içerisinde mevcut blokla en çok örtüşen blok bulunmaktadır. İki blok arasındaki değişim bir hareket vektörü ile ifade edilmektedir [7].

Bir video kodlama sisteminde, çerçeve-içi kestirim veya çerçeveler-arası kestirim yöntemleriyle elde edilen tahmini blokların, orijinal bloklarla aynı olması

istenmektedir. Ancak, pratik bir video kodlama sisteminde tahmini blok ile orijinal blok arasında artıklık (residue) olarak isimlendirilen farklılıklar oluşmaktadır. Orijinal blok ile tahmin bloğu arasındaki mutlak farklar ayrıca dönüşüm ve kuantalama işlemine tabi tutulmaktadır. Alıcıdaki tahmin farklarının kodlayıcıda aynı resmi oluşturacağından emin olmak için kuantalanmış sinyal ters işlemlere tabi tutularak yeniden oluşturulmuş (reconstructed) referans çerçeveler elde edilmektedir. Video kodlama standartları alıcıdaki videonun orijinal videoya en yüksek benzerlikte olması için tahmin işlemlerini farklı blok yapıları için tekrarlı olarak yapmaktadırlar [6]. Videodaki uzamsal ve zamansal artıklıkların ortaya çıkartılması video kodlama süresinin önemli bir kısmını oluşturmaktadır.

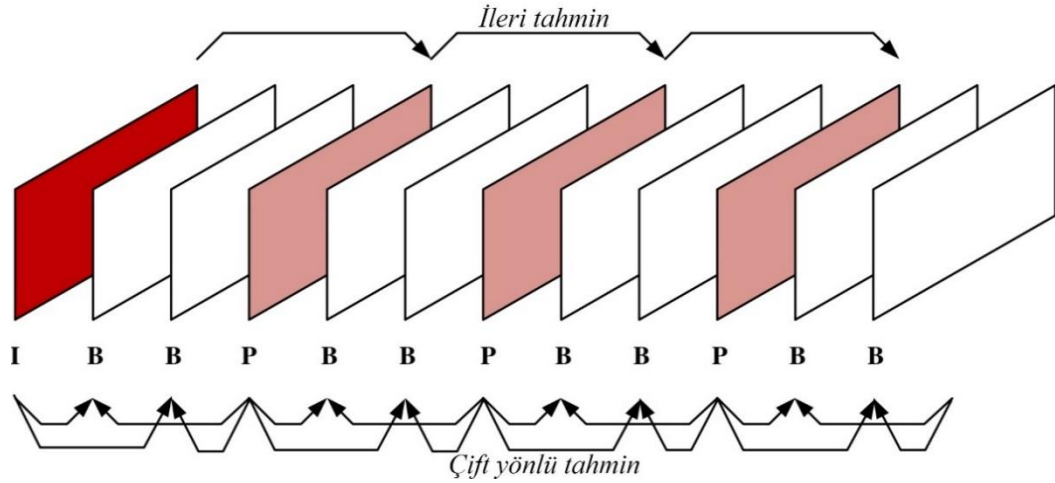
Video işaretlerindeki istatistiksel fazlalıklar tekrarlı bitlerden kaynaklanmaktadır. Tekrar eden bitler iletim hattına verilmeden önce entropi kodlama aşamasında aritmetik kodlama ve uzunluk kodlaması benzeri yöntemlerle azaltılmaktadır. En bilinen bit azaltma yöntemlerinde birisi olan uzunluk kodlamasında (length coding) tekrar eden bit değerleri yerine tekrar eden 0 ve 1 bitlerinin sayıları kodlanmaktadır.



Şekil 1.4. Video kod çözme blok yapısı

Sıkıştırılmış video bilgilerinin alıcıda tekrar çerçevelere dönüştürülme işlemi Şekil 1.4'te gösterildiği gibi kodlama işleminin tersi biçiminde olmaktadır. Bit akışı bilgileri ayrıştırılarak entropi kod çözme işlemi yapılmaktadır. Çerçeve-içi ve çerçeveler arası bilgiler ve ters dönüşümle elde edilen tahmin farkları kullanılarak video çerçeveleri elde edilmektedir. Yeniden oluşturulmuş çerçeveler ise kestirim işlemlerinde referans olarak kullanılmak üzere tutulmaktadır. Bütün video çerçeveleri elde edilene kadar aynı işlemler devam etmektedir [6]. Birçok video kodlama formatında I, P ve B olmak üzere üç farklı çerçeve türü kullanılmaktadır.

Çerçeve-içi (Intra, I) kodlanmış resimler, kodlanan başka çerçeveler tarafından referans olarak kullanılmaktadır. I-çerçeveler kendi içinde bağımsız birer resim gibi kodlandıkları için zamansal hata yayılımından etkilenmemektedirler. Bununla birlikte I-çerçeveler diğer çerçevelere kıyasla daha düşük sıkıştırma oranlarına sahiptirler. Çerçeveler-arası (Inter) resimler, tahmin (predicted, P) çerçevesi ve çift-yönlü tahmin (bidirectional predicted, B) çerçevesi olmak üzere iki türdedir. P-çerçevesi de I-çerçevesi gibi diğer çerçevelere referans olabilmektedir. P-çerçevesi kendisinden önceki referans çerçeveler kullanılarak hareket vektörleri ve tahmin artıkları ile oluşturulmaktadır. B-çerçevesi mevcut çerçeve ile I, P veya B çerçeveleri arasındaki fark bilgisini içermektedir. B-çerçevesi de P-çerçevesine benzer biçimde oluşturulmaktadır.



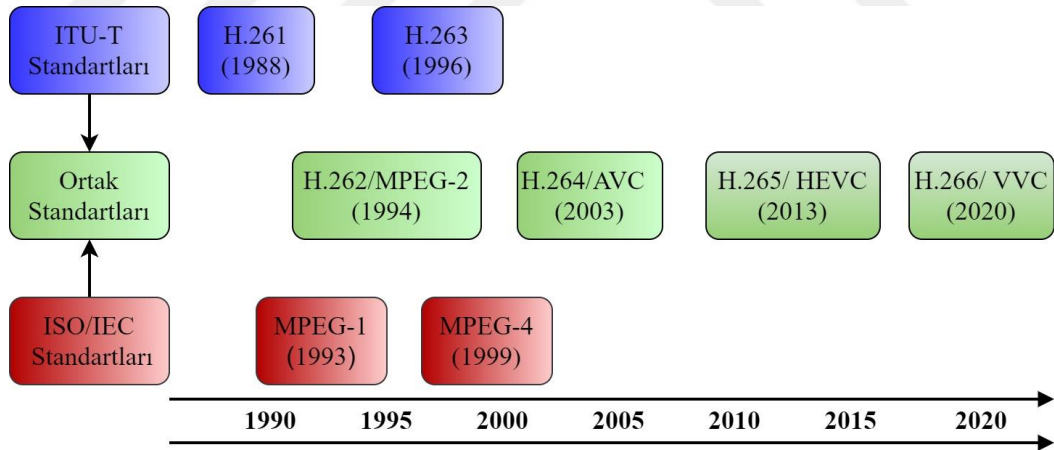
Şekil 1.5. Örnek GOP yapısı [8].

Kodlanmış videolarda kod çözücünün hangi çerçeveleri referans olarak alacağı ve çerçevelerin işlenme sırası bir resim grubu (Group of Picture, GOP) ile belirlenmektedir. Şekil 1.5'te örnek bir resim grubu yapısı verilmektedir. GOP, videonun kodlanma özelliğine göre ardışık çerçevelerden oluşmaktadır. GOP içindeki çerçevelerin türleri videonun kodlanma hızına bağlı olarak değişmektedir. Kod akışında yeni bir GOP'a denk gelindiğinde, sonraki kareleri çözmek için önceki GOP'taki çerçevelere ihtiyaç kalmaz [8].

1.3. Video Kodlama Standartları

Videoların sıkıştırma yöntemlerinin standartlaşma süreci video teknolojisinin gelişip yaygınlaşması üzerinde büyük etkiye sahiptir. Video kodlama standartları daha düşük

bit hızları kullanarak görsel kaliteyi en üst düzeyde tutmayı amaçlamaktadırlar. Başarılı bir video kodlama yönteminin kendisinden önceki standartlara verimlilik anlamında bariz üstünlükler sağlaması gerekmektedir. Video kodlama standartlarının geliştirilmesine öncü iki grup vardır. Bu gruplardan birisi Uluslararası Telekomünikasyon Birliği Telekomünikasyon Standardizasyon kolu (ITU-T) tarafından oluşturulan Video Kodlama Uzmanları Grubu (VCEG), diğeri de Uluslararası Standardizasyon Organizasyonu (ISO) ve Uluslararası Elektroteknik Komisyon (IEC) ortaklığında kurulan Hareketli Resim Uzmanları Grubudur (MPEG). VCEG ve MPEG grupları ayrı ayrı standartlar geliştirmenin yanı sıra ortak çalışmalar da yapmışlardır. VCEG tarafından H.261 ve H.263, MPEG tarafından MPEG-1 ve MPEG-4 standartları geliştirilmiştir. Her iki grubun ortak çalışmalarıyla H.262/MPEG-2, H.264/AVC-MPEG-4 Part-10 ve H.265/HEVC standartları geliştirilmiştir. Son yıllarda geliştirilen en verimli video sıkıştırma formatları olarak 2003 yılında ilan edilen H.264/AVC (Advance Video Coding) ve aynı sıkıştırma oranında ondan yaklaşık iki kat daha fazla bit tasarrufu sağlayan [9], 2013 yılında ilan edilen, H.265/HEVC (High Efficiency Video Coding) formatları göze çapmaktadır.

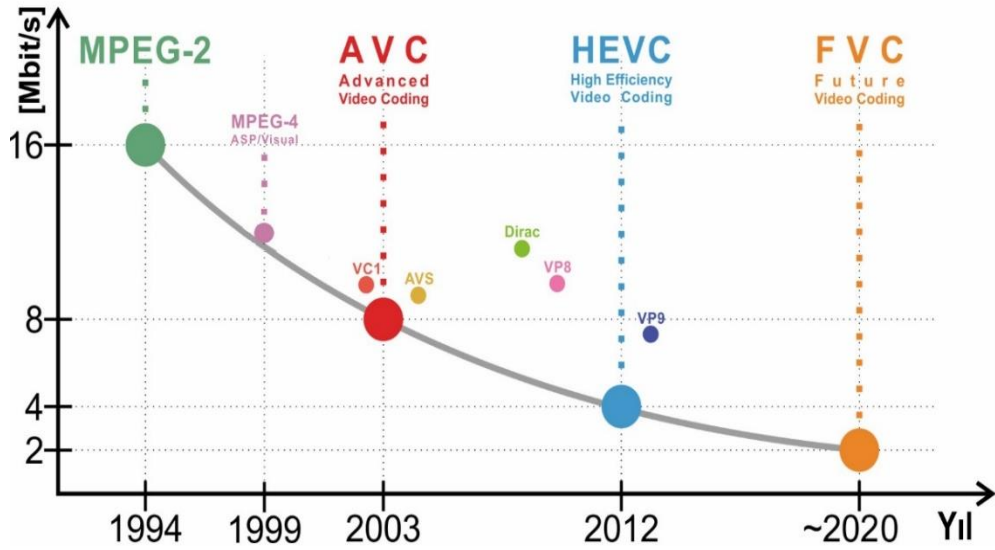


Şekil 1.6. VCEG ve MPEG tarafından geliştirilen video kodlama standartları

VCEG ve MPEG tarafından kurulan Ortak Video Uzmanları Takımı (JVET, Joint Video Experts Team) tarafından geleceğin video kodlaması ismiyle başlatılan çalışmalar sonucunda ortaya çıkan H.266/VVC (Versatile Video Coding) standardının 2020 yılının sonuna kadar ilan edilmesi beklenmektedir. VCEG tarafından geliştirilen standartların H.26x olarak adlandırılma sebebi standartların sıralamasıyla ilgilidir. Şöyle ki ITU-T'nin A ile Z arasında gruplanan standartlarından görsel-işitsel çoklu ortam sistemleriyle ilgili standartları H-serisinde yer almaktadır. H-serisi içinde

hareketli videoların kodlanmasıyla ilgili standartlar H.260-H.279 aralığındadır. Şekil 1.6'da VCEG ve MPEG grupları tarafından geliştirilen video kodlama standartları gösterilmiştir.

ITU-T ve ISO/IEC standartlarının dışında bazı gruplar kendi video uygulamalarında kullanmak üzere çeşitli video standartları geliştirmişleridir. Bunlardan Microsoft öncülüğünde geliştirilen VC-1 [10], BBC yayın organı tarafından geliştirilen Dirac [11], Google öncülüğünde geliştirilen VP8 ve VP9 [12] ve Çin devleti tarafından geliştirilen AVS [13] en çok öne çıkanlardır. VP9 verimlilik bakımından H.264 ile yakın başarımlar sunarken H.265/HEVC'ye göre oldukça geride kalmaktadır [14, 15]. Normal videolar için kodlama başarımları HEVC ile kıyaslanabilir düzeyde olan AVS'nin son versiyonu AVS-2 özellikle gözetleme (surveillance) videoları için çok iyi başarımlara sahiptir [16, 17].



Şekil 1.7. Full HD bir içerik için video kodlama standartlarının başarımları [18]

Şekil 1.7'de video sıkıştırma standartlarının Full HD bir videoyu sıkıştırma başarımları verilmiştir. Burada görüldüğü gibi ham videoya göre sıkıştırma başarımları oldukça iyi durumdadır. Geliştirilen standartlar sayesinde aynı kalitedeki video için bit oranı değerleri 10 yılda bir yaklaşık %50 aşağı düşerek iyileştirilmektedir [18]. Bu ise hem depolama hem de gerçek zamanlı uygulamalardaki video iletişim kalitesi için oldukça önemli bir gelişmedir. Geliştirilen video kodlama standartlarının en önemlileri aşağıda özetlenmiştir.

1.3.1. H.261

ITU-T'nin VCEG grubu tarafından 1988 yılında ilan edilen H.261, pratikte kullanışlı olan ilk video kodlama standardı olup, ISDN (Tümleşik Hizmetler Sayısal Ağı) üzerinden video konferans için tasarlanmıştır. CIF (352×288) ve QCIF (176×144) video çözünürlüklerinin desteklendiği H.261'de I ve P çerçeveler 4:2:0 kroma yapısıyla kullanılmıştır. Temel işlem birimi olarak 16×16 makroblok yapısının ilk defa kullanıldığı bu kodlamada I-çerçeveler görüntü sıkıştırmadaki gibi kodlanmıştır. Ayrıca, çerçeveler arası hareket dengelemeli zamansal kestirim kullanılarak bit oranı azaltılmaya çalışılmıştır [19].

1.3.2. MPEG-1

MPEG komitesinin geliştirdiği ilk halka açık video kodlama standardı olan MPEG-1 CD üzerine sayısal video kaydı için kullanılmıştır. Bu formatta çerçeveler I, P ve B çerçevelerinden oluşan GOP yapısı ilk defa kullanılmıştır. Üç farklı çerçeve biçimde kodlanan video dizisindeki B-çerçeveleri hem kendinden önceki çerçeveler hem de kendisinden sonra gelecek çerçevelere göre kestirim yaparak sadece yeni bölümlerin bit dizisine dahil edilmesini sağlamıştır [20]. Bunun yanında hareket kestirimi için yarım piksel (half-pixel, half-pel) özelliği kullanılarak daha verimli ve daha doğru eşleştirme yapılması sağlanmıştır. MPEG-1 ile birlikte veri depolamada önemli bir aşama kaydedilmiştir. MPEG-1'in 3. katmanı olarak geliştirilen MP3 (MPEG Audio Layer 3) ses dosyalarının sıkıştırılmasında önemli bir yeniliktir [7].

1.3.3. H.262/MPEG-2

MPEG ve VCEG grupları tarafından ortaklaşa geliştirilen ilk standart olan, H.262 veya MPEG-2 Part 2 olarak da bilinen MPEG-2 Sayısal Video Yayınında ve DVD'lerde kullanılan sıkıştırma formatıdır. MPEG-1'deki video kalitesini arttırmak için 1994 yılında MPEG-2 önerilmiştir. Kalite kontrolü için farklı profiller ve farklı seviyede yapıları kullanması, değişken bit oranını desteklemesi, kullandığı kodlama algoritması daha gelişmiş yapıda olması öne çıkan özelliklerindendir. 4:2:0 kroma yapısının yanında 4:2:2 ve 4:4:4 yapılarını da kullanarak daha gelişmiş renk sistemine geçmiştir. MPEG-2 televizyon yayınlarının ses ve görüntü sistemlerini sıkıştırmada kullanılması sebebiyle uzun yıllar boyu popülerliğini korumuştur [21].

1.3.4. H.263

VCEG tarafından geliştirilen H.263, iletim ağdan bağımsız olarak geliştirilmekle birlikte düşük bit oranlı sıkıştırma formatı kullanan video konferans uygulamaları için kullanılmıştır. H.263'ün en önemli özelliği görüntü kalitesini önemli ölçüde iyileştirmesidir. H.261'e göre daha büyük çerçeve boyutlarının desteklenmesinin yanı sıra yarım piksel özelliği kullanılarak hareket vektörleri elde edilmiştir. Bu özellikleri sayesinde H.261 ile kıyaslandığında aynı video kalitesi için yarı yarıya bit tasarrufu sağlamıştır [7].

1.3.5. MPEG-4

1999 yılında ISO/IEC standardı olarak önerilen MPEG-4 daha önceki standartlar MPEG-1 ve MPEG-2'nini kullandığı piksel tabanlı işlemlerinden farklı olarak içerik tabanlı erişimi desteklemektedir. Diğer bir deyişle, MPEG-4'ün getirdiği en önemli yenilik video içerisindeki nesnelerin çerçeveden bağımsız olarak kodlanmasıdır. Videoda dikdörtgen çerçeveyi bir bütün olarak kodlamak yerine, içerikteki nesneler ayrı ayrı sıkıştırılarak televizyon, web uygulamaları ve animasyon grafikleri için uyumlu ve esneklik yeteneklerine sahip çoklu ortam nesneleri oluşturma imkanı sunmuştur [22]. Aynı bit oranındaki video için MPEG-2 göre iki kat daha fazla görüntü kalitesi sağlamaktadır [23].

1.3.6. H.264/AVC

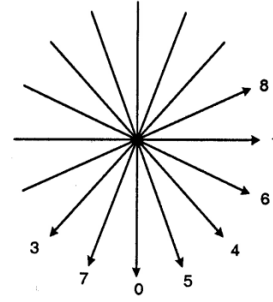
MPEG ve VCEG grupları tarafından ortaklaşa geliştirilen ikinci standart olan, H.264/AVC (Advance Video Coding, İleri video kodlama) aynı zamanda MPEG-4 Part 10 olarak da bilinir. İlk versiyonu 2003 yılında yayınlanan AVC, geliştirildiği zaman en verimli video kodlama formatlarından biri olarak öne çıkmıştır. MPEG-2'ye göre %50 civarında bit tasarrufu sağlaması geniş bir kullanım alanına yayılmasını sağlamıştır. Sunduğu bu üstün sıkıştırma başarımı, internet videolarının, HD televizyon yayıncılığının, mobil servislerin ve video temelli birçok uygulamanın yaygınlaşmasını sağlamıştır [24].

AVC'nin kendisinden önceki video kodlama standartlarından daha iyi başarımla sergilemesinin birçok etkeni vardır. Çerçeve-içi ve çerçeveler arası kestirime getirilen

yeniliklere ek olarak dönüşüm işlemi, entropi kodlama ve son aşamadaki bloklanmayı önlemek için yapılan değişiklikler bulunmaktadır.

İlk olarak çerçeve içi kestirime farklı bir yaklaşım getirilerek, dönüşüm katsayıları yerine mevcut resmin daha önce kodlanmış bloklarının uzamsal düzlemdeki komşuluk değerleri kullanılmıştır. Mevcut bloğun komşu blokların kenarlarından tahmin işlemi için 9 farklı çerçeve-içi kestirim modu kullanılmıştır. Şekil 1.8’de 4×4 bloklar için çerçeve içi kestirim biçimi gösterilmektedir. Burada 4×4 bloktaki a-p arasındaki piksel değerlerini tahmin etmek için komşu bloklara ait A-Q örneklerinden en uygun olanları verilen dokuz mod yardımıyla tespit edilmektedir. Blok değerlerinin en iyi biçimde tahmin edildiği mod, o bloğun çerçeve-içi kestirim modu olarak seçilmektedir. Bu şekilde yapılan tahmin işlemi çerçevelerin kalitesinde önemli bir artış sağlamaktadır.

Q	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				



Şekil 1.8. H.264/AVC Çerçeve içi kestirim modları [24].

AVC’de sıkıştırma başarımını arttırmak için çerçeve-içi ve çerçeveler arası kestirimde değişken makroblok yapıları kullanılmıştır. Çerçeve-içi kestirimde 16×16 bloklara ek olarak 4×4 blok boyutu da kullanılmıştır. Çerçeveler-arasındaki hareket kestirimi ve hareket dengelemesinde ise mevcut 16×16 makroblok yapısı yerine 16×16, 16×8, 8×16, 8×8 makrobloklar kullanılmaktadır. Ayrıca 8×8 boyutuna bölünen makrobloklar 8×8, 8×4, 4×8 ve 4×4 bloklar halinde kullanılabilir. En uygun blok boyutunu seçmek için oran-bozunum (rate-distortion) metodu kullanılmaktadır. Değişken blok seçimi, sabit blok boyutuna göre %15’e kadar bit oranında tasarruf sağlamaktadır. Hareket dengelemesi için yapılan yeniliklerden birisi de referans çerçeve sayısının artırılmasıdır. Eski kodlama yöntemlerinin sadece bir önceki çerçeveden yaptığı hareket dengelemesinden farklı olarak AVC birden fazla referans çerçeve kullanarak kodlama başarımını arttırmıştır.

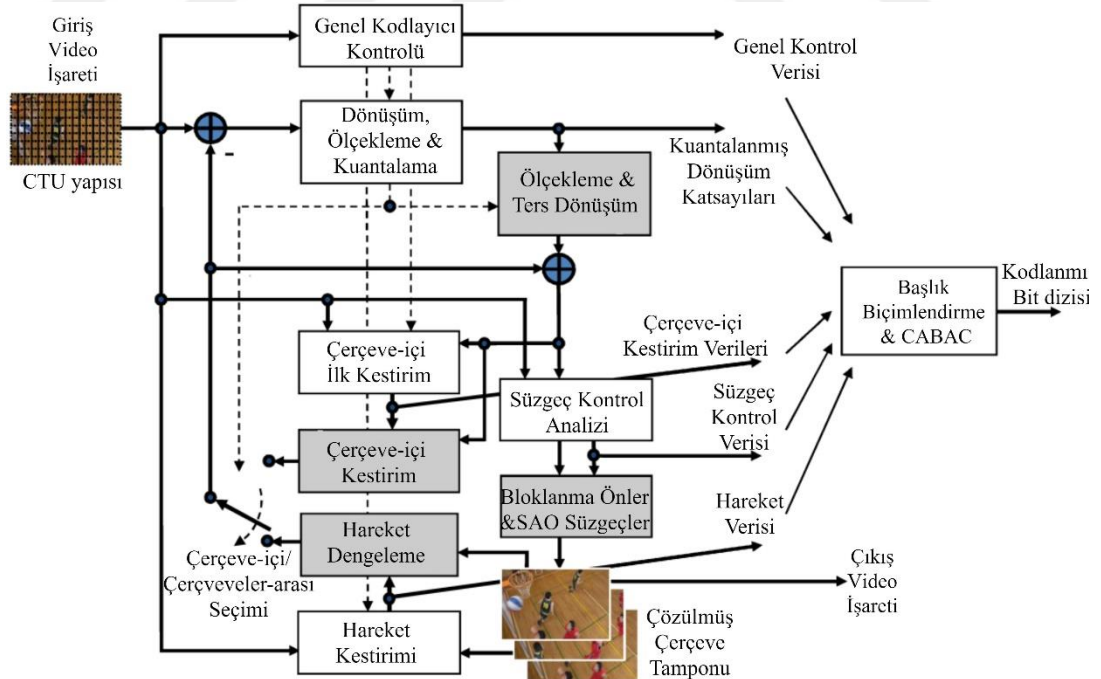
Entropi kodlama aşamasında daha verimli bir kodlama için içerik uyarlamalı ikili aritmetik kodlamayı (context-adaptive binary arithmetic coding, CABAC) kodlama ve içerik uyarlamalı değişken uzunluklu kodlama (context-adaptive variable length coding, CAVLC) olmak üzere iki yöntem kullanılmıştır.

AVC’de bütün işlemler blok tabanlı yapılmaktadır. Bu sebeple kodlanmış videoların kodu çözüldüğünde çerçeveler arasında bloklanmalar şeklindeki belirgin geçişler meydana gelmektedir. Bu problemin giderilmesi için AVC döngüsü içerisinde bloklanmayı önleyen (deblocking) süzgeç kullanılarak çerçevelerin görsel kalitesi iyileştirilmiştir [24].

H.265/HEVC, Yüksek Verimli Video Kodlama tez çalışmanın alt yapısını oluşturduğu için ayrı bir bölüm olarak sunulmuştur.

2. H.265/HEVC

H.265/HEVC (High Efficiency Video Coding, Yüksek Verimli Video Kodlama), VCEG ve MPEG gruplarının ortaklığında kurulan JCT-VC (Joint Collaborative Team on Video Coding) tarafından geliştirilen video kodlama standardıdır [25]. 2013'te ilk sürümü ilan edilen standardın, Kasım 2019'da mevcut sürümü (versiyon 7) çıkmıştır. Bir önceki standart olan H.264/AVC'ye ve Google'ın geliştirdiği VP9 standardına kıyasla %50 civarında bit tasarrufu sunmaktadır [14]. Artan video çözünürlüklerine ve artan paralel işlem mimarilerine yönelik çözüm getirmeye odaklanan HEVC, Ultra HD 4K (3840×2160) görüntü kalitesine uyumlu ve Ultra HD 8K (8192×4320) görüntü kalitesine kadar çözünürlük desteği sağlamaktadır. Söz dizimi (syntax) yapısının esnekliği sebebiyle mevcut veya yeni çıkan uygulamalara kolayca uygulanabilmektedir. Bu sebepten ötürü, birçok uygulamada H.265/HEVC standardına geçilmektedir.



Şekil 2.1. HEVC video kodlayıcı yapısı [25].

Şekil 2.1'de HEVC video kodlayıcı yapısı verilmektedir. Kodlama işleminde öncelikle her çerçeve bloklara ayrılmaktadır. Videodaki ilk çerçeve her zaman I-çerçeve olarak

kodlandıktan sonra geriye kalan çerçeveler için kodlama profiline bağlı olarak genellikle çerçeveler-arası kestirim özellikleri için kullanılmaktadır. Çerçeveler-arası hareket kestirimde, seçilen referans çerçeve ve hareket vektörü bilgileri elde edilmektedir. Her iki kestirim işlemi sonrası orijinal giriş bloğu ile kestirilen blok arasındaki farktan oluşan artıklık işareti uzamsal tamsayı dönüşümüne tabi tutulmaktadır. Kuantalanmış dönüşüm katsayıları, çerçeve-içi ve çerçeveler-arası kestirim bilgileri, blok bilgileri ve genel kontrol bilgilerinin entropi kodlama işlemi sonrası kodlanmış bit dizisi elde edilmektedir. Kod çözücü çıkışında elde edilecek resmin, kodlayıcı içerisinde ters işlemler ile oluşturularak, orijinal giriş resmi ile en yüksek benzerlikte olması sağlanmaktadır.

HEVC kodlayıcı sistemi daha önceki video kodlama standartlarına benzemesine rağmen kodlama verimliliği, kayıplara dayanıklılık, paralel mimarilere uygulanabilirlik ve sistemlerle kolay bütünleşebilirlik için kodlama yapısında çeşitli yenilikler içermektedir. Bu yenilikler aşağıda özetlenmiştir [25].

Daha önceki standartlarda makroblok yapısı kullanılırken HEVC’de her bir video çerçevesi dilimlere (slice) ayrılır, daha sonra bu dilimler 64×64 piksellik kodlama ağaç birimlerine (CTU, Coding Tree Unit) bölünmektedir. CTU, dörtlü ağaç yapısına benzer bir şekilde kodlama birimlerine (CU, Coding Unit) bölünmektedir. CU’lar ise çerçeve arası veya çerçeve içi kestirim birimlerine (PU, Prediction Unit) ve dönüşüm birimlerine (TU, Transform Unit) dönüşmektedir.

Resim içerisinde CU’ların ne kadar bölüneceğine PU’lardan elde edilen çerçeve-içi veya çerçeveler-arası kestirim bilgileriyle karar verilmektedir. Kestirim işleminden arta kalan bilgiler TU’lar ile kodlanmaktadır. CU ile aynı boyutta olan veya daha alt boyutlara bölünebilen TU bilgileri, 32×32 , 16×16 , 8×8 veya 4×4 piksel boyutları için ayrık kosinüs dönüşümüne (Discrete Cosine Transform, DCT) benzer tamsayı temelli bir dönüşümle elde edilmektedir. Buna ek olarak çerçeve-içi kestirim işleminde 4×4 blokların artıklık bilgileri ayrık sinüs dönüşümü (Discrete Sine Transform, DST) formundan üretilen bir tamsayı dönüşümü ile bulunmaktadır.

HEVC’de Hareket vektörlerinin zamansal ve uzamsal komşu bloklarından elde edildiği bir atlama (skip) moduna ek olarak gelişmiş atlama ve doğrudan hareket çıkarımı kullanılmaktadır. Ayrıca gelişmiş hareket vektörü tahmini kullanılarak komşu

PU'lerden ve referans çerçevelerden hareket kestirimi için en olası birkaç aday türetilmektedir. Hareket uyumlama için AVC'ye benzer şekilde çoklu referans resimleri kullanılırken, hareket vektörleri için çeyrek piksel hassasiyeti kullanılmaktadır.

Çerçeve-içi uzamsal kestirim işleminde, AVC'ye benzer biçimde komşu blokların kullanıldığı HEVC'de kestirim modlarının sayısı artırılarak toplam 35 mod (33 açışal mod, düzlemsel (planar) ve DC) kullanılmaktadır. Bir PU bloğu içerisindeki mod, kendisinden önceki PU blokların modlarına da bağlı olan, olası modlar içerisinde belirlenmektedir.

Kuantalama işlemindeki ölçekleme matrisleri farklı boyutlardaki TU'lara uyumlu olacak şekilde kullanılmaktadır. Entropi kodlama işleminde ise AVC'deki CABAC yapısında çeşitli iyileştirmeler yapılmıştır. Böylece paralel işlem mimarisinde daha hızlı çıktı, sıkıştırma başarımı ve bellek gereksinimini azaltmak gibi avantajlar sağlanmıştır.

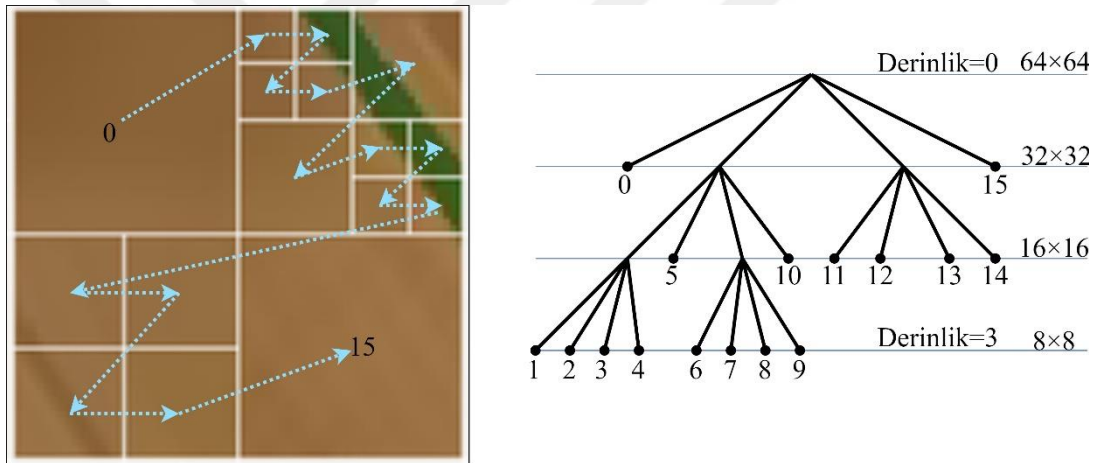
HEVC standardında tasarım özelliklerindeki yenilikler uygulama ve ağ ortamlarındaki çeşitliliğe uyumu artırmanın ve veri kayıplarına karşı dayanıklılığı artırmanın yanı sıra daha hızlı işlemlere olanak vermiştir. Bunun için AVC'deki dilimli yapı, çeşitli türlerdeki meta verileri, ağ soyutlama katmanı söz dizimi yapısı gibi özellikleri kullanılmaya devam etmiştir. Ayrıca AVC'deki kodlanmış verinin çözülmesinde kullanılan dizi ve resim parametre setlerine ek olarak video parametre setleri kullanılmıştır.

2.1. HEVC'de Blok Bölme ve Dörtlü Ağaç Yapısı

HEVC'de daha iyi bir sıkıştırma başarımı için blokların CTU, CU, PU ve TU olarak bölünmesi önerilmiştir. Bu blokların YCbCr uzayındaki iki boyutlu parlaklık ve renk bileşenlerine ait blokları CTB, CB, PB ve TB olarak adlandırılmaktadır. Örneğin bir CTU içerisinde bir parlaklık CTB ve iki tane kroma CTB bulunmaktadır [26]. Bu şekilde kullanılan kodlama ağaç yapısı video sıkıştırma için ilave verimlilik sağlamaktadır. Kodlama işlemi için girişten alınan video dizisi ilk olarak çerçevelere ayrılmaktadır. Kodlama için seçilen konfigürasyona göre GOP'lar oluşturularak ve GOP içerisindeki çerçeveler birden fazla CTU'dan oluşan dilime ayrılarak kodlama

işlemleri başlatılır. Bir çerçevedeki ana kodlama birimi olan CTU'nun kodlama işlemleri tamamlandıktan sonra diğer bir CTU'ya geçilir. Çerçeve içerisindeki tüm CTU'lar kodlanarak, entropi kodlamaya kadar olan süreç benzer biçimde gerçekleştirilir.

CTU boyutları kullanılan kodlama profiline bağlı olarak 64×64 , 32×32 , 16×16 veya 8×8 piksel olabilmektedir. Blok boyutlarının büyümesi yüksek çözünürlüklü videolardaki bölgelerin daha iyi kodlanmasını sağlamaktadır. CTU'lar içerdikleri karakteristik bölgelerin özelliklerine göre CU'lara bölünmektedir. Boyutları 8×8 piksele kadar bölünebilen CU'lar çerçeve-içi ve çerçeveler-arası kestirim işlemleri için kullanılan ana birimlerdir. Dolayısıyla CU boyutları için kestirim işlemleri sonucu elde edilen maliyet fonksiyonuna göre karar verilmektedir.

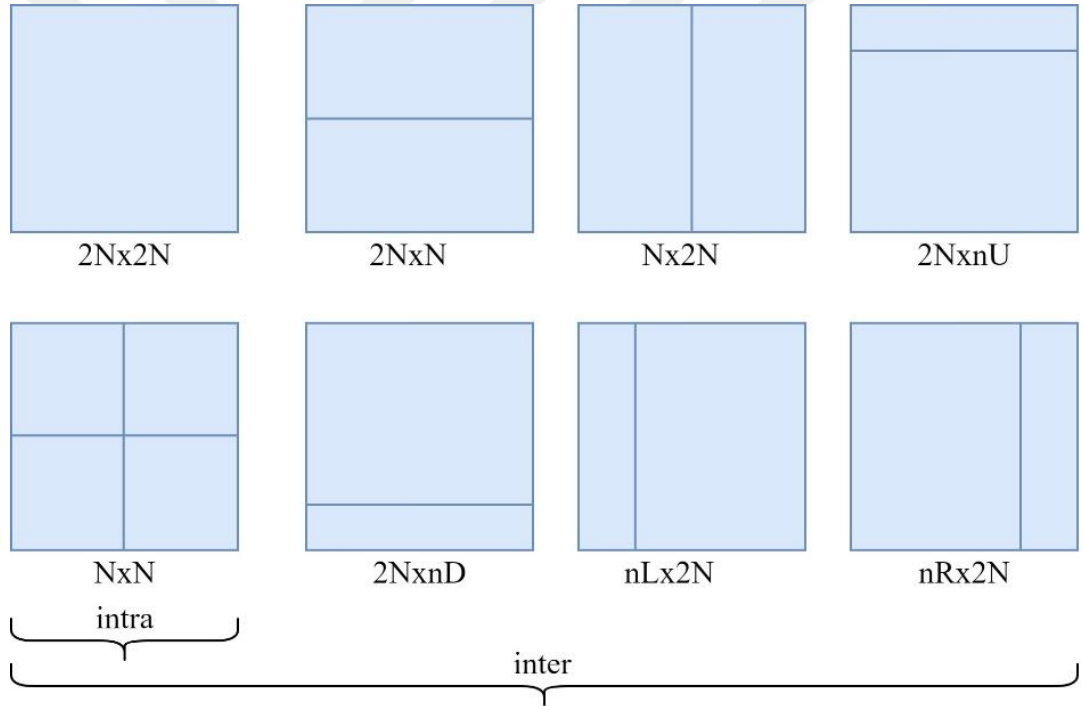


Şekil 2.2. 64×64 bir CTU için CU boyutları ve dörtlü ağaç yapısı

Bölünme boyutlarının kararı için 64×64 piksel boyutundaki CU'da kestirim işlemleri için en düşük maliyeti veren kestirim bilgileri hesaplandıktan sonra CU bir alt derinlik (depth) boyutunda dört tane CU'ya bölünmektedir. Bölünen blokların her birisi için aynı şekilde en düşük maliyete sahip modlar bulunur. Bölünme işlemi özyinelemeli (recursive) olarak 8×8 boyutuna kadar devam ederek her seviyedeki maliyet değerleri elde edilmektedir. En düşük kodlama maliyetlerine sahip CU bloklarına göre CTU'nun bölünmesine karar verilmektedir [26].

Şekil 2.2'de örnek bir video çerçevesindeki bir CTU'nun içerisindeki CU'lar ve onların dörtlü ağaç yapıları gösterilmiştir. 64×64 piksellik CTU maliyet fonksiyonu karşılaştırmaları sonucunda 4 adet 32×32 CU'ya bölünmüştür. İlk ve son CU'ların

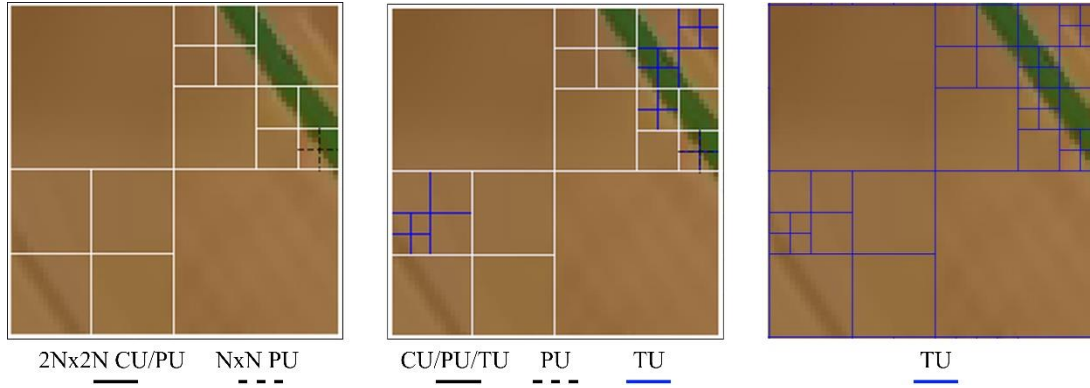
boyutu daha alt derinliklere bölünmezken, ikinci 32×32 boyutlu CU, 16×16 ve 8×8 boyutlu alt CU'lara ve üçüncü 32×32 boyutlu CU, 16×16 boyutlu alt CU'lara bölünmüştür. CTU içerisinde CU'ların kodlanma sırası özyinelemeli kodlama yaklaşımına uygun olarak dörtlü ağaç yapısına göre yapılmaktadır. Burada piksel renk dağılımının değiştiği bloklar düşük CU boyutları seçilirken, değişimin az olduğu homojen yerlerde daha büyük boyutlu CU'lar seçilmiştir. Bu CU'ların içerisinde kestirim bilgilerini tutan bir, iki veya dört adet PU bulunmaktadır. Şekil 2.3'te HEVC'de kullanılan PU boyutları gösterilmektedir. PU'nun boyutunun ne olacağı CU'nun kodlanma türüne bağlıdır. Çerçeve-içi kodlamada $2N \times 2N$ boyutlu bir CU, mevcut bloğu komşu bloklardan kestirmek için bir adet $2N \times 2N$ PU'dan veya dört adet $N \times N$ PU'dan kullanmaktadır [27].



Şekil 2.3. Çerçeve-içi (intra) ve çerçeveler-arası (inter) PU Modları

Çerçeveler-arası kodlamada ise hareket uyumlama şeması kullanılarak mevcut bloğu kestirmek için karesel $2N \times 2N$ ve $N \times N$, simetrik $2N \times N$, $N \times 2N$ ve asimetrik $2N \times nU$, $2N \times nD$, $nL \times 2N$ ve $nR \times 2N$ boyutlu PU'lar kullanılmaktadır. Bu modların haricinde çerçeveler-arası CU yapısında atlama (skip) modu PU bulunmaktadır. PU boyutunun $2N \times 2N$ olduğu bu modda hareket vektör farkları ve artıklık bilgisi sıfıra eşittir. Her PU'nun kendine ait çerçeve-içi kestirim modu ve en muhtemel mod bilgileri, çerçeveler-arası hareket vektör farkı, referans indeksi, atlama indeksi ve bayrağı gibi

bilgileri bulunmaktadır [26]. PU'ların belirlenmesinden sonra artıklık bilgilerini içeren TU'lar farklı bir dörtlü ağaç yapısına göre bölünmektedirler.



Şekil 2.4. 64x64 bir CTU'nun CU, PU ve TU yapısı

Artıklık bloğu, dörtlü ağaç yapısında bölünmektedir ve bu artıklık dörtlü ağacı (residual quad tree, RQT) yapısındaki her bir TU'ya tamsayı dönüşümü uygulanarak dönüşüm katsayıları elde edilmektedir. Şekil 2.4'te örnek bir CTU'nun bölünme yapısı gösterilmiştir. Şekildeki PU'lardan sadece birisi bir CU'lardan farklı derinlikte bölünmüştür. Geriye kalan Tüm PU'lar CU'lar ile aynı derinliğe sahiptir. TU'lar ise bazı bloklarda CU ve PU ile aynı derinliğe sahipken, bazı bloklarda daha fazla derinlikte bölünmüştür. Şekilde görülen başka bir hususta şudur: CU'lar 8x8 piksel boyutuna kadar bölünebilirken, PU ve TU blokları 4x4 piksel boyutlarına kadar bölünebilmektedir.

2.2. Çerçeve-içi Kestirim ve Mod Seçimi

HEVC'de çerçeve içi kestirim işlemi AVC'ye benzer olarak, daha önce kodlanmış bloklardan uzamsal dışadeğerleme (extrapolation) işlemi ile yapılmaktadır. Herhangi bir bloğun değerlerinin kestirmek için, o bloğun üstündeki ve solundaki blokların değerleri kullanılmaktadır. Kestirilecek değerleri $P_{x,y}$ olan bir blok için referans örnekler $R_{x,y}$ Şekil 2.5'teki gibi alınmaktadır. Herhangi bir bloğun referans örnekleri için üst veya sol bloklar bulunmuyorsa en yakın referans bloklara bakılır. Eğer henüz kodlanmış herhangi bir referans değer yoksa tüm referans değerleri bit sayısına göre temsili bir ortalama değere (mesela 8 bitlik resim için 128) eşitlenir. Referans değerden, kestirilecek değeri bulmak için çerçeve-içi kestirim modu (intra prediction mode, IPM) kullanılmaktadır. HEVC'de düzlemsel (planar) mod, DC mod ve açısıl

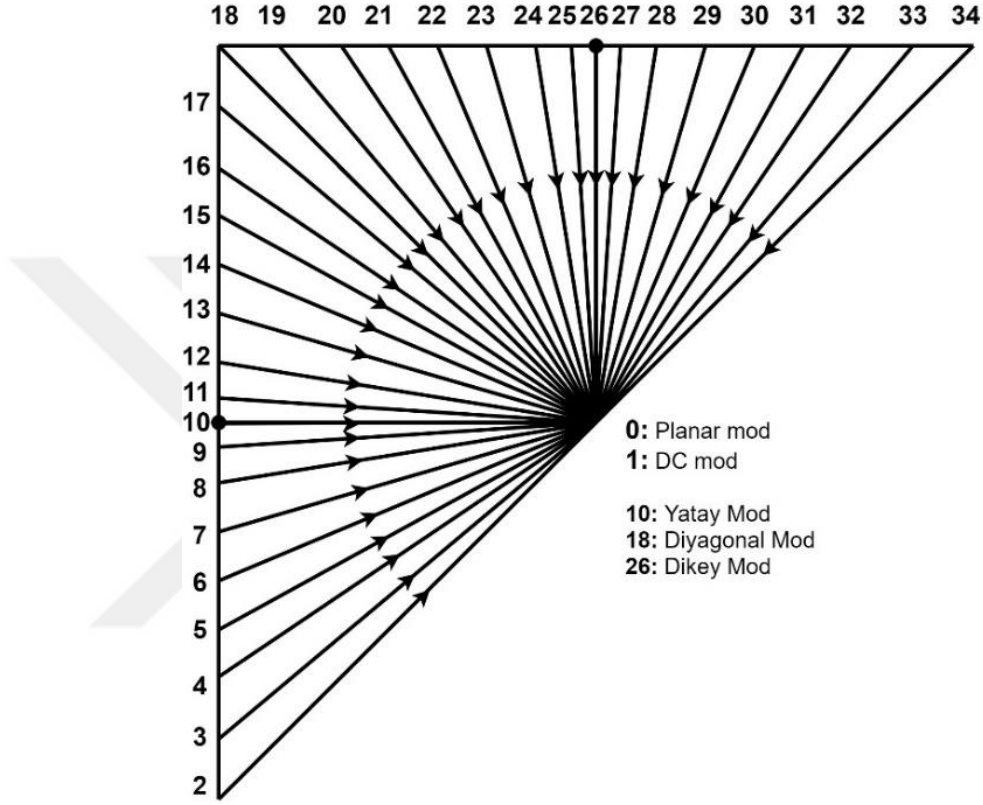
$R_{0,0}$	$R_{1,0}$	$R_{2,0}$	...	$R_{N,0}$	$R_{N+1,0}$	...	$R_{2N,0}$
$R_{0,1}$	$P_{1,1}$	$P_{2,1}$	...	$P_{N,1}$			
$R_{0,2}$	$P_{1,2}$	$\ddots$		$\vdots$			
$\vdots$							
$R_{0,N}$	$P_{1,N}$	...		$P_{N,N}$			
$R_{0,N+1}$							
$R_{0,2N}$							

Şekil 2.5. $N \times N$ 'lik bloktaki elemanların kestiriminde kullanılan referans örnekler [28]

modlar olmak üzere toplam 35 IPM desteklenmektedir. Şekil 2.6'te HEVC'deki modlar ve yönleri gösterilmektedir. Geçerli bir PU bloğu için seçilen IPM belirlendikten sonra o bloğun tüm piksel değerleri ($P_{x,y}$) referans örneklerden ($R_{x,y}$) direk kopyalanarak veya dışađerleme yapılarak belirlenir. Kopyalanacak referans değeri IPM yönündeki referanstan alınmaktadır. Açısız modlar haricinde seçilebilen IPM'lerden DC modu kenar geçişleri için kaba bir tahmin sağlarken, düzlemsel mod blok sınırları boyunca süreklilikleri korumayı amaçlamaktadır. Mod seçim işlemi kodlama zamanının önemli bir kısmını kapsamaktadır. Çünkü her bir CTU'da olası tüm PU yapıları için IPM'ler sırayla denenmektedir. Süreci hızlandırmak amacıyla geçerli PU modu için iki aşamalı değerlendirme yapılmaktadır. Birinci aşamada, N adet en muhtemel mod kaba bir mod seçim yöntemiyle seçilmektedir. Bu süreçte 35 modun tümü Eşitlik (2.1)'deki oran-bozunum maliyet fonksiyonuna göre değerlendirilerek sıralanmaktadır.

$$C = D_{\text{had}} + \lambda \cdot R_{\text{mode}} \quad (2.1)$$

Burada D_{had} bir PU için artıklık işaretinin Hadamard dönüşümünün mutlak toplamını, R_{mode} ise kestirim modu için kullanılacak bit sayısını göstermektedir. λ , Lagrangian çarpanını ifade etmekte olup, video sıkıştırma için kullanılan kuantalama parametresi (quantization parameter, QP) kullanılarak bulunan bir değerdir. Oran-bozunum maliyet değeri en düşük olan modlar, muhtemel N tane mod listesine alınır.



Şekil 2.6. HEVC’de kullanılan çerçeve-içi kestirim modları

IPM seçiminde muhtemel N tane mod belirlenirken maliyet fonksiyonuyla belirlenen modların haricinde üstteki ve soldaki bloklara ait IPM modları da kullanılmaktadır. Bir mod üstteki PU’dan, bir mod soldaki PU’dan, diğer modlar ise blok için kabaca hesaplanan 35 mod içerisinde seçilerek N tane olası mod belirlenir. Eğer üstteki ve soldaki PU’lara ait mod aynı ise N tane muhtemel modlar, o mod en yakınındaki diğer modlardan oluşturulur. N değeri PU boyutuna göre değişmekte olup 4×4 , 8×8 , 16×16 , 32×32 ve 64×64 bloklar için sırasıyla 8, 8, 3, 3 ve 3 olarak alınmaktadır. Blok boyutu küçüldükçe piksel değerleri daha kritik yerleri ifade ettiği için daha detaylı kontrol yapılmaktadır. Mod seçiminin ikinci aşamasında ise yeniden oluşturulan artıklık işareti için tam bir oran-bozunum maliyeti N tane aday mod için kontrol edilir. En

düşük RD maliyetine sahip kesitirim modu en son kestirim modu olarak seçilir ve ilgili PU'nu IPM modu olarak belirlenir. Her PU için yeni bir IPM modu seçilirken, PU içerisindeki tüm TU'lar da aynı mod değerine sahip olmaktadır [28].

2.3. Dönüşüm ve Kuantalama

Daha önceki standartlarda olduğu gibi HEVC standardında da artıklık bilgileri için dönüşüm kodlaması kullanılmaktadır. 4×4 , 8×8 , 16×16 veya 32×32 boyutunda olabilen TU artıklık bloğu için dönüşüm işlemi yapılmaktadır. TU boyutlarının çoklu olması uygulama karmaşıklığını arttırmasına rağmen sıkıştırma başarımını iyileştirmektedir. İki-boyutlu dönüşüm işlemi yatayda ve düşeyde yapılan bir-boyutlu dönüşümler ile yapılmaktadır. İlgili blok için elde edilen dönüşüm katsayıları, kuantalanır. Kuantalanmış dönüşüm katsayıları (Quantized Transform Coefficient, QTC), entropi kodlama işlemi sonrası kodlanmış bit dizisine dâhil edilerek alıcıya/kod çözücüye gönderilir. Kod çözücü tarafta, entropi kodları çözüldükten sonra elde edilen QTC'ler ters kuantalamaya ve ters dönüşüme tabi tutularak, kuantalanmış örneklerin artıklık bilgileri geri elde edilmektedir. Elde edilen artıklık bilgileri çerçeve-içi ve çerçeveler-arası kestirim bilgileriyle oluşturulan bloğa eklenerek, ilgili bloğun piksel değerleri yeniden oluşturulmaktadır.

Sıkıştırmadaki verimliliği ve verimli uygulama biçimi gibi özellikleri sebebiyle, sıkıştırma işlemlerinde DCT kullanılır. HEVC'de farklı biçimdeki gösterimleri kullanan üreticilerinin sebep olabileceği kodlayıcı ve kod çözücü uyumsuzluklarını ve bit gösterimlerindeki kaymaları önlemek için çekirdek dönüşüm matrislerini doğrudan DCT kullanmak yerine DCT'ye sonlu yaklaşık benzerlik gösteren tamsayı dönüşümü kullanılmaktadır. Tam sayı matrisini elde etmenin en kolay yollarından birisi, DCT ile elde edilen matris elemanlarının ölçeklenerek en yakın tam sayılara yuvarlamaktır. Kolaylık için sadece bir tane 32×32 boyutlu tamsayı matrisi kullanılmaktadır. HEVC referans yazılımında belirli tamsayıların matrisin satır ve sütunlarına yerleştirilmesiyle 32×32 boyutlu çekirdek tamsayı matrisi oluşturulmaktadır. Diğer blok yapıları için kullanılan çekirdek matrisler, 32×32 matristen örneklenerek elde edilmektedir. Bu özelliğin en önemli katkısı farklı matrislerin kullanılmasından kaynaklanan uygulama maliyetlerini azaltması ve aynı çarpanların farklı dönüşüm boyutlarında yeniden

kullanılabilirliği [29]. Eşitlik (2.2)'de 8×8 TU için kullanılan tamsayı dönüşüm matrisi verilmiştir.

$$S_{8 \times 8} = \begin{bmatrix} 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\ 89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 \\ 83 & 36 & -36 & -83 & -83 & -36 & 36 & 83 \\ 75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 \\ 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\ 50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 \\ 36 & -83 & 83 & -36 & -36 & 83 & -83 & 36 \\ 18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 \end{bmatrix} \quad (2.2)$$

HEVC'de 4×4 bloklardaki çerçeve-içi kestirim modlarının artıklık işaretlerini dönüştürmek için DCT yerine alternatif olarak DST'den türetilen bir tamsayı dönüşümü kullanılmaktadır. DST tarzı dönüşümün kullanılması işlem karmaşıklığını arttırmadığı halde çerçeve-içi kestirimdeki bit oranını yaklaşık %1 azaltmaktadır. DST tarzı tamsayı dönüşümü sadece 4×4 TU'nun parlaklık bloğuna uygulanmaktadır [30]. Eşitlik (2.3)'de 4×4 TU için kullanılan tamsayı dönüşüm matrisi verilmiştir.

$$S_{4 \times 4} = \begin{bmatrix} A & B & C & D \\ C & C & 0 & -C \\ D & -A & -C & B \\ B & -D & C & -A \end{bmatrix} = \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix} \quad (2.3)$$

R bir artıklık bloğu, S ise R ile aynı boyutlu tamsayı dönüşüm matrisi olmak üzere, bir TU'nun QTC değerleri Eşitlik (2.4)'teki gibi elde edilmektedir.

$$R_{QTC} = \text{round}((S \times R \times S^T) / (Q_{\text{step}})) \quad (2.4)$$

$$Q_{\text{step}} = (2^{1/6})^{QP-4} \quad (2.5)$$

Burada $\times$ matris çarpımını, T matris transpozunu, “round” yuvarlama işlemini ve Q_{step} QP'ye bağlı kuantalama adımını ifade etmektedir. QP değeri 8-bitlik video dizileri için 0-51 arasında değişmektedir.

2.4. Test Konfigürasyonları ve Profiller

Ham videoların HEVC ile kodlanması için JCT-VC tarafından HM (HEVC Test Model) olarak adlandırılan referans yazılım geliştirilmiştir. Belirli aralıklarla yeni

sürümleri yayınlanan yazılım, geliştiricilerin erişimine sunulmuştur [31]. HM referans yazılımında video özelliklerine göre farklı ortak test konfigürasyonları bulunmaktadır [32]. Bu konfigürasyonlar I, P ve B çerçevelerinin kullanımı ve GOP yapısına göre değişmektedir. Kodlanacak videoda hangi özellikler kullanılacaksa, referans yazılımında ona göre konfigürasyon ayarları seçilmelidir. HM referans yazılımı için kullanılan All-Intra, Random Access, Low Delay P ve Low Delay B olmak üzere dört farklı konfigürasyon koşulu kullanılmaktadır. Ayrıca ilk çerçeve için tanımlanan QP değeri sonraki çerçevelere için tanımlanan bir QP ofset değeri ile değiştirilebilmektedir [33].

HM referans yazılımı için kullanılan dört farklı yapılandırmanın özellikleri:

All Intra: Bu konfigürasyonda tüm çerçeveler I olarak kabul edilir ve uzamsal olarak kodlanır. QP ofset değeri sıfır olarak alındığı için bütün çerçeveler aynı QP ile kodlanmaktadır. All Intra konfigürasyonunda zamansal artıklık bilgileri giderilmediği için bit oranı nispeten yüksektir.

Random Access: Kodlama verimliliğinin en iyi olduğu konfigürasyondur. İlk çerçeve I olarak kodlanırken, hiyerarşik B olarak kodlanmakta olan diğer çerçeveler piramitsel bir yapıda rastgele erişimli olarak sıralanmaktadır. Olası hataların tespiti ve rastgele erişimi kolaylaştırmak için yaklaşık her 1 saniyede bir tane I çerçeve konulmaktadır. QP ofset değerinin çerçevelere göre farklılık göstermesi sebebiyle çerçevelerin kodlandığı QP değerleri de değişmektedir. Çerçevelerin erişim sırası ve QP ofset değerleri HM referans yazılımının konfigürasyon dosyaları içerisinde tanımlanmıştır. Referans çerçeve için hem önceki hem de sonraki çerçevelerin kullanılması sebebiyle Random Access kodlamada zamansal yükün önemli kısmı hareket kestiriminden kaynaklanmaktadır [34].

Low Delay P: GOP içerisindeki ilk çerçeve I olarak, diğer çerçeveler ise P olarak kodlanmaktadır. Kestirim için sadece önceki çerçevelerin kullanıldığı Low Delay P kodlamada çerçevelerin QP değerleri QP ofset değerine göre değişiklik göstermektedir.

Low Delay B: GOP içerisindeki ilk çerçeve I olarak, diğer çerçeveler ise B olarak kodlanmaktadır. Kestirim için sadece önceki B çerçevelerini kullanan bu

konfigürasyon Low Delay P yapısına benzemekle beraber, B çerçevelerin özelliği sebebiyle daha yüksek kodlama verimliliğine sahiptir.

HM referans yazılımının farklı özellikteki videolara uyum sağlaması için Main, Main 10 ve Main Still Picture şeklinde üç ana profil önerilmiştir. Video kodlamada kullanılan ana profil olan “Main Profil”, bit derinliği 8-bit olan çerçevelere uyumlu olup 4:2:0 kroma örneklemesini kullanmaktadır. “Main 10 Profili”, bit derinliği 10-bit olan çerçevelere uyumlu olup, 4:2:2 kroma örneklemesini kullanmaktadır. “Main Still Picture Profil”, 8-bitlik görüntülerin sıkıştırılması için önerilmiştir. Main Profil’e benzer özellik taşıyan bu profile videonun tek bir çerçeveden oluştuğu kabul edilir. Bir I çerçeve gibi kodlanan görüntü daha önceki görüntü sıkıştırma yöntemlerine göre daha düşük bit oranıyla ifade edilebilmektedir. HEVC referans yazılımında ana profil yapılarına ek olarak 12-bit, 16-bit, 4:4:4 kroma örnekleme, yüksek hacim, 3D gibi özelliklere sahip videolar için de çeşitli profiller de önerilmiştir. Bu profiller, aralık genişletme (range extension), ölçeklenebilir uzantı (scalable extension), çoklu görünüm (multi-view) ve 3D gibi özellikler için önerilmiştir. Ayrıca farklı kod çözücülere ve farklı bellek özelliklerine göre bit akışına bazı kısıtlamalar veya anahtar parametreler koyan çeşitli seviyeler (level) de kullanılmaktadır. Bir seviye için kısıtlamalar örnekleme hızı, resim boyutu, sıkıştırma oranı, bit oranı gibi bazı özelliklerin durumlarına göre belirlenmektedir [31].

2.5. HEVC Başarım Değerlendirme Ölçütleri

Video sıkıştırma yöntemlerinde başarım verimliliği için hem görsel kalite bozunumu hem de bit sıkıştırma oranı birlikte değerlendirilmektedir. Kodlama sonrası yeniden oluşturulan videolardaki görsel kalite bozunumunu ölçmek için Tepe Sinyal Gürültü Oranı (Peak Signal-to-Noise Ratio, PSNR) değerine bakılmaktadır. PSNR, orijinal çerçeve ve kodlandıktan sonra yeniden oluşturulan çerçeve arasındaki

$$MSE = \frac{1}{Y \times G} \sum_{m=1}^Y \sum_{n=1}^G [f_i(m,n) - f_o(m,n)]^2 \quad (2.6)$$

Ortalama Karesel Hata (Mean Square Error, MSE) değerinin tersinin logaritmik ölçekte desibel (dB) olarak gösterilmesiyle elde edilmektedir. Orijinal giriş çerçevesi f_i , çıkış çerçevesi f_o olan bir video için MSE değeri Eşitlik (2.6) ile elde edilmektedir.

Burada Y ve G sırasıyla çerçevenin yüksekliğini ve genişliğini göstermektedir. PSNR değeri ise,

$$\text{PSNR (dB)}=10 \cdot \log_{10}(P_{\max}^2/\text{MSE}) \quad (2.7)$$

şeklinde hesaplanmaktadır. $P_{\max}$ resimdeki piksellerin alabileceği en büyük değer olup, kullanılan bit sayısına göre değişmektedir. Piksel değeri n-bit ile ifade edilen bir resimde $P_{\max}$ değeri Eşitlik (2.8)'teki gibi bulunur.

$$P_{\max}=2^n-1 \quad (2.8)$$

Yüksek PSNR değeri benzerliğin yüksek olduğunu yani bozunumun (distortion) az olduğunu ifade etmektedir. Bozunum arttıkça PSNR değeri de düşmektedir. RGB çerçevedeki PSNR, üç bileşenin PSNR değerlerinin ortalamalarıyla bulunmaktadır. YUV video çerçevelerinde ise parlaklık ve renk bileşenleri farklı ağırlıklara sahip oldukları için ayrı ayrı bit oranı ve PSNR karşılaştırması yapmak başarımlı sonuç hakkında tam fikir vermemektedir. Bu sebeple HEVC kodlamada YUV video çerçevesinin PSNR değeri PSNR_{YUV} Eşitlik (2.9)'deki gibi hesaplanmaktadır.

$$\text{PSNR}_{YUV}=\frac{6 \cdot \text{PSNR}_Y+\text{PSNR}_U+\text{PSNR}_V}{8} \quad (2.9)$$

Burada PSNR_Y , PSNR_U ve PSNR_V ifadeleri sırasıyla video çerçevesindeki Y, U ve V bileşenlerinin PSNR değerlerini göstermekte olup, her bileşen için Eşitlik (2.7)'deki gibi hesaplanmaktadır. Video dizisinin PSNR değeri ise çerçeve PSNR değerlerinin ortalamasıyla elde edilmektedir [4]. Farklı QP değerleri için elde edilen PSNR ve bit oranı değerlerinin birlikte gösterildiği oran-bozunum grafiklerinde başarımlı değerlendirmesinde sadece bir değer üzerinden karşılaştırma yapmak doğru olmamaktadır. Bu sebeple iki farklı eğri arasındaki farkı daha iyi hesaplayabilmek için Bjøntegaard metriği [35] olarak bilinen PSNR ve bit oranının birlikte değerlendirildiği yöntem kullanılmaktadır.

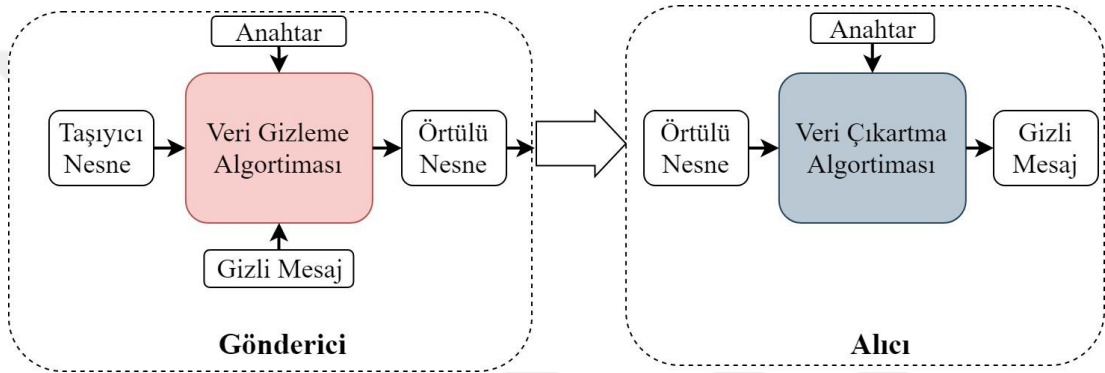
3. VERİ GİZLEME

Günümüzde internet kullanımının her yere ayılması, hücresel veri altyapısının gelişmesiyle mobil internet kullanımının artması, çeşitli medya uygulamalarına kolay şekilde erişmeyi sağlamıştır. Böyle yaygın kaynağın olması, gizli ve güvenli haberleşme yöntemlerinin daha fazla kullanılmasına da fırsat sunmaktadır. Ayrıca bilgi güvenliği yöntemleriyle, sayısal medya uygulamalarının güvenliğini sağlamak da önemli bir çalışma alanı haline gelmiştir. Steganografi, damgalama ve kriptografi gibi yöntemler bilgi güvenliği ve gizli haberleşme için kullanılan yaklaşımlardır.

Kriptografi veya şifrelemede amaç mesajı şifreleyerek alıcıya göndermektir. Üçüncü kişiler, şifreleme sonrası karmaşık bir hal alan mesajı herhangi bir şekilde anlamaz. Bu durum avantajlı gibi görünse de bazı sıkıntıları beraberinde getirmektedir. Şifreli mesajı alıcıdan önce gören üçüncü kişiler, gizli haberleşme olduğunu düşündüğü zaman, bu mesajı alıcıya gitmeden önce bozabilmektedir. Öte yandan, şifreli mesaj, kullanılan şifreleme algoritması çok gelişmiş olsa bile güçlü bilgisayarlar ve yeterli zaman ile elde edilme imkânına sahiptir. Şifrelemenin diğer bir dezavantajlı yönü ise şifreleme ve şifre çözme için güçlü algoritmalar kullandıkça harcanan zaman ve kullanılan bellek miktarlarının oldukça artmasıdır. Şifrelemedeki bu dezavantajlar daha kolay biçimde yapılabilecek güvenli haberleşme yöntemlerine yönelmiştir [36].

Veri gizleme, güvenli haberleşme için bir nesne içerisine çeşitli bilgilerin gizlenmesi için kullanılan bir yöntemdir. Özellikle askeri birimler, istihbarat örgütleri gibi legal birimlerin yanı sıra illegal örgütler de haberleşmek için veri gizleme yöntemlerine başvurumaktadırlar. Veri gizlemenin şifrelemeden ayrılan en önemli yanı, mesajı içeren nesnenin ilk bakışta hiçbir şüphe uyandırmamasıdır. Veri gizlemede gizli haberleşme ve içerik koruması olmak üzere iki farklı amaç vardır. Temel veri gizleme yaklaşımının blok yapısı Şekil 3.1’de gösterilmektedir. Mahkum problemi (prisoner problem) [37] olarak da bilinen bu yaklaşımda gönderici ve alıcı arasında yapılan gizli bir haberleşme bulunmaktadır. Göndericinin yolladığı tüm nesneler iletişim ortamındaki gardiyan tarafından kontrol edilmektedir. Gardiyan örtülü nesnede

herhangi bir şüpheli durumu gördüğünde, onu bozarak mesajın alıcıya gitmesine engel olmaktadır. Dolayısıyla göndericinin alıcıya, gardiyanın fark edemeyeceği ve şüphelenemeyeceği biçimde gizli mesaj iletmek için bir veri gizleme algoritması kullanması gerekmektedir. Bu algoritmayı kullanarak gizli mesajı (secret message) taşıyıcı bir örtü nesnesinin (cover object) içerisine gizledikten sonra elde edilen örtülü nesne (covered object), iletim ortamından alıcıya göndermektedir. Alıcı iletim ortamından aldığı örtülü nesneyi, daha önceden bildiği veri çıkartım algoritmasından geçirip, gizli mesajı elde etmektedir. Veri gizlemede haberleşmenin güvenliğini arttırmak için ayrıca bir anahtar kullanılabilir.



Şekil 3.1. Temel veri gizleme yaklaşımı

3.1. Veri Gizleme Türleri

Veri gizlemenin alt dallarından olan steganografi gizli haberleşme için kullanılırken, damgalama ise içerik koruması için kullanılmaktadır. Görünmez haberleşme sanatı olarak da adlandırılan steganografi terimi, Yunanca kökenli olup, örtülü veya gizli anlamındaki “steganos” ve yazı anlamındaki “graphia” kelimelerinin birleşiminden meydana gelmektedir. Türkçe’de, gizli yazı [2] ve sır örtme [38] olarak da kullanılan steganografideki örtülü nesne için stego nesne (stego object) ifadesi de kullanılmaktadır. Ülkelerin askeri birimleri, istihbarat örgütleri gibi resmi kurumların yanı sıra illegal yapıların bir çoğu da steganografi ile haberleşme yöntemlerini kullanmaktadır. Başarılı bir veri gizleme çalışmasında mesaj kapasitesi, fark edilmezlik ve saldırılara karşı dayanıklılık en önemli üç parametre olarak öne çıkmaktadırlar [39]. Steganografi yöntemlerinde genellikle diğer iki parametre değerlendirildikten sonra saldırılara dayanıklılığa bakılmaktadır. Gizli mesajı taşıyan örtü nesnesindeki değişimler insan görsel sistemiyle veya temel özneliliklerin

kontrolüyle anlaşılırsa, haberleşmeyi takip eden kimseler tarafından bu mesajın iletilmesi engellenecek veya örtülü nesne saldırıya uğrayarak mesaj içeriği bozulacaktır. Dolayısıyla steganografide esas olan yüksek kapasite ve fark edilmezliktir, onun için gizli mesaj kapasitesini azaltmaksızın, stego nesnenin değişiminin mümkün oldukça düşük seviyede tutulması gerekmektedir.

Veri gizlemenin diğer bir türü olan damgalamada (watermarking) amaç örtü nesnesinin korumaktır. Telif hakkı koruması, kopyalama kontrolü, içerik doğrulama, sahiplik bilgisini ispatlama gibi amaçlarla yapılan damgalamada, gizli mesaj olarak kullanılan filigran veya damga (watermark) fark edilmez olmak zorunda değildir. Burada asıl amaç iletim ortamında yapılacak bozucu saldırılarda damganın bozulmamasıdır. Dolayısıyla damgalamada korunmak isteyen medya içerisine hak sahibine ait, gözle görülür ve görülmez biçimde bir damga eklenmektedir. Eklenen damganın saldırılar karşısında dayanıklı olması ve taşıyıcı içerisinden geri elde edilebilmesi damgalamanın en önemli önceliklerinden birisidir. Güvenlik amaçlı bazı uygulamalarda ise taşıyıcı nesneye en ufak bir müdahale olduğunda damganın bozulması amaçlanmaktadır. Böylece haberleşme kanalında en ufak bir müdahale durumunda içeriğe erişim engellenmektedir. Damgalama son yıllarda özellikle ticari içeriklerin korunması için başvurulan bir yöntem olmuştur. Örneğin sinema salonlarında oynatılan filmlerin dışarı sızmasını önlemek için her salonda oynatılan kopyaya ayrı damgalar eklenebilmektedir. Benzer olarak, bireysel üyelik usulüyle satış yapan maç yayın şirketleri, canlı yayınlanan maçların belli zamanlarında damgalar kullanarak, herhangi bir korsan yayının hangi kullanıcı tarafından yapıldığını tespit etmeye çalışmaktadırlar. Damgalamanın önemli özellikleri, orijinal dosyaya ihtiyaç duymaksızın hızlı gizlenmesi ve hızlı çıkarılması, dayanıklı olması ve bazı uygulamalarda görünmez olmasıdır. Literatürde normal görüntüler [40-44], tıbbi görüntüler [45], tıbbi uygulamalar [46], ses uygulamaları [47, 48], metin [49] ve e-devlet uygulamaları [50] için çeşitli damgalama çalışmaları yapılmıştır.

Veri gizleme için taşıyıcı örtü nesnesi olarak birçok farklı sayısal dosya çeşidi kullanılabilir. Burada temel amaç, gizleme kapasitesini arttıran ve değişimleri en düşük düzeyde tutan örtü dosyaları seçmektir. Metin, ses, internet protokol paketleri, görüntü ve video dosyaları genellikle kullanılan örtü nesneleridir.

Metin içerisinde gizli mesaj iletme en eski veri gizleme yöntemlerinden biridir. Belli sıradaki kelimelerin veya harflerin seçilmesi, harf kaydırma, satır kaydırma veya daha önceden belirlenen bir örüntüye göre metin içindeki bölgelerin seçilmesiyle yapılabilmektedir. Metin içerisinde gizleme kapasitesi düşük olmakla beraber günümüzde farklı yazı dilleri için veri gizleme yöntemleri mevcuttur [51-54].

Ses içerisinde yapılan veri gizlemede, sesin düşük frekans bileşenleri kullanılmaktadır. İnsan kulağının düşük değişimlere karşı hassas olması seste yapılan gizlemelerin en kritik noktasıdır. Genellikle tek boyutlu sinyal olarak alınan ses içerisinde faz kodlama, düşük bit kodlama, yayılı spektrum yöntemi, yankı gizleme yöntemi, ton ekleme yöntemi ve gürültü ekleme yöntemi kullanılmaktadır [55, 56].

İnternet protokol paketlerini kullanan veri gizleme yöntemleri genellikle ağdan iletilecek olan paketlerin başlık bilgilerine gizli mesajlar eklemektedirler [57]. Ayrıca web sitelerinin kaynak kodlarındaki boşluklara ve etiket bilgilerine veri gizleyen yöntemlerde mevcuttur [58].

3.2. Görüntülerde Veri Gizleme

Görüntüler, veri gizlemede en çok kullanılan örtü dosyalarıdır. Gerek önceki türlere göre sundukları yüksek kapasite, gerek üzerlerinde daha kolay işlem yapılabilme özellikleri görüntüleri popüler hale getirmiştir. Görüntüleri taşıyıcı olarak kullanan veri gizleme çalışmaları uzamsal düzlem veya dönüşüm düzleminde yapılmaktadır. Uzamsal düzlem yöntemleri kolay uygulanabilmekle birlikte, saldırılara veya bozucu etkilere karşı daha hassastır. Dönüşüm düzleminde yapılan veri gizlemeler ataklara karşı daha dayanıklıdır fakat gizleme kapasiteleri uzamsal düzlem yöntemlerine göre daha az olabilmektedir. Görüntülerde uzamsal düzlemde veri gizlerken, en düşük değerlikli biti (least significant bit, LSB) değiştirme, piksel değer farkı (pixel value differencing, PVD) [59], değişim yönü kullanımı (exploiting modification direction, EMD) [60], histogram kaydırma gibi yöntemler öne çıkmaktadır [61]. Bu yöntemler görüntünün sadece parlaklık kanalına uygulandığı gibi tüm renk kanallarına da uygulanabilmektedir. Dönüşüm düzleminde yapılan veri gizleme çalışmalarında genellikle DCT [62, 63] ve dalgacık dönüşümü [64] temelli yöntemler öne çıkmaktadır.

İnsan görsel sisteminin görüntülerdeki düşük değişimleri algılamaması esasına dayanan LSB yönteminde, ikilik sayıya çevrilen gizli mesajın her bir biti 8-bitlik gri piksellerdeki en anlamsız bitlerle değiştirilmektedir. Örneğin örtü dosyasının parlaklık değerlerine, gizli mesaj olarak “z” harfinin gizleneceğini varsayalım. Bilgisayar dilinde “z” harfinin ikilik sayı karşılığı “01111010” şeklindeki sekiz bit olmaktadır. Bu bitlerin her birisini örtü dosyasındaki 8-bitlik bir pikselin en anlamsız bitiyle değiştirmek yeterli olmaktadır. Böylece “z” harfi görüntüdeki toplam sekiz pikselin en anlamsız bitlerine gizlenmiş olur. Alıcı tarafta ilgili sekiz pikselin, en anlamsız bitleri sırayla çıkartılarak, gizli mesaj geri elde edilebilmektedir. Bu yöntem, görüntü piksellerinde gözle görülmeyecek şekilde küçük değişimlere sebep olmaktadır. Kullanılan bu yöntemin güvenliğini arttırmak için, gizli mesaj bir anahtar yardımıyla şifrelendikten sonra örtü dosyasına gizlendiğinde, haberleşme kanalında LSB değerlerini kontrol eden birisi herhangi anlamlı bir mesaj bulamamaktadır.

LSB ile yapılan veri gizleme çalışmalarında, gizleme kapasitesini arttırmak için çeşitli yöntemler geliştirilmiştir. Örneğin [65] çalışmasında gizli mesajın her bir karakteri 24-bitlik RGB resimdeki bir piksele gizlenmiştir. Bu amaçla gizli karakterin sekiz bitinin, üç biti R bileşeni, üç biti G bileşeni, iki bit ise B bileşeni olmak üzere piksellerin en anlamsız bitlerine gizlenmiştir. Bu yöntemde görüntü kalitesi daha fazla değişmekle birlikte, gizli mesaj kapasitesi artmaktadır. Çünkü normal LSB yönteminde bir karakter için sekiz piksel kullanılırken bu yöntemde sadece bir piksel kullanılmaktadır.

Görüntülerde veri gizleme için kullanılan bir diğer yöntem olan histogram dağılımı kullanma ise daha çok tersinir (reversible) veri gizleme amacıyla kullanılmaktadır. Tersinir veri gizleme yöntemlerinde, alıcı gizli mesajı elde ettikten sonra stego dosyayı kullanarak örtü dosyasını geri elde edilebilmektedir. [66] çalışmasında önerilen histogram dağılımı yönteminde, görüntüdeki histogram dağılımındaki tepe değeri ile 254 arasındaki değerler bir sağa kaydırılarak, tepe noktasının yanı boşaltılmıştır. Pikseller sırayla taranarak, tepe değerine denk gelince, gizli mesaj 0 ise tepe değeri aynı tutulmuş veya gizli mesaj 1 ise tepe değeri bir arttırılmıştır. Tepe noktası bilgisinin de ayrı bir yere gizlendiği stego görüntü alıcıya gönderilmektedir. Alıcı sırayla taradığı görüntüden tepe değerine denk gelen yerlerden 0, tepe değerinin bir fazlası olan değerden ise 1 gizli mesajını çıkarmaktadır. Ayrıca tepe noktasının solundaki değerler tekrar bir azaltılarak piksellere eski haline dönüştürülmektedir.

Gizli mesaj elde edildikten sonra, kaydırılan tüm histogram değerleri de eski haline getirilerek orijinal örtü görüntüsü geri elde edilmektedir. Histogram yöntemlerinin en önemli dezavantajları, gizleme kapasitelerinin tepe noktası değeriyle sınırlı olması ve alıcıya tepe değeri bilgisini gönderme zorunluluğudur [67]. Bunları gidermek için [68] çalışmasında tepe noktası yerine, sağındaki değer kullanılırken, [69] çalışması kapasiteyi arttırmak için çoklu seviye histogram kullanmıştır.

Görüntü veri gizlemenin kullanıldığı diğer bir alan da tıbbi görüntülerdir. Gelişen teknolojiyle birlikte çevrimiçi sistemler üzerinden hastalarla veya başka doktorlarla paylaşılan tıbbi görüntü sayıları da artmıştır. Hem hastalara hem doktorlara büyük kolaylıklar sağlayan bu sistem önemli bir güvenlik problemini de beraberinde getirmektedir. Kişilerin hastalık bilgilerini içeren görüntülerin istenmeyen kişilere karşı korunması ihtiyacının yanı sıra birçok sağlık kuruluşu telif hakkı ve dijital imza gibi bilgileri tıbbi görüntülere eklemek istemektedir. Tıbbi veri gizleme ve damgalama bu amaç için kullanılan çözümlerin başında gelmektedir [70]. Tıbbi görüntülerde veri gizleme için önerilen yöntemlerde LSB tabanlı [71], DCT tabanlı [72], dalgacık tabanlı [73], histogram tabanlı [74] ve komşu piksellerin fark genişlemesi (difference expansion, DE) tabanlı [75] veri gizleme çalışmaları öne çıkmaktadır.

3.3. Videolarda Veri Gizleme

Yüksek miktarda uzamsal ve zamansal kapasite sunmalarından dolayı veri gizleme uygulamalarında kullanılan önemli taşıyıcılardır. Son yıllarda artan video paylaşımı ile birlikte videolara gizli veri ekleme işlemi popüler hale gelmiştir. Videolarda veri gizleme genel olarak sıkıştırılmış videolarda veya ham videolarda veri gizleme şeklinde iki başlıkta incelenmektedir. Ham videolarda veri gizleme işleminde genellikle herhangi bir sıkıştırma işlemi yapılmamış videolar kullanılmaktadır. Bu yöntemde video çerçevelerine doğrudan gizli mesaj eklenmektedir.

3.3.1. Ham videolarda veri gizleme

Önceki bölümlerde bahsedildiği gibi videolar ardışık resimlerden meydana gelir. Dolayısıyla ham videolarda veri gizleme işlemi görüntülerdeki yöntemlere benzer biçimde yapılmaktadır. Bunun için video, öncelikle çerçevelere dönüştürülmektedir. Daha sonra her bir çerçeve bağımsız birer görüntü gibi değerlendirilerek veri gizleme

işlemi yapılmaktadır. Gizleme işlemi bittikten sonra, çerçeveler tekrar bir araya getirilerek stego video oluşturulmaktadır [76].

Ham videolara veri gizleme için uzamsal düzlemi kullanan yöntemler incelendiğinde LSB temelli yöntemler dikkat çekmektedir. [77] çalışmasında 8 bitlik bir karakter gizlemek için her bir video çerçevesindeki piksellerin RGB renk kanallarından R kanalının son 3 bitini, G kanalının son 3 bitini ve B kanalının son 2 bitini kullanmıştır. [78] çalışmasında ise video çerçevesindeki pikselin RGB kanallarının her birinin son 2 bitine veri gizlenmiştir. [79] çalışmasında ise, video çerçevesi içerisinde Knight Tour algortması ile seçilen piksellere LSB yöntemiyle veri gizlenmiştir. [80]'deki çalışmada Kanade-Lucas-Tomasi takip algoritması ile yüz bölgesi tespit ve takip edilerek ilgi bölgesi tanımlanmıştır. Daha sonra video çerçevelerindeki yüz bölgesinin RGB bileşenlerine veri gizlenmiştir. [81] çalışmasında ise, video çerçevesinde yer alan insan vücudunun cilt bölgeleri tespit edilerek ilgili kısımlara dalgacık kuantalama tekniği ile veri gizlenmiştir. [82] çalışmasında ise, video çerçevelerindeki hareketli nesneleri tespit edilip, hareketli nesnelerin olduğu bölgelere gizli veri eklenmiştir.

Literatürde yapılan bazı çalışmalarda, video dosyasının tamamına veri gizlemek yerine belirli çerçeveler seçilip çerçevelere veri gizlenmiştir. [83]'de sahne değişimine göre bir video çerçevesi seçmiştir. [84]'teki çalışmada video bloklara ayrılmaktadır. Veri gizlenecek çerçevelerin seçimi, blok boyutu, gizli mesaj boyutu ve video boyutlarına bağlı olarak yapılmaktadır. [85]'te hızlı bir şekilde çerçeve seçmek için gradyan büyüklük benzerlik sapması (gradient magnitude similarity deviation) yöntemi kullanılmıştır. [86]'da belirlenen bir indeks yardımıyla bir çerçeve seçilmiştir. [87]'deki ham videodaki istatistiksel özelliklere göre anahtar çerçeve seçilmiştir.

Ham videolarda dönüşüm düzlemi yöntemleriyle veri gizleme yapılan çalışmalarda, çerçevelerin dönüşüm katsayıları elde edildikten sonra bu katsayılara veriler gizlenmektedir. Daha sonra ters dönüşümle yeniden oluşturulan çerçeveler videoya dâhil edilmektedir. [88] çalışması videodaki her çerçeveye sırayla ayrı dalgacık dönüşümü (discrete wavelet transform, DWT) uygulayıp, düşük alt frekans bileşenlerine bir damga gizlemiştir. Çerçevelere DWT uygulayan diğer bir çalışmada [89] ise sınıflandırma yöntemleriyle seçilen bloklar kuantalanmış ve gizli bir damga kuantalanmış katsayılar içerisine eklenmiştir. [90]'da YUV videoya DWT uygulanmış

ve uygun bir alt bant seçilmiştir. Hem seçilen alt banda hem gizlenecek damgaya Hilbert dönüşümü uygulanarak gizli veri çerçeve içerisine eklenmiştir.

3.3.2. Kodlanmış videolarda veri gizleme

Kodlanmış videolarda veri gizleme için genellikle video sıkıştırma formatları kullanılmaktadır. Ham videolardan farklı olarak veri gizleme işlemi video kodlama sürecinin bir parçası olarak yapılmaktadır. Haberleşme sürecinde, sıkıştırılmış video bilgilerinin kullanılmasından dolayı, veri gizleme için kodlanmış videolar daha çok tercih edilmektedir. Bununla birlikte bu şekildeki veri gizleme çalışmaları, doğrudan çerçevelere yapılan gizleme yöntemlerine göre daha zahmetlidir. Bunun en büyük sebeplerinden birisi kullanılacak video kodlayıcı yapısına hâkim olmanın da gerekmesidir. Aksi durumda, yapılan gizleme işleminin video yapısında kalıcı bozulmalara sebep olabileceği öne çıkan önemli bir husustur. Kodlanmış videolarla yapılan çalışmaların büyük kısmı örtü dosyası olarak MPEG-2, MPEG-4 ve H.264/AVC kullanılmıştır. Yakın dönemde yapılan çalışmalarda ise H.265/HEVC videolarını örtü dosyası olarak kullanan yöntemler çoğalmıştır.

MPEG-2 videolarında, doğrudan bit dizisine, DCT katsayılarına ve hareket vektörlerine gizleme yapılmış çalışmalar bulunmaktadır. [91] çalışmasında video bit dizisine değişken uzunluklu kod kelimeleri ile veri gizlenmiştir. [92] çalışmasında DCT katsayılarının içerisine gizli mesajlar eklenmiştir. [93] çalışmasında ise hareket vektörlerinin faz açılarında değişiklikler yapılarak mesaj gizlenmiştir.

MPEG-4 videolarında yapılan çalışmalarda DCT katsayıları ve hareket vektörleri kullanımı öne çıkmıştır. [94] çalışmasında MPEG-4 videolarda rastgele seçilen makrobloklardaki DCT katsayılarından orta frekans bölgesinde olanlara veri gizlenmiştir. [95] çalışmasında hareket vektörleri özelliklerine göre ikiye ayrıldıktan sonra, gizli mesajlar seçilen vektörlerin içerisine eklenmiştir.

Örtü dosyası olarak H.264/AVC videoları kullanan veri gizleme çalışmalarında, kodlama işleminin farklı aşamalarına müdahaleler edilmiştir. Çerçeve-içi ve çerçeveler-arası kestirimlerde kestirim modları, blok yapıları ve hareket vektörlerine müdahaleler; dönüşüm katsayılarında veya kuantalama parametrelerinde değişiklikler ve entropi kodlama aşamasındaki kod yapılarında değişiklikler yapmak gibi çeşitli yöntemlerle videoların içerisine gizli mesajlar eklenmiştir [96]. Gizlenecek bite göre

4x4 bloklar arasında 4x4 makrobloğun kestirim modu değiştirmeyi öneren [97] çalışması çerçeve-içi bloklar ile gizli bitler arasında eşleştirme tabloları kullanmıştır. [98] çalışmasında ise en iyi kestirim modu ile gizli bit arasında ilişki kurularak veri gizlenmiştir. Bir başka çalışmada [99] entropi kodlama aşamasında CAVLC içerisindeki bayrakları ve seviyeleri değiştirerek bilgi gizlenmiştir. Hareket vektörlerinin değiştirilmesini öneren [100] çalışması geri dönüştürülebilir veri gizleme özelliğini sağlamak için histogram tabanlı gizleme ile değiştirilmiştir. Bu amaçla hareket vektörlerinin yatay ve düşey değerleri görüntülerde veri gizlemek için kullanılan histogram kaydırma yöntemiyle değiştirilmiştir. AVC videolarında makrobloklar, daha önce kodlanan blokların değerlerinden kestirildiği için I-çerçevedeki dönüşüm katsayılarına yapılan veri gizlemeler, makrobloklar arası hata yayılımına sebep olmaktadır. Bu problemi aşmak için [101] çalışmasında, hata yayılımını önleyen bir yöntem geliştirilmiştir. [102] çalışmasının çeşitli eklemelerle iyileştirdiği bu yöntemde, blok içerisindeki DCT katsayılarına gizlenen mesajın komşu bloklar üzerine hata yayması önlenmiştir. Bu amaçla, komşu blokların kullandığı kestirim moduna göre, geçerli bloğun en sağ sütununu veya en alt satırını değiştirmeyecek katsayılarda değişiklikler yapılmıştır.

3.4. Steganaliz

Haberleşme kanalında üçüncü kişilerce, stego nesnelerdeki gizli mesaj varlığının kontrol edilmesi ve tespiti, mümkün ise gizli mesajın stego nesneden geri elde edilmesi veya gizli mesajı bozmak için steganaliz (steganalysis) kullanılmaktadır. Steganalizin amacı veri gizleme uygulamalarının illegal haberleşmedeki amaçlarını ortadan kaldırmak veya geliştirilen uygulamaların dayanıklılığını test etmektir. Steganaliz işleminde veri dosyasında görsel veya istatistiksel olarak değişen kısımların nitelikleri kontrol edilmektedir. Son yıllarda internet üzerinden suç teşkil eden faaliyetler için denetim talebi ciddi artış göstermiştir. Bu kapsamda önleyici güvenlik tedbirleri almak adli bilişim alanının önemli araştırma geliştirme konularından birisi olmuştur [2].

Steganalizi yapan kişinin steganografi uygulama araçlarının yöntemleri hakkında detaylı bilgi sahibi olması doğru işlem yapması açısından önemlidir. Etkili bir steganaliz için bu alanlardaki yeni çalışmaların ve eğilimlerin daima dikkate alınması gerekmektedir. Genel olarak bakıldığında, steganaliz aslında steganografi uygulamalarına karşı saldırı olarak değerlendirilmektedir. Steganalizin hedefi,

gizlenen mesajın olup olmadığına dair çeşitli kanıtlar elde etmek ve gizli haberleşmeye karşı yöntemler geliştirmektir. Bu amaçla gizli mesajı taşıma haberleşme kanalında tespit ettikten sonra mümkünse gizli mesajı çıkartarak elde etmek, mümkün değilse de stego dosyayı bozarak haberleşmeyi kesmektir. Esasen sadece gizli veri varlığının tespiti bile veri gizlemenin en önemli amacı olan gizli haberleşmeyi önlemeye yetmektedir.

Steganaliz yöntemleri, [103] çalışmasında imza analizi ve istatistiksel analiz olarak iki kısma ayrılmıştır. Veri gizleme ile oluşturulan stego medyadaki değişimler genelde insan gözünün hissedemeyeceği kadar düşüktür. Buna rağmen kullanılan bazı gizleme yöntemleri tüm örtü dosyalarının özelliklerinde aynı bozulmalara veya özel değişimlere yol açmaktadırlar. Bu özel değişim ve bozulmalar veri gizleme yönteminin imzası olarak adlandırılmaktadır. İmza tabanlı steganaliz yapan kişi, bu imzanın ne olduğunu tespit ederse, haberleşme kanalındaki tüm medyalarda bu imzayı arayabilir. Bu imzaya sahip olan medyaların gizli mesaj taşıdığı aşikâr olduğundan, bu medyalar tespit edildiğinde, alıcıya gitmesi engellenebilir veya farklı bir tedbir uygulanabilir.

İstatistiksel steganalizde ise gizli veri varlığını tespit için stego dosyanın istatistiksel özelliklerindeki değişimlere bakılmaktadır. Böylece herhangi bir karmaşık işlem yapmadan gizli veri tespit edilebilmektedir. İstatistiksel steganaliz uygulamaları matematiksel yaklaşımlara dayandığı için imza steganaliz çözümlerine göre daha güvenilir ve daha doğru sonuçlar vermektedir.

3.5. Veri Gizleme Başarım Değerlendirmesi Yöntemleri

Veri gizleme yöntemlerinde veri gizleme sonucu elde edilen stego dosyaların kalite kontrolü için öznel (subjective) yöntemler ve nesnel (objective) yöntemler olmak üzere çeşitli yöntemler kullanılmaktadır. Öznel değerlendirme yöntemleriyle, gizlenen mesajın stego dosyada oluşturduğu değişikliklerin insan duyularıyla algılanabilir olup olmadığı değerlendirilmektedir. Ses dosyası kullanan yöntemlerde öznel değerlendirme insan kulağıyla yapılırken, metin, görüntü, video gibi dosyaları kullanan yöntemlerde insan görsel sistemiyle değerlendirme yapılmaktadır. Bu amaçla sadece stego dosya kontrol edilebildiği gibi, orijinal ve stego dosyaların karşılaştırmalı değerlendirmeleri de yapılabilmektedir.

Veri gizleme yöntemleriyle elde edilen stego dosyaların nesnel değerlendirilmesi için PSNR, MSE, SSIM (Structural Similarity Index) [104] ve veri gizleme kapasitesi gibi çeşitli parametreler değerlendirilmektedir. SSIM resim veya video çerçevelerindeki parlaklık (luminance), karşıtlık (contrast) ve yapı (structure) bileşenleri üzerinden aradaki benzerliğe bakan bir kalite ölçütüdür. Orijinal görüntü çerçevesi f_i , veri gizlenmiş stego çerçevesi f_o olmak üzere SSIM Eşitlik (3.1)'deki gibi hesaplanmaktadır.

$$SSIM(f_i, f_o) = \frac{(2\mu_i\mu_o + C_1)(2\sigma_{io} + C_2)}{(\mu_i^2 + \mu_o^2 + C_1)(\sigma_i^2 + \sigma_o^2 + C_2)} \quad (3.1)$$

Burada μ_i ve μ_o ortalama, σ_i^2 ve σ_o^2 varyans ve σ_{io} kovaryans değerlerini göstermektedir. C_1 ve C_2 parametreleri parlaklık değerine bağlı sabit katsayıları ifade etmektedir. Benzerliğin en yüksek olduğu durumda SSIM, 1 değerini alırken, benzerlik düştükçe SSIM değeri sıfıra doğru düşmektedir.

Herhangi bir veri sıkıştırma işleminin kullanılmadığı veri gizleme yöntemlerinde, PSNR ve SSIM değerlerinin mümkün oldukça yüksek çıkması, MSE değerinin de düşük olması, veri gizleme sonucunda nesnel kalitenin yüksek olduğu anlamına gelmektedir. Kodlanmış videolarda ise durum biraz daha farklıdır. Ham YUV video herhangi bir gizleme olmaksızın doğrudan kodlandığında elde edilen PSNR değeri $PSNR_O$ ve aynı YUV video veri gizleme sonrası kodlandığında elde edilen PSNR değeri $PSNR_S$ olsun. Veri gizleme yönteminin video kalitesinde ne kadar bozucu etki yaptığını Eşitlik (3.2)'deki PSNR farkı ($\Delta PSNR$) ile değerlendirilmektedir. $\Delta PSNR$ değerinin sıfıra çok yakın negatif değerlere sahip olması, veri gizleme yönteminin örtü dosyasını çok bozmadığını göstermektedir. Kodlanmış videolarda $\Delta PSNR$ değeri pozitif olabilmektedir. Bunun sebebi PSNR kıyasının YUV video çerçeveleri ile kodlanmış video çerçeveleri arasında yapılmasıdır. Dolayısıyla veri gizleme sonrası kodlanan video çerçeveleri, veri gizleme olmadan kodlanan video çerçevelerine göre orijinal YUV videoya daha fazla benzerlik gösterebilir.

$$\Delta PSNR = PSNR_S - PSNR_O \quad (3.2)$$

Veri gizleme yönteminin kapasitesi değerlendirilirken iki farklı yaklaşım kullanılmaktadır. İlk yaklaşımda, yöntemin maksimum gizleme kapasitesi, doğrudan

örtü dosyasına gizlenen bit sayısı ile verilmektedir. İkinci yaklaşımda ise, piksel başına düşen gizli bit miktarı (bit per-pixel, bpp) ile veri gizleme kapasitesinin verimliliği değerlendirilmektedir.

Kodlanmış videolarda veri gizleme sonrası bit oranı artışının mümkün oldukça az olması gerekmektedir. Dolayısıyla veri gizleme olmaksızın elde edilen bit oranı bitrate_o ve veri gizleme sonrası elde edilen bit oranı bitrate_s olmak üzere yüzdelik bit oranı artışı Eşitlik (3.3)'teki gibi elde edilmektedir.

$$\Delta\text{bitrate}(\%) = \frac{\text{bitrate}_s - \text{bitrate}_o}{\text{bitrate}_o} \times 100 \quad (3.3)$$

Veri gizleme yönteminin kendi içinde değerlendirilmesinde faydalı olan bu yöntem literatür yöntemleriyle kıyaslamada pek adil olmamaktadır. Çünkü bit oranı artışı doğrudan gizleme kapasitesiyle bağlantılıdır. Gizlenen veri miktarı arttıkça bit oranı da artmaktadır. Bit oranı kıyaslamasının adil bir biçimde yapılabilmesi için [105] çalışmasında bit hızındaki artış oranı (bitrate increase ratio, BIR) parametresinin kullanılması önerilmiştir. Videoda gizlenen bit miktarı E_c olmak üzere, BIR değeri Eşitlik (3.4)'teki gibi hesaplanmaktadır.

$$\text{BIR} = \frac{\Delta\text{bitrate}}{E_c} = \frac{\text{bitrate}_s - \text{bitrate}_o}{E_c \times \text{bitrate}_o} \times 100 \quad (3.4)$$

BIR değeri ne kadar düşük olursa gizleme başarımı da o kadar iyi demektir. Birçok yöntemde E_c , $\Delta\text{bitrate}$ değerine kıyasla oldukça büyük olduğu için BIR değeri çok düşük ondalık değerlerde bulunmaktadır. Bu sebeple, bulunan değerlerin daha okunabilir olması amacıyla BIR yerine $\text{BIR}(\times 10^{-3})$ gösterimi kullanılmaktadır.

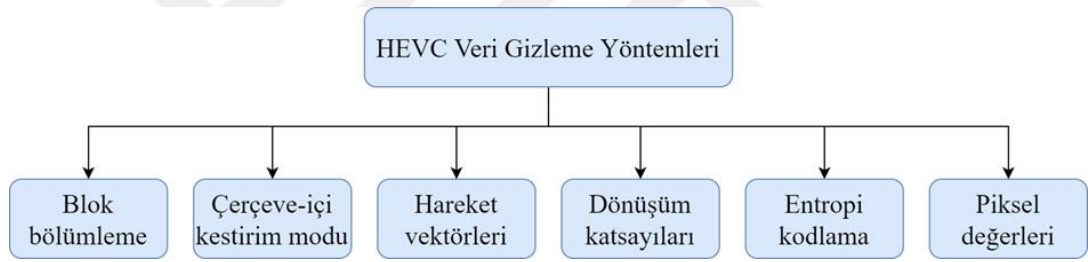
Son olarak, klasik veri gizleme yöntemi olan steganografinin kurgusunda stego nesnenin herhangi bir saldırıya veya bozulmaya maruz kalmayacağı varsayılır. Dolayısıyla steganografi yöntemlerinde gömü verisi doğru çıkması gerektiği için, gömü verisinin kalitesi tekrardan değerlendirmeye ihtiyaç yoktur. Fakat damgalama yöntemlerinde örtü dosyası içerisine gizlenmiş mesajın başarımlı değerlendirilmesi için bazı yöntemler bulunmaktadır. Çünkü damgalama yöntemlerinin iletişim ortamında gürültü, sıkıştırma, kırpma gibi çeşitli saldırılara maruz kalacağı düşünüldüğünden normalize karşıt korelasyon (normalized cross correlation, NCC) ve bit hata oranı (bit

error rate, BER) deęerleri kullanılarak gm verisindeki kayıp oranları hesaplanmaktadır. Gndericinin gizledięi mesaj ile alıcıda ıkarılan mesajlar arasındaki hatanın bit sayısının tespiti iin BER ve mesajların benzerlik oranı iin NCC en sık deęerlendirilen parametrelerdir [2].



4. HEVC VERİ GİZLEME ÇALIŞMALARI

Veri gizlemede taşıyıcı örtü dosyası olarak HEVC videolarını kullanan çalışmalar sınırlıdır. Yapılan çalışmalarda, veri gizlemeyi gerçekleştirmek için farklı yöntemler kullanılmıştır. Bu yöntemlerde, blok bölümlleme, çerçeve-içi kestirim modu, hareket vektörleri, dönüşüm katsayıları ve entropi kodlama aşamalarında değişiklikler yapılmıştır. Bazı çalışmalarda ise doğrudan video çerçevelerindeki piksellere veri gizlemiştir. Şekil 4.1’de HEVC videolarında veri gizleme yöntemleri sınıflandırılmaktadır. Literatürde HEVC videolarında veri gizleme çalışmalarının yanı sıra sınırlı sayıda steganaliz çalışması da yapılmıştır. Bu yöntemler aşağıda alt bölümler halinde özetlenmektedir.



Şekil 4.1. HEVC videolarında veri gizleme yöntemleri

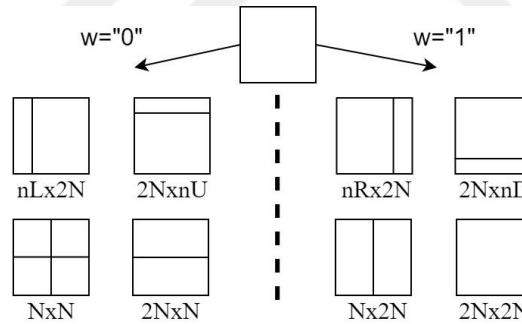
4.1. Blok Bölümlleme Yapısını Değiştirme

Blok bölümlleme yapısını değiştiren çalışmalar genellikle CTU içerisindeki CU veya PU yapısını değiştirerek veri gizlemiştir. PU yapısını değiştiren çalışmalar çerçeve-içi veya çerçeveler-arası blokları kullanmışlardır.

[106] çalışmasında CU boyutunu adaptif olarak değiştirmeye dayalı bir veri gizleme yapılmıştır. HEVC’de PU blok boyutları hesaplanırken kullanılan oran-bozunum optimizasyonu yerine önceden tanımlı eşleştirme kurallarına göre gizli mesajla uyumlu blok boyutları seçilmektedir. Bu çalışmada [106], gizlenecek verinin 0 veya 1 olması durumuna göre PU boyutlarını değiştirmektedir. I-çerçeve için en düşük blok boyutu 4×4 veya 8×8 oluyorken, P ve B-çerçeveleri için blok boyutu gizli bilginin 0 veya 1 olması durumuna göre adaptif olarak ayarlanmıştır. Daha sonra ise sıfırdan

farklı olan DCT katsayılarının tek-çift olması duruma ve TU boyutuna göre LSB değiştirme yöntemiyle veri gizlenmiştir.

[107]'de veri gizleme teknikleri kullanılarak HEVC videoları için kimlik doğrulama yöntemi önerilmiştir. Çalışmada sözde rastgele numara üretici ile 512-bitlik gizli anahtar üretilmiştir. Sonraki aşamada sıfırdan farklı DCT katsayıları gizli bir anahtar ile değiştirilerek her CTU için katsayılar üretilmekte ve her bir CTU için bir kısım istatistikler hesaplanmaktadır. Gizlenecek verinin (w) 0 veya 1 olma durumuna göre blok boyutları Şekil 4.2'deki gibi sırasıyla $2N \times N$, $2N \times uN$, $nL \times 2N$, $N \times N$ veya $N \times 2N$, $2N \times nD$, $nR \times 2N$, $2N \times 2N$ şeklinde değiştirilmektedir. Bu işlemlerden sonra birinci ve ikinci aşama doğrulama yapılmaktadır. Kimlik doğrulama için yapılan birinci aşama için, önerilen CU boyut seçimi kullanılarak 32 bitlik doğrulama kodları belirlenmiştir. Doğrulama şemasını güçlendirmek için kullanılan ikinci aşamada ise, katsayı üretme aşamasının benzeri bir yöntemle doğrulama kodları tespit edilmektedir. Videoda herhangi bir müdahale veya değişiklik olması durumunda blok yapılarında bozulma olacağı için, değişimin kolayca tespit edileceği öngörülmüştür.

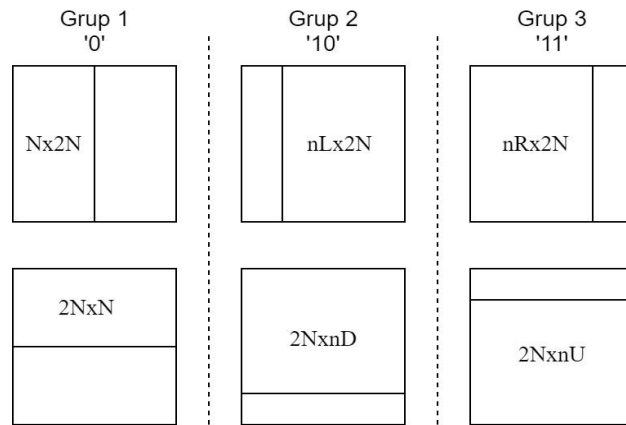


Şekil 4.2. Gizlenecek mesaj (w) verisine göre CU boyutları [107]

[108] çalışmasında P-çerçevelerindeki PU bloklarının bölünme modlarına dayalı bir veri gizleme önerilmiştir. Bu yöntemde ilk olarak değişecek moddan kaynaklanan hata miktarının en az düzeyde kalması için farklı PU modları için maliyet analizi yapılmaktadır. Yöntemde, gizli bit dizisinde üç farklı ihtimalde sayılar olacağı düşünüldükçe işlem yapılmıştır. Bit dizisinde ardışık olarak gelen bitler 0, 10 ve 11 şeklinde 3 grupta ifade edilmiştir. 01 değeri farklı bir bakış açısıyla ele alınarak, 0 ayrı grup, 1 ise kendisinden sonra gelen bitle birlikte gruplanmıştır. Çalışmanın ana mantığı, mevcut PU bölünme şekli yataysa, yatayda; dikeyse, dikeyde gizli bilgiye

göre tanımlanan bölünme biçimlerinden birine değiştirmektir. Gizlenecek gizli veri 0 ise PU modu $2N \times N$ veya $N \times 2N$ olarak, 10 ise PU modu $2N \times uN$ veya $nL \times 2N$, 11 ise PU modu $2N \times nD$ veya $nR \times 2N$ olarak değiştirilmiştir. Blok yapısının bozulmasını en düşük düzeyde tutmak için yatay modlar ($2N \times \dots$) kendi aralarında, dikey modlar ($\dots \times 2N$) kendi aralarında değiştirilmiştir. Veri çıkartım aşamasında sadece P çerçevesindeki PU bölünme modları kontrol edilerek gizli bitler geri elde edilmiştir.

[109] çalışması, P-çerçevelerindeki PU bloklarının bölünme modlarına dayalı olarak önerilen [108] çalışmasının geliştirilmiş halidir. 64, 32 ve 16 boyutlu PU'lar yatay veya dikey bölünme şekline göre 0, 10, 11 gizli bitleri için Şekil 4.3'teki gibi değiştirilmektedir. 8×8 PU'larda ise blok bölünme boyutu doğrudan gizli mesaja göre 0 için $N \times N$, 10 için $N \times 2N$ ve 11 için $2N \times N$ olarak değiştirilmiştir. Kapasite ihtiyacına göre videolar düşük ve yüksek çözünürlüklü olarak gruplanmıştır. Düşük çözünürlüklü videolarda 8×8 CU'lardaki PU'lar, 8×8 ve 16×16 CU'lardaki PU'lar ve 8×8 , 16×16 ve 32×32 CU'lardaki PU'lar olmak üzere üç farklı biçimde veri gizlenmiştir. Yüksek çözünürlüklü videolarda ise bu üç gruba ek olarak tüm CU'ların içindeki PU'ların olduğu dördüncü bir grup daha kullanılmıştır. Sadece 8×8 CU'lardaki PU'ların kullanıldığı durumlarda PSNR değerleri nispeten daha iyi iken, kapasite miktarı CU türü arttıkça daha da artmıştır. Çalışmada kapasite artışıyla birlikte bit oranı değerinde de biraz artış olduğu görülmüştür.



Şekil 4.3. 16×16 , 32×32 ve 64×64 CU boyutları için PU bölünme modlarının gizli bitlere göre gruplanması [109]

Çerçeveler arası tahmin işlemindeki PU bloklarının bölünme modlarına dayalı bir başka veri gizleme de [110] çalışmasında önerilmiştir. Burada ilk olarak 8×8 ve 16×16

boyutlu CU blokları seçilmiştir. Daha sonra, bu CU'lar içerisindeki PU bölünme modları uzamsal bir veri gizleme yöntemi olan EMD ile kodlanarak veri gizlenmiştir. Gizlenecek veri miktarına göre seçilen PU grubundan sadece bir tanesinin bölünme modu değiştirilerek daha az değişimle daha fazla veri gizlenmiştir. Çalışmanın en önemli katkısı çerçeveler arası kestirim bloklarındaki düşük kapasiteyi iyileştirmesidir.

[112]'de HEVC söz dizimi aşamasında videoların blok bölünme kararını değiştirmeye dayalı bir mesaj gizleme yöntemi önerilmiştir. Önerilen yöntem 32×32 ve 16×16 CU blokları üzerinde uygulanmıştır. Her bir CU için komşu bloklardan öznitelikler hesaplanarak bir model oluşturulmuştur. Gizli mesaj bitleri CU'ların bölünme kararına ve model bilgisine göre CU bölünme bayraklarına gizlenmiştir. Bir CTU içerisinde 16 tane 16×16 CU olduğu için, her birisinin bölünme bilgisi için bir bayrak bilgisi tutulmaktadır. Dolayısıyla bir CTU içerisinde 16 adet değiştirilebilecek bayrak bulunmaktadır. Benzer şekilde 32×32 CU'lar için 4 adet bayrak vardır. Gizli mesaj bitlerine göre bölünme bayrakları 0 veya 1 olarak değiştirilmiştir. Mesaj çıkarılma aşamasında stego bloğun gerçek bölünme bilgisi hesaplanıp, mevcut bayrakla karşılaştırılarak gizli bitler elde edilmiştir.

4.2. Çerçeve-içi Kestirim Modunu Değiştirme

Veri gizleme için videonun çerçeve-içi kestirim modlarının değiştirilmesini öneren çalışmalar, genellikle geçerli PU bloğunun en iyi modunu başka bir mod ile değiştirmişlerdir. Bu çalışmaların büyük çoğunluğu HEVC kodlayıcı yapısında erişmesi en kolay blok olması sebebiyle, PU içerisindeki 4×4 blokların modlarıyla işlem yapmışlardır.

[113] çalışmasında 4×4 blokların çerçeve-içi optimal kestirim modları ile gizli bilgi arasında eşleştirme yaparak veri gizlenmiştir. Bloklar için hesaplanan optimal kestirim modlarının olasılık dağılımları hesaplanmış ve ilgili modun yüksek olasılıklı ilk 5 alt modu alınarak 6 elemanlı bir küme oluşturulmuştur. 35 tane kestirim modu için 35 farklı küme oluşturularak, kümelerin kendi aralarında uzamsal benzerlikleri hesaplanmıştır. Alt modları birbirine en çok benzeyen modlardan 4'lü gruplar yapılmış ve modlar sıralanmıştır. Diğer taraftan da iki bitlik gizli mesajlar bu modlarla eşleştirilerek bir tablo elde edilmiştir. Gizlenecek her iki bitlik bilgi için 4×4 bloğun

kestirim modu kontrol edilmiş ve o modun olduğu 4'lü grup tespit edilmiştir. Gizli bilgi değerine göre ilgili sıradaki mod seçilip 4×4 bloğun yeni kestirim modu olarak belirlenmiştir. Veri çıkartım işleminde de aynı eşleştirme tabloları kullanılarak her bir bloktan gizli bilgiler elde edilmiştir.

[114]'te HEVC için çerçeve-içi kestirim modunun hata kontrol kodlamasında kullanılan blok kodlarına dayalı olarak değiştirildiği bir veri gizleme çalışması yapılmıştır. Bu çalışmada, (4, 3) standart dizi kod çözme tablosunda 3 bitlik veriler için 4'er bitlik eş kümelerin olduğu tablolar elde edilmiştir. Eşleştirme mantığına dayanan çalışmada 4×4 PU bloklarda optimal çerçeve-içi kestirim modlarından 4 tanesi elde edilip bunların parite bitleri ile gizlenecek bilgi bitleri arasında Hamming mesafelerine bakılmıştır. Gizlenecek veriye göre optimal kestirim modu diğer alt optimal modlardan biriyle değiştirilmiştir.

[115] çalışmasında, HEVC çerçeve-içi kestirimde kullanılan 35 farklı modun açılış farklarından yararlanarak veri gizleme yapılmıştır. Çalışmada öncelikle her bir modun yatay ile olan açılışları bir tabloya aktarılmıştır. Ayrıca gizli bilgiler ile belirli açılış aralıkları arasında eşleştirme tablosu oluşturulmuştur. Veri gizleme için ardışık iki 4×4 PU bloğunun kestirim modlarına bakılmıştır. Gizlenecek veriye göre eşleştirme tabloları kullanılarak iki bloğun kestirim modları arasında olması gereken açılış farkına göre ikinci bloğun kestirim modu değiştirilmiştir. Yeni kestirim modunun değiştirilen moda en yakın mod olmasına dikkat edilmiştir. Aksi durumlarda video kalitesi etkilenmektedir. Veri çıkartılırken I-çerçeveler çözülerek, ardışık 4×4 PU'ların kestirim modaları bulunmuş ve eşleştirme tabloları yardımıyla gizli veriler elde edilmiştir.

[116], çerçeve-içi kestirim modlarının kullanıldığı yöntem Hamming+1 adı verilen en düşük değerlikli bit ve ardışık bloklardan sadece birinin değiştirilmesi temelli bir çalışmadır. Çalışmada ardışık 4 tane PU bloğunun kestirim modları ve gizli mesaj bitleri arasında bir ilişki kurulmuş, eşitliği sağlama durumuna göre bloklardan birinin kestirim modu değiştirilmiştir.

[117]'de HEVC videolarında veri gizlemeden kaynaklı oluşan hata yayılımını azaltmak için bir yöntem önerilmiştir. Önerilen yöntem, 4×4 PU bloklarının kestirim modlarına veri gizleme yaparken hata artışını en düşük seviyede tutmayı amaçlamıştır.

Daha önceki çalışmalarda mod değişikliği yapılırken, blok için en uygun mod, bir sonraki en uygun mod ile değiştirilerek görsel kalite korunmuştur. Bu çalışmada ise, 0 ve 1 dışındaki ardışık kestirim modları ikili gruplar haline getirilmiştir. Gizli veriye göre bloğun kestirim modu sonraki en iyi mod yerine, gruptaki diğer mod ile değiştirilmiştir. Önerilen yöntemde, kodlama verimliliğini kontrol etmek için gizleme yapılacak blokların değerlendirilmesinde Sendrome-Trellis kodları [118] kullanılmıştır. 1-51 arasındaki QP değerleri için kapasite ve kapasiteye göre bit oranı artış oranları değerlendirilmiştir. 4×4 PU'lar için yapılan bu yöntem, [119]'da tüm PU blok boyutlarına uyarlanarak kapasite ve bit oranlarında daha iyi değerlere ulaşılmıştır.

[120]'de 4×4 bloklardaki çerçeve-içi kestirim modlarının dağılımları analiz edilerek, CU derinlik bilgisi ve PU bölünme modu temelli bir taşıyıcı örtü nesnesi seçme yöntemi önerilmiştir. Değiştirilecek blokların yer bilgileri rastgele bir dizi yardımıyla seçilmiştir. Kestirim modları maliyet değerlerine göre muhtemel modlar ve geriye kalanlar olmak üzere iki gruba ayrılmıştır. Gizli mesaj bitleri ve gruplanan mod bilgilerine göre video kalitesi üzerinde en az bozucu etkiyi oluşturacak modlar değiştirilerek gizleme yapılmıştır. Çalışmada veri gizleme kapasitesini arttırmak için matris kodlama yöntemi kullanılmıştır.

4.3. Hareket Vektörlerine Veri Gizleme

Hareket vektörlerinde yapılacak değişiklikler videonun yapısını daha gözle görülür biçimde bozmaktadır. Bu sebeple hareket vektörlerine yapılacak veri gizleme çalışmalarında, değiştirilecek hareket vektörü sayısı sınırlıdır. Bu ise hareket vektörü temelli çalışmaların diğer yöntemlere göre daha düşük veri gizleme kapasitesine sebep olmaktadır.

[121] çalışmasında, daha önce görüntüler için [122]'de önerilmiş olan genelleştirilmiş fark genişlemesi (difference expansion, DE) kullanılarak hareket vektörlerine veri gizlenmiştir. DE iki farklı piksel değerinin arasında bir genişleme meydana getirerek işlem yapmaktadır. Bu çalışmada ise iki farklı piksel yerine hareket vektörlerinin yatay ve düşey bileşenleri kullanılmıştır. CTU içerisindeki atlama modu (veya birleştirme modu) dışındaki hareket vektörlerinin yatay ve düşey bileşenleri DE ile değiştirilerek işlem yapılmıştır. N-li hareket vektör grubunun $2N+1$ adet uzamsal noktalarının değerleri elde edildikten sonra bu değerlerle DE için fark vektörü oluşturulmuştur.

Oluşturulan fark vektörünün LSB değerlerine gizli bitler eklenmiştir. Veri çıkartımı aşamasında da yine DE yöntemi tersine işletilerek hem gizli mesaj elde edilmiş hem de bozulan hareket vektörleri düzeltilmiştir.

[123] çalışmasında, DE yöntemi optimize edilerek hareket vektörlerine tersinir veri gizlenmesi önerilmiştir. Klasik DE'ye göre fark genişlemesi daha düşük değere ayarlanarak bozulmanın artması önlenmiştir. Veri gizleme için kullanılacak hareket vektörlerinin seçimi farklı PU blok boyutlarına göre farklı eşik değerleriyle yapılmıştır. Birleştirme modu dışındaki hareket vektörlerinden, genliği eşik değerinden büyük olanlar seçilerek DE işlemlerine tabi tutulmuşlardır. Çalışmada, gizleme kapasitesini arttırmak için ikinci bir gizleme aşaması daha uygulanmıştır. Bu aşamada ise hareket vektörlerinin son bitine LSB ile veri gizlenmiştir. Veri çıkartım aşamasında öncelikle birleştirme modu olmayan hareket vektörlerinin LSB kısmındaki gizli bitler çıkarılmıştır. Daha sonra hareket vektörleri için DE veri çıkartım işlemleri uygulanmıştır. Çalışmanın sonuçları klasik DE gizleme yöntemiyle PSNR, SSIM, bit oranı ve kapasite açısından kıyaslanmıştır.

Genel olarak, doğrudan hareket vektörlerine veri gizlemek bit oranı artışına sebep olmaktadır. Bu sebeple [124] çalışmasında, CU bölünme türüne göre seçilen farklı eşik değerleriyle seçilen hareket vektörleri kullanılarak veri gizleme yapılmıştır. İlk olarak, genliği belirlenen eşik değerinden daha büyük olan hareket vektörü seçilmiştir. Daha sonra, bir hareket vektörü fark kümesi elde etmek için hareket vektörü ile bir sonraki hareket vektörü ortalaması arasındaki fark hesaplanmıştır. Hareket vektörünün iki bileşeninin ince ayarının büyüklüğünü temsil eden değerlerden bir modülasyon vektörü oluşturulmuştur. Verinin gizlenmesini gerçekleştirmek için hareket vektörüne modülasyon vektörü eklenerek yeni bir hareket vektörü oluşturulmuştur. Veri çıkartım işlemi eşik değerinden büyük hareket vektörlerinin ve modülasyon vektörünün belli koşulları sağlamasıyla yapılmıştır. Orijinal videoyu elde etmek için ise gizli mesajın 1 olduğu yerlerde hareket vektöründen, modülasyon vektörü geri çıkarılmıştır.

[125]'te HEVC için hareket vektörü setinin, yeni bir hareket vektörü uzayındaki noktalara eşleşme ilişkisini tanımlayan, hareket vektör uzayı kodlaması temelli bir veri gizleme önerilmiştir. Veri gizleme, hareket vektörü setinin eşleştirme değerinin değiştirilmesiyle gerçekleştirilmiştir. N elemanlı vektör bileşenleri içerisinde

çoğunlukla biri değiştirilerek, gizli bit grubunu gizlemeyi sağlayan yüksek verimliliğe sahip bir yöntemdir. Hareket vektörünün yatay ve düşey bileşenlerinden N elemanlı bir vektör oluşturulabilmesi için CTU içerisinde en az $N/2$ tane PU bloğu ve onların birleştirme modu haricindeki hareket vektörlerinin $N/2$ tane olması gerekmektedir. Seçilen hareket vektörlerinin bileşenleri $2N+1$ ile yapılan mod işlemi sonucu yeni bir hareket vektörü uzayına alınmaktadır. İlgili CTU'ya gizlenecek $\log_2(2N+1)$ tane bit değerinin onluk karşılığı bulunur. Hareket vektörü uzayına ait eşleştirme değerinin hesaplanıp gizli bitlerin onluk değerine eşitlenmesi gerekmektedir. Bu amaçla hareket vektörü uzayındaki bileşenlerden sadece birinin artırarak veya azaltarak eşitlik sağlanmıştır. Farklı gizleme kapasitelerine göre elde edilen çalışma sonuçları, bir hareket vektörüne doğrudan bir veya iki bit gizleme sonuçlarıyla karşılaştırılmıştır. [125]'teki deneysel sonuçlara göre, bu yöntemde, kıyaslanan yöntemlere göre daha yüksek gizleme oranı ve daha düşük bit oranı elde edilmiştir.

4.4. Dönüşüm Katsayılarını Kullanma

Videodaki artıklık bilgilerini içeren TU bloklarındaki dönüşüm katsayıları, HEVC veri gizleme için kullanılan en popüler yerlerdir. Birçok çalışma, HEVC kodlama ağaç yapısında bilgilerine en kolay erişilebilen bloklar olan 4×4 TU'lardaki katsayıları kullanmıştır.

HEVC videolarıyla yapılan ilk çalışmalardan biri olan [126]'da HEVC kodlama aşamasında kuantalanmış dönüşüm katsayılarına, en düşük değerlikli bit gizleme yöntemiyle damga verisi saklanmaktadır. Bu yöntemde veri gizleme için sadece sıfır olmayan katsayılar kullanılmıştır. Aksi durumda sıfır olan birçok katsayı değiştiği için sıkıştırma verimliliği azalmaktadır. Veri gizlenecek yerlerin tespiti için eşik değeri olarak 1 kullanılmıştır. Mutlak değeri, eşik değerinden daha büyük olan katsayıların tamamı çift hale getirilmiştir. Her bir katsayıya gizli mesajın bir biti sırasıyla eklendikten sonra kodlanmış video elde edilmiştir. Kod çözme aşamasında gizli mesajı çıkartmak için, eşik değerinden büyük katsayıların 2'ye göre modu alınarak gizli veri elde edilmiştir.

Dönüşüm katsayılarına yapılan HEVC veri gizleme çalışmalarının birçoğuna öncü olan [127] ve [128]'de HEVC videoları için ayrık kosinüs ve sinüs dönüşümleri (DCT ve DST) tabanlı hata yayılımından bağımsız veri gizleme uygulaması geliştirilmiştir.

Yazarlar daha önce [101]'de H.264/AVC için önerilen ve kendileri tarafından bazı eklemelerle H.264/AVC'ye uygulanan [102] yöntemini, HEVC'ye uyarlamışlardır. Bu yöntemde HEVC'de çerçeve-içi kestirim yapılırken, bir PU'nun hesaplanması aşamasında, bu PU'nun sağ üst, sağ, sağ alt, alt ve sol alttaki PU'larda bulunan piksel değerleri kullanılmaktadır. Bir bloğa veri gizlemek aynı zamanda bu blokta hatalı piksellerin oluşmasına sebebiyet vermektedir. Dolayısıyla eğer komşu bloklarda herhangi bir hata varsa aynı hata mevcut bloğa da yayılmaktadır. Aynı şekilde mevcut PU'dan da sol alt, alt, sağ alt, sağ ve sağ üst bloklara hata yayılımı devam edecektir. Bu durumda yatay, düşey ve köşegen olmak üzere 3 farklı yönde 5 farklı hata yayılım örüntüsü meydana gelmektedir. Bu 5 durum için hataya maruz kalan pikseller olduğu gibi, hatadan bağımsız piksellerde elde edilmektedir. Önerilen yöntem 4×4 , 8×8 , 16×16 , 32×32 ve 64×64 PU'larda bulunan TU'larda veri gizlenince hataya sebebiyet vermeyecek yerleri bulmaya çalışmaktadır. Bu amaçla her bir blok için tüm komşu blokların çerçeve-içi kestirim modları kontrol edilmektedir. Bütün komşu blokların kestirim modlarına bakılarak elde edilen örüntülerden 5 farklı ihtimale göre veri gizleme yöntemleri oluşturulmuştur. Bu yöntemler kullanılarak çerçeve-içi kestirimden arta kalan QDCT ve QDST katsayılarının değiştirilmesi esaslı veri gizleme işlemi yapılmıştır. Sıfırdan farklı dönüşüm katsayılarının tek veya çift olması ve gizlenecek bitin durumuna göre katsayılar bir ekleme veya çıkarma yapılmıştır. Veri çıkartılırken de benzer örüntüler tekrar hesaplanarak her bir bloktan gizli bitler geri elde edilmiştir. [128]'deki çalışma kapsamlı olmasına rağmen, orijinal HEVC video kodlamaya göre işlem yükü oldukça fazla ve karmaşıktır. Bunun nedeni, her bir PU bloğu için komşu blokların kestirim modlarının tek tek kontrol edilip, o bloğun hata yayılım örüntüsüne karar verilmesidir. Ayrıca modların elde edilmesi ve veri gizleme işlemi için ise her bir video en az iki defa kodlanmaktadır.

[129] çalışmasında, 4×4 TU bloklarında sıfırdan farklı olan dönüşüm katsayılarına veri gizlenmiştir. Katsayı ve blok seçimi için ardışık I çerçevelerindeki hareket uyumlu blokların tespit edilmiştir. I çerçevelerine en yakın konumdaki ardışık P çerçevelerindeki 4×4 TU bloklarının hareket vektörleri bulunmuştur. Bulunan vektörler 4×4 I-çerçevesinin sözde hareket vektörü olarak atanmıştır. Bu hareket bilgileri ve sıfır olmayan katsayı bilgileri kullanılarak veri gizleme yapılmıştır. Her video çerçevesi içerisine 100 bit gizlenerek sonuçlar değerlendirilmiştir.

[130] çalışmasında ise daha önce hata yayılımına karşı önerilmiş olan [128]'deki yöntem ile veri gizlenmiştir. Gizli mesajın alıcıda doğru biçimde elde edilmesi için hata kontrol kodlama yöntemleri kullanılmıştır. Gizli mesaj BCH (Bose-Chaudhuri-Hocquenghem) hata kontrol kodlamasıyla kodlanmıştır. Buna ek olarak tekrarlı (repetition) kodlama yöntemiyle aynı mesaj birden fazla tekrarlar HEVC videolarına gizlenmiştir. Alıcı tarafında, videodan çıkartılan gizli bitler, kod çözme işlemiyle ayrıştırılarak gizli mesaj elde edilmiştir. Tekrarlı gizlenen mesaj, videonun bit oranı değerini olması gerekenden daha fazla arttırmıştır. Ek olarak, bu yöntemle elde edilen videoya çok düşük seviyede Gauss gürültüsü eklenerek gizli mesajın çıkarılma durumu incelenmiştir. Sonuçta, gürültüye maruz kalan stego videodan gizli mesaj büyük oranda doğru olarak elde edilmiştir.

[131]'de HEVC ile kodlanmış videonun 4×4 çerçeve-içi kestirim bloklarına bir damga gizlenmiştir. Veri gizlemek için ilk olarak I-çerçeveler seçilmiş, daha sonra ise uygun 4×4 bloklar seçilmiştir. Bu amaçla öncelikle I-çerçeveye en yakın P-çerçevenin içinde bulunan hareket vektörleri kullanılarak, I-çerçeve için sanal hareket vektörleri hesaplanmıştır. Sanal hareket vektörü sıfırdan farklı olan bloklar veri gizleme için seçilmiştir. Seçilen çerçeve içerisinden bir anahtar yardımıyla rastgele bloklar seçilmiş ve o blokların sıfırdan farklı DCT katsayıları değiştirilmiştir. Çalışmanın sonuçları incelendiğinde, HEVC videolarına yapılan veri gizlemenin AVC videolarına yapılan gizlemeye göre daha iyi görsel kalite sunduğu görülmüştür. Çalışma sonuçları ayrıca video kalite metriği ve bit oranı değişimi açısından da incelenmiştir. Çalışmanın son aşamasında ise, damgalama çalışmalarında dayanıklılığı test etmek için kullanılan bazı ataklar stego videoya uygulanmıştır. Stego video çerçevelerine düşük yoğunluklu gürültü ve basit süzgeçleme işlemleri uygulandıktan sonra gizli damganın ne kadar doğrulukta çıkarıldığı test edilmiştir.

[132]'de, videodaki 4×4 bloklar içerisine veri gizlemek için ilgili bloğun çerçeve-içi kestirim modu esas alınmıştır. Bloкта değiştirilecek katsayıların sırası kestirim moduna göre belirlenmiştir. 4×4 bloklar için; yatay modlu (6-14) olan bloklarda ilk satırdaki katsayılar, dikey modlu (22-30) olan bloklarda ilk sütundaki katsayılar, diğer modlu bloklarda ise köşegendeki katsayılar değiştirilmiştir. Farklı QP değerleri için gizleme sonuçları PSNR ve bit oranı açısından incelenmiştir.

[133] çalışmasında I-çerçevesinde bulunan 8×8 TU bloklarında orta frekans bölgesinde yer alan DCT katsayılarına veri gizlenmiştir. Basit rastgele ikili bir diziyle şifrelenen gizli bitler doğrudan TU içerisindeki katsayılara eklenmiştir. Değişen TU katsayılarının alıcıda orijinal hale getirilebilmesi için gizleme işleminde kullanılan katsayı önceki ve sonraki değerlerle birlikte, belli kurallar çerçevesinde değiştirilmektedir. PSNR ve bit oranı artış sonuçları MPEG-4 videoları için önerilen bir çalışma ile kıyaslanmıştır.

[134]'te veri gizleme işlemi için 4×4 çerçeve-içi TU bloklarının katsayıları değiştirilmiştir. Fakat diğer hata yayılımını önleyen yöntemler gibi belirli katsayıları değiştirmek yerine farklı bir yol izlenmiştir. I çerçevelerinde piksel dağılımı homojen olan bölgeler tespit edilip, bu bloklara gizleme yapılmıştır. I çerçevede yapılan gizlemeden kaynaklı olarak P çerçevelerindeki hareket vektörlerinin bozulmaması için, ilgili bloktaki hareket vektörü en yakın başka bir vektör ile değiştirilmiştir. Bu yöntemde her çerçeveye 100 bit gizlenmiştir.

[135] çalışmasında, biri HEVC sıkıştırma aşaması için, biri de HEVC sıkıştırmaya dayanlı olmak üzere iki farklı yöntem önerilmiştir. İlk yöntemde videodaki I çerçevenin 4×4 , 8×8 , 16×16 bloklarındaki kuantalanmış dönüşüm katsayılarına tekil değer ayrıştırma (SVD) yöntemiyle gizli bilgiler eklenmiştir. Video içerisine gizlenecek damga Arnold kaotik yöntemiyle şifrelendikten sonra 8×8 bloklara bölünmüştür. Veri gizleme kapasitesini arttırmak için bölünen bloklar önerilen dönüşüm yöntemi ve görüntü sıkıştırma yöntemiyle sıkıştırılarak 4×4 , 8×8 ve 16×16 TU'lar için farklı tekrarlı uzunluk kodlamalarıyla (run-length coding) yeni değerler oluşturulmuştur. Oluşturulan değerler TU blokları içerisinde sıfırdan farklı son katsayının arkasına eklenerek gizlenmiştir. Veri çıkartım aşamasında da entropi kod çözümü sonrasında aynı bloklardaki sıfırdan farklı en son katsayılar bulunarak tekrarlı uzunluk kodlamalarına eklenmiştir. [135]'teki ikinci yöntem olan HEVC sıkıştırmaya dayanlı veri gizleme yönteminde ise SVD ve Tüm Faz Biortogonal Dönüşümü [136] kullanılmıştır. Burada video çerçevesinin Y kanalı Tüm Faz Biortogonal Dönüşümü ile dönüştürülüp, tüm blokların DC katsayılarıyla yeni bir matris elde edilmiştir. Elde edilen matris SVD ile dönüştürülmüş ve gizli mesaj bilgisi SVD dönüşüm katsayılarına eklenmiştir. Çalışmada gizli bilgilerin görüntüyü en az oranda değiştirmesi için iki defa SVD kullanılarak katsayılar dönüştürülmüştür. Çeşitli QP

değerleri için sıkıştırılan videoda PSNR değeri ve korelasyon katsayıları incelenmiştir. Ayrıca stego videoya çok düşük yoğunluğa sahip tuz ve biber gürültüsü ile gauss gürültüsü uygulanarak gizli mesajın elde edilmesi durumu incelenmiştir.

Veri gizleme ve video şifrelemenin bir arada sunulduğu [137] çalışması, stego videonun orijinal haline dönüştürülebildiği tersinir (reversible) bir çalışmadır. Veri gizleme için $N \times N$ TU'ların içerisindeki dönüşüm katsayıları kullanılmıştır. TU içerisindeki DC katsayıya dokunulmazken, AC katsayılar gizli bitler eklenmektedir. Bunun için öncelikle genliği 1'den büyük bütün katsayılar 1 eklenerek ötelenmiştir. Böylece histogram kaydırma çalışmalarına da benzer biçimde 1'in yanındaki 2 katsayısı boşaltılmıştır. Güvenlik amacıyla, şifrelenen gizli bitler genliği 1 olan TU katsayılarına eklenerek yeni blok elde edilmektedir. Ayrıca, videonun hareket vektör farklarının işaret ve genlik bilgileri bir anahtar yardımıyla ayrılabilir biçimde şifrelenmiştir. İlave olarak, artık sinyaldeki katsayıların işaret bilgileri de şifrelenmektedir. Böylece stego video elde edilmektedir. Veri çıkartım aşamasında şifresi çözülen artık sinyal ve hareket vektör farkları elde edilmiştir. Veri çıkartım işlemi için öncelikle videoda şifrelenen kısımlar çözülmektedir. Daha sonra $N \times N$ TU içerisindeki değeri 1 veya 2 olan dönüşüm katsayılarına bakılarak gizli mesajın 0 veya 1 değeri elde edilmektedir. Veri çıkartımı bittikten sonra, genliği 1'den büyük olan bütün TU katsayıların genliği 1 azaltılarak stego videodaki katsayılar orijinal videodaki katsayılarla dönüştürülmüştür. Bu çalışma, benzer yöntemlerden farklı olarak hata yayılımını dikkate almamaktadır ve dolayısıyla video çerçevelerini bozmaktadır. Bu bozulmanın anlaşılmaması için bozulan videoyu şifreleyerek, çerçeveleri anlaşılmaz biçimde alıcıya göndermiştir. Alıcıda da şifre çözülüp gizli mesaj çıkarıldıktan sonra video orijinal haline çevrilmiştir.

[138]'de 4×4 TU'lara, komşu bloklarının çerçeve-içi kestirim modlarına göre seçilen modülasyon şemasıyla veri gizlenmiştir. Bu çalışma daha önce Ma ve arkadaşları tarafından H.264/AVC için [101]'de önerilen, Chang ve arkadaşları tarafından da H.265/HEVC için [128]'de kullanılan hata yayılımını önleyen veri gizleme yöntemi kısmen kullanılmıştır. Kodlanmış HEVC bit dizisinin entropi kod çözme işleminden sonra, uygun bloklar ve uygun katsayılar seçilmiştir. Seçilen bloklardaki katsayılar önerilen yöntemle göre değiştirilerek veri gizlenmiştir. Çalışmada katsayıların değiştirilmesi için komşu blokların modları üç farklı mekanizmayla kontrol edilmiştir.

Geçerli 4×4 TU bloğunun satır veya sütunlardaki katsayılarına gizleme yapılabilmesi için komşu blokların kestirim modlarının belli aralıkta olması gerekmektedir. Şöyle ki; sol alt ve alt PU bloklarının kestirim modları sırasıyla $\{2,3, \dots, 26\}$ ve $\{2,3, \dots, 10\}$ ise satırlar, üst sağ ve sağ bloklar için $\{10,11, \dots, 34, DC\}$ ve $\{26,27, \dots, 34\}$ ise sütunlar değiştirilmiştir. Her satır veya sütundaki bir katsayı veri gizlemek için kullanılırken, diğer iki tanesi ise hata yayılımını önlemek için kullanılmıştır. Ek olarak, eğer yukarıda kontrol edilen dört komşu bloğun yanında, sağ alt bloğun kestirim modu $\{2,3, \dots, 10, 26, 27, \dots, 34, DC\}$ grubunda ise komşulara bir hata yayılımı olmayacağından geçerli bloğun tüm katsayılarına gizleme yapılabilir. Çalışmanın sonuçları hata yayılımına karşı bir önlem olmadan yapılan gizleme sonucuyla kıyaslanmıştır. Burada dikkat çeken yönlerden birisi veri gizleme işleminin video kodlama aşamasında değil de kod çözme aşamasında yapılmasıdır. Veri gizleme sonrası tekrar entropi kodlama işlemi uygulanmıştır. Veri çıkartma işlemi için entropi kod çözme işlemi sonrası 4×4 TU bloğun komşu PU bloklarıyla olan ilişkisi kontrol edilmiştir. Belirlenen modülasyon yöntemine göre bloktaki katsayının tek veya çift olma durumuna bakılarak gizli bitler elde edilmiştir. Bu yöntemde TU katsayılarına doğrudan veya çeşitli eşik değerleriyle veri gizlenmiş ve sonuçları değerlendirilmiştir. Bu çalışma her ne kadar yüksek kapasiteli görünse bile, uygun koşulların sağlandığı blok sayısı azdır. Ayrıca her bir TU bloğu için hem veri gizleme aşamasında, hem de veri çıkartma aşamasında komşu blokların modlarını kontrol ettiği için çok fazla işlem yükü bulunmaktadır.

Aynı yazarlar [138] çalışmasının neredeyse aynısını [139]'da da yapmışlardır. Bu çalışmada ise gizli mesaj bitleri BCH hata kontrol kodlaması yöntemiyle kodlandıktan sonra komşu blokların durumuna göre, 4×4 TU bloğundaki DST katsayılarına gizlenmiştir. Stego video içerisinden çıkarılan gizli bitler BCH kod çözme işleminden geçirildikten sonra gizli mesaj elde edilmiştir. Elde edilen gizli mesajın kalitesi ve sıkıştırma ataklarına karşı dayanıklılığı incelenmiştir.

[140]'ta HEVC videolarının doğrulama kontrolü için veri gizlenmiştir. Bu amaçla çerçeve-içi 4×4 TU blokları iki alt kümeye ayrılmıştır. Alt kümelerden birisi doğrulama kodunun üretilmesi için, diğeri oluşturulan kodun gömülmesi için kullanılmıştır. Blokların seçimi uzamsal analizler yardımıyla belirlenen eşiklerle yapılmıştır. TU bloğundaki pozitif ve negatif nicemlenmiş DCT katsayıların sayısı arasındaki ilişkilere

göre oluşturulan doğrulama kodu ve damga bilgisi, diğer kısımdaki DST katsayılarına gömülmektedir. Gizlenen damganın ve doğrulama bilgisinin çıkarılma işlemi ise entropi kod çözmenin hemen sonrasında yukarıdaki işlemlerin tam tersi biçimde işlemektedir. Videodaki damga çeşitli ataklarla test edilmiş ve başarımlar değerlendirilmiştir.

[141]'de video içerisindeki 4×4 , 8×8 ve 16×16 TU bloklarındaki katsayılara veri gizlenmiştir. 4×4 TU'ların tümü kullanılırken, 8×8 ve 16×16 TU'ları seçmek için Jensen-Shannon diverjansı ve ikinci momente dayalı bir ölçüt kullanılmıştır. Gizli mesaj saldırılardan sonra daha doğru biçimde çıkarılması için BCH kodu ve Turbo kodlarıyla kodlandıktan sonra matris kodlama yöntemiyle gizlenmiştir. Yapılan deneysel sonuçlarda Turbo kodlarının BCH kodlarına göre daha etkili sonuçlar verdiği gözlenmiştir. Bu yöntem blok içerisindeki katsayıları matris kodlama için gruplara ayırıp, her gruptaki katsayıdan sadece birini değiştirmiştir. Gizleme işleminde değişen katsayılar az olmakla birlikte, veri gizleme yöntemi komşu bloklar üzerinde hata yayılımına sebep olmuştur. Ayrıca gizlenen veri miktarı çok düşük kalmıştır.

[142]'de daha önce önerilen [138] ve [139] çalışmaları farklı bir şekilde ele alınmıştır. Önerilen yöntemin veri gizleme biçimi diğer iki yöntemle tamamen aynı olmakla birlikte sadece gizli mesajın kodlanması değiştirilmiştir. Bu çalışmada gizli mesaj Shamir'in sır paylaşım algoritmasıyla kodlanarak alt parçalara bölünmüştür. Sır paylaşım yöntemlerindeki ana amaç mesajın belirli sayıda parçasını bir araya getirerek bütün mesajı elde etmektir. Elde edilen parçalar komşu bloklara hata yayılımı yapmayacak şekilde TU katsayıları içerisine gizlenmiştir. Stego videoda kayıpların mesajın dayanıklılığını nasıl etkilediği değerlendirilmiştir.

4.5. Entropi Kodlama Aşamasında Veri Gizleme

Bu bölümdeki çalışmalarda, entropi kodlama sürecindeki sembol kodlama biçimleri veya entropi kodlama sonrasında elde edilen söz dizimi elemanları değiştirilmektedir. Özellikle söz dizimi elemanlarının değiştirilmesiyle gerçekleştirilen veri gizleyen çalışmalar dönüşüm katsayıları, hareket vektörleri, bayrak değerleri gibi çeşitli bilgileri kullanmışlardır.

Video sıkıştırmanın kayıplı son kısmı olan entropi kodlama aşamasında videonun çerçeve-içi ve çerçeveler-arası kestirim modları, artıklık bilgisi ve hareket vektörleri sıkıştırılmaktadır. CABAC kodlamada kullanılan yöntemlerde semboller 0 ile bitmektedir. Dolayısıyla HEVC’de kodlanan sembolün sonuna gelinip gelinmediğinin kontrolü için 0 varlığına bakılmaktadır. Bu yaklaşımdan yola çıkan [143] çalışmasında gizli veriye kaos tabanlı şifreleme tekniğiyle ön işlem yapılmıştır. Daha sonra CABAC ile yapılan sıkıştırma aşamasında veri gizleme yapılmıştır. Bu amaçla sembolün son bitinin bir sonraki sembolle birlikte kodlandığı bit oranını arttırmayan, bit yerine koyma algoritması (CBIB, Bit substitution with constant bitrate) geliştirilmiştir. Eğer 1 olan yer 0 olursa sembolün sonunun geldiği kabul edilmiş, benzer olarak 0 olan yer 1 olursa sembol bir sonraki sembol ile birlikte kodlanmıştır. Kodlanmış hareket vektör farklarının video steganografi için uygun yerler olduğunu tespit edilmiş. Kaliteyi kontrol için bir eşik kullanarak veri gizleme için uygun CBIB’ler tespit edilip hareket vektör farklarının mutlak değerinin, eşikten büyük olduğu yerlere gizli mesaj bitleri eklenmiştir. Bu yöntem bit oranını arttırmıyor olsa da veri gizleme kapasitesi oldukça düşüktür.

[144]’te yapılan çalışmada ise, videodaki blok yapılarının hesaplandığı döngü bittikten sonra elde edilen söz dizimi elemanları kullanılarak veri gizlenmiştir. Farklı derinliklere sahip CU’lar için kodlanmış hareket vektör farklarına veya DCT katsayılarına gizli bilgiler eklenmiştir. Veri gizleme için klasik LSB yöntemi kullanılmıştır. Çerçeveler-arası seviyede veri gizlenecek yerin tespiti yapılırken, hata yayılımını en aza indirgeyen çerçeveler seçilmiştir. Bunun için de diğer çerçeveler tarafından en az miktarda referans alınan çerçeveler seçilmiştir. Böylece veri gizleme sonrası çerçevenin hata yayılım etkisi en düşük seviyede tutulmuştur.

[145] çalışmasında entropi kodlama sonrası elde edilen bit dizilerine veri gizleme yapılmıştır. Çalışma her ne kadar veri gizleme yapıldığını iddia etse de, bazı eksiklikler bulunmaktadır. Çünkü sadece veri gizlemenin mümkün olup olmadığı incelenmiştir. Veri gizleme sonrası videonun kalitesi ve veri gizleme yönteminin başarımı değerlendirilmemiştir.

[146]’daki çalışmada, HEVC videoda entropi kodlama aşamasında şifreleme ve veri gizleme bir arada kullanılmıştır. Veri gizleme işlemi için ise dönüşüm katsayıları

üzerinde deęişiklik yapılmıřtır. Kuantalanmıř katsayıların Exp-Glomb koduyla kodlanmasıyla oluřan bitlerde deęişiklik yapılarak, video ierisine belirli kurallara gre gizli mesaj eklenmiřtir. Videodaki sz dizimi elemanlarından ereve-ii kestirim modları, kuantalanmıř dnřm katsayıları, hareket vektr farkları řifrelenmiřtir. Bu alıřmanın devamı nitelięinde [147] alıřması yapılmıřtır. Entropi kodlama ařamasında kuantalanmıř katsayılara veri gizleme yntemi iin kullanılan yntem iyileřtirilmiřtir. řifreleme iřleminde ise nceki yntemden farklı olarak kuantalanmıř dnřm katsayıları ve hareket vektr farkları řifrelenmiřtir. Her iki alıřmada da řifreleme sonrası elde edilen stego videodaki frame yapısı grsel olarak bozulmuřtur. Gizli mesaj ıkartıldıktan sonra video eski haline dnřtrtlmektedir. Bu alıřma entropi kodlama ařamasındaki bitlere veri gizlerken video yapısını kısmen bozmaktadır. Bu ise istenmeyen bir durumdur ama bozulan ereve yapısı řifreleme iřlemiyle perdelenmiřtir.

4.6. Doğrudan Piksel Deęerlerine Veri Gizleme

Bu blmde yapılan alıřmalar klasik video veri gizleme gibi doęrudan HEVC video erevesindeki piksel deęerleri kullanılmıřtır. Bunun yanında, bazı alıřmalar ereve piksellerine gizleme yaptıktan sonra HEVC kodlamasını bir atak tr olarak kullanıp, gizli verinin sıkıřtırılmıř video ierisinden geri elde edilme durumunu incelemiřlerdir.

[148]’deki alıřmada, ham videolar MJPEG, H.264/AVC ve H.265/HEVC video formatlarına gre kodlandıktan sonra elde edilen yeni videoların erevesine LSB, PVD gibi temel yntemlerle veri gizlenmiřtir. Veri gizleme videolarda  farklı yerde yapılmıřtır. Bunlar sırasıyla sadece B-erevesi, P+B erevesi veya I+P+B erevesidir. alıřmanın bařarım deęerlendirmesi iin gizlenen veri sonrası videoların maksimum ortalama tutarsızlık ve PSNR deęerlerine bakılmıřtır.

[149]’da ham videoyu iki farklı řekilde kodlanarak oluřturulan HEVC ve AVC video erevesinin piksel deęerlerine kod kelimeleri deęiřtirme yntemiyle veri gizlenmiřtir. Daha sonra rastgele retilmiř anahtar ile řifrelenmiř olan videoların ereve-ii kestirim modları, hareket vektr farkı deęerleri ve artıklık bilgileri řifrelendikten sonra alıcıya gnderilmiřtir.

[150] çalışmasında, YUV video çerçevelerine yapılan veri gizlemenin HEVC kodlamaya karşı dayanıklılığı incelenmiştir. Yüksek kaliteli videoların içerisine kanal logosu ve videonun meta bilgilerinin gizlenmesi için uzamsal düzlemde işlemler yapılmıştır. Hareket değişikliği tespitine göre elde edilen I çerçevelerinin parlaklık kısımlarındaki düşük ve yüksek piksel değerleri kullanılmıştır. Çerçeveleri birbirinde ayırmak için iki farklı eşik değerine dayalı adaptif değişim tespit algoritması kullanılmıştır. Bu çalışmada HEVC kodlama sistemi bir saldırı aracı olarak, önerilen yöntemin sıkıştırmaya dayanıklılığını tespit için kullanılmıştır.

Daha önce [139]'da önerilen çalışmada, gizlenen bilgilerin kod çözme aşamasında blok yapısını değiştirmesini önlemek amacıyla [151] çalışmasında farklı bir yöntem önerilmiştir. Çalışmada dönüşüm katsayıları yerine, onların uzamsal düzlemdeki piksel değerleri kullanılmıştır. Çerçeve-içi artıklık bilgisini içeren 4×4 blokta pikseller ile piksellerin ortalama değerleri arasındaki ilişkiye göre blokta bir değeri değiştiren bir yöntem kullanılmıştır. Verinin gizlendiği yerlerin konum bilgisi ayrıca tutularak alıcıya gönderilmiştir. Sıkıştırılmış videolar için kod çözme aşamasında, ham videolar için ise kodlama aşamasında gizleme yapılmıştır. Damgalama yönteminin dayanıklılığı için stego videoya çeşitli ataklar uygulanarak sonuçlar değerlendirilmiştir.

[152] çalışmasında, HEVC ile kodlanan videodaki PU'lara denk gelen 4×4 bloklar seçilmiştir. Gizli mesaj bitleri rastgele seçilen 4×4 bloklardaki bir piksele eklenmiştir. Her bir gizli bit için rastgele çerçeve, satır ve sütunlar üretilerek oradaki 4×4 bloğun içindeki en sol ve en üst pikselin ikinci en yüksek değerlikli bit değeri değiştirilmiştir. Elde edilen stego video HEVC ile sıkıştırılarak başarımlar sonuçları değerlendirilmiştir.

4.7. Literatür Çalışmalarının Karşılaştırılması

Bu bölümde, yukarıdaki bölümlerde anlatılan HEVC veri gizleme çalışmaları genel hatlarıyla karşılaştırılarak Tablo 4.1'de verilmiştir. Bu çalışmalar tezin yazıldığı zamana kadar erişilebilen veri gizleme çalışmaların büyük çoğunluğu içermektedir. Bütün yöntemler kronolojik sırayla, video kodlamanın hangi aşamasında ve hangi çerçevesinde yapıldıkları, neleri değiştirdikleri, gizleme yöntemi ve kullanılan yöntemle göre blok boyutları (CU, PU veya TU) açısından kıyaslanmıştır.

Tablo 4.1. Literatürdeki HEVC veri gizleme yöntemlerinin karşılaştırılması

Yöntem	Yıl	Çerçeve Tipi	Kodlama Aşaması	Gizlenen Yer	Gizleme Biçimi	Blok Boyutu
[127]	2013	I	Dönüşüm	TU katsayıları	Satır veya sütun değiştirme	Hepsi
[128]	2014	I	Dönüşüm	TU katsayıları	Satır veya sütun değiştirme	Hepsi
[126]	2014	I	Dönüşüm	TU katsayıları	Katsayı LSB değiştirme	Hepsi
[106]	2014	I	Blok bölümlleme	PU modu	Bölme şeklini değiştirme	Hepsi
[113]	2014	I	Çerçeve-içi kestirim	Kestirim modu	Sonraki en iyi moda geçiş	4×4
[114]	2014	I	Çerçeve-içi kestirim	Kestirim modu	Gruplanan modlar arası geçiş	4×4
[115]	2014	I	Çerçeve-içi kestirim	Kestirim modu	Gizli mesaj göre mod açısı değişimi	4×4
[107]	2015	I	Blok bölümlleme	PU modu	Bölme şeklini değiştirme	Hepsi
[143]	2015	I, P, B	Entropi kodlama	CABAC sembolleri ve Hareket vektör farkları	Sembolleri değiştirme	-
[144]	2015	I, P, B	Entropi kodlama	Söz dizimi elemanları	Kodlanmış bitleri değiştirme	-
[145]	2015	I, P, B	Entropi kodlama	Söz dizimi elemanları	Bayrak değeri değiştirme.	-
[116]	2015	I	Çerçeve-içi kestirim	Kestirim modu	Sonraki en iyi moda geçiş	4×4
[129]	2015	I, P	Dönüşüm	TU katsayıları ve Hareket vektörleri	Katsayı ve hareket vektörü değiştirme	4×4
[148]	2016	-	-	Piksel değerleri	Parlaklık değeri değiştirme	-
[149]	2016	-	-	Piksel değerleri	Parlaklık değeri değiştirme ve şifreleme	-
[130]	2016	I	Dönüşüm	TU katsayıları	Satır veya sütun değiştirme	Hepsi
[131]	2016	I, P	Dönüşüm	TU katsayıları	Katsayı değiştirme	4×4
[132]	2016	I	Dönüşüm	TU katsayıları	Satır veya sütun değiştirme	4×4
[121]	2017	P	Çerçeveler-arası kestirim	Hareket vektörleri	Hareket vektörü değiştirme	Hepsi

Tablo 4.1 (Devam). Mevcut HEVC veri gizleme yöntemlerinin karşılaştırılması

Yöntem	Yıl	Çerçeve Tipi	Kodlama Aşaması	Gizlenen Yer	Gizleme Biçimi	Blok Boyutu
[117]	2017	I	Çerçeve-içi kestirim	Kestirim modu	Gruplanan modlar arası geçiş	4×4
[133]	2017	I	Dönüşüm	TU katsayıları	Katsayı değiştirme	8×8
[134]	2017	I, P	Dönüşüm	TU katsayıları ve Hareket vektörleri	Katsayı ve hareket vektörü değiştirme	4×4
[108]	2018	P	Blok bölümlleme	PU modu	Bölme şeklini değiştirme	Hepsi
[112]	2018	I	Blok bölümlleme	CU modu	Bölme bayrağı değiştirme	32×32 16×16
[150]	2018	-	-	Piksel değerleri	Parlaklık değeri değiştirme	-
[123]	2018	P	Çerçeveler-arası kestirim	Hareket vektörleri	Hareket vektörü değiştirme	Hepsi
[124]	2018	P	Çerçeveler-arası kestirim	Hareket vektörleri	Hareket vektörü değiştirme	Hepsi
[125]	2018	P	Çerçeveler-arası kestirim	Hareket vektörleri	Hareket vektörü değiştirme	Hepsi
[135]	2018	I	Dönüşüm	TU katsayıları	Katsayı değiştirme	4×4, 8×8, 16×16
[137]	2018	I	Dönüşüm	TU katsayıları	Katsayı değiştirme	Hepsi
[139]	2018	I	Dönüşüm	TU katsayıları	Satır veya sütun değiştirme	4×4
[109]	2019	P	Blok bölümlleme	PU modu	Bölme şeklini değiştirme	Hepsi
[151]	2019	I	-	Piksel değerleri	TU'ya denk gelen pikselleri değiştirme	4×4
[155]	2019	P	Blok bölümlleme	PU modu	Bölme şeklini EMD ile değiştirme	Hepsi
[146]	2019	I, P, B	Entropi kodlama	Söz dizimi elemanları	Katsayı değiştirme	-
[119]	2019	I	Çerçeve-içi kestirim	Kestirim modu	Gruplanan modlar arası geçiş	Hepsi
[120]	2019	I	Çerçeve-içi kestirim	Kestirim modu	Mod değiştirme	4×4
[140]	2019	I	Dönüşüm	TU katsayıları	Katsayı değiştirme	4×4

Tablo 4.1 (Devam). Mevcut HEVC veri gizleme yöntemlerinin karşılaştırılması

Yöntem	Yıl	Çerçeve Tipi	Kodlama Aşaması	Gizlenen Yer	Gizleme Biçimi	Blok Boyutu
[138]	2019	I	Dönüşüm	TU katsayıları	Satır veya sütun değiştirme	4×4
[141]	2019	I	Dönüşüm	TU katsayıları	Katsayı değiştirme	4×4, 8×8, 16×16
[147]	2020	I, P, B	Entropi kodlama	Söz dizimi elemanları	Kodlanmış bitleri değiştirme	-
[152]	2020	-	-	Piksel değerleri	PU'ya denk gelen pikselleri değiştirme	4×4
[142]	2020	I	Dönüşüm	TU katsayıları	Satır veya sütun değiştirme	4×4

4.8. Steganaliz Çalışmaları

HEVC veri gizleme yöntemlerine karşı geliştirilen steganaliz çalışmaları oldukça azdır. Steganaliz çalışmalarında en büyük problem hazır kodlanmış video veri setlerini bulabilmektir. Çünkü HEVC için yapılacak steganaliz için, içerisine veri gizlenmiş HEVC bit dizileri ve videolarının oluşturulması gerekmektedir. Bu tarz işlemler için uygun veri gizleme yöntemini bulup HEVC kodlama sistemine uyarlamak ise oldukça zaman almaktadır. Artan veri gizleme çalışmaları sonrasında HEVC steganaliz çalışmalarının da artacağı öngörülmektedir.

[153]'teki çalışmada, HEVC videolarında steganaliz çalışması olarak görünmesine rağmen çalışmada esasen uzamsal düzlemde bir steganaliz yapılmıştır. Mevcut olan veri gizleme yöntemleri kullanılarak video piksellerine eklenen gizli mesajın varlığı kontrol edilmiştir. Bu amaçla videolardaki hareket vektörlerinin çerçeveler arası bilgileri kullanılarak çeşitli öznitelikler çıkartılmıştır. Sınıflandırıcı yardımıyla stego videolardaki değişimler kontrol edilerek steganaliz için kullanılacak öznitelikler bulunmuştur. Bu özniteliklerin kontrol edilmesiyle veri gizleme olup olmadığı tespit edilmiştir.

[154]'teki çalışmada, genel olarak HEVC videolarında steganalizin nasıl yapılabileceğine dair prensipler üzerinde değerlendirmeler yapılmıştır. Ayrıca LSB'lere yapılan gizlemeler için bir video steganaliz çalışması yapılmıştır. Çalışma,

HEVC videolarının piksellerine yapılan gizlemenin öznitelik çıkarımı ve sınıflandırma ile tespit edilebileceğini göstermiştir.

[155]'teki çalışmada, daha önce HEVC'de PU bloklarının bölünmesinde değişiklik yaparak veri gizleme öneren [108] yöntemine karşı bir steganaliz çalışması yapılmıştır. Gizli veri varlığını tespit etmek için öncelikle videodaki P çerçevelerinin PU bölme modları çıkartılmıştır. Orijinal ve stego videolarına ait her bir PU bölme modundan sınıflandırma için bir öznitelik grubu oluşturulmuştur. Destek vektör makinası sınıflandırıcısı kullanılarak öznitelikler sınıflandırılmış ve stego videolar tespit edilmiştir.



5. TEZ KAPSAMINDA YÜKSEK KAPASİTE VE YÜKSEK VERİM İÇİN GELİŞTİRİLEN YÖNTEMLER

Bu tez çalışmasında, HEVC videolarına veri gizleme kapasitesini arttıran ve videoların aslına benzerlik oranını yüksek tutan yöntemler geliştirilmiştir. Küçük bloklardaki değişimlerin daha az bozulmaya sebep olması, insan görsel sistemini buralardaki değişimlere karşı daha az duyarlı kılmaktadır. Bu yüzden, bu tez kapsamında yapılan veri gizleme çalışmaları için 4×4 artıklık bloklarının QTC'leri kullanılmaktadır. HEVC'deki 8×8 ve daha büyük TU blokları tamsayı dönüşüm matrisleri için DCT katsayılarını kullanılırken, 4×4 TU blokları için DST katsayıları kullanılmaktadır. Dolayısıyla tez kapsamında 4×4 TU bloklarının kuantalanmış DST (QDST) katsayılarına veri gizlenmektedir. Tezdeki özgün noktalardan birisi, görüntüler için önerilen bir veri gizleme yönteminin TU bloklarına uyarlanmasıdır. İkinci özgün nokta ise, QDST katsayılarına veri gizlenirken kullanılan hata yayılımını önleyen yöntemle farklı bir bakış açısı getirilerek, yöntemin basitleştirilmesidir.

Tablo 5.1. Tez çalışmasında kullanılan test videoları ve özellikleri

Video dizisi	Çözünürlük	Çerçeve hızı	Bit derinliği
Traffic	2560×1600	30fps	8 bit
PeopleOnStreet	2560×1600	30fps	8 bit
Cactus	1920×1080	50fps	8 bit
BQTerrace	1920×1080	60fps	8 bit
Johnny	1280×720	60fps	8 bit
FourPeople	1280×720	60fps	8 bit
RaceHorsesC	832×480	30fps	8 bit
BasketballDrill	832×480	50fps	8 bit
Keiba	416×240	30fps	8 bit
BasketballPass	416×240	50fps	8 bit

Tez kapsamında yapılan tüm çalışmalar için HM referans yazılımının HM16.15 sürümü ve HM yazılımının test video dizileri [32] kullanılmaktadır. Önerilen yöntemlerin başarımlarını analizleri için HM16.15 ile elde edilen PSNR ve bit oranı değerleri kullanılmaktadır. Deneysel çalışmalarda HM test modeli için sunulan video dizilerinden bazıları kullanılmıştır. Kullanılan video dizilerinin özellikleri Tablo 5.1'de, orijinal videoların ilk çerçeve örnekleri ise Şekil 5.1'de verilmiştir.



Traffic



PeopleOnStreet



Cactus



BQTerrace



Johnny



FourPeople



RaceHorsesC



BasketballDrill



Keiba



BasketballPass

Şekil 5.1. Tez çalışmasında kullanılan test videoların ilk çerçeveleri

5.1. Kullanılan Veri Gizleme Yöntemi

Tez çalışmalarında QDST katsayılarına veri gizleme için, daha önce görüntülere veri gizlemek için önerilen matris kodlama (matrix encoding, ME) [156] metodu kullanılmaktadır. ME, geleneksel veri gizleme yöntemlerine göre yüksek gizleme verimliliğine sahiptir. ME genellikle görüntüler içerisine veri gizleme için kullanılırken [157, 158], bazı çalışmalarda videolara veri gizleme için de kullanılmıştır [159]. Klasik veri gizleme yöntemlerinde bir taşıyıcı eleman değiştirilerek bir mesaj biti gizlenirken, matris kodlamada bir taşıyıcı değiştirilerek birden fazla bit gizlenmektedir.

ME'nin genel gösteriminde c ve m sırasıyla taşıyıcı sayısı ve gizli bit sayısını, 1 değeri ise değişecek eleman sayısını göstermek üzere, $(1, c, m)$ şeklinde gösterilmektedir. c ile m arasındaki ilişki Eşitlik (5.1)'de verilmiştir.

$$c=2^m-1 \quad (5.1)$$

Dolayısıyla ME'de m tane biti gizlemek için c tane taşıyıcıdan sadece birisini değiştirmek yeterlidir. Veri gizleme sonrası, değişen bit sayısının taşıyıcıya oranını ifade eden değişim yoğunluğu, D Eşitlik (5.2)'deki gibi olmaktadır.

$$D=\frac{1}{c+1}=\frac{1}{2^m} \quad (5.2)$$

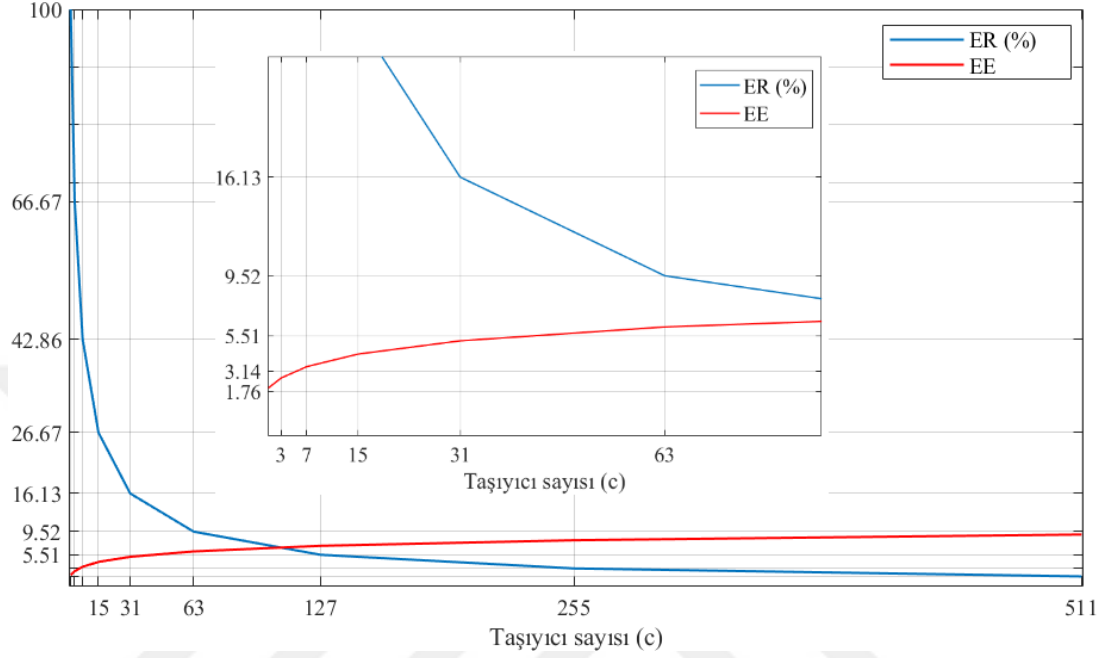
ME'de veri gizleme başarımı için en önemli parametreler, gizlenen bit sayısı ile taşıyıcı sayısı arasındaki gizleme oranı (embedding rate, ER) ve gizleme verimliliğidir (embedding efficiency, EE). Gizlenen bit sayısının taşıyıcı sayısına oranını veren ER,

$$ER=\frac{m}{c}=\frac{m}{2^m-1} \quad (5.3)$$

şeklinde hesaplanmaktadır. Gizleme oranının, değişim yoğunluğuna oranı ise EE'yi vermektedir.

$$EE=\frac{ER}{D}=\frac{c+1}{c} \cdot m=\frac{2^m}{2^m-1} \cdot m \quad (5.4)$$

EE ile ER'nin taşıyıcı sayısına göre değişimi Şekil 5.2'de verilmektedir. Eşitlik (5.3) ve (5.4)'ten anlaşılacağı üzere taşıyıcı sayısı olan c arttıkça ER azalırken, EE artmaktadır.



Şekil 5.2. ME'de gizleme oranı ve gizleme verimliliği arasındaki ilişki

ME'de $c+1$ tane durum bulunmakta olup, bunlardan birinde hiçbir taşıyıcı değiştirilmezken, geriye kalan c tane durumun her birisinde ise sadece bir taşıyıcı değiştirilmektedir. Tüm durumların elde edilmesi için ikili Hamming kodları kullanılmaktadır. Buradaki temel mantık, hata kontrol kodlamalarındaki hata düzeltme yöntemlerine benzemektedir. $H_{m \times c}$ Hamming kod matrisi ve $r_{1 \times c}$ taşıyıcı vektörü olmak üzere, bağımlılık vektörü d , Eşitlik (5.5)'teki gibi olmaktadır.

$$d = Hr^T \text{ mod } 2 \quad (5.5)$$

burada $d_{m \times 1}$, taşıyıcı işaret ile mesaj işareti $b_{m \times 1}$ arasındaki bağımlılık vektörünü göstermektedir. Bağımlılık vektörü ve mesaj vektörü kullanılarak değiştirilecek olan taşıyıcının, vektör içerisindeki pozisyonu p , elde edilmektedir.

$$p = \text{bin2dec}(d \oplus b) \quad (5.6)$$

burada $\text{bin2dec}()$ işlemi ikili sayılardan onluk sayılara dönüşümü, $\oplus$ ise özel-veya (ex-or) mantıksal operatörünü göstermektedir. p değeri bulunduktan sonra taşıyıcı

vektöründeki ilgili taşıyıcı eleman Eşitlik (5.7)'deki gibi değiştirilerek mesaj bitleri gizlenmektedir. p sıfır olduğunda hiçbir katsayıyı değiştirilmemektedir.

$$r_{\text{yeni}}(p) = \begin{cases} r(p)+1 & r(p) \geq 0 \text{ ise} \\ r(p)-1 & r(p) < 0 \text{ ise} \end{cases} \quad (5.7)$$

Veri gizleme sonrası stego vektör elde edilmektedir. Stego vektör içerisinde gizli mesajı çıkartma işlemi de veri gizlemeye benzer biçimde yapılmaktadır. Alıcıda r_{yeni} stego vektörü alındıktan sonra gizli mesaj,

$$b' = Hr_{\text{yeni}}^T \quad (5.8)$$

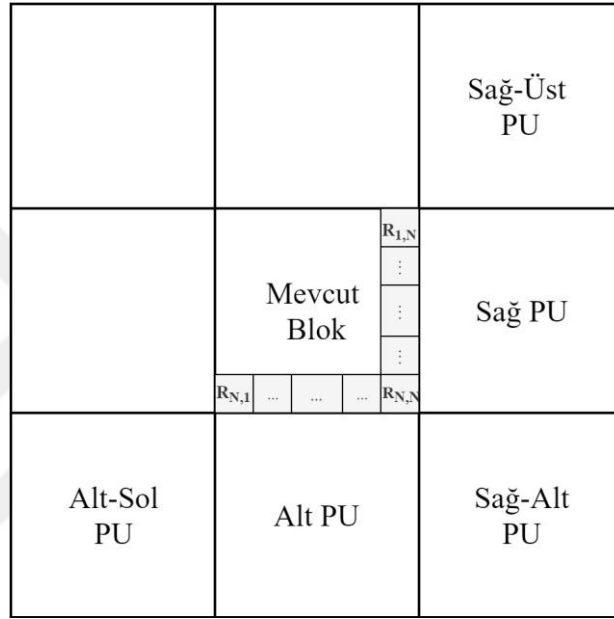
eşitliği ile elde edilmektedir. Burada $b'_{m \times 1}$ stego vektörden çıkarılan mesajı göstermektedir. Herhangi bir saldırı olmadığı sürece çıkartılan mesaj, gizlenen mesajın aynısı olmaktadır.

5.2. Kullanılan Hata Yayılımını Önleme Yöntemi

Tez çalışmasında, 4×4 TU'lara ait QDST katsayılarına veri gizlenmektedir. Daha önce Bölüm 2.2'te anlatıldığı gibi HEVC'de çerçeve-içi piksellerin kestirimi için komşu blokların değerleri kullanılmaktadır. Her bloğun piksel değerleri yeniden oluşturulmuş üst bloklardaki en alt satır veya sol bloklardaki en sağ sütun değerlerinden IPM yardımıyla kestirilmektedir. Daha sonra, kestirilen değerlere artıklık işareti de eklenerek blok değerleri kabaca elde edilmektedir. Mevcut bir PU bloğun değerleri üst ve sol bloklardan kestirildiği gibi, alt ve sağ bloklar da yeniden oluşturulmuş mevcut bloğun değerlerini kullanmaktadır. Bu bloklar, mevcut $N \times N$ boyutlu bloğun en sağ sütunundaki $R_{i,N} (i=1, \dots, N)$ veya en alt satırındaki $R_{N,j} (j=1, \dots, N)$ değerlerinden açısız moda göre kopyalanarak veya ekstrapolasyonla oluşturulmaktadır. Şekil 5.3'te bir PU bloğunun değerlerini kullanan komşu bloklar gösterilmektedir.

Mevcut bloktaki referans piksel değerlerine herhangi bir müdahale olması durumunda komşu bloklar üzerine yatay, dikey ve çapraz olmak üzere üç farklı hata yayılımı olabilmektedir. Yatay hata yayılımında, $R_{i,N} (i=1, \dots, N)$ piksellerindeki hata, sağ-üst PU ve sağ PU üzerinde hataya sebep olur. Dikey hata yayılımında, $R_{N,j} (j=1, \dots, N)$ piksellerindeki hata, sol-alt PU ve alt PU üzerinde hataya sebep olur. Çapraz hata

yayılımında ise $R_{N,N}$ pikselindeki hata, sağ-alt PU üzerinde hata yayılımına sebep olur. Hangi blokta ne zaman hata yayılımı olacağı ise komşu blokların IPM değerine bağlıdır. Yatay hata yayılımı, sağ-üst PU $\{0, 2-10\}$ ve sağ PU $\{0, 1-25\}$ modlarından biriyle kestirildiğinde ortaya çıkmaktadır. Dikey hata yayılımı, sol-alt PU $\{0, 26-34\}$, alt PU $\{0, 1, 11-34\}$ modlarından biriyle kestirildiğinde ortaya çıkmaktadır. Çapraz hata yayılımı ise sağ-alt PU $\{0, 1, 11-25\}$ modlarından biriyle kestirildiğinde ortaya çıkmaktadır.



Şekil 5.3. Kestirim için mevcut bloğun referans değerlerini kullanan komşu PU bloklar

Yukarıdaki bilgiler ışığında açıkça görülmektedir ki eğer herhangi bir bloğun artıklık işareti içerisine gizli veri eklenirse, o blok komşu bloklar üzerine hata yayılımına sebep olmaktadır. [128] çalışması komşu PU modlarının durumuna göre oluşan yatay, dikey veya çapraz hata örüntülerinden yola çıkarak, mevcut bloğun $R_{i,N}$ veya $R_{N,j}$ piksellerini koruyacak şekilde beş farklı veri gizleme şeması önermiştir. Bu işlemler için veri gizlenecek her bir bloğun komşu bloklarının tek tek kontrol edilmesi gerekmektedir.

Bu tez çalışmasında, [128] çalışmasındaki yöntemler basitleştirilerek, HM kodlarında daha kolay uygulama yapılması sağlanmaktadır. Çünkü [128] ve [138] gibi yöntemler TU katsayılarına veri gizlemek için, videoyu iki defa kodlamışlardır. Birinci kodlamada veri gizleme yapılmaksızın sadece blokların analizi ve sınıflandırılması

yapılırken, ikinci kodlamada blokların sınıflandırılma durumuna göre veri gizleme şemaları kullanılmıştır. Bu tez kapsamında önerilen yöntem sayesinde blok analizi yapılmaksızın veri gizlendiği için videoyu iki defa kodlama zorunluluğu ortadan kalkmaktadır. Ayrıca kod çözücü tarafta komşu blok yapılarının analizine ve blok bilgilerinin ayrı bir dosyada tutulup veri gizleme aşamasında kontrol edilmesine ihtiyaç kalmamaktadır.

Tez kapsamında, veri gizlenecek tüm blokların en sağ sütundaki ve en alt satırdaki piksel değerleri hata yayılımına karşı korunmaktadır. Dolayısıyla önceki yöntemlere göre, açısız modlarla komşu blokların durumunu kontrol etme ihtiyacı ortadan kalkmaktadır. Artıklık (residual) işaretini bulunduran 4×4 TU bloğu R, tamsayı DST matrisi S ve QP değerlerine bağlı olarak elde edilen ölçekleme parametresi α olarak alındığında, TU bloğunun QDST katsayıları Eşitlik (5.9)'daki gibi hesaplanmaktadır.

$$R_{QDST} = \frac{SRS^T}{\alpha} \quad (5.9)$$

Kodlama aşamasında komşu blokların kestirimi için gerekli olan yeniden oluşturulmuş (reconstruction) blok değerleri, mevcut PU bloğu ile yeniden oluşturulan artıklık (reconstruction residual, R_{REC}) bloğunun toplamından elde edilmektedir. R_{REC} ters kuantalama ve ters dönüşüm ile Eşitlik (5.10)'deki gibi elde edilmektedir.

$$R_{REC} = S^T (R_{QDST} \times \alpha) S \quad (5.10)$$

HEVC'de 4×4 TU bloğu için R_{QDST} matrisi ve QDST katsayıları, Eşitlik (5.11)'deki gibi olmaktadır.

$$R_{QDST} = \begin{bmatrix} q_{11} & q_{21} & q_{31} & q_{41} \\ q_{12} & q_{22} & q_{32} & q_{42} \\ q_{13} & q_{23} & q_{33} & q_{43} \\ q_{14} & q_{24} & q_{34} & q_{44} \end{bmatrix} \quad (5.11)$$

Veri gizleme işlemi için R_{QDST} katsayılarına 4×4 boyutlu bir ϵ gizli mesaj matrisi eklendiğinde yeni QDST katsayıları Eşitlik (5.12)'deki gibi olmaktadır.

$$R_{QDST}^{yeni} = R_{QDST} + \epsilon \quad (5.12)$$

Gizli mesajı içeren yeni artıklık işareti ters kuantalama ve ters dönüşümle yeniden oluşturularak, R_{REC}^{yeni} Eşitlik (5.13)'teki gibi elde edilmektedir.

$$R_{REC}^{yeni} = S^T (R_{QDST} \times \alpha) S + S^T (\varepsilon \times \alpha) S \quad (5.13)$$

Bu durumda, orijinal artıklık bloğu ile veri gizlenmiş artıklık bloğu arasındaki hata Eşitlik (5.14)'deki gibi elde edilmektedir.

$$R_{hata} = R_{REC}^{yeni} - R_{REC} = S^T (\varepsilon \times \alpha) S \quad (5.14)$$

Komşu blokların açısal modlarını kontrol etme zorunluluğunu ortadan kaldırmak için, [128] çalışmasında sadece bazı bloklar için kullanılan yöntem, bu çalışmada tüm bloklara uygulanmaktadır. Bu amaçla, öyle bir ε matrisi seçilmektedir ki, R_{hata} matrisinin en sağ ve en alt piksellerinin hiçbirisi değişmesin. Yani $i=0,1,2,3$, ve $j=0,1,2,3$ için $R_{hata}(i,0)=0$ ve $R_{hata}(0,j)=0$ olsun. Bu elde edilirse mevcut bloktan komşu bloklara hata yayılımı her halükarda önlenir. Değişken tamsayı değeri t için ε matrisi Eşitlik (5.15)'teki gibi olduğunda R_{hata} Eşitlik (5.16)'daki gibi elde edilmektedir (Bkz. Ek-A). Burada C değeri Eşitlik (2.3)'te verilen tamsayı DST matrisindeki sabit bir katsayıdır.

$$\varepsilon = \begin{bmatrix} t & 0 & -t & t \\ 0 & 0 & 0 & 0 \\ -t & 0 & t & -t \\ t & 0 & -t & t \end{bmatrix} \quad (5.15)$$

$$R_{hata} = \alpha \times t \times \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 9C^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.16)$$

R_{hata} matrisinin en sağ ve en alt değerleri değişmediği için veri gizleme sonrası yeniden oluşturulan bloktan komşu bloklara hata yayılımı olmayacaktır. Dolayısıyla, veri gizleme için Eşitlik (5.15)'teki ε matrisi kullanılarak 4×4 boyutlu TU'lardaki QDST katsayılarına veri gizlenebilmektedir. R_{hata} matrisinin değerini en düşük düzeyde tutmak, piksellerdeki değişimi azaltmak için önemlidir. Eşitlik (5.16)'daki α ve C değerleri sabit, t değeri değişkendir. Bu sebeple tez çalışmasında en düşük değişim için t değeri pozitif katsayı için 1 olarak, negatif katsayı için -1 olarak alınmıştır.

Taşıyıcı olarak 4×4 TU bloğunda sol üstteki ilk katsayı (q_{11}) kullanılmaktadır. Veri gizleme sonrası hata yayılımını önlemek için TU'ya Eşitlik (5.15)'teki matris değerleri eklenmektedir. Matriste q_{11} 'e denk gelen değer veri gizleme için, geriye kalan değerler ise hata yayılımını önlemek için kullanılmaktadır.

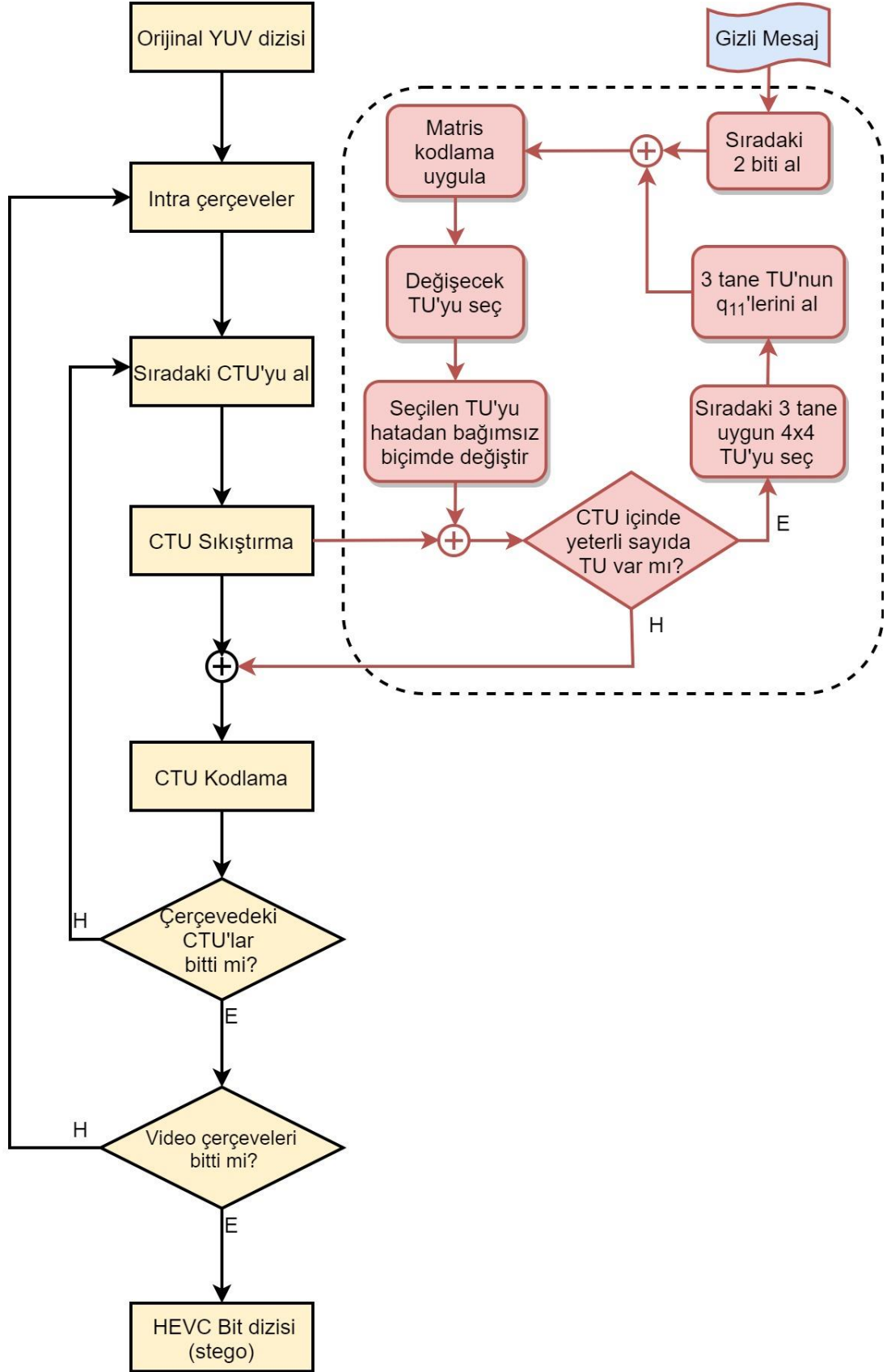
5.3. Yüksek Kapasite Temelli Veri Gizleme Yöntemi

Çalışmanın bu kısmında HEVC videolarında yüksek kapasiteli veri gizleme yapılırken videodaki görsel değişimin düşük oranda kalacağı bir yöntem önerilmiştir. Veri gizleme için taşıyıcı olarak 4×4 TU blokları kullanılmaktadır. 4×4 blokların seçilmesindeki en önemli sebep, insan görsel sisteminin küçük bloklardaki değişimi daha zor algılamasıdır. Önceki bölümde anlatıldığı gibi, gizli mesaj olarak Eşitlik (5.15)'teki ε matrisi kullanılarak, mevcut PU içerisindeki TU bloğuna veri gizlenince komşu bloklardaki IPM değerlerini kontrol etmeye ihtiyaç ortadan kalmaktadır.

5.3.1. Önerilen yöntem

HEVC videolarında 4×4 TU bloklarında hata yayılımı olmadan, yüksek kapasiteli veri gizleme yapabilmek için ME (1, c, m) şeması kullanılmaktadır yani, c adet TU içerisine m adet bit gizlenmektedir. Klasik veri gizleme ile c tane TU'ya c tane bit gizlenebilmektedir. Fakat bu durumda toplam c tane TU'nun tamamında değişiklik yapılması gerekmektedir. Önerilen yöntemde ise c tane TU'ya m bit gizlenirken sadece 1 tane TU değiştirilmektedir. Gizleme kapasitesinin en yüksek oranda tutulması için m değeri 2 bit olarak alınmaktadır. O zaman c değeri de 3 olmaktadır. Dolayısıyla önerilen yöntem ME(1, 3, 2) olarak ifade edilmektedir.

Şekil 5.4'te önerilen yöntemin akış şeması yer almaktadır. Bu şemada HEVC kodlamanın CTU sıkıştırma işlemi aşamasında, CTU'daki 4×4 TU'lar kontrol edilir. QDST katsayılarından en az biri sıfırdan farklı olan TU'ların sayısı ve tamamının yerleri bulunur. Veri gizleme yöntemine uygun olarak ilk 3 tane 4×4 TU'nun her birindeki q_{11} katsayıları alınır. Alınan katsayıların LSB değerleri, matris kodlama işlemlerinde kullanılmak üzere tespit edilir. Matris kodlama sonucu değiştirilmesine karar verilen TU'nun QDST katsayılarına hata yayılımını önleyen ε matrisi eklenerek veri gizlenir ve veri gizleme için sıradaki 3 tane TU'ya geçilir.



Şekil 5.4. Önerilen yöntemin akış şeması

Bu işlemler CTU içerisindeki uygun TU sayısı 3'ten az olana kadar tekrar eder. CTU içerisindeki TU'lar bitince ilgili CTU'nun kodlanma işlemi tamamlanır ve diğer CTU'ya geçilerek benzer işlemler tekrar eder. Çerçeveadaki tüm CTU'ların kodlanması bitince bir sonraki çerçeveye geçilerek veri gizleme işlemine devam edilir. Videodaki tüm çerçevelere veri gizlenene kadar bu işlemler devam ederek veri gizlenmiş HEVC bit dizisi elde edilir.

Veri gizleme işleminde değişecek TU seçimi için ilk olarak gizli mesaj bitleri Eşitlik (5.17)'deki gibi $b_{2 \times 1}$ 'lik vektörlere ayrılmaktadır.

$$b_{2 \times 1} = (b_1, b_2) \quad (5.17)$$

Bağımlılık vektörünü bulmak için Hamming kod matrisi $H_{2 \times 3}$ Eşitlik (5.18)'deki gibi alınmaktadır.

$$H_{2 \times 3} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad (5.18)$$

Bu durumda bağımlılık vektörünü bulmak için taşıyıcı örtü vektörünün oluşturulması gerekmektedir. Bu tez çalışmasının özgün noktalarından biri olan $r_{1 \times 3}$ taşıyıcı vektörü için TU'lar içerisindeki QDST katsayılarının (q_{11}) LSB değerleri kullanılmaktadır. TU seçimi için her I-çerçevesinin parlaklık bileşenine ait CTU içerisindeki TU'lar $c=3$ 'lü gruplara ayrılmaktadır. İlgili TU'ların LSB değerlerinden oluşturulan taşıyıcı vektör,

$$r_{1 \times 3} = (q_{TU1}, q_{TU2}, q_{TU3}) \quad (5.19)$$

şeklinde elde edilmektedir. Bu değerler Eşitlik (5.5)'te yerine yazıldığında,

$$d_1 = (q_{TU1} + q_{TU3}) \bmod 2 \quad (5.20)$$

$$d_2 = (q_{TU1} + q_{TU3}) \bmod 2$$

$$p = (d_2 \oplus b_2) \times 2^1 + (d_1 \oplus b_1) \times 2^0 \quad (5.21)$$

şeklinde bulunmaktadır. Tablo 5.2'de hangi TU'nun, hangi koşullarda değişeceği verilmiştir. Değişecek TU bulunduktan sonra, ilgili TU'ya Eşitlik (5.15)'teki ε matrisi eklenerek hata yayılımından bağımsız biçimde veri gizlenmiştir. Bütün video

çerçevelerine veri gizleme işlemi tamamlandıktan sonra stego video alıcıya gönderilmektedir.

Tablo 5.2. ME(1,3,2) için değişecek bloğun seçimi

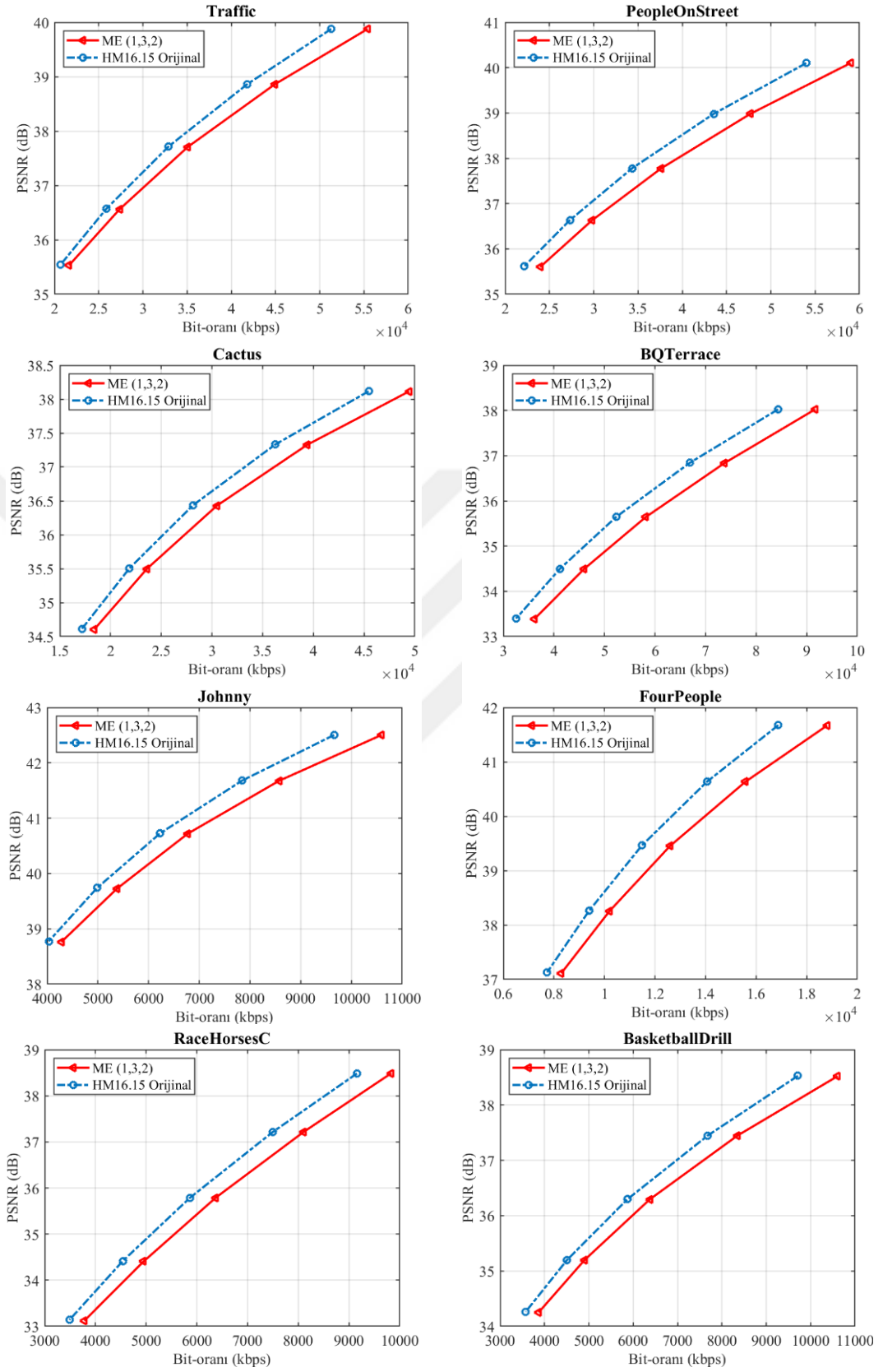
Kontrol	p	Değiştirilen TU
$d_2=b_2$ ve $d_1=b_1$ ise	$\text{bin2dec}(00)=0$	-
$d_2=b_2$ ve $d_1 \neq b_1$ ise	$\text{bin2dec}(01)=1$	TU_1
$d_2 \neq b_2$ ve $d_1=b_1$ ise	$\text{bin2dec}(10)=2$	TU_2
$d_2 \neq b_2$ ve $d_1 \neq b_1$ ise	$\text{bin2dec}(11)=3$	TU_3

Alıcıdaki veri çıkartma işlemi de veri gizlemeye benzer biçimde yapılmaktadır. HEVC video kod çözme işlemi aşamasında stego bit dizisinden intra çerçeveler çözülür ve sırasıyla CTU'ların kodları çözülmeye başlanır. Çözülen CTU içindeki TU'lar 3'lü gruplar halinde sıralanarak her birinin q_{11} katsayılarının LSB değerleri elde edilerek veri gizlenmiş taşıyıcı vektörü r_{yeni} oluşturulur. Daha sonra Hamming kod matrisi $H_{2 \times 3}$ ve r_{yeni} kullanılarak Eşitlik (5.8)'e göre gizlenen 2 bit elde edilir. CTU'lardaki bütün 3'lü TU gruplarından çıkarılan 2 bitlik mesajlar birleştirilerek gizli mesaj oluşturulur.

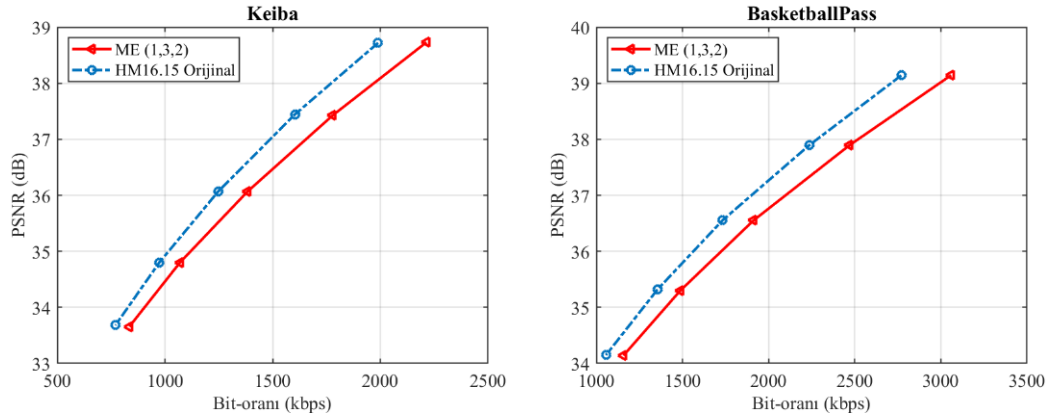
5.3.2. DeneySEL çalışmalar

HM16.15 referans yazılımında çeşitli özelliklerdeki video dizileri encoder_intra_main konfigürasyonunda kodlanarak sonuçlar elde edilmiştir. Literatür çalışmalarıyla dengeli bir kıyaslama yapabilmek için tüm videolarda çerçeve sayısı 20 olarak alınmıştır.

Her bir video dizisi için 28, 30, 32, 34 ve 36 gibi farklı QP değerleriyle sonuçlar elde edilmiştir. Bütün videolar ilk olarak, veri gizleme olmaksızın bu QP değerleriyle orijinal biçimde kodlanmıştır. Daha sonra aynı QP değerleri için, önerilen yüksek kapasiteli veri gizleme yöntemi videolara uygulanmıştır. Her bir I-çerçevenin parlaklık kanalına ait CTU'lar sırayla elde edilerek veri gizleme yöntemi gerçekleştirilmiştir. Veri gizlemenin başarımı için nesnel ve öznel ölçütlere bakılmıştır. Nesnel başarımlar değerlendirilmesinde, farklı QP değerleri için önerilen veri gizleme yöntemi ve orijinal video kodlama işleminin sonuçlarıyla elde edilen oran-bozunum grafikleri Şekil 5.5'te gösterilmektedir. Şekil 5.5'teki grafikler incelendiğinde aynı QP değerleri için PSNR değerlerinin neredeyse aynı olduğu, buna karşılık bit oranı değerinde artış olduğu görülmektedir. Bit oranındaki artışın en önemli sebebi gizlenen veri miktarının yüksek olmasıdır. Bu sebeple gizlenen veri kapasitesine göre bit oranı değişimi için kullanılan



Şekil 5.5. Önerilen yöntemin oran-bozunum başarımı



Şekil 5.5. (Devam) Önerilen yöntemin oran-bozunum başarımı

BIR ile başarımlar değerlendirilmiştir. Tablo 5.3'te BasketballPass dizisi için başarımlar sonuçları ve benzer bir çalışma olan [138] yöntemiyle karşılaştırması verilmiştir.

Tablo 5.3. BasketballPass test dizisinin farklı QP değerleri için veri gizleme başarımları

QP	Δ PSNR (dB)		BIR($\times 10^{-3}$)		Kapasite (bit)	
	Önerilen Yöntem	[138]	Önerilen Yöntem	[138]	Önerilen Yöntem	[138]
28	0,030	-0,200	0,52	0,86	19928	2920
30	-0,001	-0,350	0,63	1,20	16530	2184
32	-0,001	-0,230	0,78	1,04	13298	1412
34	-0,020	-0,380	0,91	1,99	10714	1184
36	-0,008	-0,210	1,15	2,68	8340	652

Önerilen yöntem, aynı videoya tüm QP değerleri için ortalama 8 kat daha fazla veri gizlemektedir. Buna rağmen, Δ PSNR değerleri incelendiğinde, önerilen yöntemin videodaki görsel kaliteyi neredeyse hiç bozmadığı görülmektedir. [138] çalışması ise tüm QP değerleri için ortalama 0,27 dB'lık bozulmaya sebep olmaktadır. Yüksek kapasiteli veri gizlenmesine rağmen BIR değeri de yarı-yarıya daha düşüktür. BIR değeri ne kadar düşük olursa gizleme başarımları da o kadar yüksektir, dolayısıyla önerilen yöntem yüksek kapasite değerine rağmen iyi başarımlar değerleri sunmaktadır.

Önerilen yöntem BasketballPass videosu için farklı QP değerlerinde etkili sonuçlar sunmaktadır. QP değerinin artmasıyla, videodaki sıkıştırma oranı arttığı için veri gizleme kapasitesi düşmektedir. Yüksek QP değerinde PSNR ve bit oranı değerleri düşük olduğu için gizlenen veri BIR değerini daha fazla arttırmaktadır.

Tablo 5.4. Önerilen yöntemin çeşitli test dizilerinde QP=32 için kapasite (bit) başarımı

Video Dizisi	Önerilen Yöntem	[138]	[139]	[109]	[125]
Traffic	236058	47300	-	10770	1560
PeopleOnStreet	393966	45972	-	64935	16110
Cactus	174512	20184	-	7500	2550
BQTerrace	337494	61644	-	4905	390
Johnny	39644	2616	-	-	-
FourPeople	68458	8500	-	-	-
RaceHorsesC	57010	6140	-	6885	1770
BasketballDrill	46710	4040	-	-	-
Keiba	14494	3664	1248	-	-
BasketballPass	13298	1412	1865	-	-

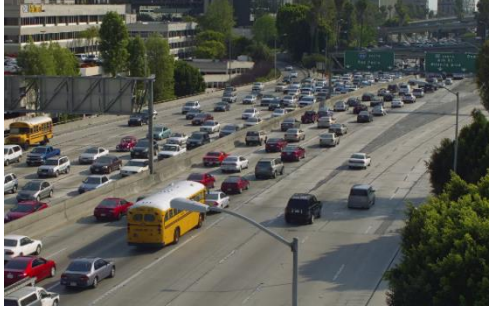
Dolayısıyla QP arttıkça BIR değeri de artmaktadır. Başarımın detaylı analizi için farklı özellikteki videolar için elde edilen sonuçlar Tablo 5.4, 5.5 ve 5.6'da verilmektedir. Tablolardaki sonuçların tamamı QP=32 için elde edilmiştir.

Tablo 5.5. Önerilen yöntemin çeşitli test dizilerinde QP=32 için Δ PSNR (dB) başarımı

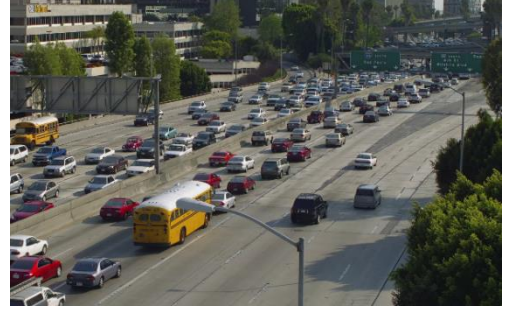
Video Dizisi	Önerilen Yöntem	[138]	[139]	[109]	[125]
Traffic	-0,005	-0,320	-	-0,048	-0,073
PeopleOnStreet	-0,001	-0,310	-	-0,125	-0,592
Cactus	-0,008	-1,060	-	-0,041	-1,672
BQTerrace	-0,005	-0,340	-	-0,026	-0,124
Johnny	-0,008	-0,420	-	-	-
FourPeople	-0,014	-0,640	-0,796	-	-
RaceHorsesC	0	-0,360	-	-0,107	-0,006
BasketballDrill	-0,001	-1,860	-2,282	-	-
Keiba	0,002	-0,340	-0,140	-	-
BasketballPass	-0,001	-0,230	-0,290	-	-

Tablo 5.4 ve Tablo 5.5 incelendiğinde önerilen yöntemin ulaştığı veri gizleme kapasitesi değerinin diğer yöntemlere göre oldukça yüksek olduğu, ayrıca önerilen yöntemin görsel kaliteyi koruduğu görülmektedir.

Önerilen yöntem, Cactus, Johnny ve BasketballDrill dizileri için en iyi sonuçları verirken, Traffic, BQTerrace ve RaceHorsesC dizilerinde en iyi sonuçları [138] çalışması, BasketballPass dizisinde en iyi sonucu [139] çalışması vermektedir. PeopleOnStreet, FourPeople ve Keiba dizilerindeki en iyi BIR başarımını ise önerilen yöntem ve [138] çalışması birlikte sunmaktadırlar.



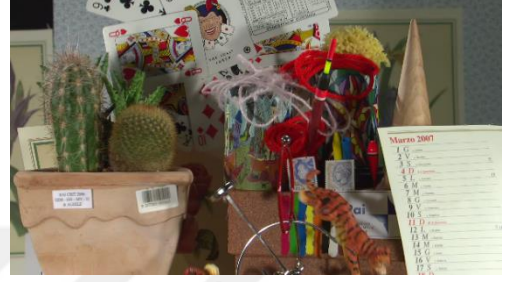
(a) QP28



(b) QP28



(c) QP30



(d) QP30



(e) QP32



(f) QP32



(g) QP34



(h) QP34



(i) QP36



(j) QP36

Şekil 5.6. Önerilen yöntemin öznel değerlendirilmesi. Orijinal kodlananlar (solda), önerilen veri gizleme ile kodlananlar (sağda). (a)-(b): Traffic QP=28, (c)-(d): Cactus QP=30, (e)-(f): Johnny QP=32, (g)-(h): RaceHorsesC QP=34 ve (i)-(j): Keiba QP=36

Veri gizleme yöntemlerinde nesnel değerlendirmenin yanında öznel değerlendirme işlemi de önemlidir. Öznel değerlendirme için video çerçevelerindeki değişim karşılaştırılmaktadır. Şekil 5.6'da orijinal kodlama ve veri gizleme sonrası kodlama sonuçlarında elde edilen videoların çerçeve örnekleri karşılaştırılmaktadır. Burada her bir video için kullanılan QP değerlerinden sadece birisi için ilk çerçevelerin karşılaştırılması gösterilmektedir.

Tablo 5.6. Önerilen yöntemin çeşitli test dizilerinde QP=32 için BIR($\times 10^{-3}$) başarımı

Video Dizisi	Önerilen Yöntem	[138]	[139]	[109]	[125]
Traffic	0,03	0,02	-	1,10	0,16
PeopleOnStreet	0,02	0,02	-	0,95	0,03
Cactus	0,05	0,06	-	1,68	0,28
BQTerrace	0,03	0,01	-	1,10	0,46
Johnny	0,22	0,37	-	-	-
FourPeople	0,14	0,14	-	-	-
RaceHorsesC	0,15	0,14	-	8,52	0,36
BasketballDrill	0,18	0,24	-	-	-
Keiba	0,76	0,76	0,83	-	-
BasketballPass	0,78	0,67	0,66	-	-

Şekil 5.6 değerlendirildiğinde, görülmektedir ki önerilen yöntemle videoya gizli mesaj eklenmesi durumunda, videoda insan görsel sistemiyle algılanabilecek bir değişim olmamaktadır. Detay kısımlarındaki küçük değişimler ise doğrudan hissedilemeyecek düzeyde kalmaktadır.

5.3.3. Sonuçlar

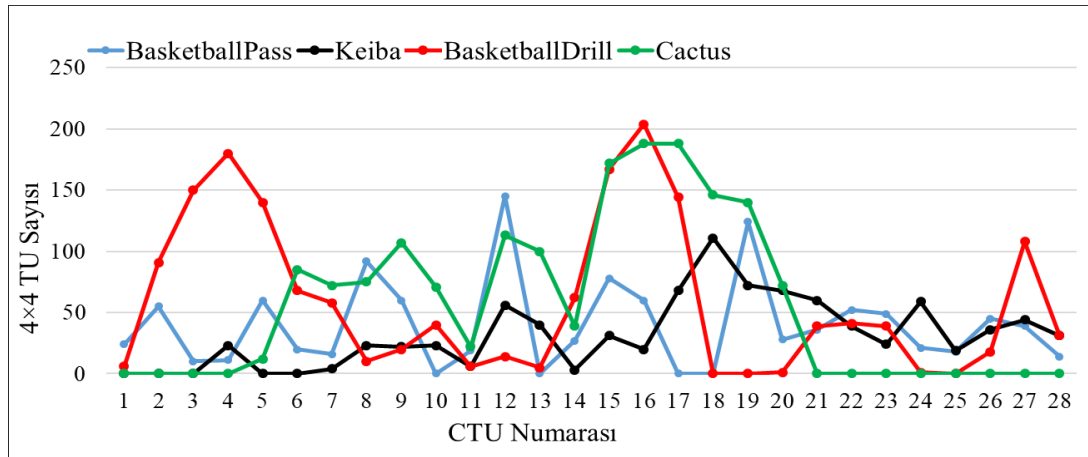
Deneyssel çalışmalar göz önüne alındığında, önerilen yöntem veri gizleme olmadan yapılan video kodlama işlemine kıyasla görsel kaliteyi korurken, gizlenen mesajın büyüklüğüne göre bit oranını ve stego videonun boyutunu arttırmıştır. Literatürdeki yöntemlerle kıyaslandığında ise önerilen yöntem BasketballPass dizisi için farklı QP değerlerinde yüksek kapasitenin yanında BIR ve Δ PSNR anlamında da daha iyi sonuçlar vermiştir. Farklı video dizileri için elde edilen sonuçlarda yüksek kapasite ve düşük bozunum özelliklerinin korunduğu ortaya çıkmıştır.

5.4. Yüksek Verimlilik Temelli Veri Gizleme Yöntemi

Bu bölümdeki çalışmaların amacı veri gizleme sonrası videodaki bit oranı artışını daha düşük seviyelerde tutmaktır. Önceki bölümde yapılan çalışmalarda veri gizleme kapasitesi oldukça yüksek çıkarken, BIR değerleri literatürdeki bazı yöntemlere yakın veya daha düşük olmaktadır. HEVC videolarında 4×4 TU bloklarında hata yayılımı olmadan daha düşük BIR değerine sahip bir veri gizleme işlemi için ME (1, c, m) şemasında taşıyıcı sayısını arttırmak en iyi çözümdür. Çünkü Şekil 5.2’den de görüldüğü gibi, taşıyıcı sayısı arttığında gizleme oranı (ER) bir miktar düşerken, gizleme verimliliği (ER) artmaktadır. Kullanılacak ME (1, c, m) şemasının tespiti için bazı ön analizlerin yapılması gerekmektedir.

5.4.1. CTU içindeki TU bloklarının analizi

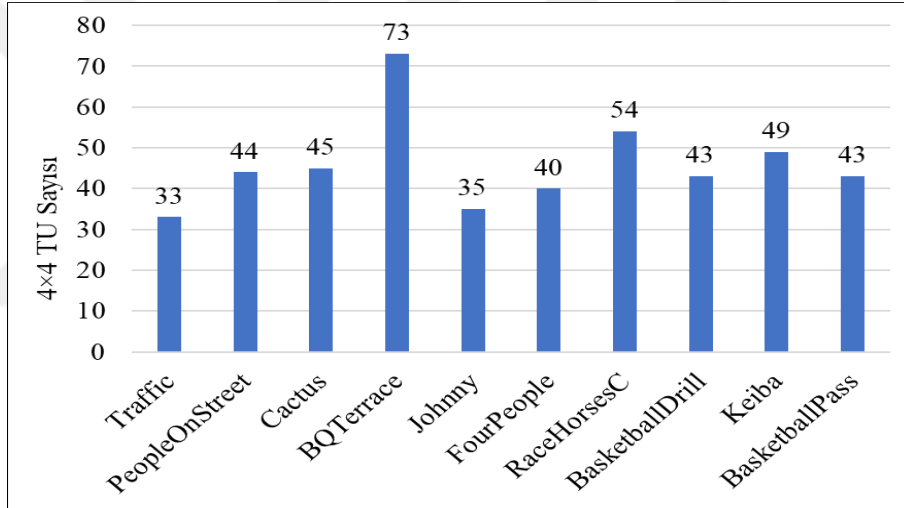
Tez çalışmasındaki bütün veri gizleme işlemleri CTU içerisinde yapılmaktadır. Veri gizleme için CTU içerisindeki TU blokları c’li gruplara ayrılmaktadır. Diğer bir ifadeyle ME (1, c, m) şemasında taşıyıcı sayısını gösteren c değişkeni CTU içerisindeki TU’lardan alınan katsayılardan oluşmaktadır. Önerilen yüksek verimliliğe sahip veri gizleme yönteminde c değerinin seçilmesi için HM referans yazılımı kullanılarak videolarda çeşitli analizler yapılmıştır.



Şekil 5.7. Örnek video dizilerinin 1. çerçevelerindeki CTU’larda bulunan 4×4 TU sayıları

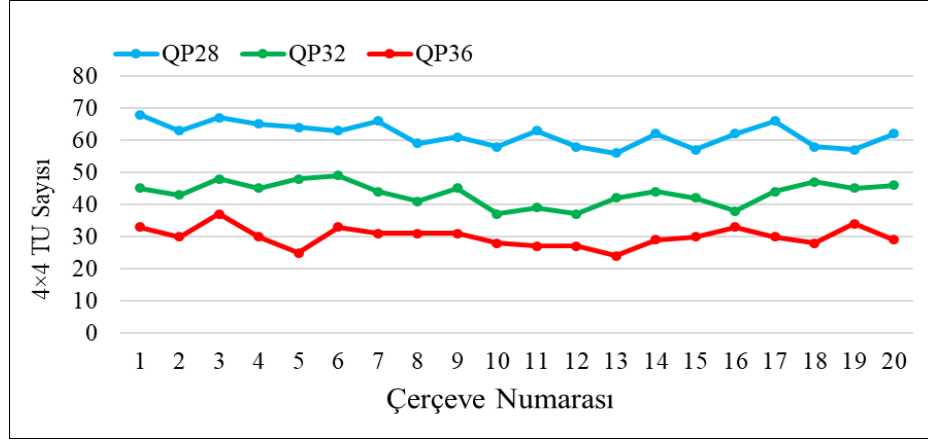
HEVC’de CTU boyutları 64×64 olduğu için bir CTU içerisinde en fazla 256 tane TU bulunabilmekle birlikte, yapılan analizlerde tamamı 4×4 TU’lara bölünen CTU sayısı nadirdir. Şekil 5.7’de QP=32 için Basketballpass, Keiba, BasketballDrill ve Cactus

video dizilerinin ilk çerçevelerindeki CTU’larda bulunan TU sayıları gösterilmektedir. Şekil 5.7 ve HEVC dörtlü ağaç yapısı göz önüne alındığında, yüksek verim için önerilen yöntemde kullanılacak ME şemasındaki, c değeri hiçbir CTU’da 255 olmadığı gibi sınırlı sayıda CTU için 127’den büyük olmaktadır. Verimi artırırken, kapasite değerinin aşırı düşmemesi için mümkün olan en fazla sayıda CTU kullanılmalıdır. Oysaki c’nin 127 olması durumunda, Keiba dizisinde uygun CTU bulunmazken, BasketballPass dizisinde sadece bir CTU, BasketballDrill dizisinde sadece altı CTU ve Cactus dizisinde sadece beş CTU veri gizleme amacıyla kullanılabilir. Oldukça anlamsız bir kapasiteye karşılık gelmesi sebebiyle $c < 127$ ve dolayısıyla $m < 7$ olmaktadır. Önerilen yöntemde, m gizli bit sayısı 3, 4, 5 veya 6 için c değerleri sırasıyla 7, 15, 31, veya 63 olabilir.



Şekil 5.8. Video dizilerinin tüm çerçevelerindeki CTU’larda bulunan ortalama 4x4 TU sayıları

Şekil 5.8’de QP=32 için kodlanan video dizilerinin tüm çerçevelerindeki CTU’ların içinde bulunan ortalama TU sayıları gösterilmektedir. Ortalamalar hesaplanırken içinde 4x4 TU bulunmayan CTU’lar hesaba dahil edilmemesine rağmen, sadece BQTerrace videosunun ortalama CTU sayısı 63’ten büyük çıkmaktadır. Her ne kadar çerçeve içindeki bazı CTU’ların değerleri 63 üzerinde olacaksa da c değişkeninin 63 olması durumunda videolardaki birçok CTU kullanılamamaktadır. Dolayısıyla ME (1, 63, 6)’nın kullanılması da uygun değildir.



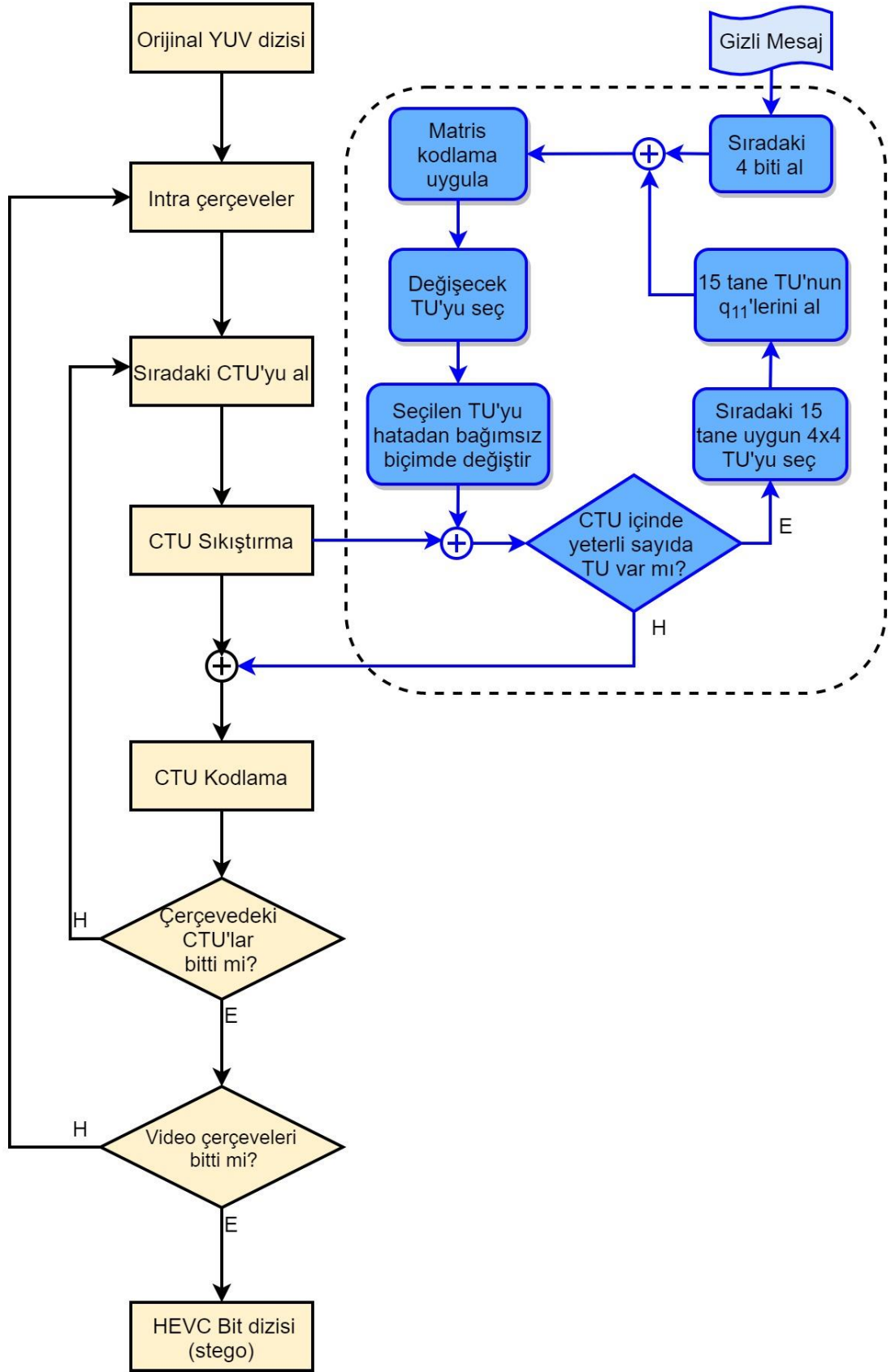
Şekil 5.9. Farklı QP’ler ile kodlanan BasketballPass dizisinin çerçevelerinde bulunan CTU’lardaki ortalama 4x4 TU sayıları

Şekil 5.9’da BasketballPass video dizisinin 28, 32 ve 36 QP değerlerine göre kodlanması durumunda CTU’ları içerisindeki TU sayıları verilmektedir. Her ne kadar düşük QP değerlerinde TU sayıları yüksek çıksa da QP değeri arttıkça aynı sayıda TU bulmak mümkün değildir. Önerilen veri gizleme için seçilecek taşıyıcı değerinin bütün kodlama özelliklerine uygun olması gerekmektedir. Yüksek verimliliğe sahip bir veri gizleme yöntemi için c değişkeninin mümkün olan en yüksek değerde olması gerekmektedir. Bununla beraber, HEVC ile kodlanan video dizilerinde yukarıda anlatılan sınırlamalar sebebiyle c değişkeninin 63 veya 31 olması durumunda uygun özelliklere sahip TU sayısı oldukça azalmaktadır. Bu sebeple ME(1, c , m) şemasında en uygun c değerinin 15 olacağı ortaya çıkmaktadır.

5.4.2. Önerilen yöntem

Önceki bölümde, gizleme verimliliğinin en yüksek oranda tutulması için c değerinin 15 olması gerektiği tespit edilmişti. Bu durumda Eşitlik (5.1)’e göre m değeri 4 bit olarak elde edilmektedir. Bu sebeple, tez çalışmasının bu kısmında önerilen yüksek verimliliğe sahip veri gizleme yöntemi için ME(1,15,4) şeması kullanılmaktadır. Bu şemada Hamming kod matrisi $H_{4 \times 15}$ Eşitlik (5.22)’deki gibi alınmaktadır.

$$H_{4 \times 15} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (5.22)$$



Şekil 5.10. Önerilen yöntemin akış şeması

Şekil 5.10’da önerilen yöntemin akış şemasında normal video kodlama işlemi devam ederken ilk olarak HEVC kodlamanın CTU sıkıştırma işlemi aşamasında, CTU’daki 4×4 TU’lar kontrol edilir. QDST katsayılarından en az biri sıfırdan farklı olan TU’ların sayısı ve tamamının yerleri bulunur. Veri gizleme yöntemi için belirlenen c değerine göre 15 tane 4×4 TU’nun her birindeki q_{11} katsayıları alınır. Alınan katsayıların LSB değerleri, matris kodlama işlemlerinde kullanılmak üzere tespit edilir. Matris kodlama sonucu değiştirilmesine karar verilen TU’nun QDST katsayılarına hata yayılımını önleyen ε matrisi eklenerek veri gizlenir ve veri gizleme için sıradaki 15 tane TU’ya geçilir. Bu işlemler CTU içerisindeki uygun TU sayısı 15’ten az olana kadar tekrar eder. CTU içerisindeki TU’lar bitince ilgili CTU’nun kodlanma işlemi tamamlanır ve diğer CTU’ya geçilerek benzer işlemler tekrar eder. Çerçeveadaki tüm CTU’ların kodlanması bitince bir sonraki çerçeveye geçilerek veri gizleme işlemine devam edilir. Videodaki tüm çerçevelere veri gizlenen kadar bu işlemler devam ederek veri gizlenmiş HEVC bit dizisi elde edilir.

Veri gizleme işleminde değişecek TU seçimi için ilk olarak gizli mesaj bitleri mesaj bitleri Eşitlik (5.23)’teki gibi $b_{4 \times 1}$ ’lik vektörlere ayrılmaktadır.

$$b_{4 \times 1} = (b_1, b_2, b_3, b_4) \quad (5.23)$$

Bağımlılık vektörünü bulmak için olan taşıyıcı örtü vektörünün oluşturulması gerekmektedir. Taşıyıcı vektörü $r_{1 \times 15}$ için TU’lar içerisindeki QDST katsayılarının (q_{11}) LSB değerleri kullanılmaktadır. I-çerçevesinin parlaklık bileşenine ait CTU içerisindeki TU’lar $c=15$ ’li gruplara ayrıldıktan sonra, ilgili TU’ların LSB değerlerinden oluşturulan taşıyıcı vektör Eşitlik (5.24)’teki gibi elde edilmektedir.

$$r_{1 \times 15} = (q_{TU1}, q_{TU2}, q_{TU3}, q_{TU4}, q_{TU5}, q_{TU6}, q_{TU7}, q_{TU8}, q_{TU9}, q_{TU10}, q_{TU11}, q_{TU12}, q_{TU13}, q_{TU14}, q_{TU15}) \quad (5.24)$$

Bu değerler Eşitlik (5.5)’te yerine yerleştirilip, gerekli hesaplamalar yapıldığında bağımlılık vektörleri,

$$\begin{aligned}
d_1 &= (q_{TU1} + q_{TU3} + q_{TU5} + q_{TU7} + q_{TU9} + q_{TU11} + q_{TU13} + q_{TU15}) \bmod 2 \\
d_2 &= (q_{TU2} + q_{TU3} + q_{TU6} + q_{TU7} + q_{TU10} + q_{TU11} + q_{TU14} + q_{TU15}) \bmod 2 \\
d_3 &= (q_{TU4} + q_{TU5} + q_{TU6} + q_{TU7} + q_{TU12} + q_{TU13} + q_{TU14} + q_{TU15}) \bmod 2 \\
d_4 &= (q_{TU8} + q_{TU9} + q_{TU10} + q_{TU11} + q_{TU12} + q_{TU13} + q_{TU14} + q_{TU15}) \bmod 2
\end{aligned} \tag{5.25}$$

şeklinde bulunmaktadır. Daha sonra, Eşitlik (5.6) yardımıyla değiştirilecek TU'nun bulunması gerekmektedir. Gizli bit değerleri ve bağımlılık değerleri kullanılarak, değişecek TU'nun yeri,

$$p = (d_4 \oplus b_4) \times 2^3 + (d_3 \oplus b_3) \times 2^2 + (d_2 \oplus b_2) \times 2^1 + (d_1 \oplus b_1) \times 2^0 \tag{5.26}$$

şeklinde hesaplanmaktadır. Hangi koşullarda hangi TU'nun, değişeceği Tablo 5.7'de gösterilmektedir.

Tablo 5.7. ME(1,15,4) için değişecek bloğun seçimi

Kontrol				p	Değiştirilen TU
$d_4=b_4$,	$d_3=b_3$,	$d_2=b_2$ ve	$d_1=b_1$ ise	0	-
$d_4=b_4$,	$d_3=b_3$,	$d_2=b_2$ ve	$d_1 \neq b_1$ ise	1	TU ₁
$d_4=b_4$,	$d_3=b_3$,	$d_2 \neq b_2$ ve	$d_1=b_1$ ise	2	TU ₂
$d_4=b_4$,	$d_3=b_3$,	$d_2 \neq b_2$ ve	$d_1 \neq b_1$ ise	3	TU ₃
$d_4=b_4$,	$d_3 \neq b_3$,	$d_2=b_2$ ve	$d_1=b_1$ ise	4	TU ₄
$d_4=b_4$,	$d_3 \neq b_3$,	$d_2=b_2$ ve	$d_1 \neq b_1$ ise	5	TU ₅
$d_4=b_4$,	$d_3 \neq b_3$,	$d_2 \neq b_2$ ve	$d_1=b_1$ ise	6	TU ₆
$d_4=b_4$,	$d_3 \neq b_3$,	$d_2 \neq b_2$ ve	$d_1 \neq b_1$ ise	7	TU ₇
$d_4 \neq b_4$,	$d_3=b_3$,	$d_2=b_2$ ve	$d_1=b_1$ ise	8	TU ₈
$d_4 \neq b_4$,	$d_3=b_3$,	$d_2=b_2$ ve	$d_1 \neq b_1$ ise	9	TU ₉
$d_4 \neq b_4$,	$d_3=b_3$,	$d_2 \neq b_2$ ve	$d_1=b_1$ ise	10	TU ₁₀
$d_4 \neq b_4$,	$d_3=b_3$,	$d_2 \neq b_2$ ve	$d_1 \neq b_1$ ise	11	TU ₁₁
$d_4 \neq b_4$,	$d_3 \neq b_3$,	$d_2=b_2$ ve	$d_1=b_1$ ise	12	TU ₁₂
$d_4 \neq b_4$,	$d_3 \neq b_3$,	$d_2=b_2$ ve	$d_1 \neq b_1$ ise	13	TU ₁₃
$d_4 \neq b_4$,	$d_3 \neq b_3$,	$d_2 \neq b_2$ ve	$d_1=b_1$ ise	14	TU ₁₄
$d_4 \neq b_4$,	$d_3 \neq b_3$,	$d_2 \neq b_2$ ve	$d_1 \neq b_1$ ise	15	TU ₁₅

Değişecek TU bulunduktan sonra, ilgili TU'ya Eşitlik (5.15)'teki ε matrisi eklenerek hata yayılımından bağımsız biçimde veri gizlenmiştir. Bütün video çerçevelerine veri gizleme işlemi tamamlandıktan sonra stego video alıcıya gönderilmektedir. Alıcıdaki veri çıkartma işlemi de veri gizlemeye benzer biçimde yapılmaktadır. HEVC video kod çözme işlemi aşamasında stego bit dizisinden intra çerçeveler çözülür ve sırasıyla

CTU'ların kodları çözülmeye başlanır. Çözülen CTU içindeki TU'lar 15'li gruplar halinde sıralanarak her birinin q_{11} katsayılarının LSB değerleri elde edilerek veri gizlenmiş taşıyıcı vektörü r_{yeni} oluşturulur. Daha sonra Hamming kod matrisi $H_{4 \times 15}$ ve r_{yeni} kullanılarak Eşitlik (5.8)'e göre gizlenen 4 bit elde edilir. CTU'lardaki bütün 15'li TU gruplarından çıkarılan 4 bitlik mesajlar birleştirilerek gizli mesaj oluşturulur.

5.4.3. Deneyisel çalışmalar

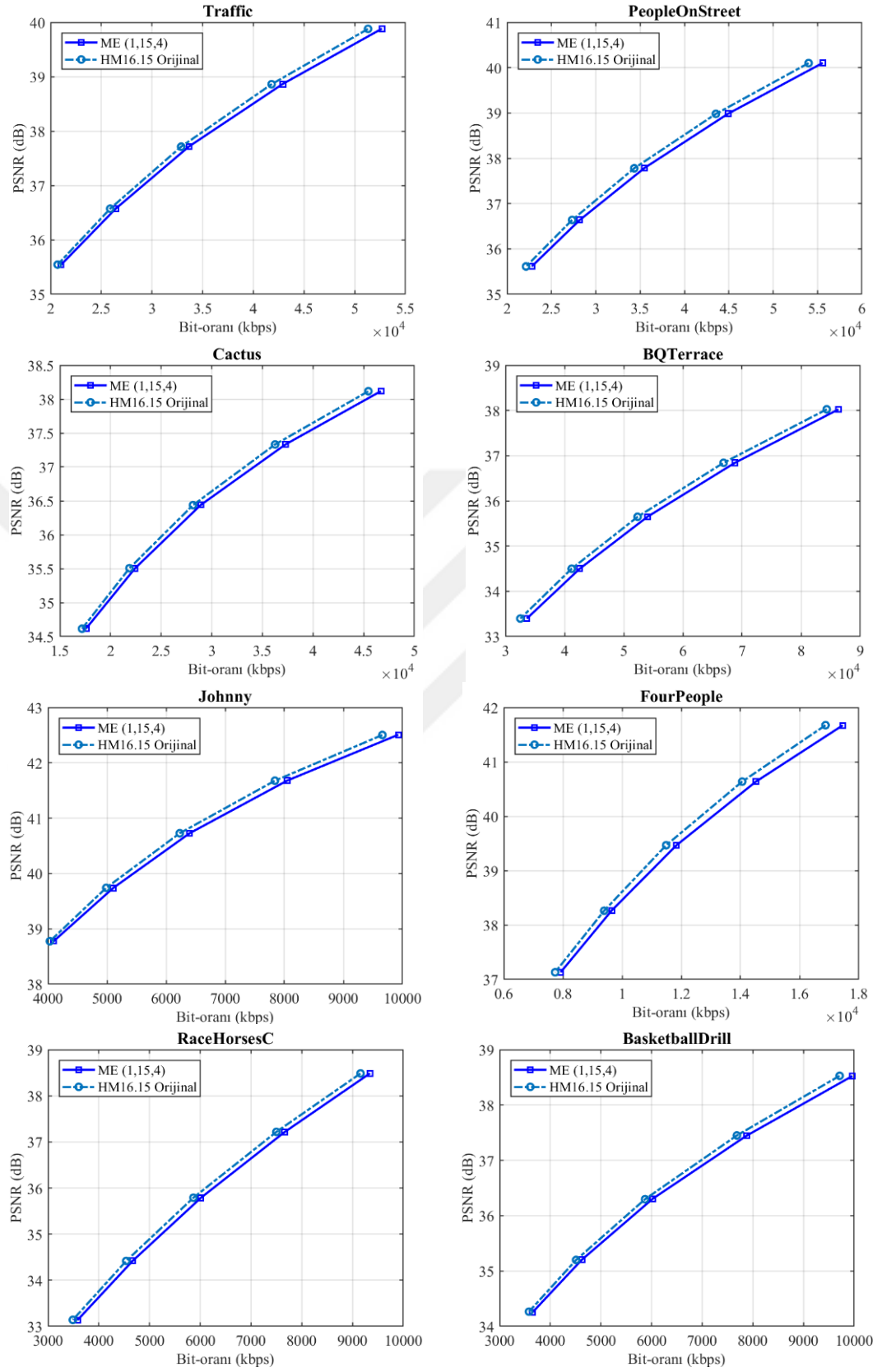
Bir önceki önerilen yöntemde olduğu gibi bu yöntemde de tüm deneyisel çalışmalar HM16.15 referans yazılımındaki encoder_intra_main konfigürasyonu ile gerçekleştirilmiştir. Farklı çözünürlüklere sahip test video dizilerinin ilk 20 çerçevesi 28, 30, 32, 34 ve 36 gibi farklı QP değerleriyle kodlanmıştır. Video kodlama işlemleri hem önerilen veri gizleme yöntemi için hem de veri gizleme olmadan orijinal HM16.15 için gerçekleştirilmiştir.

Önerilen veri gizleme yöntemi, her bir I-çerçevenin parlaklık kanalından elde edilen CTU'ların içindeki TU'lara uygulanmaktadır. Tablo 5.7'ye göre değişecek TU tespit edildikten sonra, ilgili TU'daki bir QDST katsayısı veri gizleme için, diğer katsayılar ise hata yayılımını önlemek için Eşitlik (5.15)'teki ϵ matrisi kullanılarak değiştirilmektedir. Şekil 5.11'de önerilen veri gizleme yönteminin orijinal video kodlama yöntemine göre oran-bozunum grafikleri gösterilmiştir.

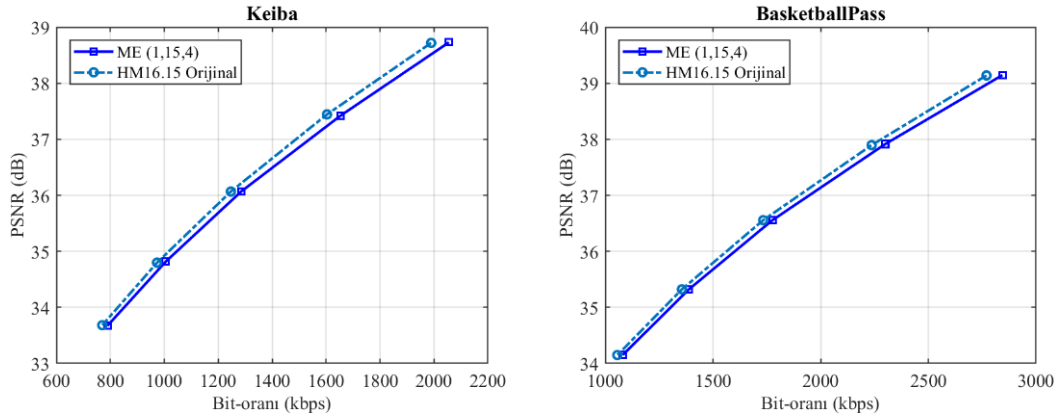
Tablo 5.8. BasketballPass test dizisinin farklı QP değerleri için veri gizleme başarımı

QP	$\Delta PSNR$ (dB)		$BIR(\times 10^{-3})$		Kapasite (bit)	
	Önerilen Yöntem	[138]	Önerilen Yöntem	[138]	Önerilen Yöntem	[138]
28	0,006	-0,200	0,37	0,86	7276	2920
30	0,019	-0,350	0,47	1,20	6064	2184
32	0	-0,230	0,53	1,04	4664	1412
34	-0,002	-0,380	0,61	1,99	3848	1184
36	0,004	-0,210	0,84	2,68	2884	652

Şekil 5.11'deki grafikler değerlendirildiğinde farklı QP değerleri için önerilen veri gizleme yöntemiyle kodlanan videolar ile veri gizleme olmadan kodlanan videolar arasında PSNR farkının çok az olduğu görülmektedir. Önerilen yöntemin yüksek verimlilik ve düşük değişim özelliklerini sağladığı bit oranı artışı değerlerinden anlaşılmaktadır.



Şekil 5.11. Önerilen yöntemin oran-bozunum başarımı



Şekil 5.11. (Devam) Önerilen yöntemin oran-bozunum başarımı

Tablo 5.8’de, bütün QP değerleri için literatürdeki benzer çalışma olan [138] yöntemiyle kıyaslanarak başarımlar değerlendirilmiştir. Tablodaki değerler incelendiğinde önerilen yöntemin, farklı QP değerleri için Δ PSNR, BIR ve kapasite anlamında oldukça üstün sonuçlar verdiği görülmektedir.

BasketballPass dizisi için farklı QP değerlerindeki BIR değerleri göz önüne alındığında, [138] çalışmasına göre gizlenen veri miktarı yaklaşık 2,5 kat daha fazla olmasına rağmen, BIR değeri bu çalışmanın yarısından daha düşüktür. Dolayısıyla önerilen yöntemin yüksek verimlilik sunduğu görülmektedir. BIR değerinin detaylı analizi için Tablo 5.9’da farklı video dizileri için elde edilen sonuçlar değerlendirilmiştir. Tablo 5.9’daki sonuçlar değerlendirildiğinde, önerilen yöntemin videolardaki BIR değerini literatür yöntemlerine göre aşağıda tuttuğu görülmektedir.

Tablo 5.9. Önerilen yöntemin çeşitli test dizilerinde QP=32 için $BIR(\times 10^{-3})$ başarımı

Video Dizisi	Önerilen Yöntem	[138]	[139]	[109]	[125]
Traffic	0,02	0,02	-	1,10	0,16
PeopleOnStreet	0,02	0,02	-	0,95	0,03
Cactus	0,04	0,06	-	1,68	0,28
BQTerrace	0,02	0,01	-	1,10	0,46
Johnny	0,18	0,37	-	-	-
FourPeople	0,11	0,14	-	-	-
RaceHorsesC	0,11	0,14	-	8,52	0,36
BasketballDrill	0,14	0,24	-	-	-
Keiba	0,54	0,76	0,83	-	-
BasketballPass	0,53	0,67	0,66	-	-

Bu sonuçlar Tablo 5.10’da verilen kapasite sonuçlarıyla birlikte değerlendirildiğinde, önerilen yöntemin farklı video dizilerinde yüksek kapasiteyle birlikte yüksek verimlilik sunduğu dolayısıyla BIR değerlerinin düştüğü görülmektedir. Sadece BQTerrace videosunda [138] çalışması daha düşük BIR başarımı sunarken, Ultra HD çözünürlüğe sahip olan Traffic ve PeopleOnStreet video dizilerinde [138] çalışmasına göre daha fazla veri gizlenmesine rağmen aynı değerler elde edilmektedir.

Tablo 5.10. Önerilen yöntemin çeşitli test dizilerinde QP=32 için kapasite (bit) başarımı

Video Dizisi	Önerilen Yöntem	[138]	[139]	[109]	[125]
Traffic	105860	47300	-	10770	1560
PeopleOnStreet	169056	45972	-	64935	16110
Cactus	72172	20184	-	7500	2550
BQTerrace	133336	61644	-	4905	390
Johnny	15320	2616	-	-	-
FourPeople	28624	8500	-	-	-
RaceHorsesC	23460	6140	-	6885	1770
BasketballDrill	18200	4040	-	-	-
Keiba	5852	3664	1248	-	-
BasketballPass	4664	1412	1865	-	-

Diğer bütün videolar için literatür yöntemlerine göre daha iyi başarımlara ulaşılmaktadır. Bunlara ek olarak, Tablo 5.11’de verilen Δ PSNR değerleri incelendiğinde, önerilen yöntemin farklı videolarda PSNR değerini neredeyse hiç değiştirmedeği ortaya çıkmaktadır. Literatürdeki yöntemlerin hepsi videoda bozunuma sebep olurken, önerilen yüksek verimliliğe sahip yöntem PSNR değerlerini 0,0023 dB iyileştirmektedir.

Önerilen yöntemin öznel değerlendirme işlemi için video çerçevelerindeki değişim kontrol edilmektedir. Veri gizleme olmaksızın farklı QP değerleriyle orijinal kodlanan video çerçeveleri ve veri gizleme sonucu elde edilen stego videoların kodlama sonuçları Şekil 5.12’de karşılaştırılmaktadır. Şekil 5.12’de, her bir QP değeri için bir video dizisinin çerçeveleri kıyaslanmaktadır. Şekil 5.12’deki çerçeve örnekleri değerlendirildiğinde, daha önce elde edilen nesnel değerlendirme sonuçlarıyla uyumlu olduğu görülmektedir.



(a) QP28



(b) QP28



(c) QP30



(d) QP30



(e) QP32



(f) QP32



(g) QP34



(h) QP34



(i) QP36



(j) QP36

Şekil 5.12. Önerilen yöntemin öznel değerlendirmesi. Orijinal kodlananlar (solda), önerilen veri gizleme ile kodlananlar (sağda). (a)-(b): PeopleOnStreet QP=28, (c)-(d): BQTerrace QP=30, (e)-(f): FourPeople QP=32, (g)-(h): BasketballDrill QP=34 ve (i)-(j): BasketballPass QP=36

Tablo 5.11. Önerilen yöntemin çeşitli test dizilerinde QP=32 için Δ PSNR (dB) başarımı

Video Dizisi	Önerilen Yöntem	[138]	[139]	[109]	[125]
Traffic	0,002	-0,320	-	-0,048	-0,073
PeopleOnStreet	0,006	-0,310	-	-0,125	-0,592
Cactus	0,001	-1,060	-	-0,041	-1,672
BQTerrace	0,001	-0,340	-	-0,026	-0,124
Johnny	0	-0,420	-	-	-
FourPeople	-0,001	-0,640	-0,796	-	-
RaceHorsesC	0	-0,360	-	-0,107	-0,006
BasketballDrill	0,008	-1,860	-2,282	-	-
Keiba	0,006	-0,340	-0,140	-	-
BasketballPass	0	-0,230	-0,290	-	-

5.4.4. Sonuçlar

Bu bölümde önerilen yöntemin hedefi yüksek kapasiteli gizlemeyi daha verimli hale getirmektir. Gizleme verimliliğini arttırmak için daha fazla veri gizlenirken, değişimin daha düşük olması gerekmektedir. Bir TU'daki katsayıyı değiştirerek birden fazla biti gizlerken, mümkün olan en az değişim için ME(1, c, m) şemasındaki en yüksek c değeri çeşitli analizler sonucu 15 olarak tespit edilmiştir.

c değerinin daha büyük seçilmesi durumunda, farklı videolarda veya farklı QP ile yapılan kodlamalarda yeterli TU sayısına sahip CTU sayısı oldukça düştüğü için gizleme kapasitesi düşeceği tespit edilmiştir. Deneysel çalışmalar, önerilen yöntem ile veri gizlenen videodaki PSNR değerinin orijinal kodlama sonuçlarına çok yakın olduğunu ortaya çıkarmaktadır. Değişen TU sayısı azalınca, bit oranı artışı azaldığı için BIR değeri de daha düşük oranlarda kalmıştır.

5.5. Kapasite ve Verimlilik Temelli Veri Gizleme Yöntemi

Bu bölümde önerilen yöntemde, yüksek kapasite ve yüksek verimlilik sunan yöntemleri dengeli duruma getirmek amaçlanmıştır. Denge için önceki bölümlerde kullanılan ME(1, c, m) şemasında c değeri yüksek kapasiteli yöntem ile yüksek verimlilik yöntemlerinin arasında kalan değer olan 7 olarak alınmaktadır.

5.5.1. Önerilen yöntem

Önerilen dengeli yöntemde taşıyıcı TU sayısı 7 olarak alınarak, Eşitlik (5.1)'e göre m değeri 3 bit olarak elde edilmektedir. Bu sebeple, tez çalışmasının bu kısmında önerilen dengeli veri gizleme yöntemi için ME(1, 7, 3) şeması kullanılmaktadır.

Önerilen yöntemin Şekil 5.13'te gösterilen akış şemasında normal video kodlama işlemi devam ederken ilk olarak HEVC kodlamanın CTU sıkıştırma işlemi aşamasında, CTU'daki 4×4 TU'lar kontrol edilir. QDST katsayılarından en az biri sıfırdan farklı olan TU'ların sayısı ve tamamının yerleri bulunur. Veri gizleme yöntemi için belirlenen c değerine göre 7 tane 4×4 TU'nun her birindeki q_{11} katsayıları alınır. Alınan katsayıların LSB değerleri, matris kodlama işlemlerinde kullanılmak üzere tespit edilir. Matris kodlama sonucu değiştirilmesine karar verilen TU'nun QDST katsayılarına hata yayılımını önleyen ε matrisi eklenerek veri gizlenir ve veri gizleme için sıradaki 7 tane TU'ya geçilir. Bu işlemler CTU içerisindeki uygun TU sayısı 7'den az olana kadar tekrar eder. CTU içerisindeki TU'lar bitince ilgili CTU'nun kodlanma işlemi tamamlanır ve diğer CTU'ya geçilerek benzer işlemler tekrar eder. Çerçeveadaki tüm CTU'ların kodlanması bitince bir sonraki çerçeveye geçilerek veri gizleme işlemine devam edilir. Videodaki tüm çerçevelere veri gizlenen kadar bu işlemler devam ederek veri gizlenmiş HEVC bit dizisi elde edilir.

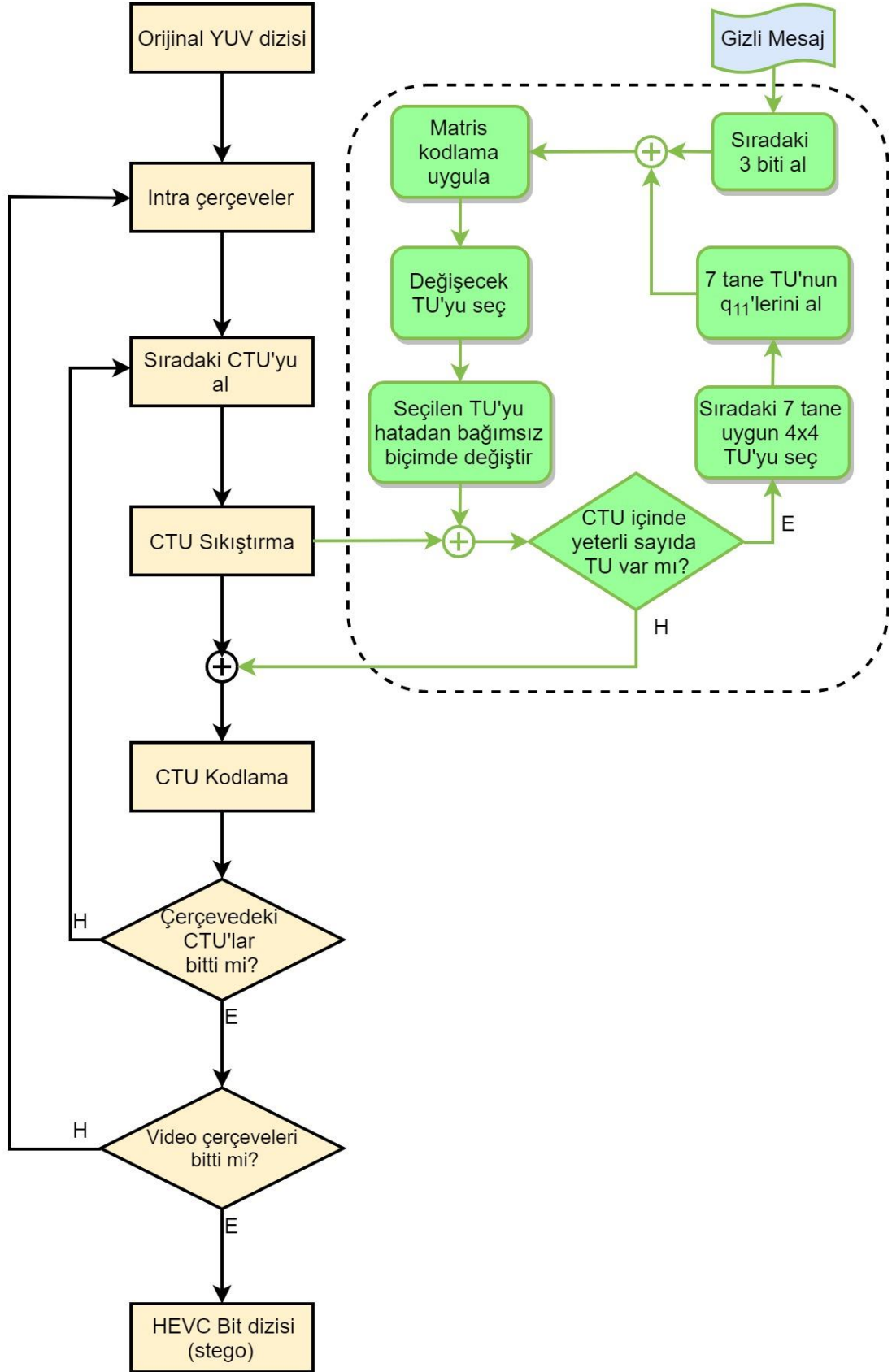
ME (1, 7, 3) şemasında Hamming kod matrisi $H_{3 \times 7}$ Eşitlik (5.27)'deki gibidir. Üçlü olarak gruplanan gizli mesaj mesaj bitleri Eşitlik (5.28)'deki gibi $b_{3 \times 1}$ 'lik vektörlere ayrılmaktadır.

$$H_{3 \times 7} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (5.27)$$

$$b_{3 \times 1} = (b_1, b_2, b_3) \quad (5.28)$$

Taşıyıcı vektörü $r_{1 \times 7}$ için TU'lar içerisindeki QDST katsayılarının (q_{11}) LSB değerleri kullanılarak Eşitlik (5.29)'daki gibi oluşturulmaktadır.

$$r_{1 \times 7} = (q_{TU1}, q_{TU2}, q_{TU3}, q_{TU4}, q_{TU5}, q_{TU6}, q_{TU7}) \quad (5.29)$$



Şekil 5.13. Önerilen yöntemin akış şeması

Hamming kod matrisi ve taşıyıcı vektörü kullanılarak elde edilen bağımlılık vektörleri Eşitlik (5.30)'daki gibi oluşmaktadır.

$$d_1=(q_{TU1}+q_{TU3}+q_{TU5}+q_{TU7})\bmod 2$$

$$d_2=(q_{TU2}+q_{TU3}+q_{TU6}+q_{TU7})\bmod 2 \quad (5.30)$$

$$d_3=(q_{TU4}+q_{TU5}+q_{TU6}+q_{TU7})\bmod 2$$

Bağımlılık vektörleri ve gizli mesaj bitleri kullanılarak değişecek TU'nun taşıyıcı vektör içerisindeki konumu Eşitlik (5.31)'deki gibi hesaplanmaktadır. Hangi koşullarda hangi TU'nun değişeceğinin daha kolay anlaşılması için Tablo 5.12'de bağımlılık vektörü ve gizli bitlerin durumuna göre değişecek TU verilmiştir.

$$p=(d_3 \oplus b_3) \times 2^2 + (d_2 \oplus b_2) \times 2^1 + (d_1 \oplus b_1) \times 2^0 \quad (5.31)$$

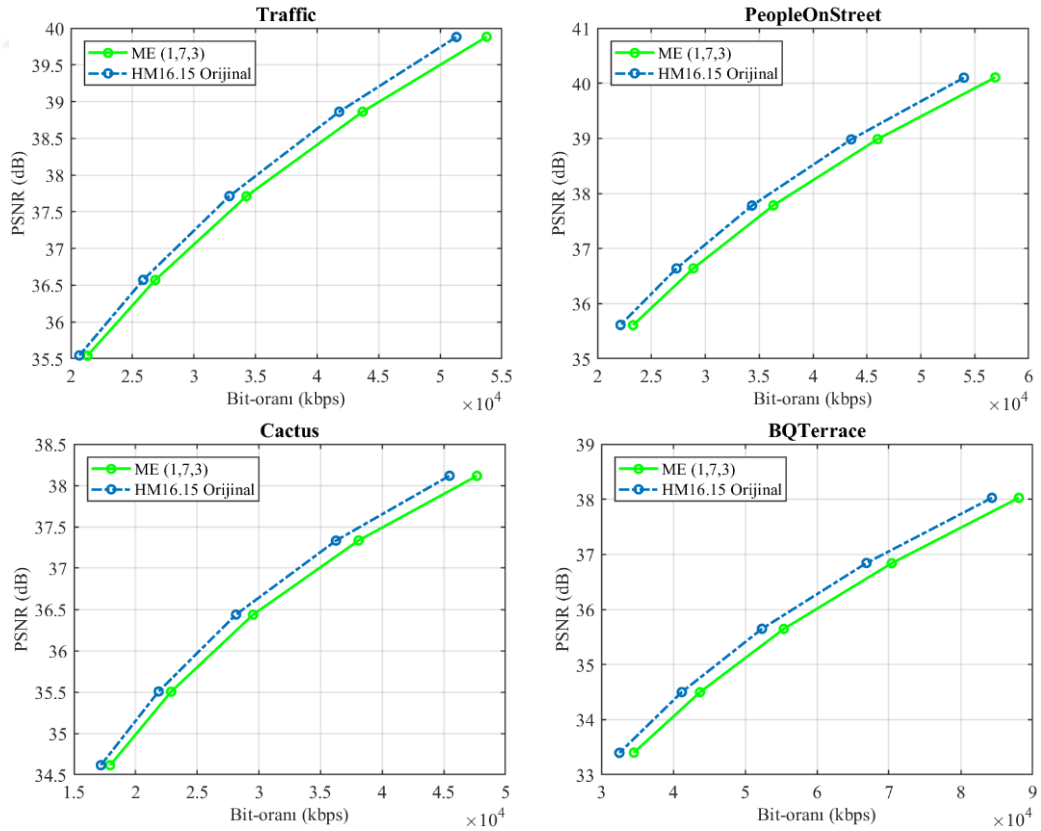
Tablo 5.12. ME(1,7,3) için değişecek bloğun seçimi

	Kontrol		p	Değiştirilen TU
$d_3=b_3$,	$d_2=b_2$ ve	$d_1=b_1$ ise	0	-
$d_3=b_3$,	$d_2=b_2$ ve	$d_1 \neq b_1$ ise	1	TU ₁
$d_3=b_3$,	$d_2 \neq b_2$ ve	$d_1=b_1$ ise	2	TU ₂
$d_3=b_3$,	$d_2 \neq b_2$ ve	$d_1 \neq b_1$ ise	3	TU ₃
$d_3 \neq b_3$,	$d_2=b_2$ ve	$d_1=b_1$ ise	4	TU ₄
$d_3 \neq b_3$,	$d_2=b_2$ ve	$d_1 \neq b_1$ ise	5	TU ₅
$d_3 \neq b_3$,	$d_2 \neq b_2$ ve	$d_1=b_1$ ise	6	TU ₆
$d_3 \neq b_3$,	$d_2 \neq b_2$ ve	$d_1 \neq b_1$ ise	7	TU ₇

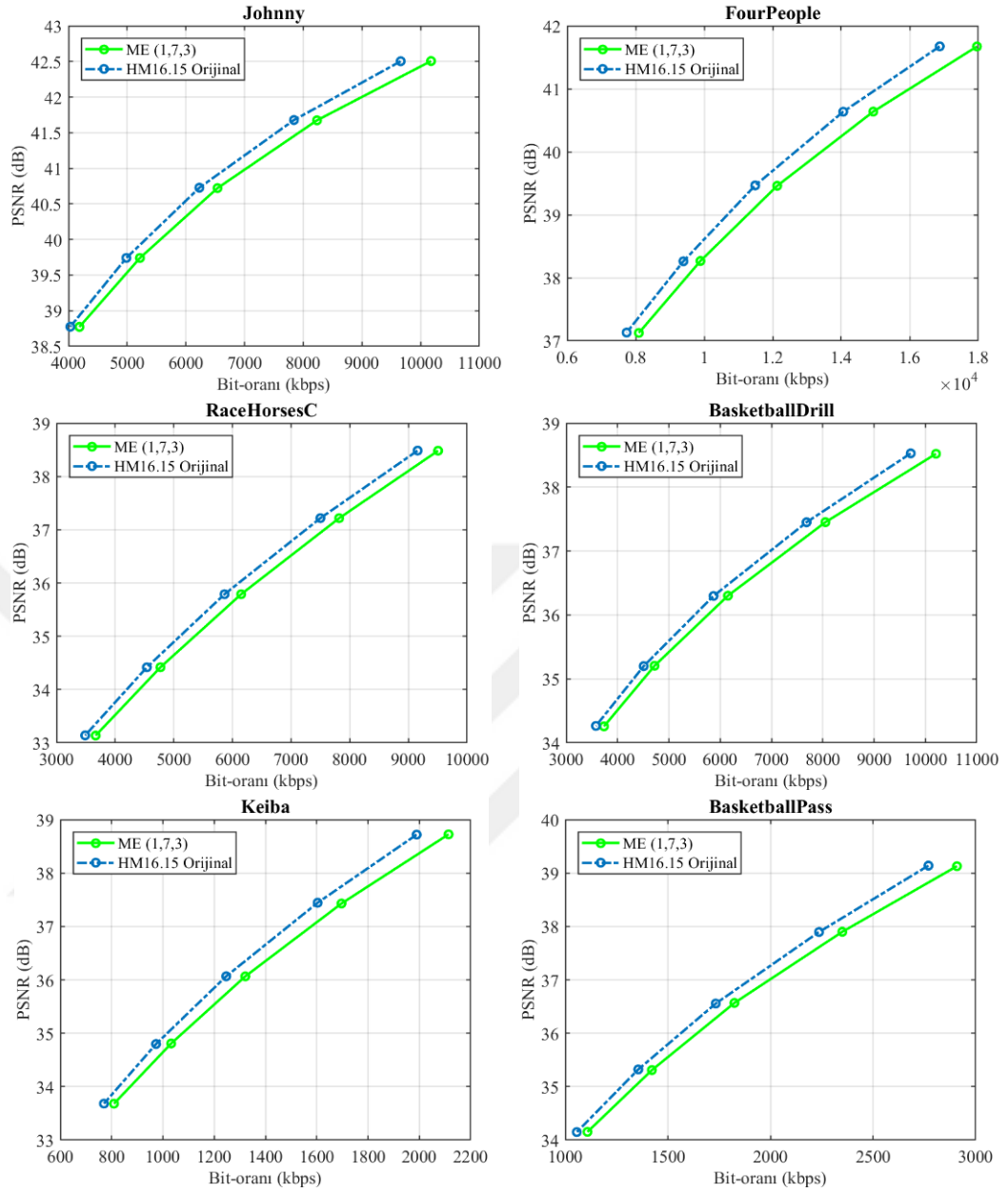
Değişecek TU bulunduktan sonra, ilgili TU'ya Eşitlik (5.15)'teki ε matrisi eklenerek hata yayılımından bağımsız biçimde veri gizlenmiştir. Bütün video çerçevelerine veri gizleme işlemi tamamlandıktan sonra stego video alıcıya gönderilmektedir. Alıcıdaki veri çıkartma işlemi de veri gizlemeye benzer biçimde yapılmaktadır. HEVC video kod çözme işlemi aşamasında stego bit dizisinden intra çerçeveler çözülür ve sırasıyla CTU'ların kodları çözülmeye başlanır. Çözülen CTU içerisindeki TU'lar 7'li gruplar halinde sıralanarak her birinin q_{11} katsayılarının LSB değerleri elde edilerek veri gizlenmiş taşıyıcı vektörü r_{yeni} oluşturulur. Daha sonra Hamming kod matrisi $H_{3 \times 7}$ ve r_{yeni} kullanılarak Eşitlik (5.8)'e göre gizlenen 3 bit elde edilir. CTU'lardaki bütün 7'li TU gruplarından çıkarılan 3 bitlik mesajlar birleştirilerek gizli mesaj oluşturulur.

5.5.2. Deneysel çalışmalar

Deneysel çalışmalarda tüm videolar çerçeve-içi özellikte kodlanmıştır. Bu amaçla HM16.15 referans yazılımındaki encoder_intra_main konfigürasyonu kullanılmıştır. Test video dizilerinin ilk 20 çerçevesi 28, 30, 32, 34 ve 36 gibi farklı QP değerleriyle kodlanarak test sonuçları elde edilmiştir. Video kodlama işlemleri hem önerilen veri gizleme yöntemi için hem de veri gizleme olmadan orijinal HM16.15 için gerçekleştirilmiştir. Veri gizleme için seçilen her çerçevedeki CTU'lar kendi içinde değerlendirilmiştir. Gizli mesaj bitleri ve CTU içerisindeki TU'lar ME(1, 7, 3) şemasına göre gruplandıktan sonra, değişecek TU tespit edilmektedir. Değişecek TU'da hata yayılımını önlemek için Eşitlik (5.15)'teki ϵ matrisi, sıfır veya daha büyük QDST katsayıları için $t=1$, sıfırdan küçük QDST katsayıları için $t=-1$ alınarak oluşturulmaktadır. Önerilen yöntemin nesnel değerlendirmesi için ilk olarak stego videonun, orijinal videoya ne kadar benzer olduğunu test edilmektedir. Bu amaçla veri gizlemesiz, orijinal kodlamaya göre oran-bozunum durumu kontrol edilmiştir. Şekil 5.14'te bu teste ait oran-bozunum grafikleri gösterilmektedir.



Şekil 5.14. Önerilen yöntemin oran-bozunum başarımı



Şekil 5.14. (Devam) Önerilen yöntemin oran-bozunum başarımı

Şekil 5.14'teki grafikler incelendiğinde farklı QP değerleri için kodlanmış stego videoların, orijinal videolara görsel kalite açısından benzerlik gösterdiği görülmektedir. Tablo 5.13'te, farklı QP değerleri için BasketballPass videosu üzerinden literatürde bulunan [138] yöntemiyle Δ PSNR, BIR ve kapasite karşılaştırması yapılmıştır. Önerilen yöntem, BasketballPass videosunda, [138] çalışmasına göre veri gizlemeden kaynaklı PSNR farkını gösteren Δ PSNR değerlerini oldukça düşük oranda tutmaktadır.

Tablo 5.13. BasketballPass test dizisinin farklı QP değerleri için veri gizleme başarımı

QP	Δ PSNR (dB)		BIR($\times 10^{-3}$)		Kapasite (bit)	
	Önerilen Yöntem	[138]	Önerilen Yöntem	[138]	Önerilen Yöntem	[138]
28	-0,010	-0,200	0,40	0,86	12555	2920
30	0,004	-0,350	0,49	1,20	10449	2184
32	0,012	-0,230	0,64	1,04	8256	1412
34	-0,012	-0,380	0,71	1,99	6942	1184
36	0,004	-0,210	0,84	2,68	5283	652

Hatta QP değerleri 30, 32 ve 34 iken küçük bir miktarda kalite iyileştirmesi sağlamaktadır. Önerilen yöntem kapasite miktarını 5-8 kat civarında arttırmasına rağmen BIR değerlerini oldukça düşük tutmaktadır.

Tablo 5.14. Önerilen yöntemin çeşitli test dizilerinde QP=32 için kapasite (bit) başarımı

Video Dizisi	Önerilen Yöntem	[138]	[139]	[109]	[125]
Traffic	177396	47300	-	10770	1560
PeopleOnStreet	279909	45972	-	64935	16110
Cactus	121635	20184	-	7500	2550
BQTerrace	222468	61644	-	4905	390
Johnny	26349	2616	-	-	-
FourPeople	47820	8500	-	-	-
RaceHorsesC	39216	6140	-	6885	1770
BasketballDrill	31302	4040	-	-	-
Keiba	10215	3664	1248	-	-
BasketballPass	8256	1412	1865	-	-

Tablo 5.15. Önerilen yöntemin çeşitli test dizilerinde QP=32 için Δ PSNR (dB) başarımı

Video Dizisi	Önerilen Yöntem	[138]	[139]	[109]	[125]
Traffic	-0,001	-0,320	-	-0,048	-0,073
PeopleOnStreet	0,004	-0,310	-	-0,125	-0,592
Cactus	-0,002	-1,060	-	-0,041	-1,672
BQTerrace	-0,002	-0,340	-	-0,026	-0,124
Johnny	-0,005	-0,420	-	-	-
FourPeople	-0,005	-0,640	-0,796	-	-
RaceHorsesC	0	-0,360	-	-0,107	-0,006
BasketballDrill	0,004	-1,860	-2,282	-	-
Keiba	-0,001	-0,340	-0,140	-	-
BasketballPass	0,012	-0,230	-0,290	-	-



(a) QP28



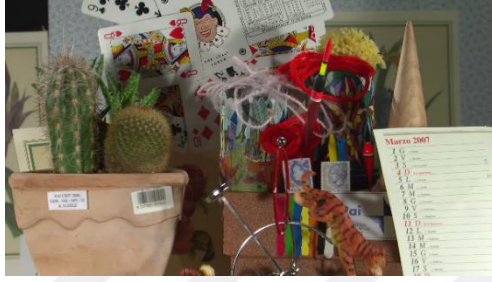
(b) QP28



(c) QP30



(d) QP30



(e) QP32



(f) QP32



(g) QP34



(h) QP34



(i) QP36



(j) QP36

Şekil 5.15. Önerilen yöntemin öznel değerlendirmesi. Orijinal kodlananlar (solda), önerilen veri gizleme ile kodlananlar (sağda). (a)-(b): BasketballDrill QP=28, (c)-(d): BQTerrace QP=30, (e)-(f): Cactus QP=32, (g)-(h): RaceHorsesC QP=34 ve (i)-(j): Traffic QP=36

Farklı video dizilerinde QP=32 için elde edilen detaylı başarımlar Tablo 5.14, 5.15 ve 5.16’da verilmiştir. Tablolar incelendiğinde veri gizleme kapasitesi ve BIR açısından elde edilen değerlerin literatür yöntemlerine göre oldukça iyi olduğu görülmektedir.

Tablo 5.16. Önerilen yöntemin çeşitli test dizilerinde QP=32 için BIR($\times 10^{-3}$) başarımları

Video Dizisi	Önerilen Yöntem	[138]	[138][139]	[109]	[125]
Traffic	0,02	0,02	-	1,10	0,16
PeopleOnStreet	0,02	0,02	-	0,95	0,03
Cactus	0,04	0,06	-	1,68	0,28
BQTerrace	0,03	0,01	-	1,10	0,46
Johnny	0,19	0,37	-	-	-
FourPeople	0,12	0,14	-	-	-
RaceHorsesC	0,12	0,14	-	8,52	0,36
BasketballDrill	0,15	0,24	-	-	-
Keiba	0,59	0,76	0,83	-	-
BasketballPass	0,64	0,67	0,66	-	-

Öznel değerlendirme için orijinal kodlama ve veri gizleme sonrası kodlama sonuçlarında elde edilen videoların çerçeve örnekleri karşılaştırılmıştır.

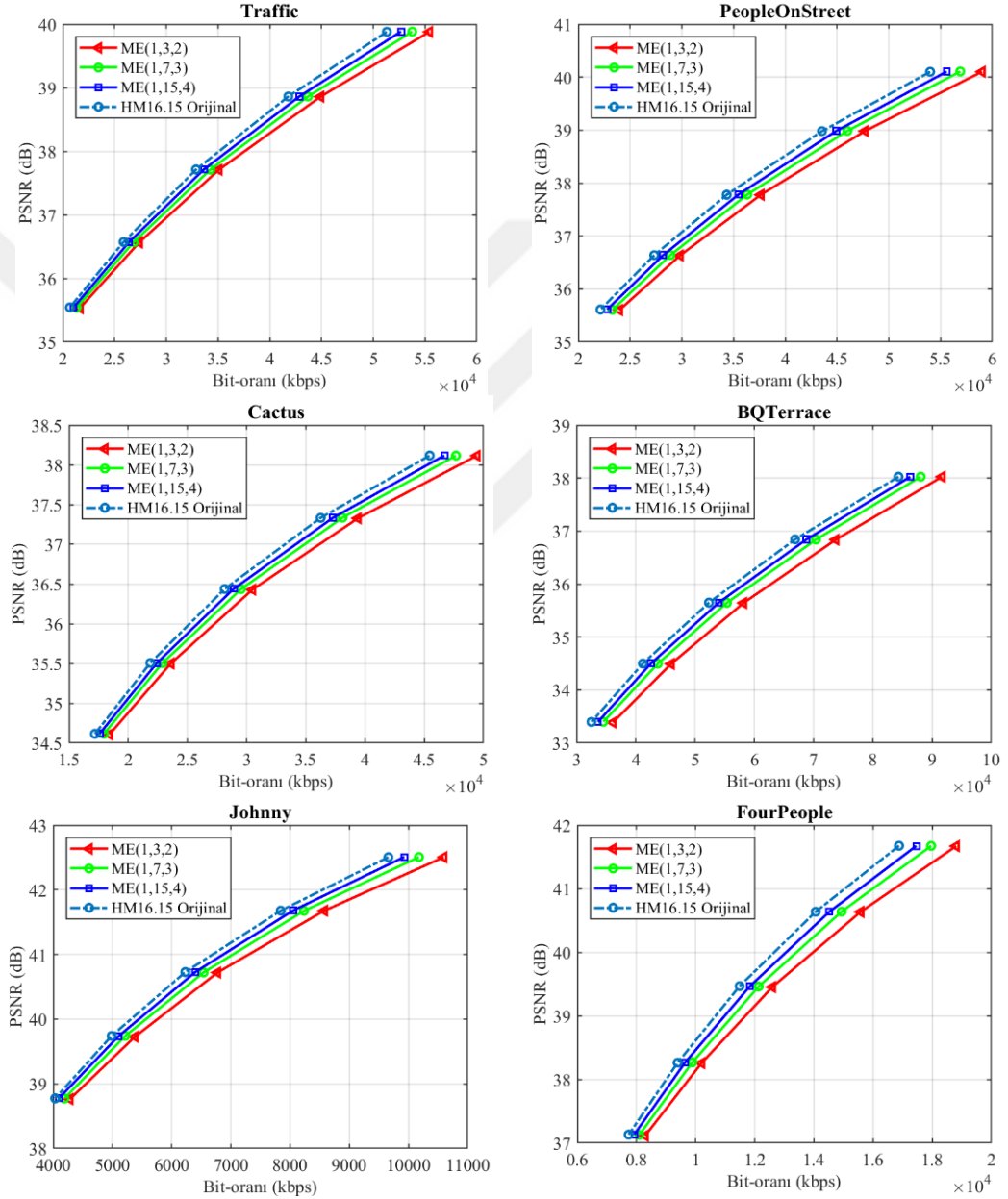
Farklı QP değerleri için örnek videoların ilk çerçevelerinin karşılaştırılması Şekil 5.15’te gösterilmektedir. Video çerçevelerinden hiç birinde gözle görülen bir bozulma veya değişim bulunmamaktadır. Dolayısıyla yapılan veri gizleme işleminin fark edilmesi mümkün değildir.

5.5.3. Sonuçlar

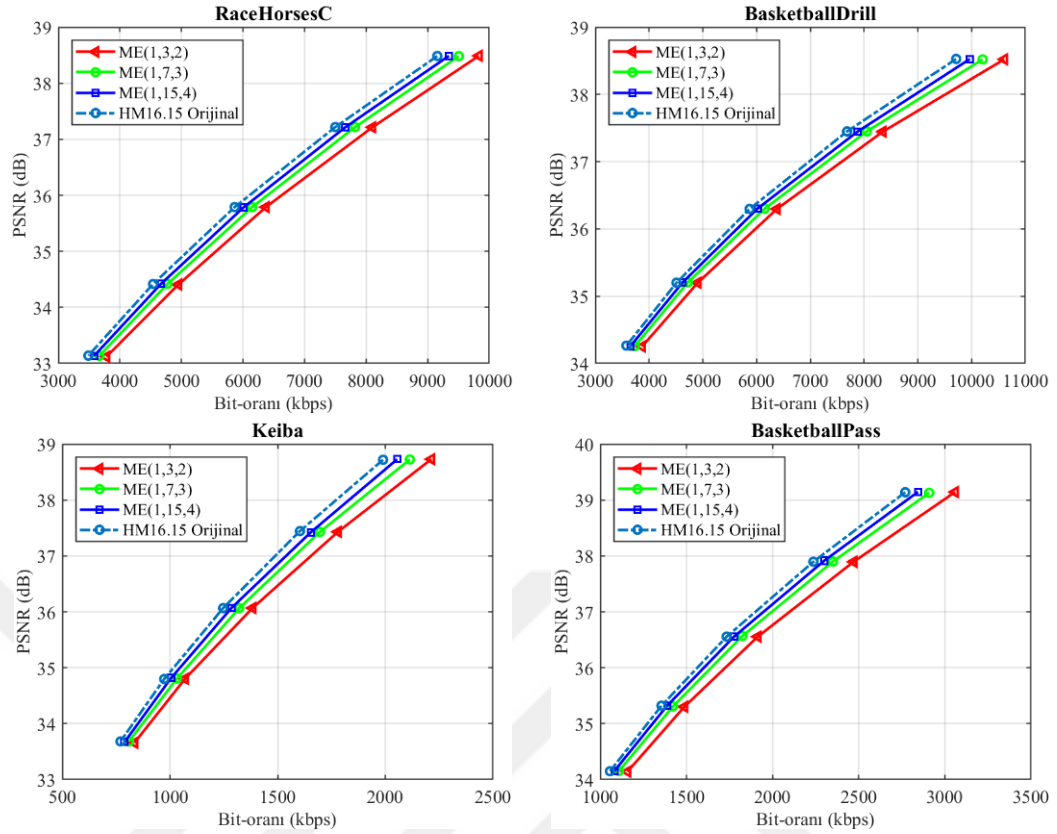
Tez çalışmasının bu kısmında önerilen yöntemin amacı, yüksek kapasiteli ve yüksek verimli veri gizleme işlemlerini bir araya getirmektir. Sadece yüksek kapasiteye odaklı yöntemde BIR değeri artarken, verimliliği hedefleyen yöntemde BIR düşerken kapasitede azalmaktadır. Bu sebeple önerilen yöntem kapasite ve verimi dengeli bir seviyeye getirerek kullanışlı bir veri gizleme sunmaktadır. Yapılan deneysel çalışmalar önerilen yöntemin literatür yöntemlerine göre her anlamda üstünlük sağladığını göstermiştir.

5.6. Karşılaştırmalar

Tez çalışmasının bu kısmında, önceki kısımlarda önerilen yüksek kapasiteli, yüksek verimli ve kapasite ile verimin dengeli olduğu veri gizleme yöntemleri kıyaslanmaktadır. Karşılaştırma için hem oran-bozunum grafikleri hem de farklı QP değerleri için elde edilen veri gizleme kapasiteleri kullanılmaktadır.

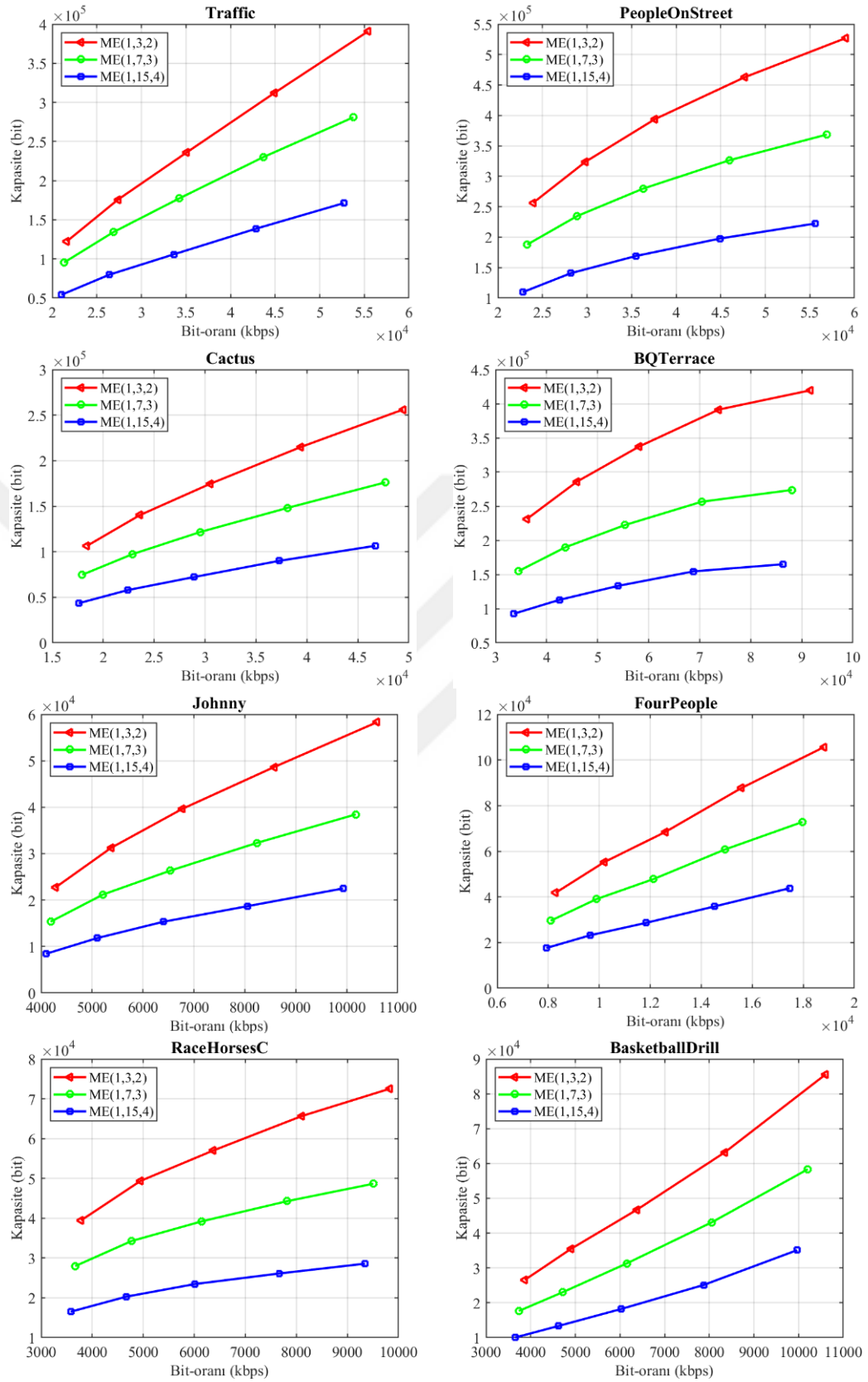


Şekil 5.16. Tez çalışmasındaki tüm yöntemlerin oran-bozunum karşılaştırmaları

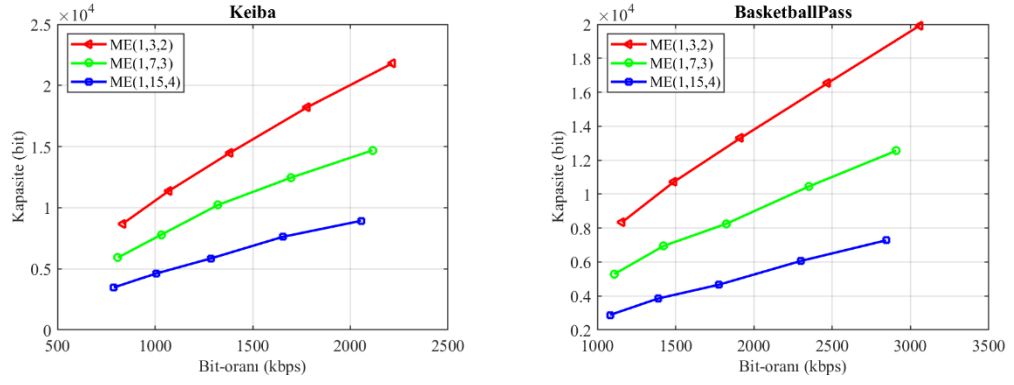


Şekil 5.16. (Devam) Tez çalışmasındaki tüm yöntemlerin oran-bozunum karşılaştırmaları

Şekil 5.16’da tez çalışması kapsamında önerilen bütün yöntemlerin oran-bozunum grafikleri üzerinden karşılaştırılması verilmektedir. Bütün yöntemlerde aynı QP değerleri için, veri gizleme sonrası elde edilen stego videoların PSNR değerleri, veri gizleme olmaksızın kodlanan videoların PSNR değerleriyle neredeyse aynı çıkmaktadır. Dolayısıyla önerilen yöntemlerle yapılan veri gizleme sonucu elde edilen videolarda gözle görülen bir bozulma oluşmayacağı için, iletim ortamında bu videolara karşı herhangi bir şüphe oluşmayacaktır. Şekil 5.16’ya göre PSNR değerleri pek değişmezken bit oranı değerleri artmaktadır. Bu artış stego videonun yapısına zarar vermemekle birlikte, veri gizleme işleminin doğası gereği iletim kanalında ek yük oluşturmaktadır. Yüksek kapasite temelli ME(1, 3, 2) yönteminde bu yük en fazla iken, yüksek verimli ME(1, 15, 4) yöntemiyle bu yük oldukça azaltılmıştır. Yüksek kapasite ve yüksek verimin bir arada olduğu ME (1, 7, 3) yönteminde bit oranı artışı, ME(1, 3, 2) yöntemine göre yaklaşık %50 azalmaktadır.



Şekil 5.17. Tez çalışmasındaki tüm yöntemlerin kapasite karşılaştırması



Şekil 5.17. (Devam). Tez çalışmasındaki tüm yöntemlerin kapasite karşılaştırması

Şekil 5.17’de önerilen yöntemlerin veri gizleme kapasiteleri kıyaslanmaktadır. QP değeri arttıkça videodaki sıkıştırma oranı arttığı için videodaki detaylar da kaybolmaktadır. Dolayısıyla video içerisinde, önerilen veri gizleme için kullanılan 4×4 blok sayısı da azalmaktadır. Bu sebeple QP değeri arttıkça veri gizleme kapasitesi düşmektedir. Öte yandan, videoların çözünürlüğü arttıkça 4×4 blok sayısının artması ile veri gizleme kapasiteleri artmaktadır.

6. SONUÇLAR VE ÖNERİLER

Bu tez çalışması kapsamında, HEVC videolarındaki 4×4 TU'lar kullanılarak hata yayılımından bağımsız yüksek kapasite ve yüksek verimlilik temelli veri gizleme yöntemleri geliştirilmiştir. Önerilen yöntemler, literatürde veri gizleme için kullanılan karmaşık ve zahmetli hata yayılımı önleme yöntemini iyileştiren basit bir yöntem sunmuştur. Mevcut literatür yöntemleri, QTC'lere veri gizlerken hata yayılımını önlemek için, ilk olarak videoyu normal kodlayıp blok yapıları ve IPM değerlerini analiz etmişlerdir. İkinci kodlamada ise veri gizlenecek bloğun komşu bloklarının IPM değerlerine göre farklı veri gizleme örüntüleri kullanmışlardır. Benzer işlemlerin veri çıkartımı için de yapılması gerektiği düşünülünce, oldukça fazla iş yükü oluşmaktadır.

Tez kapsamında kullanılan hata yayılımını önleme yöntemi mevcut yöntemlerin iyileştirilip basitleştirilmesi esasına dayanmaktadır. Önerilen yöntemlerde, komşu blokların IPM değerlerinin kontrol edilmesi ihtiyacını da ortadan kaldırılmıştır. Dolayısıyla videonun sadece veri gizleme için kodlanması yeterli olup başka bir analiz yapmaya ihtiyaç kalmadığından iş yükü oldukça azaltılmıştır.

Tez kapsamında TU bloklarındaki QDST katsayıları hata yayılımını önleyecek bir matris ile değiştirilmektedir. Veri gizleme yapabilmek için ME (1, c, m) şemasına farklı bir bakış açısı getirilerek, m adet bit gizlemek için c tane TU bloğunun kullanılması önerilmiştir. Veri gizleme yöntemi için Bölüm 5.3'te önerilen ME (1, 3, 2) yöntemi ile farklı videolar için oldukça yüksek kapasitelere erişilmiştir. Ayrıca literatürdeki benzer yöntemlere göre kat kat daha fazla veri gizleme kapasitesine erişilmiştir. HEVC videolarında veri gizlemenin daha verimli yapılabilmesi için Bölüm 5.4'te önerilen yüksek verimlilik temelli yöntem için videolardaki CTU'lar içerisinde bulunan 4×4 TU sayıları analiz edilmiştir. Yapılan analizler sonucunda videolardaki TU sayıları farklılık göstermekle birlikte, kullanılan bütün QP değerlerine ve farklı videolara en uygun veri gizleme yöntemi belirlenmiştir. Bölüm 5.4'te önerilen bu yöntem ile yapılan veri gizleme sonrası elde edilen BIR değerleri hem bir önceki bölümde önerilen yüksek kapasite temelli yöntemle göre hem de

literatürdeki farklı yöntemlere göre oldukça düşük sonuçlar sunmuştur. Bölüm 5.5'te önerilen üçüncü yöntemde, kapasite ve verimlilik birlikte dikkate alınmaktadır. Böylece hem kapasite yüksek seviyede tutulmakta hem de BIR değerleri aşağılara çekilerek gizleme verimliliği artırılmaktadır. Bölüm 5.6'da yapılan karşılaştırmalar, bundan sonraki çalışmalarda kapasite ve BIR değerleri arasında denge gözetilmek istendiğinde Bölüm 5.5'te önerilen yöntemin kullanılmasının daha uygun olacağını göstermektedir.

Önerilen yöntemlerin her birisi literatürdeki çalışmalara göre üstün Δ PSNR başarımı sergilemektedir. Yani, önerilen yöntemler sonucunda elde edilen videolardaki çerçevelerin hiç birindeki görsel kalite, insan görsel sistemiyle fark edilecek bir değişiklik olmamıştır. Dolayısıyla, stego videolarda herhangi bir bozulma olmadığından, önerilen yöntemlerde veri gizlemenin fark edilmezlik ilkesine sadık kalınmıştır. Bu noktada önemli bir hususun altını tekrar çizmek gerekir. Normal şartlarda veri gizleme sonrası taşıyıcı nesnenin kalitesinde düşüş olması beklenmektedir. Oysaki, önerilen yöntemlerde bazı videolar için Δ PSNR başarımları pozitif olarak çıkmaktadır. Yani veri gizleme sonrası elde edilen kodlanmış stego videonun kalitesi, veri gizleme olmadan kodlanan videonun kalitesinden daha iyi çıkmaktadır. Her ne kadar aradaki fark sembolik olsa da, bu değerler tutarlıdır. Çünkü buradaki PSNR değeri girişteki YUV video çerçeveleri ile çıkıştaki kodlanmış video çerçeveleri arasındaki tepe-sinyal gürültü oranını ifade etmektedir. Dolayısıyla 4x4 artıklık bloklarına gizlenen verilerin etkisiyle kodlanmış videonun orijinal YUV videoya daha fazla benzmesi olasıdır.

HEVC video uygulamalarının önümüzdeki yıllarda da piyasaya hakim olacağı öngörüsünden yola çıkarak, gelecekte yapılacak çalışmalarda, daha yüksek kapasitelere ihtiyaç olduğunda 8x8 ve 16x16 TU'ların kullanılabileceği değerlendirilmektedir. Önerilen veri gizleme yöntemlerinin kısmen veya tamamen kullanılmasıyla, video kodlamanın hareket kestirimi, entropi kodlaması ve örtüşme önler süzgeçleme gibi farklı aşamaları için veri gizleme yöntemleri geliştirilebileceği düşünülmektedir. Bu tez kapsamında yapılan veri gizlemenin amacının gizli haberleşme olacağı kabul edilmiştir. Bu sebeple haberleşme kanalında stego videoya herhangi bir saldırı yapılmayacağı varsayılmıştır. Gelecekte, önerilen yöntemin çeşitli saldırılara karşı test edilerek içerik koruması ve telif hakkı doğrulama gibi damgalama

senaryolarına uyumluluđu incelenebilecek bir alıřma konusudur. Ayrıca nerilen yntemlerin hata kontrol kodlama yntemleriyle birlikte kullanılmasıyla dayanıklı damgalama yntemleri geliřtirilebilir.



KAYNAKLAR

- [1] Sencar H. T., Ramkumar M., Akansu A. N., *Data Hiding Fundamentals and Applications*, Elsevier Academic Press, 1.st ed., San Diego, 2004.
- [2] Yalman Y., Ertürk İ., Çetin Ö. ve Akar F., *Veri Gizleme*, Beta, 1.baskı, İstanbul, 2014.
- [3] Sullivan G. J., Introduction, Editors: Sze V., Budagavi M., Sullivan G. J., *High Efficiency Video Coding (HEVC) Algorithms and Architectures*, Springer, 1.st ed., Cham, 1-12, 2014.
- [4] Ohm J., Sullivan G. J., Schwarz H., Tan T. K., Wiegand T., Comparison of the Coding Efficiency of Video Coding Standards-Including High Efficiency Video Coding (HEVC), *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, **22**(12), 1669-1684.
- [5] Lederer S., 2019 Video Developer Report - The Future of Video, *Bitmovin*, 2019, 1-36.
- [6] Correa G., Assuncao P., Agostini L., Cruz L. A. D., *Complexity-Aware High Efficiency Video Coding*, Springer, 1.st ed., Cham, 2016.
- [7] Akramullah S., Digital Video Concepts, Methods, and Metrics: Quality, Compression, Performance, and Power Trade-off Analysis, 1.st ed., Apress Open, Online, 2014.
- [8] Bull D. R., Communicating Pictures: A Course in Image and Video Coding, 1.st ed., Academic Press, Oxford, 2014.
- [9] Tan, T. K., Weerakkody R., Mrak M., Ramzan N., Baroncini V., Ohm J., Sullivan G. J., Video Quality Evaluation Methodology and Verification Testing of HEVC Compression Performance, *IEEE Transactions on Circuits And Systems For Video Technology*, **26**(1), 2016, 76-90.
- [10] Sullivan G. J., DirectX Video Acceleration Specification for Windows Media Video v8, v9 and vA Decoding (Including SMPTE 421M "VC-1"), *Microsoft Corporation*, 2007, 1-102.
- [11] Borer T., Davies T., Dirac Video Compression using Open Technology, *BBC Research & Development*, 2005, 1-9.
- [12] Grange A., Rivaz P., Hunt A., VP9 Bitstream & Decoding Process Specification- v0.6, *Google Inc*, 2016, 1-163.

- [13] Gao W., AVS standard - Audio Video Coding Standard Workgroup of China, *14th Annual International Conference on Wireless and Optical Communications (WOCC 2005)*, New Jersey, USA, 22-23 Apr. 2005.
- [14] Grois D., Marpe D., Mulayoff A., Itzhaky B., Hadar O., Performance comparison of H.265/MPEG-HEVC, VP9, and H.264/MPEG-AVC encoders, *2013 Picture Coding Symposium (PCS)*, San Jose, CA, USA, 8-11 Dec. 2013.
- [15] Rerabek M., Ebrahimi T., Comparison of Compression Efficiency Between HEVC/H.265 and VP9 based on Subjective Assessments, *SPIE Optical Engineering Applications*, San Diego, California, USA, 18-21 Aug 2014.
- [16] Chen Y., Zhou Y., Wen J., Efficient Software HEVC to AVS2 Transcoding, *Information*, 2016, 7(3), 53.
- [17] Ma S., Huang T., Reader C., Gao W., AVS2 ? Making Video Coding Smarter [Standards in a Nutshell], *IEEE Signal Processing Magazine*, 2015, 32(2), 172-183.
- [18] Karwowski D., Grajek T., Klimaszewski K., Stankiewicz O., Stankowski J., Wegner K., 20 Years of Progress in Video Compression – from MPEG-1 to MPEG-H HEVC, *8th International Conference Image Processing and Communications(IP&C 2016)*, Bydgoszcz, Poland, 7-9 Sep. 2016.
- [19] Gupta A., Goyal D., Hemrajani N., Performance Analysis of Various Video Compression Techniques, *International Journal of Science and Research (IJSR)*, 2013, 2(8), 335-338.
- [20] Abomhara M., Khalifa O. O., Zakaria O., Zaidan Z., Zaidan B. B., Rame A., Video Compression Techniques: An Overview, *Journal of Applied Sciences*, 2010, 10(16), 1834-1840.
- [21] Schwarz H., Wiegand T., Video Coding: Part II of Fundamentals of Source and Video Coding, *Foundations and Trends in Signal Processing*, 2016, 10(1-3), 1–346.
- [22] Parithi T. I, Murugan V., Balasubramanian R., A Performance Analysis of Video Compression Standards MPEG & H.264, *International Journal of Advance Engineering and Research Development*, 2017, 4(8), 627-633.
- [23] Puri A., Eleftheriadis A., MPEG-4: An object-based multimedia coding standard supporting mobile applications, *Mobile Networks and Applications*, 1998, 3, 5–32.
- [24] Wiegand T., Sullivan G. J., Bjontegaard G., Luthra A., Overview of the H.264/AVC Video Coding Standard, *IEEE Transactions on Circuits And Systems For Video Technology*, 2003, 13(7), 560-576.
- [25] Sullivan G. J., Ohm J., Han W., Wiegand T., Overview of the High Efficiency Video Coding (HEVC) Standard, *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, 22(12), 1649-1668.

- [26] Kim I., Min J., Lee T., Han W., Park J., Block Partitioning Structure in the HEVC Standard, *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, **22**(12), 1697-1706.
- [27] Yuan Y., et al., Quadtree Based Nonsquare Block Structure for Inter Frame Coding in High Efficiency Video Coding, *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, **22**(12), 1697-1706.
- [28] Lainema J., Bossen F., Han W., Min J., Ugur K., Intra Coding of the HEVC Standard, *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, **22**(12), 1792-1801.
- [29] Budagavi M., Fuldseth A., Bjøntegaard G., Sze V., Sadafale M., Core Transform Design in the High Efficiency Video Coding (HEVC) Standard, *IEEE Journal of Selected Topics in Signal Processing*, 2013, **7**(6), 1029-1041.
- [30] Budagavi M., Fuldseth A., Bjøntegaard G., HEVC Transform and Quantization, Editors: Sze V., Budagavi M., Sullivan G. J., *High Efficiency Video Coding (HEVC) Algorithms and Architectures*, Springer, 1.st ed., Cham, 141-169, 2014.
- [31] Bossen F, Flynn D., Sharman K, Sühring K., HM Software Manual, *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*, JCTVC-Software Manual, 1-35, 2020.
- [32] Bossen F, Common test conditions and software reference configurations, *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11*, JCTVC-L1110, 1-4, 2013
- [33] Tabatabai A., et al., Compression Performance Analysis in HEVC, Editors: Sze V., Budagavi M., Sullivan G. J., *High Efficiency Video Coding (HEVC) Algorithms and Architectures*, Springer, 1.st ed., Cham, 275-302, 2014.
- [34] Bossen F, Bross B., Sühring K., Flynn D., HEVC Complexity and Implementation Analysis, *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, **22**(12), 1685-1696.
- [35] Bjøntegaard G., Calculation of average PSNR differences between RD-curves, *ITU-T , VCEG-M33*, 2001, 1-4.
- [36] Cox I., Miller M., Bloom J., Fridrich J., Kalker T., *Digital Watermarking and Steganography*, Morgan Kaufmann Publishers Inc., 2nd. ed., San Francisco, 2007.
- [37] Simmons G. J., The Prisoners' Problem and the Subliminal Channel, Editor: Chaum A., *Advances in Cryptology*, Springer, 1.st ed., Boston, 51-67, 1984.
- [38] Güzeldereli E, Doğan F, Çetin Ö, Medikal Görüntü İçerisine Tıbbi Bilgilerin Gömülmesi İçin Yeni Bir Yaklaşım, *Sakarya University Journal of Science*, 2013, **17**(2), 286-277.

- [39] Fridrich J., *Steganography in Digital Media: Principles, Algorithms, and Applications*, Cambridge University Press, 1st. ed.. USA. 2009.
- [40] Huang H. C., Chu S. C., Pan J. S., Huang C. Y., Liao B. Y., Tabu search based multi-watermarks embedding algorithm with multiple description coding, *Information Science*, 2011, **181**(16), 3379-3396.
- [41] Chuan Q., Ping J., Xinpeng Z., Jing D., Jinwei W., Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy, *Signal Processing*, 2017, **138**, 280-293.
- [42] Singh A. K., Improved hybrid algorithm for robust and imperceptible multiple watermarking using digital images, *Multimed Tools Appl*, 2017, **76**, 8881-8900.
- [43] Prasad S., Pal A. K., A Secure Fragile Watermarking Scheme for Protecting Integrity of Digital Images. *Iran J Sci Technol Trans Electr Eng*, 2020, **44**, 703–727.
- [44] Hsu L. Y., Hu H. T., Blind watermarking for color images using EMMQ based on QDFT, *Expert Systems with Applications*, 2020, **149**, 113225.
- [45] Shehab A., et al., Secure and Robust Fragile Watermarking Scheme for Medical Images, *IEEE Access*, 2018, **6**, 10269-10278.
- [46] Zear A., Singh A. K., Kumar P., A proposed secure multiple watermarking technique based on DWT, DCT and SVD for application in medicine, *Multimed Tools Appl*, 2018, **77**, 4863-4882.
- [47] Liang X., Xiang S., Robust reversible audio watermarking based on high-order difference statistics, *Signal Processing*, 2020, **173**, 107584.
- [48] Yaslan Y., Günsel B., A context-aware mobile application framework using audio watermarking, *Multimedia Systems*, 2020, **26**, 323–337.
- [49] Khedr W. I., A novel Bloom-filter-based scheme for secure text ownership protection. *Wireless Networks*, 2020, **26**, 3831–3845.
- [50] Horng S., Rosiyadi D., Fan P., Wang X., Khan M. K., An adaptive watermarking scheme for e-government document images. *Multimed Tools Appl*, 2014, **72**, 3085-3103.
- [51] El-Rahman S. A., Text Steganography Approaches Using Similarity of English Font Styles, *International Journal of Software Innovation*, 2019, **7**(3), 29-50.
- [52] Al-Nofaie S. M. A., Gutub A. A., Utilizing pseudo-spaces to improve Arabic text steganography for multimedia data communications. *Multimed Tools Appl*, 2020, **79**, 19-67.

- [53] Khairullah M., A novel steganography method using transliteration of Bengali text, *Journal of King Saud University - Computer and Information Sciences*, 2019, **31**(3), 348-366.
- [54] Liu Y., Wu J., Xin G., Multi-keywords carrier-free text steganography method based on Chinese Pinyin, *International Journal of Computational Science and Engineering*, 2020, **21**(2), 202-209.
- [55] Tan D., Lu Y., Yan X., Wang X., A Simple Review of Audio Steganography, *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chengdu, China, 15-17 Mar. 2019.
- [56] Alsabhany A. A., Ridzuan F., Azni A. H., "The Adaptive Multi-Level Phase Coding Method in Audio Steganography, *IEEE Access*, 2019, **7**, 129291-129306.
- [57] Liu Z., Jiang Y., Qian P., A Data-Hiding Method Based on TCP/IP Checksum, Editors: Jeong H., Obaidat M. S., Yen N., Park J., *Advances in Computer Science and its Applications Lecture Notes in Electrical Engineering*, Springer, **279**, Berlin, 1039-104, 2013.
- [58] Chou Y. C., Liao H. C., A Webpage Data Hiding Method by Using Tag and CSS Attribute Setting, *2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, Kitakyushu, Japan, 27-29 Aug. 2014.
- [59] Wu D. C., Tsai W. H., A steganographic method for images by pixel-value differencing, *Pattern Recognition Letters*, 2003, **24**(10), 1613-1626.
- [60] Zhang X., Wang S., Efficient Steganographic Embedding by Exploiting Modification Direction, *IEEE Communications Letters*, 2006, **10**(11), 781-783.
- [61] Pradhan A., Sekhar K. R., Swain G., Digital image steganography using LSB substitution, PVD, and EMD, *Math. Prob. Eng*, 2018, **2018**, 1804953.
- [62] Kim S., Huang F., Kim H. J., Reversible Data Hiding in JPEG Images Using Quantized DC, *Entropy*, 2019, **21**, 835.
- [63] Biswas R., Bandyapadhyay S. K., Random selection based GA optimization in 2D-DCT domain color image steganography. *Multimed Tools Appl*, 2020, **79**, 7101-7120.
- [64] Gutub A., Al-Shaarani F., Efficient Implementation of Multi-image Secret Hiding Based on LSB and DWT Steganography Comparisons, *Arab J Sci Eng*, 2020, **45**, 2631-2644
- [65] Solak S., Altınışık U., Image steganography based on LSB substitution and encryption method: Adaptive LSB+3, *J. Electron. Imaging*, 2019, **28**, 043025.

- [66] Ni Z., Shi Y.Q., Ansari N., Su W., Reversible Data Hiding, *IEEE Transactions on Circuits and Systems for Video Technology*, 2006, **16**(3), 354-362.
- [67] Kurnaz H, Konyar M. Z., Sondaş A., Yakın Histogramlar Temelli Yeni Bir Hibrit Veri Gizleme Yöntemi, *Avrupa Bilim ve Teknoloji Dergisi*, 2020, **18**, 683-694.
- [68] Islamy C. C., Ahmad T., Histogram-based multilayer reversible data hiding method for securing secret data, *Bulletin of Electrical Engineering and Informatics*, 2019, **8**(3), 1128-1134.
- [69] Jabbari F., Sedaaghi M. H., Adaptive multilevel technique based on histogram shifting for reversible data hiding, *Multidimensional Syst. Signal Process*, 2019, **30**(3), 1563-1581.
- [70] Konyar M. Z., Öztürk S., Reed Solomon Coding-Based Medical Image Data Hiding Method against Salt and Pepper Noise, *Symmetry*, 2020, **12**, 899.
- [71] Memon N. A., Gilani S. A. M., Watermarking of chest CT scan medical images for content authentication, *Int. J. Comput. Math*, 2011, **88**, 265-280.
- [72] Parah S. A., Sheikh J. A., Ahad F., Loan N.A., Bhat G. M., Information hiding in medical images: A robust medical image watermarking system for E-healthcare, *Multimed Tools Appl*, 2017, **76**, 10599-10633.
- [73] Al-Dmour H., Al-Ani A. A, Medical Image Steganography Method Based on Integer Wavelet Transform and Overlapping Edge Detection. Editors: Arik S., Huang T., Lai W., Liu Q., *Lecture Notes in Computer Science*; **9492**, Springer, Cham, 436-444, 2015.
- [74] Er F., Yalman Y. Adaptive Multiple Peak and Histogram based Reversible Data Hiding. *Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA2019)*, Istanbul, Turkey, 6–9 Nov. 2019.
- [75] Tian J., Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.*, 2003, **13**, 890–893.
- [76] Mstafa R. J., Elleithy K. M., Compressed and raw video steganography techniques: A comprehensive survey and analysis, *Multimed Tools Appl*, 2017, **76**, 21749-21786.
- [77] Eltahir M.E., Kiah L.M., Zaidan B.B., High rate video streaming steganography, *International Conference on Information Management and Engineering, (ICIME'09)*, Kuala Lumpur, Malaysia, 3-5 Apr. 2009.
- [78] Hanafy A., Salama G., Mohasseb Y. Z., A secure covert communication model based on video steganography, *IEEE Military Communications Conference (MILCOM 2008)*, San Diego, CA, USA, 16-19 Nov. 2008.

- [79] Younus Z. S., Younus G. T., Video steganography using knight tour algorithm and LSB method for encrypted data, *Journal of Intelligent Systems*, 2019, **29**(1), 1216-1225.
- [80] Mstafa R. J. , Elleithy K. M., A video steganography algorithm based on Kanade-Lucas-Tomasi tracking algorithm and error correcting codes, *Multimed Tools Appl*, 2016, **75**, 10311-10333.
- [81] Sadek M. M., Khalifa A. S., Mostafa M. G. M., Robust video steganography algorithm using adaptive skin-tone detection, *Multimed Tools Appl*, 2017, **76**(2), 3065-3085.
- [82] Dalal M., Singh M., Kumar A., Charu, Juneja M., An Approach of Data Hiding in Video Steganography using Object Detection, *International Journal of Engineering and Advanced Technology (IJEAT)*, 2019, **8**(5), 2460-2466.
- [83] Paul R., Acharya A. K., Yadav V. K., Batham S., Hiding large amount of data using a new approach of video steganography, *Confluence 2013: The Next Generation Information Technology Summit (4th International Conference)*, Noida, India, 26-27 Sep. 2013.
- [84] Bhardwaj A., Verma V. S., Jha R. K., Robust video watermarking using significant frame selection based on coefficient difference of lifting wavelet transform, *Multimed Tools Appl*, 2018, **77**, 19659-19678.
- [85] Himeur Y., Boukabou A., A robust and secure key-frames based video watermarking system using chaotic encryption, *Multimed Tools Appl*, 2018, **77**, 8603-8627.
- [86] Bhautmage P., Jeyakumar A., Dahatonde A., Advanced Video Steganography Algorithm, *International Journal of Engineering Research and Applications (IJERA)*, 2013, **3**(1), 1641-1644.
- [87] Khan N., Gorde K. S., Video steganography by using statistical key frame extraction method and LSB technique, *Int. J. Innov. Res. Sci. Eng. Technol*, 2015, **4**(10), 10410-10417.
- [88] Farri E., Ayubi P., A blind and robust video watermarking based on IWT and new 3D generalized chaotic sine map, *Nonlinear Dyn*, 2018, **93**, 1875–1897.
- [89] Yassin N. I., Salem N. M., El Adawy M. I., QIM blind video watermarking scheme based on Wavelet transform and principal component analysis, *Alexandria Engineering Journal*, 2014, **53**, 833-842.
- [90] Agilandeewari L., Ganesan K., An Efficient Hilbert And Integer Wavelet Transform Based Video Watermarking, *Journal of Engineering Science and Technology*, 2016, **11**(3), 327-345.

- [91] Lu C. S., Chen J.-R., Fan K. C., Real-time frame-dependent video watermarking in VLC domain, *Signal Process., Image Commun.*, 2005, **20**(7), 624–642.
- [92] Choi D., Do H., Choi H., Kim, T. A blind MPEG-2 video watermarking robust to camcorder recording. *Signal Process.*, 2010, **90**, 1327-1332.
- [93] Fang D. Y., Chang L. W., Data hiding for digital video with phase of motion vector, *2006 IEEE International Symposium on Circuits and Systems*, Island of Kos, Greece, 21-24 May 2006.
- [94] Barni M., Bartolini F., Checcacci N., Watermarking of MPEG-4 video objects, *IEEE Transactions on Multimedia*, 2005, **7**(1), 23-32.
- [95] Chen X, Zhou P., Jing X., Shi J., The video water marking scheme based on MPEG-4 motion vector, *2010 International Conference on Intelligent Computing and Integrated Systems*, Guilin, China, 22-24 Oct. 2010.
- [96] Tew Y., Wong K., An Overview of Information Hiding in H.264/AVC Compressed Video, *IEEE Transactions on Circuits and Systems for Video Technology*, 2014, **24**(2), 305-319
- [97] Hu Y., Zhang C., Su Y., Information Hiding Based on Intra Prediction Modes for H.264/AVC, *2007 IEEE International Conference on Multimedia and Expo*, Beijing, China, 2-5 July 2007.
- [98] Xu D., Wang R., Wang J., Prediction mode modulated data-hiding algorithm for H.264/AVC, *J Real-Time Image Proc.*, 2012, **7**, 205–214.
- [99] Ke N., Weidong Z., A video steganography scheme based on H.264 bitstreams replaced, *2013 IEEE 4th International Conference on Software Engineering and Service Science*, Beijing, China, 23-25 May 2013.
- [100] Niu K., Yang X., Zhang Y., A novel video reversible data hiding algorithm using motion vector for H.264/AVC, *Tsinghua Science and Technology*, 2017, **22**(5), 489-498.
- [101] Ma X. J., Li Z. T., Tu H., Zhang B., A data hiding algorithm for H.264/AVC video streams without intra frame distortion drift, *IEEE Trans. Circ. Syst. Video Technol.*, 2010, **20**(10), 1320-1330.
- [102] Lin T. J., Chung K. L., Chang P. C., Huang Y. H., Liao Y. M., Fang C. Y., An improved DCT-based perturbation scheme for high capacity data hiding in H.264/AVC intra frames, *J. Syst. Softw.*, 2013, **86**(3), 604–614.
- [103] Nissar A., Mir A. H., Classification of steganalysis techniques: A study, *Digit. Signal Process.*, 2010, **20**(6), 1758–1770
- [104] Wang Z., Simoncelli E. P., Bovik A. C., Multiscale structural similarity for image quality assessment, *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*, Pacific Grove, CA, USA, 9-12 Nov. 2003.

- [105] Mansouri A., Aznaveh A.M., Torkamani-Azar F., Kurugollu F., A low complexity video watermarking in H.264 compressed domain, *IEEE Trans. Inf. Forensics Secur.*, 2010, 5(4), 649–657.
- [106] Tew Y., Wong K., Information Hiding in HEVC Standard Using Adaptive Coding Block Size Decision, *IEEE International Conference on Image Processing (ICIP)*, Paris, France, 27-30 Oct. 2014.
- [107] Tew Y., Wong K., Phan R. C. W., HEVC video authentication using data embedding technique, *2015 IEEE International Conference on Image Processing (ICIP)*, Quebec City, Canada, 27-30 Sep. 2015.
- [108] Xie W., Yang Y., Li Z., Wang J., Zhang M., An Information Hiding Algorithm for HEVC Videos Based on PU Partitioning Modes, *4th International Conference on Cloud Computing and Security*, Haikou, China, 8-10 June 2018.
- [109] Yang Y., Li Z., Xie W., Zhang Z., High capacity and multilevel information hiding algorithm based on PU partition modes for HEVC videos. *Multimed. Tools Appl.*, 2019, 78, 8423–8446.
- [110] Li Z., Meng L., Jiang X., Li, Z., High Capacity HEVC Video Hiding Algorithm Based on EMD Coded PU Partition Modes, *Symmetry*, 2019, 11, 1015.
- [111] Zhang X., Wang S., Efficient Steganographic Embedding by Exploiting Modification Direction, *IEEE Communications Letters*, 2006, 10(11), 781-783,
- [112] Shanableh T., Altering split decisions of coding units for message embedding in HEVC, *Multimed Tools Appl.*, 2018, 77, 8939-8953.
- [113] Wang J., Wang R. D., Li W., Xu D., Huang M., A large-capacity information hiding method for HEVC video, *The 3rd International Conference on Computer Science and Service System*, Bangkok, Thailand. 13-15 June 2014.
- [114] Wang J., Wang R. D., Li W., Xu D., Huang M., An information hiding algorithm for HEVC based on intra-prediction mode and block code, *Sensors & Transducers*, 2014, 177(8), 230-237.
- [115] Wang J., Wang R. D., Xu D., Li W., An information hiding algorithm for HEVC based on angle differences of intra prediction mode, *Journal of Software*, 2015, 10(2), 213-221.
- [116] Xu J., Wang R. D., Huang M.L., Wang J., Xu D., An information hiding algorithm for HEVC based on intra-prediction modes and Hamming+1, *Journal of Computational Information Systems*, 2015, 11(15), 5587-5598.
- [117] Dong Y., Jiang X., Sun T., Xu D., Coding Efficiency Preserving Steganography Based on HEVC Steganographic Channel Model, Editors: Kraetzer C., Shi Y. Q., Dittmann J., Kim H., *Lecture Notes in Computer*

Science, Digital Forensics and Watermarking, **10431**, Springer, 149-162, 2017.

- [118] Filler T., Judas J., Fridrich J., Minimizing embedding impact in steganography using trellis-coded quantization, *Proceedings of SPIE, Media Forensics and Security II*, 2010, **7541**, 754105.
- [119] Dong Y., Sun T., Jiang X., A High Capacity HEVC Steganographic Algorithm Using Intra Prediction Modes in Multi-sized Prediction Blocks, Editors: Yoo C., Shi YQ., Kim H., Piva A., Kim G., *Lecture Notes in Computer Science, Digital Forensics and Watermarking*, **11378**, Springer, 233-247, 2019.
- [120] Wang J., Jia X., Kang X., Shi Y., A Cover Selection HEVC Video Steganography Based on Intra Prediction Mode, *IEEE Access*, 2019, **7**, 119393-119402.
- [121] Yang J., Peng B., An optimized algorithm based on generalized difference expansion method used for HEVC reversible video information hiding, 2017 *IEEE 17th International Conference on Communication Technology (ICCT)*, Chengdu, China, 27-30 Oct. 2017.
- [122] Alattar A. M., Reversible watermark using the difference expansion of a generalized integer transform, *IEEE Transactions on Image Processing*, 2004, **13**(8), 1147-1156.
- [123] Peng B., Yang J., A Reversible Information Hiding Method Based on HEVC, *IFAC Conference on Bio-Robotics*, Beijing, China, 13–15 July 2018.
- [124] Yang J., Peng B., A Novel Video Reversible Data Hiding Algorithm Using Motion Vector for HEVC, *International Conference on Information Hiding and Image Processing*, Manchester, England, 22–24 Sep. 2018.
- [125] Yang J, Li S., An efficient information hiding method based on motion vector space encoding for HEVC, *Multimed Tools Appl.*, 2018, **77**(10), 11979-12001.
- [126] Swati S., Hayat K., Shadid Z., A Watermarking Scheme for High Efficiency Video Coding (HEVC), *Plos One*, 2014, **9**(8), e105613.
- [127] Chang P., Chung K., Chen J., Lin C., Lin T., An error propagation free data hiding algorithm in HEVC intra-coded frames, *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, Kaohsiung, Taiwan, 29 Oct-1 Nov. 2013.
- [128] Chang P., Chung K., Chen J., Lin C., Lin T., A DCT/DST-based error propagation-free data hiding algorithm for HEVC intra-coded frames, *J. Vis. Commun. Image R.*, 2014, **25**, 239–253.
- [129] Gaj S., Sur, A., Bora P. K., A Robust Watermarking Scheme Against Re-compression Attack for H.265-HEVC, *2015 Fifth National Conference on*

Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), Patna, India, 16-19 Dec. 2015.

- [130] Elrowayati A., Abdullah F. L., Abdmanaf A., Alfagi A. S., Robust HEVC video watermarking scheme based on repetition-BCH syndrome code, *Int J Softw Eng Appl*, 2016, **10**(1), 263–270.
- [131] Dutta T., Gupta H. P., A robust watermarking framework for High Efficiency Video Coding (HEVC) - Encoded video with blind extraction process, *Journal of Visual Communication and Image Representation*, 2016, **38**, 29-44.
- [132] Jo K., Lei W., Li Z., A Watermarking Method by Modifying QTCs for HEVC, *Computing Performance and Communication systems*, 2016, **1**, 8-16.
- [133] Gui F., Xue H., A Reversible Data Hiding Scheme For HEVC, *10th International Symposium on Computational Intelligence and Design*, Hangzhou, China, 9-10 Dec. 2017.
- [134] Gaj S., Kanetkar A., Sur A., Bora P. K., Drift-Compensated Robust Watermarking Algorithm for H.265/HEVC Video Stream, *ACM Trans. Multimedia Comput. Commun. Appl.*, 2017, **13**(1), 11:1-11:24.
- [135] Wang C., Shan R., Zhou X., Anti-HEVC Recompression Video Watermarking Algorithm Based on the All Phase Biorthogonal Transform and SVD, *IETE Technical Review*, 2018, **35**(1), 1-17.
- [136] Hou Z. X., Wang C. Y., Yang A. P., All phase biorthogonal transform and its application in JPEG-like image compression, *Signal Processing: Image Communication*, 2009, **24**(10), 791-802.
- [137] Long, M., Peng, F., Li, H., Separable reversible data hiding and encryption for HEVC video, *J Real-Time Image Proc*, 2018, **14**, 171–182.
- [138] Liu Y., Liu S., Zhao H. Liu S., A new data hiding method for H.265/HEVC video streams without intra-frame distortion drift, *Multimed Tools Appl*, 2019, **78**, 6459–6486.
- [139] Liu Y., Zhao H., Liu S., Feng C., Liu S., A Robust and Improved Visual Quality Data Hiding Method for HEVC, *IEEE Access*, 2018, **6**, 53984-53997.
- [140] Kaur G., Kasana S. S., Sharma M. K., An efficient authentication scheme for high efficiency video coding/H.265, *Multimed Tools Appl*, 2019, **78**, 21245–21271.
- [141] Biswas K., A Robust and High Capacity Data Hiding Method for H.265/HEVC Compressed Videos with Block Roughness Measure and Error Correcting Techniques, *Symmetry*, 2019, **11**, 1360.
- [142] Liu S., Xu D., A robust steganography method for HEVC based on secret sharing, *Cognitive Systems Research*, 2020, **59**, 207–220.

- [143] Jiang B., Yang G., Chen W., A CABAC Based HEVC Video Steganography Algorithm without Bitrate Increase, *Journal of Computational Information Systems*, 2015, **11**(6), 2121–2130
- [144] Van L. P., De Praeter J., Van Wallendael G., De Cock J., Van de Walle R., Out-of-the-loop information hiding for HEVC video, *2015 IEEE International Conference on Image Processing (ICIP)*, Quebec City, Canada, 27-30 Sep. 2015.
- [145] Ogawa K., Ohtake G., Watermarking for HEVC/H.265 stream, *2015 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, USA, 9-12 Jan. 2015.
- [146] Xu D., Commutative Encryption and Data Hiding in HEVC Video Compression, *IEEE Access*, 2019, **7**, 66028-66041.
- [147] Guan B., Xu D., Li Q., An Efficient Commutative Encryption and Data Hiding Scheme for HEVC Video, *IEEE Access*, 2020, **8**, 60232-60245.
- [148] Priya S., Amritha P. P., Information Hiding in H.264, H.265, and MJPEG, *International Conference on Soft Computing Systems, Advances in Intelligent Systems and Computing*, Chennai, India, 2016.
- [149] Shaikh S., Sayyad S., Data Hiding in Encrypted HEVC/AVC Video Streams, *International Journal of Advanced Research in Computer and Communication Engineering*, 2016, **5**(8), 60-65.
- [150] Mohammed A. A., Ali N. A., Robust video watermarking scheme using high efficiency video coding attack, *Multimed Tools Appl*, 2018, **77**, 2791-2806.
- [151] Zhou, Y., Wang C., Zhou, X., An Intra-Drift-Free Robust Watermarking Algorithm in High Efficiency Video Coding Compressed Domain, *IEEE Access*, 2019, **7**, 132991-133007.
- [152] Galiano D. R., Del Barrio A.A., Botella G., Cuesta D., Efficient embedding and retrieval of information for high-resolution videos coded with HEVC, *Computers and Electrical Engineering*, 2020, **81**, 106541.
- [153] Tasdemir K., Kurugollu F., Sezer S., A steganalysis system utilizing temporal pixel correlation of HEVC video, *2015 23rd Signal Processing and Communications Applications Conference (SIU)*, Malatya, Turkey, 16-19 May. 2015.
- [154] Wu K., Research of Video Steganalysis Algorithm Based on H265 Protocol, *2015 International Conference on Energy, Materials and Manufacturing Engineering (EMME 2015)*, Kuala Lumpur, Malaysia, 15-16 Oct. 2015.
- [155] Li Z., Meng L., Xu S., Li Z., Shi Y., Liang Y., A HEVC Video Steganalysis Algorithm Based on PU Partition Modes, *CMC-Computers Materials & Continua*, 2019, **59**(2), 563-574.

- [156] Crandall R., Some notes on steganography, *Steganography Mailing List*, 1998, 1-6.
- [157] Fan L., Gao T., Yang Q., Cao Y., An extended matrix encoding algorithm for steganography of high embedding efficiency, *Comput. Electr. Eng.*, 2011, **37**(6), 973–981.
- [158] Westfeld, A., F5-A Steganographic Algorithm, Editor: Moskowitz I. S., *Information Hiding. Lecture Notes in Computer Science*, **2137**, Springer, Berlin, 289–302, 2001.
- [159] Shanableh T., Matrix encoding for data hiding using multilayer video coding and transcoding solutions. *Signal Process. Image Commun.*, 2012, **27**, 1025–1034.





EKLER

EK-A

Hata Yayılmını Önleme Yöntemi

Veri gizleme işlemi için 4×4 TU'ya eklenen değerlerin 4×4 TU'daki en sağ sütunu ve en alt satırı değiştirmedeğinin ispatı:

Önerme:

$$S_{4 \times 4} = \begin{bmatrix} A & B & C & D \\ C & C & 0 & -C \\ D & -A & -C & B \\ B & -D & C & -A \end{bmatrix} = \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix},$$

$$\epsilon_{4 \times 4} = \begin{bmatrix} \epsilon_{00} & \epsilon_{01} & \epsilon_{02} & \epsilon_{03} \\ \epsilon_{10} & \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{20} & \epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{30} & \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{bmatrix},$$

α sabit bir değer ve $R_{4 \times 4} = S^T (\epsilon \times \alpha) S$ olmak üzere,

$$\text{Eğer } \epsilon_{4 \times 4} = \begin{bmatrix} t & 0 & -t & t \\ 0 & 0 & 0 & 0 \\ -t & 0 & t & -t \\ t & 0 & -t & t \end{bmatrix} \text{ olursa, } R_{4 \times 4} \text{ matrisinin en sağ sütunu ve en alt satırı sıfır}$$

olur. Yani, $R_{i3} = 0$ ($i=0,1,2,3$) ve $R_{3j} = 0$ ($j=0,1,2,3$) olur.

$$\text{Ayrıca, } R_{4 \times 4} = \alpha \times t \times \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 9C^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ olarak elde edilir.}$$

İspat:

$R_{4 \times 4} = S^T (\epsilon \times \alpha) S$ iken α skaler bir değer olduğu için $R_{4 \times 4} = \alpha \times S^T \epsilon S$ olarak yazılabilir.

İşlem kolaylığı için α değeri ilk başta göz ardı edilmekte olup, ispat sürecinin sonunda eşitliğe eklenecektir.

$$R = S^T \epsilon S = \begin{bmatrix} 29 & 74 & 84 & 55 \\ 55 & 74 & -29 & -84 \\ 74 & 0 & -74 & 74 \\ 84 & -74 & 55 & -29 \end{bmatrix} \begin{bmatrix} \epsilon_{00} & \epsilon_{01} & \epsilon_{02} & \epsilon_{03} \\ \epsilon_{10} & \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{20} & \epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{30} & \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{bmatrix} \begin{bmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{bmatrix}$$

$R_{i3}=0$ ($i=0,1,2,3$) için, $S^T(\varepsilon S)$ işlemi yapıldığında

$$R_{03}=29(84\varepsilon_{00} - 74\varepsilon_{01}+55\varepsilon_{02} - 29\varepsilon_{03})+74(84\varepsilon_{10} - 74\varepsilon_{11}+55\varepsilon_{12} - 29\varepsilon_{13})+ \\ 84(84\varepsilon_{20} - 74\varepsilon_{21}+55\varepsilon_{22} - 29\varepsilon_{23})+55(84\varepsilon_{30} - 74\varepsilon_{31}+55\varepsilon_{32} - 29\varepsilon_{33})$$

$$R_{13}=55(84\varepsilon_{00} - 74\varepsilon_{01}+55\varepsilon_{02} - 29\varepsilon_{03})+74(84\varepsilon_{10} - 74\varepsilon_{11}+55\varepsilon_{12} - 29\varepsilon_{13}) - \\ 29(84\varepsilon_{20} - 74\varepsilon_{21}+55\varepsilon_{22} - 29\varepsilon_{23}) - 84(84\varepsilon_{30} - 74\varepsilon_{31}+55\varepsilon_{32} - 29\varepsilon_{33})$$

$$R_{23}=74(84\varepsilon_{00} - 74\varepsilon_{01}+55\varepsilon_{02} - 29\varepsilon_{03})+0(84\varepsilon_{10} - 74\varepsilon_{11}+55\varepsilon_{12} - 29\varepsilon_{13}) - \\ 74(84\varepsilon_{20} - 74\varepsilon_{21}+55\varepsilon_{22} - 29\varepsilon_{23})+74(84\varepsilon_{30} - 74\varepsilon_{31}+55\varepsilon_{32} - 29\varepsilon_{33})$$

$$R_{33}=84(84\varepsilon_{00} - 74\varepsilon_{01}+55\varepsilon_{02} - 29\varepsilon_{03}) - 74(84\varepsilon_{10} - 74\varepsilon_{11}+55\varepsilon_{12} - 29\varepsilon_{13})+ \\ 55(84\varepsilon_{20} - 74\varepsilon_{21}+55\varepsilon_{22} - 29\varepsilon_{23}) - 29(84\varepsilon_{30} - 74\varepsilon_{31}+55\varepsilon_{32} - 29\varepsilon_{33})$$

$$R_{i3} = (84\varepsilon_{00} - 74\varepsilon_{01}+55\varepsilon_{02} - 29\varepsilon_{03}) \times \begin{bmatrix} 29 \\ 55 \\ 74 \\ 84 \end{bmatrix} + (84\varepsilon_{10} - 74\varepsilon_{11}+55\varepsilon_{12} - 29\varepsilon_{13}) \times \begin{bmatrix} 74 \\ 74 \\ 0 \\ -74 \end{bmatrix} \\ + (84\varepsilon_{20} - 74\varepsilon_{21}+55\varepsilon_{22} - 29\varepsilon_{23}) \times \begin{bmatrix} 84 \\ -29 \\ -74 \\ 55 \end{bmatrix} + (84\varepsilon_{30} - 74\varepsilon_{31}+55\varepsilon_{32} - 29\varepsilon_{33}) \times \begin{bmatrix} 55 \\ -84 \\ 74 \\ -29 \end{bmatrix} \\ = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 29 \\ 55 \\ 74 \\ 84 \end{bmatrix}, \begin{bmatrix} 74 \\ 74 \\ 0 \\ -74 \end{bmatrix}, \begin{bmatrix} 84 \\ -29 \\ -74 \\ 55 \end{bmatrix} \text{ ve } \begin{bmatrix} 55 \\ -84 \\ 74 \\ -29 \end{bmatrix} \text{ vektörleri lineer bağımsızdırlar.}$$

Dolayısıyla yeni eşitlikler,

$$84\varepsilon_{00} - 74\varepsilon_{01}+55\varepsilon_{02} - 29\varepsilon_{03}=0$$

$$84\varepsilon_{10} - 74\varepsilon_{11}+55\varepsilon_{12} - 29\varepsilon_{13}=0$$

$$84\varepsilon_{20} - 74\varepsilon_{21}+55\varepsilon_{22} - 29\varepsilon_{23}=0$$

$$84\varepsilon_{30} - 74\varepsilon_{31}+55\varepsilon_{32} - 29\varepsilon_{33}=0$$

şeklinde olur.

$84\varepsilon_{00} - 74\varepsilon_{01} + 55\varepsilon_{02} - 29\varepsilon_{03} = 0$ eşitliği $\varepsilon_{00} = -\varepsilon_{02} = \varepsilon_{03}$ ve $\varepsilon_{01} = 0$ için sağlanır.

$84\varepsilon_{10} - 74\varepsilon_{11} + 55\varepsilon_{12} - 29\varepsilon_{13} = 0$ eşitliği $\varepsilon_{10} = -\varepsilon_{12} = \varepsilon_{13}$ ve $\varepsilon_{11} = 0$ için sağlanır.

$84\varepsilon_{20} - 74\varepsilon_{21} + 55\varepsilon_{22} - 29\varepsilon_{23} = 0$ eşitliği $\varepsilon_{20} = -\varepsilon_{22} = \varepsilon_{23}$ ve $\varepsilon_{21} = 0$ için sağlanır.

$84\varepsilon_{30} - 74\varepsilon_{31} + 55\varepsilon_{32} - 29\varepsilon_{33} = 0$ eşitliği $\varepsilon_{30} = -\varepsilon_{32} = \varepsilon_{33}$ ve $\varepsilon_{31} = 0$ için sağlanır.

Sonuç olarak $R_{i3}=0$ ($i=0,1,2,3$) eşitliği $\varepsilon_{4 \times 4} = \begin{bmatrix} \varepsilon_{00} & 0 & -\varepsilon_{00} & \varepsilon_{00} \\ \varepsilon_{10} & 0 & -\varepsilon_{10} & \varepsilon_{10} \\ \varepsilon_{20} & 0 & -\varepsilon_{20} & \varepsilon_{20} \\ \varepsilon_{30} & 0 & -\varepsilon_{30} & \varepsilon_{30} \end{bmatrix}$ durumunda elde

edilir.

$R_{3j}=0$ ($j=0,1,2,3$) için, $S^T(\varepsilon S)$ işlemi yapıldığında,

$$R_{30} = 29(84\varepsilon_{00} - 74\varepsilon_{10} + 55\varepsilon_{20} - 29\varepsilon_{30}) + 74(84\varepsilon_{01} - 74\varepsilon_{11} + 55\varepsilon_{21} - 29\varepsilon_{31}) + 84(84\varepsilon_{02} - 74\varepsilon_{12} + 55\varepsilon_{22} - 29\varepsilon_{32}) + 55(84\varepsilon_{03} - 74\varepsilon_{13} + 55\varepsilon_{23} - 29\varepsilon_{33})$$

$$R_{31} = 55(84\varepsilon_{00} - 74\varepsilon_{10} + 55\varepsilon_{20} - 29\varepsilon_{30}) + 74(84\varepsilon_{01} - 74\varepsilon_{11} + 55\varepsilon_{21} - 29\varepsilon_{31}) - 29(84\varepsilon_{02} - 74\varepsilon_{12} + 55\varepsilon_{22} - 29\varepsilon_{32}) - 84(84\varepsilon_{03} - 74\varepsilon_{13} + 55\varepsilon_{23} - 29\varepsilon_{33})$$

$$R_{32} = 74(84\varepsilon_{00} - 74\varepsilon_{10} + 55\varepsilon_{20} - 29\varepsilon_{30}) + 0(84\varepsilon_{01} - 74\varepsilon_{11} + 55\varepsilon_{21} - 29\varepsilon_{31}) - 74(84\varepsilon_{02} - 74\varepsilon_{12} + 55\varepsilon_{22} - 29\varepsilon_{32}) + 74(84\varepsilon_{03} - 74\varepsilon_{13} + 55\varepsilon_{23} - 29\varepsilon_{33})$$

$$R_{33} = 84(84\varepsilon_{00} - 74\varepsilon_{10} + 55\varepsilon_{20} - 29\varepsilon_{30}) - 74(84\varepsilon_{01} - 74\varepsilon_{11} + 55\varepsilon_{21} - 29\varepsilon_{31}) + 55(84\varepsilon_{02} - 74\varepsilon_{12} + 55\varepsilon_{22} - 29\varepsilon_{32}) - 29(84\varepsilon_{03} - 74\varepsilon_{13} + 55\varepsilon_{23} - 29\varepsilon_{33})$$

$$R_{3j} = (84\varepsilon_{00} - 74\varepsilon_{10} + 55\varepsilon_{20} - 29\varepsilon_{30}) \times \begin{bmatrix} 29 \\ 55 \\ 74 \\ 84 \end{bmatrix} + (84\varepsilon_{01} - 74\varepsilon_{11} + 55\varepsilon_{21} - 29\varepsilon_{31}) \times \begin{bmatrix} 74 \\ 74 \\ 0 \\ -74 \end{bmatrix} + (84\varepsilon_{02} - 74\varepsilon_{12} + 55\varepsilon_{22} - 29\varepsilon_{32}) \times \begin{bmatrix} 84 \\ -29 \\ -74 \\ 55 \end{bmatrix} + (84\varepsilon_{03} - 74\varepsilon_{13} + 55\varepsilon_{23} - 29\varepsilon_{33}) \times \begin{bmatrix} 55 \\ -84 \\ 74 \\ -29 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$\begin{bmatrix} 29 \\ 55 \\ 74 \\ 84 \end{bmatrix}, \begin{bmatrix} 74 \\ 74 \\ 0 \\ -74 \end{bmatrix}, \begin{bmatrix} 84 \\ -29 \\ -74 \\ 55 \end{bmatrix}$ ve $\begin{bmatrix} 55 \\ -84 \\ 74 \\ -29 \end{bmatrix}$ vektörleri lineer bağımsızdırlar. Dolayısıyla yeni eşitlikler,

$$84\varepsilon_{00} - 74\varepsilon_{10} + 55\varepsilon_{20} - 29\varepsilon_{30} = 0$$

$$84\varepsilon_{01} - 74\varepsilon_{11} + 55\varepsilon_{21} - 29\varepsilon_{31} = 0$$

$$84\varepsilon_{02} - 74\varepsilon_{12} + 55\varepsilon_{22} - 29\varepsilon_{32} = 0$$

$$84\varepsilon_{03} - 74\varepsilon_{13} + 55\varepsilon_{23} - 29\varepsilon_{33} = 0$$

şeklinde olur.

$84\varepsilon_{00} - 74\varepsilon_{10} + 55\varepsilon_{20} - 29\varepsilon_{30} = 0$ eşitliği $\varepsilon_{00} = -\varepsilon_{20} = \varepsilon_{30}$ ve $\varepsilon_{10} = 0$ için sağlanır.

$84\varepsilon_{01} - 74\varepsilon_{11} + 55\varepsilon_{21} - 29\varepsilon_{31} = 0$ eşitliği $\varepsilon_{01} = -\varepsilon_{21} = \varepsilon_{31}$ ve $\varepsilon_{11} = 0$ için sağlanır.

$84\varepsilon_{02} - 74\varepsilon_{12} + 55\varepsilon_{22} - 29\varepsilon_{32} = 0$ eşitliği $\varepsilon_{02} = -\varepsilon_{22} = \varepsilon_{32}$ ve $\varepsilon_{12} = 0$ için sağlanır.

$84\varepsilon_{03} - 74\varepsilon_{13} + 55\varepsilon_{23} - 29\varepsilon_{33} = 0$ eşitliği $\varepsilon_{03} = -\varepsilon_{23} = \varepsilon_{33}$ ve $\varepsilon_{13} = 0$ için sağlanır.

Sonuç olarak $R_{3j}=0$ ($i=0,1,2,3$) eşitliği $\varepsilon_{4 \times 4} = \begin{bmatrix} \varepsilon_{00} & \varepsilon_{01} & \varepsilon_{02} & \varepsilon_{03} \\ 0 & 0 & 0 & 0 \\ -\varepsilon_{00} & -\varepsilon_{01} & -\varepsilon_{02} & -\varepsilon_{03} \\ \varepsilon_{00} & \varepsilon_{01} & \varepsilon_{02} & \varepsilon_{03} \end{bmatrix}$

durumunda elde edilir.

$R_{4 \times 4}$ matrisinin en sağ sütununun ve en alt satırının sıfır olması için $R_{i3}=0$ ve $R_{3j}=0$ durumlarının aynı anda sağlanması gerekmektedir. Bunun için elde edilen $\varepsilon_{4 \times 4}$ matrisleri birleştirilmektedir.

$$R_{i3}=0 \text{ için elde edilen } \varepsilon_{4 \times 4} = \begin{bmatrix} \varepsilon_{00} & 0 & -\varepsilon_{00} & \varepsilon_{00} \\ \varepsilon_{10} & 0 & -\varepsilon_{10} & \varepsilon_{10} \\ \varepsilon_{20} & 0 & -\varepsilon_{20} & \varepsilon_{20} \\ \varepsilon_{30} & 0 & -\varepsilon_{30} & \varepsilon_{30} \end{bmatrix} \text{ ve } R_{3j}=0 \text{ için elde edilen}$$

$$\varepsilon_{4 \times 4} = \begin{bmatrix} \varepsilon_{00} & \varepsilon_{01} & \varepsilon_{02} & \varepsilon_{03} \\ 0 & 0 & 0 & 0 \\ -\varepsilon_{00} & -\varepsilon_{01} & -\varepsilon_{02} & -\varepsilon_{03} \\ \varepsilon_{00} & \varepsilon_{01} & \varepsilon_{02} & \varepsilon_{03} \end{bmatrix} \text{ matrisleri birleştirildiğinde } \varepsilon_{01} = 0, \varepsilon_{10} = 0 \text{ ve}$$

$\varepsilon_{00} = -\varepsilon_{02} = \varepsilon_{03} = -\varepsilon_{20} = \varepsilon_{30}$ olarak bulunur. Dolayısıyla,

$$\varepsilon_{4 \times 4} = \begin{bmatrix} \varepsilon_{00} & 0 & -\varepsilon_{00} & \varepsilon_{00} \\ 0 & 0 & 0 & 0 \\ -\varepsilon_{00} & 0 & \varepsilon_{00} & -\varepsilon_{00} \\ \varepsilon_{00} & 0 & -\varepsilon_{00} & \varepsilon_{00} \end{bmatrix} \text{ olarak oluşur.}$$

$$\varepsilon_{00}=t \text{ olarak kabul edildiğinde, } \varepsilon_{4 \times 4} = \begin{bmatrix} t & 0 & -t & t \\ 0 & 0 & 0 & 0 \\ -t & 0 & t & -t \\ t & 0 & -t & t \end{bmatrix} \text{ olur.}$$

İşlem kolaylığı için ilk başta göz ardı edilen α değeri yerine yazılıp R matrisinin tüm

$$\text{değerleri hesaplanırsa } R_{4 \times 4} = \alpha \times S^T \varepsilon S = \alpha \times t \times \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 9C^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ olarak elde edilir.}$$

KİŞİSEL YAYIN VE ESERLER

- [1] **Konyar M. Z.**, Akbulut O., Öztürk S., Matrix encoding-based high-capacity and high-fidelity reversible data hiding in HEVC, *Signal Image and Video Processing*, 2020, **14**, 897-905.
- [2] **Konyar M. Z.**, Öztürk S., Reed Solomon Coding-based Medical Image Data Hiding Method against Salt and Pepper Noise, *Symmetry*, 2020, **12**(6), 899.
- [3] Mamak U, **Konyar M. Z.**, Solak S., Uçar M. H. B., Gerçek Zamanlı Yüz Tanıma Tabanlı Personel Kontrol ve Takip Sistemi Tasarımı, *Avrupa Bilim ve Teknoloji Dergisi*, 2020, **19**, 497-504.
- [4] Kurnaz H., **Konyar M. Z.**, Sondaş A., Yakın Histogramlar Temelli Yeni Bir Hibrit Veri Gizleme Yöntemi. *Avrupa Bilim ve Teknoloji Dergisi*, 2020, **18**, 683-694.
- [5] **Konyar M. Z.**, İlkin S., Çelik N., Sondaş A., Steganografi için En Uygun Resmi Belirleyen Uygulama Arayüz Tasarımı, *İleri Teknoloji Bilimleri Dergisi*, 2018, **7**(1), 83-89.
- [6] **Konyar M. Z.**, Akbulut O., Öztürk S., HEVC’de Matris Kodlama Tabanlı Veri Gizleme, *3. Uluslararası Bilgisayar Bilimleri ve Mühendisliği Konferansı (UBMK)*, Saraybosna, Bosna Hersek 20-23 Eylül 2018.
- [7] **Konyar M. Z.**, Öztürk S., HEVC Videolarında Kullanılan Damgalama Yöntemlerin Genel Değerlendirmesi, *III. Uluslararası Mesleki Ve Teknik Bilimler Kongresi*, Gaziantep, Türkiye, 21-22 Haziran 2018.
- [8] **Konyar M. Z.**, Öztürk S., Hasta Bilgilerini Tıbbi Görüntülere Gizlemek için Yeni Bir Yaklaşım, *III. Uluslararası Mesleki Ve Teknik Bilimler Kongresi*, Gaziantep, Türkiye, 21-22 Haziran 2018.
- [9] Akbay K., **Konyar M. Z.**, İlkin S., Sondaş A., Data hiding using shuffle algorithm and LSB method, *IEEE 26th Signal Processing and Communications Applications Conference (SIU2018)*, İzmir, Türkiye, 2-5 Mayıs 2018.
- [10] **Konyar M. Z.**, Öztürk S., Comparison of Data Hiding Methods on HEVC/H.265, *International Conference on Engineering Technologies (ICENTE 2017)*, Konya, Türkiye, 7-9 Aralık 2017.
- [11] Çelik N., **Konyar M. Z.**, İlkin S., Sondaş A., Steganografi İçin En Uygun Resmi Belirleyen Uygulama Arayüz Tasarımı, *Uluslararası Mühendislik Araştırmaları Sempozyumu (UMAS)*, Düzce, Türkiye, 11-13 Eylül 2017.
- [12] **Konyar M. Z.**, Ertürk S., Ultrason Görüntülerinin Sözcük Renklendirilmesi. *IEEE 25th Signal Processing and Communications Applications Conference (SIU2017)*, Antalya, Türkiye, 15-18 Mayıs 2017.

- [13] **Konyar M. Z.**, Şayli Ö., Ertürk S., (2015). Pseudocoloring Ultrasound Images with Two Dimensional Discrete Wavelet Transform, *Tıp Teknolojileri Ulusal Kongresi (TIPTEKNO)*, Bodrum, Türkiye, 15-18 Ekim 2015
- [14] **Konyar M. Z.**, Ertürk S., (2015). Enhancement of Ultrasound Images with Bilateral Filter and Rayleigh CLAHE, *IEEE 23th Signal Processing and Communications Applications Conference (SIU2015)*, Malatya, Türkiye, 16 - 19 Mayıs 2015.



ÖZGEÇMİŞ

İlköğretimi sırasıyla Atatürk İlköğretim ve 100. Yıl İlköğretim Okullarında okuduktan sonra, 2002 yılında kazandığı Ağrı Anadolu Lisesi'nden 2006 yılında mezun oldu. Aynı yıl yerleştiği Kocaeli Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü'nü, 2010 yılında, Elektronik ve Haberleşme Mühendisi olarak tamamladı. Ocak 2011'de Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı'nda yüksek lisans eğitimine başlayıp, Şubat 2014'te Ultrason Görüntülerinin Sözde Renklendirilmesi isimli Yüksek Lisans Tezi ile Elektronik ve Haberleşme Mühendisi Yüksek Mühendisi unvanını aldı.

Şubat 2014'te Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı'nda başladığı doktora eğitimine halen devam etmekte olup, doktora eğitimi TÜBİTAK Bilim İnsanı Destek Programları Başkanlığı (BİDEB) tarafından 2211/A Genel Yurt İçi Doktora Burs Programıyla desteklenmektedir.

Ocak 2012'den beri Kocaeli Üniversitesi Teknoloji Fakültesi Bilişim Sistemleri Mühendisliği Bölümü'nde Araştırma Görevlisi olarak çalışmaktadır.