



**T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



DOKTORA TEZİ

SYMPES Yöntemi ile Video Görüntülerinin Sıkıştırılması

Türker Togay AKSOY

DANIŞMAN

Prof. Dr. Bekir Sıddık Binboğa YARMAN

Biyomedikal Mühendisliği Anabilim Dalı

Biyomedikal Mühendisliği Programı

Mayıs, 2023

BÜTÇE DESTEKLERİ

-SYMPES Yöntemi ile Video Görüntülerinin Sıkıştırılması

Bu tez çalışması için herhangi bir kurumdan bütçe desteği alınmamıştır.



TEŞEKKÜR

Öncelikle tez çalışmamda eşsiz bilgisi ve düşüncelerinden yararlandığım danışman hocam çok değerli Prof. Dr. Bekir Sıddık Binboğa YARMAN'a eğitimim boyunca bana kattıkları için sonsuz teşekkürlerimi sunarım. Tez izleme komitemde yer alan ve tüm tez süresince değerli fikirleriyle yönlendiren, her zaman desteklerini hissettiğim çok değerli Prof. Dr. Osman Nuri UÇAN ve Prof. Dr. Fırat KAÇAR hocalarıma teşekkürü bir borç bilirim. Çok değerli canım hocam Doç. Dr. Aysel Ersoy YILMAZ'a ise her anımda yanımda olduğu ve bıkmadan usanmadan bana katlandığı için çok teşekkür ederim. Bu süreçte çalışmaktan büyük keyif aldığım hem bilgisi hem de manevi desteğiyle yanımda olan çok değerli dostum Dr. Öğr. Üyesi Ali SARIKAŞ'a çok teşekkür ederim. Tezimin son döneminde tanıdığım, desteğini ve samimiyetini esirgemeyen arkadaşım Osman KORKMAZ'a teşekkürü bir borç bilirim. Hayatım boyunca maddi ve manevi desteklerini hissettiğim canım aileme en içten teşekkürü bir borç bilirim.

Mayıs 2023

Türker Togay AKSOY

İÇİNDEKİLER

Sayfa No

TEZ KABUL VE ONAYI.....	ii
BEYAN	iii
BÜTÇE DESTEKLERİ	iv
TEŞEKKÜR.....	v
İÇİNDEKİLER.....	vi
ŞEKİL LİSTESİ	x
TABLO LİSTESİ.....	xvii
SİMGE VE KISALTMA LİSTESİ	xxiv
ÖZET	xxvi
ABSTRACT	xxviii
1. GİRİŞ.....	1
2. KAVRAMSAL ÇERÇEVE	3
2.1. GÖRÜNTÜ İŞLEME ALTYAPISI.....	3
2.2. GÖRÜNTÜ ÇEŞİTLERİ	4
2.2.1 Gri Tonlamalı Görüntü (Grayscale Image)	4
2.2.2. İkili Görüntü (Binary Image)	5
2.2.4. Aşınma (Erosion)	6
2.2.5. Genişleme (Dilation)	6
2.2.6. Morfolojik Geriçatma.....	7
2.2.7. Sınır Temizleme	7
2.2.8. Bağlı Bileşenler	8
2.3. GÖRÜNTÜ İYİLEŞTİRME	8
2.3.1 Histogram	8
2.4. FİLTRELER	9
2.6. GÖRÜNTÜ SIKIŞTIRMA	17
2.6.1. Görüntü Sıkıştırma Yöntemleri.....	17
2.6.2. Kayıpsız Görüntü Sıkıştırma Yöntemleri.....	20

2.7. KAYIPSIZ SIKIŞTIRMADA KULLANILAN GÖRÜNTÜ DOSYA FORMATLARI	23
2.7.1. GIF	23
2.7.2. TIFF.....	24
2.7.3. PNG	25
2.7.4. Kayıplı Görüntü Sıkıştırma Yöntemleri.....	25
2.8. GÖRÜNTÜ BÖLÜTLEMEYLE BİRLİKTE KULLANILAN KODLAMA BİÇİMLERİ.....	30
2.8.1. Koşu-Uzunluğu Kodlama.....	30
2.8.2. Zincir Kodlama	32
2.9. AMAÇ FONKSİYONLU ALGORİTMALARLA GÖRÜNTÜ SIKIŞTIRMA SONUÇLARI.....	32
2.10. BÖLGE BÜYÜTME TEMELLİ ALGORİTMALARLA GÖRÜNTÜ SIKIŞTIRMA SONUÇLARI.....	34
2.11. VİDEO KODLAMA.....	34
2.11.1. Video CODEC'i.....	40
2.12. TAHMİN MODELİ	41
2.12.1. Zamansal Tahmin.....	42
2.12.2. Mekansal Model: Tahmin İçeriği	42
2.12.3. İmaj Modeli	42
2.13. VİDEO SIKIŞTIRMA	44
2.14. HAREKETLİ GÖRÜNTÜ SIKIŞTIRMA.....	46
2.15. HAREKETLİ GÖRÜNTÜ SIKIŞTIRMA YÖNTEMLERİ.....	49
2.15.1. Kayıpsız Hareketli Görüntü Sıkıştırma Yöntemi	49
2.16. KAYIPLI HAREKETLİ GÖRÜNTÜ SIKIŞTIRMA YÖNTEMİ	50
2.16.1. MPEG-1	50
2.16.2. MPEG-2	50
2.16.3. MPEG-4	51
2.16.4. H-264.....	51
2.16.5. H-265.....	53
3. YÖNTEM	56
3.1. KULLANILAN MATEMATİKSEL MODEL.....	56
3.1.1. Sürekli Zaman Bölgesinde Tanım.....	56
3.1.2. Ayrık Zaman Bölgesinde Tanım	57
3.2. SYMPES YÖNTEMİ İLE VİDEO GÖRÜNTÜLERİN KODLANMASI.....	58

3.2.1. Sympes Metodu ile Video Görüntülerin Modellenmesi	58
3.2.2. Enerji ve Yapı Bloklarının indirgenmesi	62
3.3. ALGORİTMA	63
3.3.1. Sympes Yöntemi ile Kodlama Algoritması	63
3.3.2. SYMPES Kod Çözme Algoritması Aşamaları.....	65
3.3.3. SYMPES Blok Görüntü Yaklaşımı.....	67
3.4. MATERYAL	68
3.4.1. Evrensel Küme Oluşturulurken Kullanılan Görüntüler	68
3.4.2. Evrensel Küme İçin Oluşturulacak Videonun İçeriğinden Bağımsız Görüntüler ve Özellikleri.....	68
3.4.3. Evrensel Küme İçin Oluşturulacak Videodan Rasgele Seçilen Görüntüler ve Özellikleri.....	69
3.4.4. SYMPES Yöntemi ile Yeniden Oluşturulacak Videoların Özellikleri	69
4. BULGULAR	70
4.1. <i>Vtg-1</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-1</i> SONUÇLARI.....	70
4.2. <i>Vtg-1</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-2</i> SONUÇLARI.....	72
4.3. <i>Vtg-1</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-3</i> SONUÇLARI.....	74
4.4. <i>Vtg-2</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-1</i> SONUÇLARI.....	76
4.5. <i>Vtg-2</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-2</i> SONUÇLARI.....	78
4.6. <i>Vtg-2</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-3</i> SONUÇLARI.....	80
4.7. <i>Vtg-3</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-1</i> SONUÇLARI.....	82
4.8. <i>Vtg-3</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-2</i> SONUÇLARI.....	84
4.9. <i>Vtg-3</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-3</i> SONUÇLARI.....	86
4.10. <i>Vtg-4</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-2</i> SONUÇLARI.....	88
4.11. <i>Vtg-5</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-3</i> SONUÇLARI.....	90
4.12. <i>Vtg-6</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-1</i> SONUÇLARI.....	92

4.13. <i>Vtg-1</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-1</i> SONUÇLARI.....	94
4.14. <i>Vtg-1</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-2</i> SONUÇLARI.....	96
4.15. <i>Vtg-1</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-3</i> SONUÇLARI.....	98
4.16. <i>Vtg-2</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-1</i> SONUÇLARI.....	100
4.17. <i>Vtg-2</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-2</i> SONUÇLARI.....	102
4.18. <i>Vtg-2</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-3</i> SONUÇLARI.....	104
4.19. <i>Vtg-3</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-1</i> SONUÇLARI.....	106
4.20. <i>Vtg-3</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-2</i> SONUÇLARI.....	108
4.21. <i>Vtg-3</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-3</i> SONUÇLARI.....	110
4.22. <i>Vtg-4</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-2</i> SONUÇLARI.....	112
4.23. <i>Vtg-5</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-3</i> SONUÇLARI.....	114
4.24. <i>Vtg-6</i> İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN <i>Vv-1</i> SONUÇLARI.....	116
5. TARTIŞMA.....	119
6. SONUÇ VE ÖNERİLER	123
KAYNAKLAR.....	124
İNTİHAL RAPORU İLK SAYFASI	134
ETİK KURUL İZİN YAZISI	135
KURUM İZNİ YAZILARI.....	136
ÖZGEÇMİŞ	137

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1:	NxM sayısal görüntü	3
Şekil 2.2:	İki boyutlu görüntüde sırasıyla 4 ve 8-bitişiklik	6
Şekil 2.3:	B yapılandırma elemanı ile X'in aşınma işlemi.....	6
Şekil 2.4:	B yapılandırma elemanı ile X'in genişleme işlemi.....	7
Şekil 2.5:	Filtre boyutu [11]	9
Şekil 2.6:	Gauss filtre çıktısı [16]	10
Şekil 2.7:	Average filtre çıktısı [12]	10
Şekil 2.8:	Medyan filtre çıktısı [17].....	11
Şekil 2.9:	8 bitlik bir görüntü için, Log dönüşümü [18].....	12
Şekil 2.10:	Log Transform filtre çıktısı [19]	13
Şekil 2.11:	Fast Fourier dönüşümü [20]	14
Şekil 2.12:	Sobel kenar belirleme filtre çıktısı [21].....	15
Şekil 2.13:	Laplace kenar belirleme filtre çıktısı [22]	15
Şekil 2.14:	Prewitt kenar belirleme filtre çıktısı [23]	16
Şekil 2.15:	Log kenar belirleme filtre çıktısı [24]	16
Şekil 2.16:	Canny kenar belirleme filtre çıktısı [25]	17
Şekil 2.17:	JBIG sıkıştırma kodlamasının blok diyagramı.....	22
Şekil 2.18:	JPEG algoritmasının genel yapısı [82].....	26
Şekil 2.19:	PNG Görüntü.....	27
Şekil 2.20:	JPEG Görüntü	27
Şekil 2.21:	Karo ve kodblok dönüşümü [85].....	29
Şekil 2.22:	Koşu kodlama gösterimi.....	30
Şekil 2.23:	Farklı görüntüler için koşu uzunluğu kodlama	31
Şekil 2.24:	Bölütleme temelli sıkıştırılan görüntü.....	33
Şekil 2.25:	Video formatları için uzamsal çözünürlükler.....	35
Şekil 2.26:	Farklı kroma alt-örnekleme biçimleri	36
Şekil 2.27:	Video kodlama blok yapısı.....	37
Şekil 2.28:	Video kod çözme blok yapısı	39

Şekil 2.29:	Örnek GOP yapısı [88].	40
Şekil 2.30:	VCEG ve MPEG tarafından geliştirilen video kodlama standartları	45
Şekil 2.31:	Full HD bir içerik için video kodlama standartlarının başarımları [98].	46
Şekil 2.32:	Blok hareket tahmini [55]	47
Şekil 2.33:	H.264/AVC Çerçeve içi kestirim modları [106].	52
Şekil 2.34:	HEVC video kodlayıcı yapısı [107].	53
Şekil 3.1:	CEPB Oluşturma Basamakları Akış Şeması [108]	63
Şekil 3.2:	SYMPES kodlama kısmı akış diyagramı	65
Şekil 3.3:	SYMPES Kod Çözme Kısmı Akış Diyagramı	67
Şekil 3.4:	<i>Vtg-1</i>	68
Şekil 3.5:	<i>Vtg-2</i>	68
Şekil 3.6:	<i>Vtg-3</i>	68
Şekil 3.7:	<i>Vtg-4</i>	69
Şekil 3.8:	<i>Vtg-5</i>	69
Şekil 3.9:	<i>Vtg-6</i>	69
Şekil 4.1:	<i>F-1</i>	70
Şekil 4.2:	<i>F-11</i>	70
Şekil 4.3:	<i>F-21</i>	70
Şekil 4.4:	<i>Yof-1</i>	71
Şekil 4.5:	<i>Yof-11</i>	71
Şekil 4.6:	<i>Yof-21</i>	71
Şekil 4.7:	<i>Vtg-1</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> için her bir Karesinin Psnr ve Mse Değerleri Grafiği	72
Şekil 4.8:	<i>F-33</i>	72
Şekil 4.9:	<i>F-44</i>	72
Şekil 4.10:	<i>F-87</i>	72
Şekil 4.11:	<i>Yof-33</i>	73
Şekil 4.12:	<i>Yof-44</i>	73
Şekil 4.13:	<i>Yof-87</i>	73
Şekil 4.14:	<i>Vtg-1</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> için her bir Karesinin Psnr ve Mse Değerleri Grafiği	74
Şekil 4.15:	<i>F-100</i>	74
Şekil 4.16:	<i>F-110</i>	74
Şekil 4.17:	<i>F-123</i>	74
Şekil 4.18:	<i>Yof-100</i>	75

Şekil 4.19:	<i>Yof-110</i>	75
Şekil 4.20:	<i>Yof-123</i>	75
Şekil 4.21:	<i>Vtg-1</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> için her bir Karesinin <i>Psnr</i> ve <i>Mse</i> Değerleri Grafiği	76
Şekil 4.22:	<i>F-4</i>	76
Şekil 4.23:	<i>F-18</i>	76
Şekil 4.24:	<i>F-28</i>	76
Şekil 4.25:	<i>Yof-4</i>	77
Şekil 4.26:	<i>Yof-18</i>	77
Şekil 4.27:	<i>Yof-28</i>	77
Şekil 4.28:	<i>Vtg-2</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> için her bir Karesinin <i>Mse</i> ve <i>Psnr</i> Değerleri Grafiği	78
Şekil 4.29:	<i>F-5</i>	78
Şekil 4.30:	<i>F-45</i>	78
Şekil 4.31:	<i>F-85</i>	78
Şekil 4.32:	<i>Yof-5</i>	79
Şekil 4.33:	<i>Yof-45</i>	79
Şekil 4.34:	<i>Yof-85</i>	79
Şekil 4.35:	<i>Vtg-2</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> için her bir Karesinin <i>Psnr</i> ve <i>Mse</i> Değerleri Grafiği	80
Şekil 4.36:	<i>F-7</i>	81
Şekil 4.37:	<i>F-97</i>	81
Şekil 4.38:	<i>F-117</i>	81
Şekil 4.39:	<i>Yof-7</i>	81
Şekil 4.40:	<i>Yof-97</i>	81
Şekil 4.41:	<i>Yof-117</i>	81
Şekil 4.42:	<i>Vtg-2</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> için her bir Karesinin <i>Psnr</i> ve <i>Mse</i> Değerleri Grafiği	82
Şekil 4.43:	<i>F-2</i>	83
Şekil 4.44:	<i>F-20</i>	83
Şekil 4.45:	<i>F-30</i>	83
Şekil 4.46:	<i>Yof-2</i>	83
Şekil 4.47:	<i>Yof-20</i>	83
Şekil 4.48:	<i>Yof-30</i>	83
Şekil 4.49:	<i>Vtg-3</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> için her bir Karesinin <i>Psnr</i> ve <i>Mse</i> Değerleri Grafiği	84

Şekil 4.50:	<i>F-34</i>	85
Şekil 4.51:	<i>F-54</i>	85
Şekil 4.52:	<i>F-74</i>	85
Şekil 4.53:	<i>Yof-34</i>	85
Şekil 4.54:	<i>Yof-34</i>	85
Şekil 4.55:	<i>Yof-74</i>	85
Şekil 4.56:	<i>Vtg-3</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> için her bir Karesinin Psnr ve Mse Değerleri Grafiği	86
Şekil 4.57:	<i>F-42</i>	87
Şekil 4.58:	<i>F-102</i>	87
Şekil 4.59:	<i>F-122</i>	87
Şekil 4.60:	<i>Yof-42</i>	87
Şekil 4.61:	<i>Yof-42</i>	87
Şekil 4.62:	<i>Yof-42</i>	87
Şekil 4.63:	<i>Vtg-3</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> için her bir Karesinin Psnr ve Mse Değerleri Grafiği	88
Şekil 4.64:	<i>F-8</i>	89
Şekil 4.65:	<i>F-17</i>	89
Şekil 4.66:	<i>F-69</i>	89
Şekil 4.67:	<i>Yof-8</i>	89
Şekil 4.68:	<i>Yof-17</i>	89
Şekil 4.69:	<i>Yof-69</i>	89
Şekil 4.70:	<i>Vtg-4</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> için her bir Video Karesinin Psnr ve Mse Değerleri Grafiği	90
Şekil 4.71:	<i>F-21</i>	91
Şekil 4.72:	<i>F-41</i>	91
Şekil 4.73:	<i>F-111</i>	91
Şekil 4.74:	<i>Yof-21</i>	91
Şekil 4.75:	<i>Yof-41</i>	91
Şekil 4.76:	<i>Yof-111</i>	91
Şekil 4.77:	<i>Vtg-5</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> için her bir Video Karesinin Psnr ve Mse Değerleri Grafiği	92
Şekil 4.78:	<i>F-9</i>	93
Şekil 4.79:	<i>F-19</i>	93
Şekil 4.80:	<i>F-29</i>	93
Şekil 4.81:	<i>Yof-9</i>	93

Şekil 4.82:	<i>Yof-19</i>	93
Şekil 4.83:	<i>Yof-29</i>	93
Şekil 4.84:	<i>Vtg-6</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> için her bir Video Karesinin Psnr ve Mse Değerleri Grafiği	94
Şekil 4.85:	<i>F-1</i>	95
Şekil 4.86:	<i>F-11</i>	95
Şekil 4.87:	<i>F-21</i>	95
Şekil 4.88:	<i>Yof-1</i>	95
Şekil 4.89:	<i>Yof-11</i>	95
Şekil 4.90:	<i>Yof-21</i>	95
Şekil 4.91:	<i>Vtg-1</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> için her bir Karesinin Psnr ve Mse Değerleri Grafiği	96
Şekil 4.92:	<i>F-33</i>	97
Şekil 4.93:	<i>F-44</i>	97
Şekil 4.94:	<i>F-87</i>	97
Şekil 4.95:	<i>Yof-33</i>	97
Şekil 4.96:	<i>Yof-44</i>	97
Şekil 4.97:	<i>Yof-87</i>	97
Şekil 4.98:	<i>Vtg-1</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> için her bir Karesinin Psnr ve Mse Değerleri Grafiği	98
Şekil 4.99:	<i>F-100</i>	99
Şekil 4.100:	<i>F-110</i>	99
Şekil 4.101:	<i>F-123</i>	99
Şekil 4.102:	<i>Yof-100</i>	99
Şekil 4.103:	<i>Yof-110</i>	99
Şekil 4.104:	<i>Yof-123</i>	99
Şekil 4.105:	<i>Vtg-1</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> için her bir Karesinin Psnr ve Mse Değerleri Grafiği	100
Şekil 4.106:	<i>F-4</i>	101
Şekil 4.107:	<i>F-18</i>	101
Şekil 4.108:	<i>F-28</i>	101
Şekil 4.109:	<i>Yof-4</i>	101
Şekil 4.110:	<i>Yof-18</i>	101
Şekil 4.111:	<i>Yof-28</i>	101
Şekil 4.112:	<i>Vtg-2</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> için her bir Karesinin Mse ve Psnr Değerleri Grafiği	102

Şekil 4.113: <i>F</i> -5	103
Şekil 4.114: <i>F</i> -45	103
Şekil 4.115: <i>F</i> -85	103
Şekil 4.116: <i>Yof</i> -5	103
Şekil 4.117: <i>Yof</i> -45	103
Şekil 4.118: <i>Yof</i> -85	103
Şekil 4.119: <i>Vtg</i> -2 Evrensel Kümesi ile Yeniden Üretilen <i>Vv</i> -2 için her bir Karesinin <i>Psnr</i> ve Mse Değerleri Grafiği	104
Şekil 4.120: <i>F</i> -7	105
Şekil 4.121: <i>F</i> -97	105
Şekil 4.122: <i>F</i> -117	105
Şekil 4.123: <i>Yof</i> -7	105
Şekil 4.124: <i>Yof</i> -97	105
Şekil 4.125: <i>Yof</i> -117	105
Şekil 4.126: <i>Vtg</i> -2 Evrensel Kümesi ile Yeniden Üretilen <i>Vv</i> -3 için her bir Karesinin <i>Psnr</i> ve Mse Değerleri Grafiği	106
Şekil 4.127: <i>F</i> -2	107
Şekil 4.128: <i>F</i> -20	107
Şekil 4.129: <i>F</i> -30	107
Şekil 4.130: <i>Yof</i> -2	107
Şekil 4.131: <i>Yof</i> -20	107
Şekil 4.132: <i>Yof</i> -30	107
Şekil 4.133: <i>Vtg</i> -3 Evrensel Kümesi ile Yeniden Üretilen <i>Vv</i> -1 için her bir Karesinin <i>Psnr</i> ve Mse Değerleri Grafiği	108
Şekil 4.134: <i>F</i> -34	109
Şekil 4.135: <i>F</i> -54	109
Şekil 4.136: <i>F</i> -74	109
Şekil 4.137: <i>Yof</i> -34	109
Şekil 4.138: <i>Yof</i> -34	109
Şekil 4.139: <i>Yof</i> -74	109
Şekil 4.140: <i>Vtg</i> -3 Evrensel Kümesi ile Yeniden Üretilen <i>Vv</i> -2 için her bir Karesinin <i>Psnr</i> ve Mse Değerleri Grafiği	110
Şekil 4.141: <i>F</i> -42	111
Şekil 4.142: <i>F</i> -102	111
Şekil 4.143: <i>F</i> -122	111
Şekil 4.144: <i>Yof</i> -42	111

Şekil 4.145: <i>Yof-42</i>	111
Şekil 4.146: <i>Yof-42</i>	111
Şekil 4.147: <i>Vtg-3</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> için her bir Karesinin Psnr ve Mse Değerleri Grafiği	112
Şekil 4.148: <i>F-8</i>	113
Şekil 4.149: <i>F-17</i>	113
Şekil 4.150: <i>F-69</i>	113
Şekil 4.151: <i>Yof-8</i>	113
Şekil 4.152: <i>Yof-17</i>	113
Şekil 4.153: <i>Yof-69</i>	113
Şekil 4.154: <i>Vtg-4</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> için her bir Video Karesinin Psnr ve Mse Değerleri Grafiği	114
Şekil 4.155: <i>F-21</i>	115
Şekil 4.156: <i>F-41</i>	115
Şekil 4.157: <i>F-111</i>	115
Şekil 4.158: <i>Yof-21</i>	115
Şekil 4.159: <i>Yof-41</i>	115
Şekil 4.160: <i>Yof-111</i>	115
Şekil 4.161: <i>Vtg-5</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> için her bir Video Karesinin Psnr ve Mse Değerleri Grafiği	116
Şekil 4.162: <i>F-9</i>	117
Şekil 4.163: <i>F-19</i>	117
Şekil 4.164: <i>F-29</i>	117
Şekil 4.165: <i>Yof-9</i>	117
Şekil 4.166: <i>Yof-19</i>	117
Şekil 4.167: <i>Yof-29</i>	117
Şekil 4.168: <i>Vtg-6</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> için her bir Video Karesinin Psnr ve Mse Değerleri Grafiği	118

TABLO LİSTESİ

Sayfa No

Tablo 2.1: Çeşitli görüntüler ve kapladıkları depolama alanları	18
Tablo 3.1: Evrensel kümeyi oluşturacak oluşturulacak videodan bağımsız veri tabanı görüntülerinin özellikleri	68
Tablo 3.2: Evrensel kümeyi oluşturacak oluşturulacak videodan alınan veri tabanı görüntülerinin özellikleri	69
Tablo 3.3: Sympes Yöntemi ile Yeniden Oluşturulacak Videoların Özellikleri	69
Tablo 4.1: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	71
Tablo 4.2: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	71
Tablo 4.3: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-1</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	71
Tablo 4.4: <i>Vtg-1</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	72
Tablo 4.5: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-2</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	73
Tablo 4.6: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	73
Tablo 4.7: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-2</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	73
Tablo 4.8: <i>Vtg-1</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	74
Tablo 4.9: Sympes Yöntemi ile Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-3</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	75
Tablo 4.10: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	75

Tablo 4.11: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	75
Tablo 4.12: <i>Vtg-1</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	76
Tablo 4.13: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	77
Tablo 4.14: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	77
Tablo 4.15: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	77
Tablo 4.16: <i>Vtg-2</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	78
Tablo 4.17: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-2</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	79
Tablo 4.18: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	79
Tablo 4.19: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	80
Tablo 4.20: <i>Vtg-2</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	80
Tablo 4.21: Sympes Yöntemi ile Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-3</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri.....	81
Tablo 4.22: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	81
Tablo 4.23: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	82
Tablo 4.24: <i>Vtg-2</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	82
Tablo 4.25: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	83
Tablo 4.26: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	83

Tablo 4.27: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	84
Tablo 4.28: <i>Vtg-3</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	84
Tablo 4.29: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-2</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	85
Tablo 4.30: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	85
Tablo 4.31: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	86
Tablo 4.32: <i>Vtg-3</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	86
Tablo 4.33: Sympes Yöntemi ile Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-3</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri.....	87
Tablo 4.34: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	87
Tablo 4.35: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	88
Tablo 4.36: <i>Vtg-3</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	88
Tablo 4.37: Sympes Yöntemi ile <i>Vtg-4</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-2</i> Videosundan Alınan Video Karelerinin Özellikleri	89
Tablo 4.38: Sympes Yöntemi ile <i>Vtg-4</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan Video Karelerinin Özellikleri	89
Tablo 4.39: Sympes Yöntemi ile <i>Vtg-4</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Karelerinin Psnr ve Mse Değerleri	90
Tablo 4.40: <i>Vtg-4</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	90
Tablo 4.41: Sympes Yöntemi ile <i>Vtg-5</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-3</i> Videosundan Alınan Video Karelerinin Özellikleri	91
Tablo 4.42: Sympes Yöntemi ile <i>Vtg-5</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan Video Karelerinin Özellikleri	91

Tablo 4.43: Sympes Yöntemi ile <i>Vtg-5</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Karelerinin Psnr ve Mse Değerleri	92
Tablo 4.44: <i>Vtg-5</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	92
Tablo 4.45: Sympes Yöntemi ile <i>Vtg-6</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	93
Tablo 4.46: Sympes Yöntemi ile <i>Vtg-6</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	93
Tablo 4.47: Sympes Yöntemi ile <i>Vtg-6</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Karelerinin Psnr ve Mse Değerleri	93
Tablo 4.48: <i>Vtg-6</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	94
Tablo 4.49: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	95
Tablo 4.50: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	95
Tablo 4.51: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-1</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	96
Tablo 4.52: <i>Vtg-1</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	96
Tablo 4.53: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-2</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	97
Tablo 4.54: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	97
Tablo 4.55: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-2</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	98
Tablo 4.56: <i>Vtg-1</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	98
Tablo 4.57: Sympes Yöntemi ile Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-3</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri.....	99
Tablo 4.58: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	99

Tablo 4.59: Sympes Yöntemi ile <i>Vtg-1</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	100
Tablo 4.60: <i>Vtg-1</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	100
Tablo 4.61: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	101
Tablo 4.62: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	101
Tablo 4.63: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	102
Tablo 4.64: <i>Vtg-2</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	102
Tablo 4.65: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-2</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	103
Tablo 4.66: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	103
Tablo 4.67: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	104
Tablo 4.68: <i>Vtg-2</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	104
Tablo 4.69: Sympes Yöntemi ile Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-3</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri.....	105
Tablo 4.70: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	105
Tablo 4.71: Sympes Yöntemi ile <i>Vtg-2</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	106
Tablo 4.72: <i>Vtg-2</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	106
Tablo 4.73: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	107
Tablo 4.74: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	107

Tablo 4.75: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	108
Tablo 4.76: <i>Vtg-3</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	108
Tablo 4.77: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-2</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	109
Tablo 4.78: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	109
Tablo 4.79: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	110
Tablo 4.80: <i>Vtg-3</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	110
Tablo 4.81: Sympes Yöntemi ile Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-3</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri.....	111
Tablo 4.82: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	111
Tablo 4.83: Sympes Yöntemi ile <i>Vtg-3</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri	112
Tablo 4.84: <i>Vtg-3</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	112
Tablo 4.85: Sympes Yöntemi ile <i>Vtg-4</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-2</i> Videosundan Alınan Video Karelerinin Özellikleri	113
Tablo 4.86: Sympes Yöntemi ile <i>Vtg-4</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan Video Karelerinin Özellikleri	113
Tablo 4.87: Sympes Yöntemi ile <i>Vtg-4</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-2</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Karelerinin Psnr ve Mse Değerleri	114
Tablo 4.88: <i>Vtg-4</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-2</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	114
Tablo 4.89: Sympes Yöntemi ile <i>Vtg-5</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-3</i> Videosundan Alınan Video Karelerinin Özellikleri	115
Tablo 4.90: Sympes Yöntemi ile <i>Vtg-5</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan Video Karelerinin Özellikleri	115

Tablo 4.91: Sympes Yöntemi ile <i>Vtg-5</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-3</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Karelerinin Psnr ve Mse Değerleri	116
Tablo 4.92: <i>Vtg-5</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-3</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	116
Tablo 4.93: Sympes Yöntemi ile <i>Vtg-6</i> Evrensel Kümesiyle Yeniden oluşturulacak <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	117
Tablo 4.94: Sympes Yöntemi ile <i>Vtg-6</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri	117
Tablo 4.95: Sympes Yöntemi ile <i>Vtg-6</i> Evrensel Kümesiyle Yeniden oluşturulan <i>Vv-1</i> Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Karelerinin Psnr ve Mse Değerleri	117
Tablo 4.96: <i>Vtg-6</i> Evrensel Kümesi ile Yeniden Üretilen <i>Vv-1</i> Video Karelerinin Ortalama Psnr ve Mse Değerleri	118
Tablo 5.1: <i>Vv-1</i> video verisi için psnr, mse değerleri	119
Tablo 5.2: <i>Vv-2</i> video verisinin psnr ve mse değerleri	119
Tablo 5.3: <i>Vv-3</i> video verisinin psnr ve mse değerleri	120
Tablo 5.4: Evrensel kümeler ile Video Verisi <i>Vv-1</i> oluşma süreleri	120
Tablo 5.5: Evrensel kümeler ile Video Verisi <i>Vv-2</i> oluşma süreleri	120
Tablo 5.6: Evrensel kümeler ile Video Verisi <i>Vv-3</i> oluşma süreleri	121
Tablo 5.7: Sympes Yöntemi ile Videoların Sıkıştırma Yüzdeleri	121

SİMGE VE KISALTMA LİSTESİ

Simgeler

Açıklama

G_j	: VF_j 'nin kazanç vektörü
L_F	: Çerçeve Boyutu
N_F	: Çerçeve Sayısı
N_{IE}	: Önceden Tanımlı Bilgi Bloğun Toplam İndis Sayısı
N_{IS}	: Önceden Tanımlı Enerji Bloğun Toplam İndis Sayısı
J_t	: Toplam Kareler Hatası
VF	: Video Verisi
$VF_j^{(s)}$	: Alt Çerçeve Vektörü
ε_t	: Yaklaşık Hata
λ	: Özdeğer Matrisi
λ_i	: i.nci Özdeğer
p_{yz}	: Y ve Z örneklemelerin Pearson Özilinti Katsayısı
Φ	: Özvektör Matrisi
$(.)^T$	: Bir Matrisin Transpozu
$(.)^{-1}$	: Bir Matrisin Ters

Kısaltmalar

Açıklama

ASCII	: Bilgi Değişimi İçin Amerikan Standart Kodlama Sistemi
AVC	: Gelişmiş Video kodlama
CABAC	: Bağlam Uyarlamalı İkili Aritmetik Kodlaması
CAVLC	: Bağlam uyarlamalı değişken uzunluklu kodlama
CEPB	: Enerji ve Yapı Öbeği Bankası- Önceden Tanımlı Vektör Bankası
CTU	: Kodlama Ağacı Ünitesi
DCT	: Ayrık Kosinüs Dönüşümü
DPCM	: Diferansiyel Darbe Kodu Modülasyonu
DWT	: Ayrık Dalgacık Dönüşümü
DST	: Ayrık Sinüs Dönüşümü
FFT	: Hızlı Fourier Dönüşümü

GIF	: Grafik Deęiřtirme Biçimi
HEVC	: Yüksek Verimlilikli Video Kodlama
JBIG	: Joint Bi-level Image Experts Group
JPEG	: Joint Photographic Experts Group
LOG	: Laplacian of Gaussian
MPEG	: Moving Picture Experts Group
MSU	: Moscow State University
PNG	: Portable Network Graphics
PU	: Öngörme Ünitesi
RGB	: Red-Green-Blue
ROI	: Region Of Interest
RLE	: Tekrarlı Uzunluk Kodlaması
VCEG	: Video Coding Experts Group
VVC	: Çok Yönlü Video Kodlama
TIFF	: Tagged Image File Format
TU	: Dönüşüm Ünitesi

ÖZET

DOKTORA TEZİ

SYMPES Yöntemi ile Video Görüntülerinin Sıkıştırılması

Türker Togay AKSOY

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

Biyomedikal Mühendisliği Anabilim Dalı

Biyomedikal Mühendisliği Programı

Danışman: Prof. Dr. Bekir Sıddık Binboğa Yarman

Video, bir dizi hareketli görüntünün elektronik ortamda kaydedilmesi ve oynatılması için kullanılan bir dijital medya formatıdır. Günümüzde video iletimi, internet teknolojileri sayesinde oldukça gelişmiş bir hale gelmiştir. İnternet bağlantısı olan herkes, dünya genelindeki farklı platformlardan video içeriklerine kolayca erişebilir ve bunları izleyebilir. Video, görüntü, ses ve diğer verileri içerebilir. Bu veriler, genellikle dijital olarak sıkıştırılır ve depolanır. Video verilerinin güvenliği hem videoyu kaydedenlerin hem de videoyu izleyenlerin gizliliğini korumak için önemlidir.

Daha önce yapılan çalışmalarda görüntülerin SYMPES metodu ile modellenmesi yapılmıştır. Bu tezde ise video diye adlandırdığımız hareketli görüntülerin SYMPES metodu ile modelleme çalışması yapılmıştır. Bu metodun en önemli özelliği, SYMPES yöntemi ile yeniden oluşturmak istediğimiz videoyu, videodan bağımsız görüntüler ve videonun herhangi bir görüntü karesinden aldığımız görüntüler ile üretmektir. Bu tezde, 2048×1536 , 2400×1800 , 960×1280 , 848×480 , 1280×720 ve 1920×1080 çözünürlük özelliğine sahip 5 farklı görüntü kullanıldı. SYMPES yöntemi ile yeniden üretmek istediğimiz 1920×1080 , 848×480 ve 1280×720 çözünürlük özelliğine sahip 3 farklı video kullanılmıştır. Bu tezde Sympes yöntemini, sütun vektör yaklaşımı ve görüntü blok yaklaşımı kullanarak yeniden video oluşturduk. Video, ilk önce $VF^{(s)}$ olarak adlandırılan video karelerine bölünür. Bu durumda, SYMPES yöntemine göre, her biri eşit L_F uzunluğunda çerçevelere yada $M \times M$ sayısında karesel bloklara bölünen işaretin her bir j . çerçevesi için $GF_j \cong G_j E_{IE} S_{IS}$, ($j=1,2,...,V_F$) şeklinde ifade edilebilir. Burada, G_j çerçeve kazancını ifade eden reel bir sabit; $S_{IST}=[S_{IS1} S_{IS2}...S_{ISLF}]$ ise en küçük ortalama karesel hata (LMS) anlamında $G_j S_{IS}$ ifadesi ile çerçevenin hemen hemen bütün enerjisini içeren bir imza (veya enerji) vektörü; E_{IE} ise $L_F \times L_F$ boyutunda köşegen

matris olup GJS ifadesinin bir zarf (veya bilgi) vektörüdür. Önceden tanımlı olan bilgi ve enerji vektörleri bu çalışmada etkin bir şekilde kullanılmıştır. $L_F=8$ olarak alınan çerçeve uzunlukları ve $M \times M=8$ karesel görüntü blokları için Psnr(Peak Signal Noise Ratio), Mse(Mean Squared Error) değerleri, videoyu yeniden oluşturma süreleri ve blok bazında sıkıştırma oranları ile incelenmiştir.

Mayıs 2023 , 166 sayfa.

Anahtar kelimeler: Symes yöntemi, Video sıkıştırma, Görüntü işleme



ABSTRACT

Ph.D. THESIS

Compression Of Video Images With SYMPES Method

Türker Togay AKSOY

İstanbul University-Cerrahpaşa

Institute of Graduate Studies

Department of Biomedical Engineering

Biomedical Engineering

Supervisor : Prof. Dr. Bekir Sıddık Binboğa YARMAN

Video is a digital media format which is used for recording and playback of a series of motion pictures electronically. Today, video transmission has become quite advanced thanks to internet technologies. Everyone with an internet connection can easily access and watch video content from different platforms around the world. Video may contain images, audio, and other data. This data is usually digitally compressed and stored. The security of video data is important to protect the privacy of those who recorded the video and those who watched the video.

In the previous studies, the images were modelled with the SYMPES method. In this thesis, the modelling study of motion pictures, which we call video, was carried out with the SYMPES method. The most important feature of this method is to produce the video we want to reproduce with the SYMPES method, with images independent of the video and the images we receive from any image frame of the video. In this thesis, 5 different images with 2048x1536, 2400x1800, 960x1280, 848x480, 1280x720 and 1920x1080 resolutions were used. 3 different videos with 1920x1080, 848x480 and 1280x720 resolutions that we wanted to reproduce with the SYMPES method. In this thesis, we recreated video, using the Sympes method, column vector approach and image block approach. The video is first divided into video frames called $VF^{(s)}$. At this stage, according to the SYMPES method, it can be expressed as $GFj \cong GjE_{IE}S_{IS}, (j=1,2,...,VF)$ for each j. frame of the sign, divided into frames of equal length LF or square blocks of $M \times M$ number. Here, Gj is a real constant expressing the frame gain; $S_{IST} = [S_{IS1} S_{IS2} ... S_{ISLF}]$ is a signature (or energy) vector containing almost all the energy of the frame, with the expression GjS_{IS} in the sense of least mean square error (LMS); E_{IE} is a diagonal matrix of size $LF \times LF$ and is an envelope (or information) vector of GjS_{IS} .

Information and energy vectors, which are predefined, are used effectively in this study. Psnr (Peak Signal Noise Ratio), Mse (Mean Squared Error) values for frame lengths taken as $LF=8$ and $M \times M=8$ square image blocks were analysed with video re-render times and block-based compression ratios.

May 2023, 166 pages.

Keywords: Sympes Method, Video Compression, Image Processing,



1. GİRİŞ

Video, bir dizi hareketli görüntünün ardışık olarak oynatılması ile elde edilen bir görsel bilgidir. Video kaydedicileri, kameralar ve diğer cihazlar tarafından yakalanan hareketli görüntüler dijital olarak saklanır ve video dosyaları olarak adlandırılır. Video dosyaları, belirli bir video formatında kaydedilir ve daha sonra oynatmak veya düzenlemek için kullanılabilir.

Video kodlama, bir video dosyasındaki verileri daha küçük bir boyuta sıkıştırmak için kullanılan bir yöntemdir. Bu, videoyu daha az alan kaplamasını sağlar ve daha hızlı bir şekilde yüklenmesine yardımcı olur. Video kodlama, iki temel yöntem kullanır: kayıpsız ve kayıplı sıkıştırma diye adlandırılır. Video kodlama, belirli standartlara uygun olarak gerçekleştirilir. Bu standartlar, farklı codec'ler ve video dosya formatları arasında uyumluluk sağlamak için gerekli olan parametreleri belirler. Bu nedenle, uygun codec ve dosya formatı seçimi, video kodlama işleminde önemli bir faktördür.

Bu tezde;

- SYMPES yöntemi ile yeniden oluşturmak istediğimiz videoların evrensel kümesini oluştururken diğer SYMPES çalışmalarından farklı olarak evrensel kümeyi sadece 1 adet görüntüden oluşturduk.
- SYMPES yöntemi ile yeniden oluşturmak istediğimiz videoları için 2 farklı yaklaşım kullandık.

1-Görüntüleri Sütun Vektörlere Ayırma ($L_F=8$ uzunluğunda çerçevelere ayrılır.)

2- Görüntüleri Bloklara Ayırma ($M \times N=8 \times 8$ bloklara ayrılır.)

- Video sinyali framelerine bölünür. Her bir frame bir görüntüdür.
- Görüntü, Sütun vektörü yaklaşımı ile, kullanıcı tarafından belirlenen çerçeve boyutuna göre, görüntünün yükseklik değerine karşılık gelen satır değerleri belirli sayıda çerçevelere bölünür.

- Görüntü, Blok Görüntü Yaklaşımı ile kullanıcı tarafından belirlenen çerçeve boyutuna göre karesel matrislere bölünerek bloklara ayrılan görüntü, yine sütun vektörüne dönüştürülür.
- Her bir bloğun SYMPES matrisleri elde edilir.
- SYMPES Kodlayıcı, her bir görüntü bloğunun Envelope (Bilgi) ve Signature (İmza) vektörleri için sırasıyla evrensel kümedeki Envelope ve Signature vektörleri arasından en yüksek benzerliğe sahip olan vektörlerin indis değerlerini üretir.
- Vericiden gönderilen bilgiler, her bir blok için çerçeve kazancı ile Envelope ve Signature vektörleri için üretilen indis değerleridir.
- Alıcı tarafta yani SYMPES Kod Çözücü'de evrensel bilgi ve enerji kütüphanelerindeki, verici tarafından gönderilen indislerine karşılık gelen değerleri ve çerçeve kazancı kullanılarak yeni görüntü kareleri elde edilmesi sağlanır.
- Elde edilen görüntü karelerinden video dosyası üretilir.

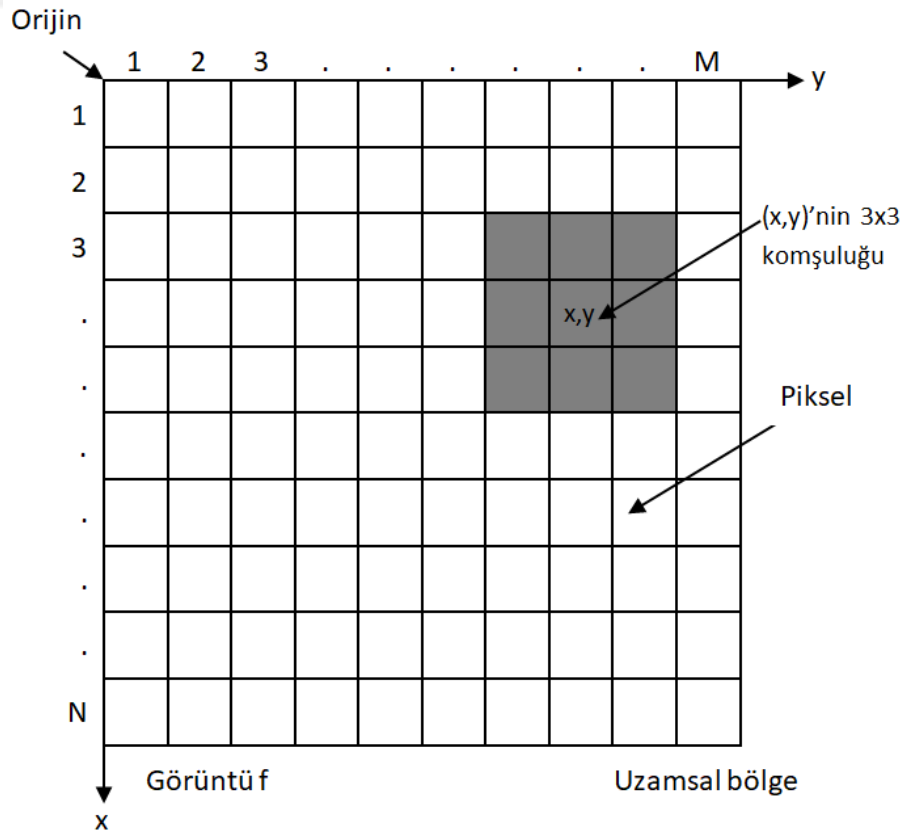
Her iki yaklaşım için Psnr (Peak Signal Noise Ratio), Mse (Mean Squared Error) değerleri, videoyu yeniden oluşturma süreleri ve blok bazında sıkıştırma oranları ile incelenmiştir.

2. KAVRAMSAL ÇERÇEVE

Görüntü; genişlik, yükseklik ve kanal sayısına sahip olan üç boyutlu nesnelerin üzerine düşen ışıklarla algılanan görünümün iki boyutlu haritasının çıkarımıdır.

2.1. GÖRÜNTÜ İŞLEME ALTYAPISI

Bir sayısal görüntü $f(x,y)$ şeklinde iki boyutlu bir fonksiyon olarak tanımlanabilir. Burada x ve y uzamsal (düzlemsel) koordinatlardır ve herhangi bir (x, y) koordinat çiftindeki f 'in genliğine görüntünün o noktadaki yoğunluğu veya gri seviyesi denir. Eğer bu tanımdaki x , y ve yoğunluk değerlerinin tamamı sonlu ve ayrık ise bu görüntüye sayısal görüntü denir. Bilgisayar biliminde sayısal görüntü işleme, bilgisayarlar ve bir algoritma kullanılarak sayısal görüntülerin işlenmesidir. Tanımda da belirtildiği gibi bir sayısal görüntü belirli bir lokasyonu ve değeri olan sonlu sayıda elemandan oluşur ve bu elemanlar piksel olarak adlandırılır [1, 2]. M sütun ve N satırdan oluşan bir sayısal görüntüde $M \times N$ adet piksel bulunur.



Şekil 2.1: $N \times M$ sayısal görüntü

Uzamsal bölge (spatial domain) terimi görüntü düzleminin kendisini ifade eder ve bu kategoride uygulanan görüntü işleme metotları piksellerin direk olarak manipülasyonu şeklinde gerçekleştirilir. Uzamsal görüntü işleme (spatial processing) yoğunluk transformasyonu (intensity transformation) ve uzamsal filtreleme (spatial filtering) olarak iki ana başlığa ayrılabilir. Yoğunluk transformasyonu tekli pikseller üzerinde işlem yaparak kontrast manipülasyonu ve eşikleme (thresholding) gibi işlemleri gerçekleştirir. Uzamsal filtreleme ise her bir pikselin belirli bir komşuluğu ile çalışarak görüntü keskinleştirme gibi işlemleri gerçekleştirir.

Morfoloji kelimesi genellikle hayvanların ve bitkilerin şekil ve yapısıyla ilgilenen biyolojinin branşını işaret eder. Matematiksel morfoloji veya basitçe morfoloji uzamsal yapıları analiz eden bir teori olarak tanımlanabilir [3]. Matematiksel morfoloji sadece bir teori olmayıp güçlü bir görüntü analiz tekniğidir. Morfoloji olarak tanımlanmasının nedeni nesneleri şekil ve biçim yönünden analiz etmesidir ve analiz yöntemi küme teorisine dayandığı için matematiksel bir yöntemdir.

Genellikle karmaşık olan görüntü işleme problemlerinin çözümünde temel morfolojik işlemler tek başına değil bir dizi işlem kombinasyonunun şeklinde uygulanır. Bu bölümde temel morfolojik işlemler ve bu işlemlerin kombinasyonu ile gerçekleştirilen sınır temizleme ve morfolojik rekonstrüksiyon gibi bazı sık kullanılan işlemler anlatılacaktır. Morfolojik işlemlere girmeden önce konunun anlaşılması için bir altyapı sağlayacak şekilde görüntü çeşitleri ve pikseller arasındaki bazı temel ilişkilerden bahsedilecektir.

2.2. GÖRÜNTÜ ÇEŞİTLERİ

2.2.1 Gri Tonlamalı Görüntü (Grayscale Image)

Gri tonlamalı görüntüleme gri renk tonlarının kullanıldığı bir görüntüleme türüdür. Dijital görüntülemede gri renk tonlarının gösterimi için sınırlı sayıda renk tonu kullanılabilir. 256 tonlu gri tonlamalı görüntü en çok kullanılan metotlardan biri olmasına karşın daha çok ve daha az tonların kullanıldığı görüntüleme metotları da vardır. Bir görüntüdeki gri renk tonları sayısının kaç bit ile ifade edildiği o görüntünün derinliğini verir. Derinliği 2 bit bir görüntüde gri renk tonlarının gösterimi için (2², 0-3) 4 farklı gri renk tonu gösterilebilir iken derinliği 8 bit bir görüntüde gri renk tonlarının gösterimi için (2⁸, 0-255) 256 farklı gri renk tonu gösterilebilir.

Gri tonlamalı görüntüler yoğunluk transformasyonu ve düzlemsel filtreleme yöntemleri ile işlenir ve ayrıca görüntünün binary'e dönüşümü yolu ile morfolojik işlemlerle de işlenebilir.

2.2.2. İkili Görüntü (Binary Image)

Bir binary görüntünün piksel değerleri, pikselin nesneye mi yoksa arka plana mı ait olup olmadığına göre 1 veya 0'dır. Genellikle siyah zemin üzerine beyaz nesne olacak şekilde kullanılabilir veya tam tersi de olabilir. Morfolojik görüntü işlemede görüntünün içindeki nesne ya da nesneler küme ya da kümeler olarak kabul edilir.

Görüntü işlemenin en temel yöntemlerinden biri eşikleme (thresholding) yöntemidir. Binary görüntüler genellikle eşikleme yöntemi kullanılarak gri tonlamalı görüntülerden elde edilirler. Arka planın siyah üzerinde beyaz nesne ya da nesneler şeklindeki görüntü gösteriminde gri ton eşiklemesi (thresholding) eşitlik 2.1.'deki uygulanır.

$$g(x,y) = \begin{cases} 1 & \text{eğer } f(x,y) > T \\ 0 & \text{eğer } f(x,y) \leq T \end{cases} \quad (2.1)$$

2.2.3. Pikseller Arası İlişkiler

$f(x, y)$ fonksiyonu olarak tanımlanan Şekil 2.2'deki görüntüde (x, y) konumundaki p pikselinin sekiz adet komşusu bulunmaktadır. Bunlar iki adet yatay, iki adet dikey ve dört adet köşegendir. P pikselinin dikey ve yatay komşuları, dört komşusu olarak tanımlanmaktadır. $N_4(P)$ şeklinde gösterilmektedir. P pikselinin 4 köşegen komşusu $N_4(p)$ şeklinde gösterilmektedir. P pikselinin iki boyutlu eksenindeki komşuları $N_8(p)$ şeklinde gösterilmektedir.

Dijital görüntülerde bitişiklik kavramı komşuluk konusuna ek olarak piksellerin yoğunluk değerlerini ilişkilendirmek içinde kullanılmaktadır. Bitişikliği tanımlarken bir V yoğunluk değeri kümesi kullanırsak ikili (binary) bir görüntüde 1 değerine sahip piksellerin bitişikliğini tanımlamak için $V = \{1\}$ olacaktır ve 4-Bitişiklik, 8-Bitişiklik kavramları aşağıdaki şekillerde gösterilmektedir.

4-Bitişiklik: Eğer bir q pikseli p 'nin 4 komşuluğu içerisinde ($N_4(P)$) ve değeri V 'den ise p ve q pikselleri 4-bitişiktir.

8-Bitişiklik: Eğer bir q pikseli p 'nin 8 komşuluğu içerisinde ($N_8(p)$) ve değeri V 'den ise p ve q pikselleri 8-bitişiktir [4].

elemanının merkezinin görüntü ile kesiştiği her bir nokta için yapılandırma elemanının kapsadığı alanların tamamıdır [6].

Şekil 2.4'te 5 birim uzunluğundaki yatay bir nesnenin 3 birim uzunluğunda dikey bir yapılandırma elemanı ile genişletilmiş hali görülmektedir. Yapılandırma elemanının merkezinin nesne ile her kesiştiği noktada kapladığı alanlar genişletilmiş görüntüde x ile işaretlenmiştir. Morfolojik genişleme nesneleri büyötmek için kullanılır ve belirli noktalardan görüntü içerisindeki nesneleri tekrar inşa eden morfolojik geri çatmanın da adımlarından biridir.

0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0		0	x	x	x	x	x	x	0	0
0	1	1	1	1	1	0		0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0		0	x	x	x	x	x	x	0	0
0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
X								B		X $\oplus$ B						
								1								
								1								
								1								

Şekil 2.4: B yapılandırma elemanı ile X'in genişleme işlemi

2.2.6. Morfolojik Geriçatma

Morfolojik geriçatma işlemi iki görüntü ve bir yapılandırma elemanını ile uygulanan morfolojik bir operasyondur. Morfolojik geriçatma işlemi görüntü içerisindeki belli noktalardan nesneleri tekrar yaparak görüntüden çıkarmayı sağlamaktadır. İşaretleyici denilen görüntü, dönüşümün başlangıç noktalarını belirleme işiyle görevliyken maske denilen görüntü ise dönüşümün sınırlarını belirlemektedir. Yapılandırma elemanı, bitişikliği tanımlamak için kullanılmaktadır [5, 7].

2.2.7. Sınır Temizleme

Uygulamalarda görüntü sınırına dokunan nesnelerin kaldırılması büyük bir problem olarak karşımıza çıkmaktadır. Bu tür problemler morfolojik geri çatma denilen yöntem kullanılarak çözülebilmektedir. Böyle bir durumda görüntü maske, görüntü sınırında bulunan nesnelerin sınıra temas ettiği yerleri gösteren görüntü ise işaretleyici olarak kullanılmaktadır. İşaretleyici, görüntü kenarlarına dokunan tüm nesneleri büyötmek için kullanılan bir kar topu niteliğinde olmaktadır. Görüntü sınırına dokunan nesneler bu işaretleyiciden genişleme ile geri çatma kullanılarak büyötmekte ve ana görüntüden çıkarılmaktadır [6].

2.2.8. Bağlı Bileşenler

Bir görüntü kümesi içerisindeki her nokta komşuluk ilişkisi ile birbirine bağlıysa bu kümeye bağlı küme denir. Binary görüntülerde bağlı bileşenlerin her birini ayrı ayrı birer nesne olarak düşündüğümüzde bağlı bileşenleri etiketlemek, nesnenin özelliklerine göre filtreleme yapabilmek için gereklidir. Bağlı bileşenleri etiketlemek için farklı yöntemler kullanılabilmektedir [6, 8].

Geri çatma işlemiyle bağlı bileşenleri çıkarma, bağlı bileşenleri etiketlemek için kullanılabilecek yöntemlerden biri sayılabilmektedir. Geri çatma işlemiyle bağlı bileşenleri çıkarma işlemine, görüntü içerisindeki ön plan piksellerinden rastgele biriyle başlanmaktadır. Pikselin bağlı bulunduğu nesne morfolojik geri çatma işlemiyle büyütülerek etiketlenir sonrasında görüntüden çıkarılmaktadır. Geri çatma işlemine görüntüden geriye kalan nesnelerden rastgele birine ait bir ön plan pikseliyle devam edilmektedir. Görüntüdeki bütün ön plan pikselleri etiketlenene kadar işlem devam etmektedir. Görüntü içerisindeki bağlı bileşenler çıkartıldıktan sonra her bağlı bileşen üzerinde nesnenin özelliğine göre filtreleme için işlem yapılabilir [9].

2.3. GÖRÜNTÜ İYİLEŞTİRME

2.3.1 Histogram

Histogram, görüntüyü oluşturan ikili kodların piksel değerlerinin yoğunluğunu gösteren ve sık kullanılan bir grafik yöntemidir. Görüntü işleme yöntemlerinde histogramlarla görüntü kalitesi artırılarak, piksel kaymaları azaltılır ve görüntünün daha belirginleştirilmesi sağlanabilir. Görüntünün netlik kazanması için histogramın daha ayrık ve düzgün bir yapıya gelmesi sağlanmalıdır. Bu işleme histogram eşitleme (histogram equalisation) denir [10]. Histogram eşitleme resimde bir yerde toplanan, düzgün dağılımlı olmayan renk seviye değerlerini dağıtarak renk bozukluğunu gideren bir tür görüntü iyileştirme tekniğidir. Bu işlem

$$S(k) = \sum_{i=0}^k \frac{n_i}{n} * (L - 1), k = 0.1.2.3 \dots, L - 1 \quad (2.3)$$

formülü ile ifade edilir.

n_i = i. seviyedeki piksel sayısı (i. tondan resimde kaç adet var?)

L: istenilen gri seviye sayısı (8 bit için 256-1=255 olur)

n = toplam piksel sayısıdır.

2.4. FİLTRELER

Görüntü işlemede görüntüler değiştirildikten sonra gelen görüntüdeki istenmeyen veri ve desenleri silme işlemine filtreleme denilir. Bu filtreleme yöntemlerini, gürültüyü azaltanlar, yokedenler, kenar ve köşeleri belirtenler olarak ayırabiliriz. Filtrelerle işlenecek görüntü üzerine çalışılarak değişik efektlerle yeni resim elde edilir. Filtreleme işlemi şu formülle elde edilebilir: Burada h fonksiyonu filtredir.

$$f'(x, y) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} h(i, j) f(x - i, y - j) \quad (2.4)$$

Filtremizi görüntüdeki piksel değerleri ile matrise denk gelenleri çarparak çarpım toplamlarını alırız ve ulaştığımız bu tek değeri filtrenin görüntüdeki merkez pikseli ile yer değiştiririz. Filtre matrisimizi, resmin ilk satır ve sütuna, son satır ve sütuna uygulamayacak şekilde hareket ettirmemiz gerekir. Şekil 2.5’’te matris üzerinde filtre boyutu gösterilmiştir.



Şekil 2.5: Filtre boyutu [11]

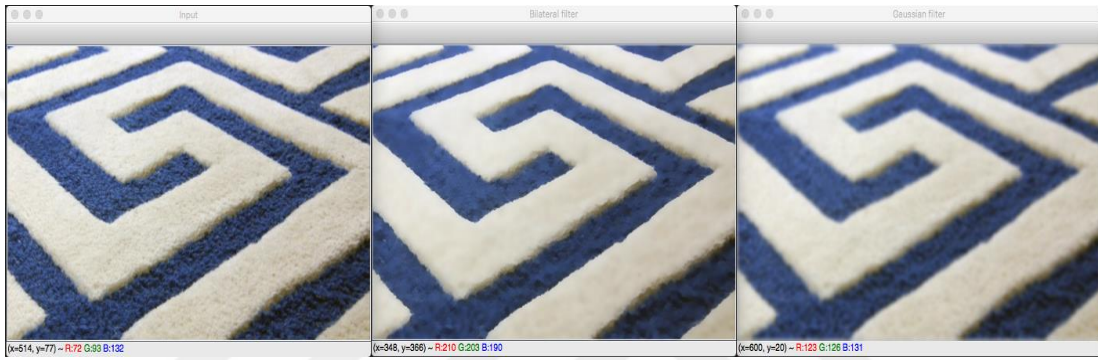
Yapılan işlemlerde filtredeki katsayılar ve filtrenin boyutu yapılan önemli bir yere sahiptir. Genelde filtrelerin boyutu 3×3 matris şeklinde alınır. Yapılan işlemde filtre boyutu ne kadar büyükse aslında hesaplanması gereken değerde artar ve detaylar kaybolur. Bu yüzden görüntü işlemede büyük boyutlu filtrelerle çalışılmak genelde tercih edilmez.

Filtrelerin katsayıları toplamı bizlere görüntü hakkında ipuçları vermektedir. Bulunan katsayılar 1 ise görüntünün parlaklığını dengede tuttuğunu, 0 ise görüntünün parlaklığının azaldığı sonucuna varırız. Katsayılar + ve - değerler bulunduruyorsa (örneğin -5,+5) kenar bilgilerini kuvvetlendirir. Eğer katsayı değerleri + değerlerden (örneğin 3,8,16...gibi) meydana gelmişse görüntü bulanıklık meydana getirir [10].

Literatürde gürültüyü azaltmak ya da yok etmek için gauss, average (mean), medyan, disk, log transform, fast fourier, motion, unsharp gibi birçok filtreler kullanılır.

Gauss: Gaussian filtresi aslında Ortalama filtrenin Gaussian dağılımını kullanarak biraz daha değiştirilmiş halidir. Gaussian filtrelemeye aynı zamanda bir fourier dönüşümüdür diyebiliriz. Gauss filtre ile sonsuz bir transfer fonksiyonuna karşılık, mekansal alanda, tarama penceresinde filtreleme yapılabilmektedir. Bu da filtrelemenin temel problemini daha kolay çözülmesini mümkün kılar.

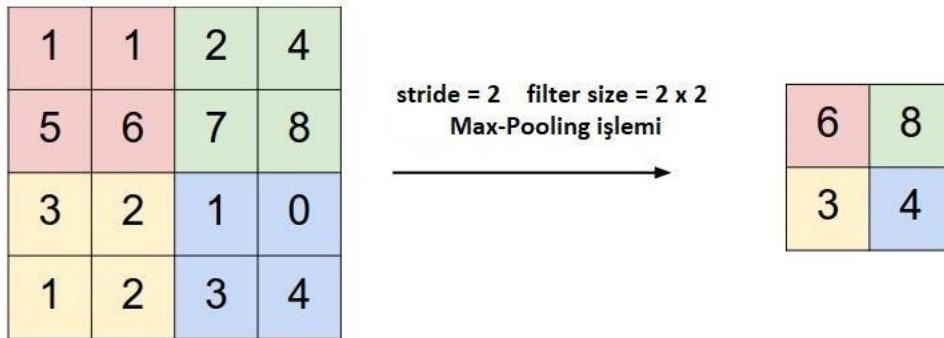
Gauss filtrenin sistemde uygulama çıktı görüntüsü Şekil 2.6’da verilmektedir.



Şekil 2.6: Gauss filtre çıktısı [16]

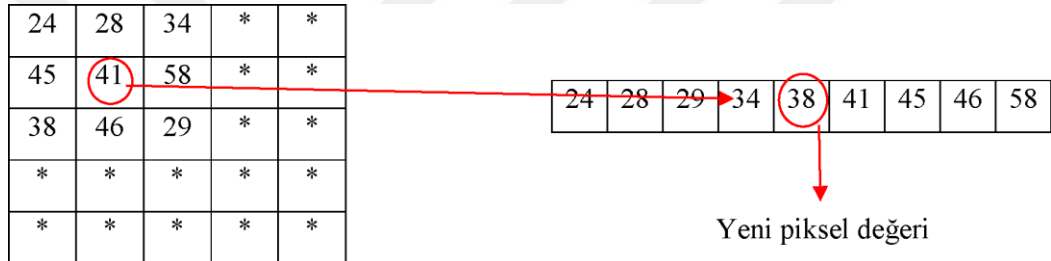
Average: Average Filtre ortalama filtre olarak adlandırılan filtredir. Bu filtre alçak geçirgen filtrelerden biridir. Görüntü üzerinde piksellerin gri ton değerlerinin değişmediği bir bölgeye filtre yerleştirildiğinde merkezdeki gri ton değeri aynı kalır. Aynı filtre, gri ton değerlerinde büyük değişiklikler olan bir bölgeye yerleştirildiğinde, merkez pikseline etrafındaki piksellerin gri ton değerlerinin ortalaması atanacağından merkez ile çevre pikseller arasındaki fark azalmış olacaktır. Bu yüzden bu tür filtreler görüntüdeki gürültüleri ortadan kaldırırlar fakat aynı zamanda görüntüyü bulanıklaştırırlar [12].

Average filtrenin sistemde uygulama çıktı görüntüsü Şekil 2.7’de verilmektedir.

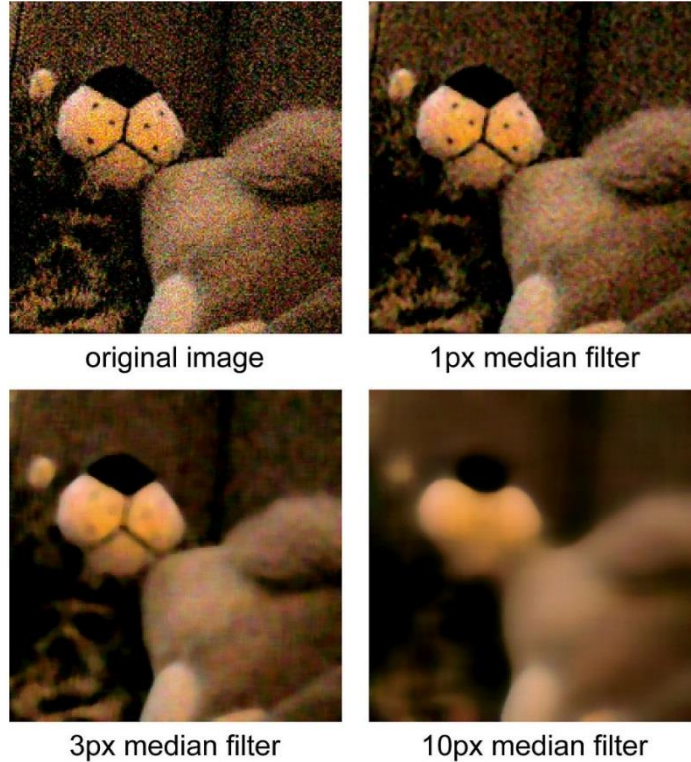


Şekil 2.7: Average filtre çıktısı [12]

Medyan: Medyan filtresi seçimdeki piksellerin parlaklığını harmanlayarak katmandaki gürültüleri azaltan filtredir. Filtrede maske matris benzer parlaklıkta pikseller arar, bitişik piksellerden çok fazla farklılık gösteren pikselleri bulur ve merkez pikselin yerine aranan piksellerin ortalama değerini yerleştirir. Bu filtre taranan görüntüdeki hareket görünümünü veya matrisin taradığı görüntüde oluşabilecek istenmeyen desenleri ortadan kaldırmak veya azaltmak için başarılı bir filtredir. Özellikle tuz biber gürültüsünü ortadan kaldırmada çok başarılı olduğu literatürlerde ispatlanmıştır. Görüntü ve sinyal işleme konularında, gürültüyü ortadan kaldırmak amacıyla sıkça kullanılan yöntemlerden birisidir. Filtrenin hedefi belirli bir pencere aralığındaki sayıların ortancasını alarak aşırı büyük atlamaları kaldırmak ve fark aralığını düşürmektir. Yani filtre uygulandıktan sonra resimde belirgin şekilde diğerlerinden ayrılan pikseller tespit edilerek temizlenmiş olur.



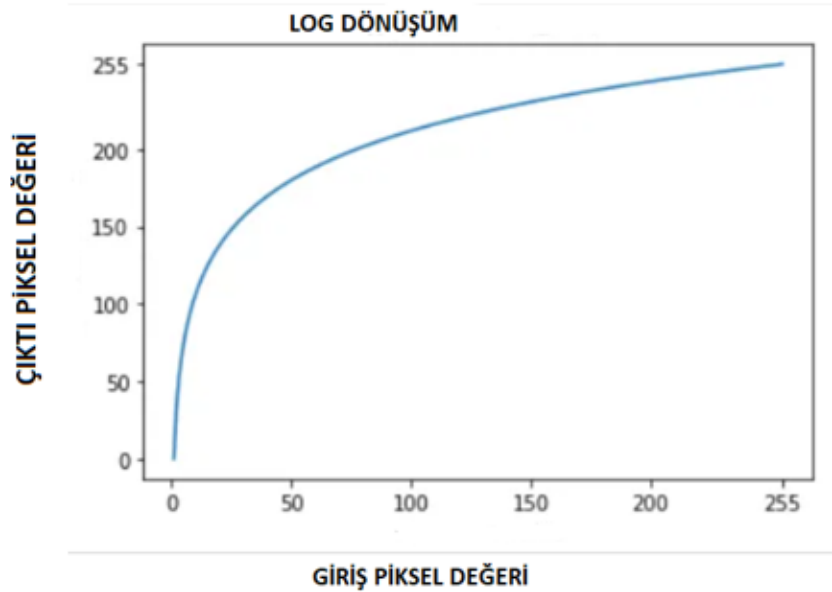
Medyan filtrenin sistemde uygulama çıktı görüntüsü Şekil 2.8’de verilmektedir.



Şekil 2.8: Medyan filtre çıktısı [17]

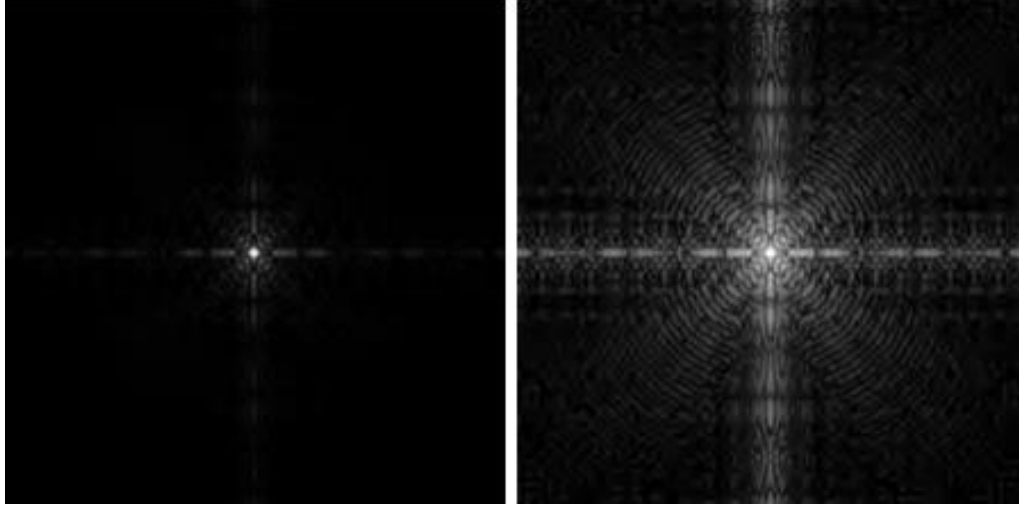
Disk: Bu filtre dairesel ortalama alma filtresi (pillbox) olarak da söylenebilir. Disk filtre uygulaması dairesel bir şeklin ortalaması hesaplanarak yapılır. Her pikselin değeri, bu pikselin çevresindeki bir daire içinde kalan tüm piksel değerlerinin ortalaması ile değiştirilir.

Log Transform: Logaritmik dönüşüm her piksel değerini logaritmasıyla değiştirmek anlamına gelir. İki tür dönüşüm içerir. Log dönüşümü ve ters log dönüşümü. Log dönüşümünün formülü $s = T(r) = c \cdot \log(1+r)$ dir. Burada s ve r, çıktının ve girdi görüntüsünün piksel değerleridir ve c sabit bir ölçüm değeridir. $c = 255/(\log(1 + \max_input_pixel_value))$ Burada C'nin değeri, kullanılan bit boyutuna karşılık gelen maksimum çıktı değerini alacak şekilde seçilir. Örneğin, 8 bitlik görüntü için, maksimum değeri 255'e eşit olacak şekilde c seçilir. Girdi görüntüsünün piksel değerlerinin her birine 1 değeri eklenir çünkü görüntüde 0 piksel yoğunluğu varsa, $\log(0)$ sonsuz demektir. Yani minimum değeri en az 1 yapmak için 1 değeri eklenir. Log dönüşümü sırasında, bir görüntüdeki koyu pikseller, kendi değerlerinden daha yüksek piksel değerlerine kıyasla genişletilir. Daha yüksek piksel değerleri, log dönüşümünde aslında bir şekilde sıkıştırılır. Log dönüşümündeki c değeri, aradığınız geliştirme türünü işleme göre ayarlar.



Şekil 2.9: 8 bitlik bir görüntü için, Log dönüşümü [18]

Bu yöntem sonucunda sistemde elde edilen görüntü çıktısı Şekil 2.10'da verilmiştir.

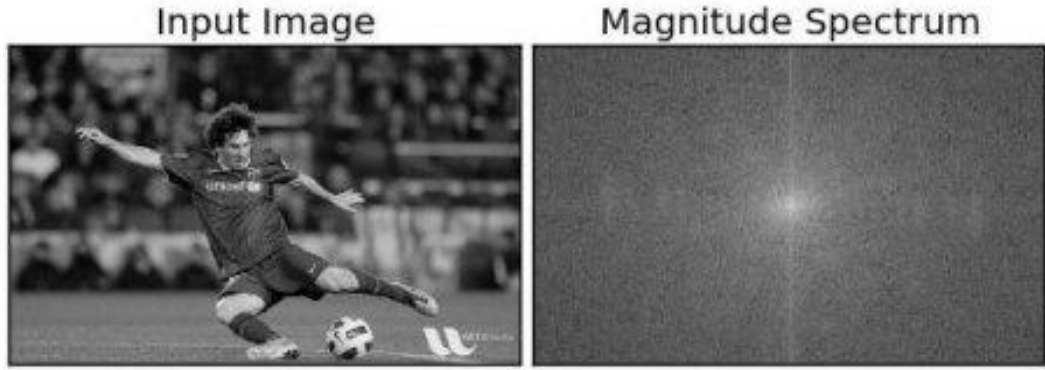


Şekil 2.10: Log Transform filtre çıktısı [19]

Fast Fourier Dönüşümü (FFT): Görüntü işlemede, piksel konumunu temsil etmenin en bilinen yolu görsel etki alanında sütun (x), satır (y) ve z (değer) 'değerleridir. Fakat bazen görüntü işleme kalıpları, uzamsal alanda yavaş hatta verimsiz bir hal alabilir ve sıkıştırma avantajları sunan farklı bir alana dönüşüm gerektirebilir. Aslında yaygın bir dönüşüm, uzaysaldan frekans alanına doğru bir yapıdır. Frekans alanı, gürültüleri gidermek, bir görüntüyü keskinleştirmek, tekrar eden desenleri analiz etmek veya özellikleri çıkarmak için, literatürde kullanılan birçok görüntü filtresinin özünü oluşturur. Frekans alanında piksel konumu, x ve y frekansları ile temsil edilir ve değeri, genlik ile ifade edilir [13].

Fourier Dönüşümü, bir sinyalin alanını zamandan frekansa doğru değiştiren matematiksel bir yöntemdir. Özellikle, çok sayıda saf frekanstan oluşan bir sinyalin ayrıştırılması için faydalı bir metottur. Hızlı Fourier Dönüşümü (FFT), bir görüntüyü uzamsal ve frekans alanı arasında dönüştürmek için yaygın olarak kullanılan ve bilinen bir yöntemdir. Matlab uygulama ortamında Fourier dönüşümü için `fft()` fonksiyonu, ters Fourier dönüşümü içinse `ifft()` komutu kullanılmaktadır. Fourier dönüşümü temel olarak N tane kompleks $x(n)$ veri serisini N tane $X(k)$ serisine aşağıdaki eşitliği kullanarak çevirmektedir.

Şekil 2.11'de araç altı görüntüsüne uygulanan Fast Fourier Dönüşümü etkisi görülmektedir.



Şekil 2.11: Fast Fourier dönüşümü [20]

2.5. KENAR BELİRLEME YÖNTEMLERİ

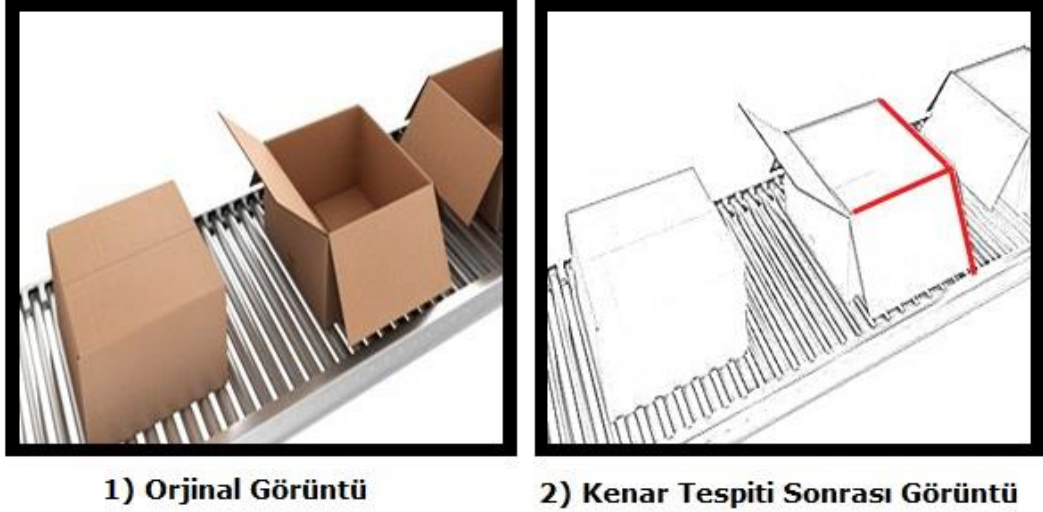
Görüntü üzerinde kenar tespiti yapmak; o görüntüdeki nesneleri tespit etmek, nesneleri saymak ve görüntünün özelliklerini belirlemek amacıyla kullanılan 2 boyutlu doğrusal filtrelerdir. Kenar belirleme algoritmaları, görüntü üzerindeki piksellerin renk değerlerinin birbirlerinden ayrıştırılmasıyla işlem yapar. Kenar belirleme algoritmalarına Prewitt, sobel, canny, log ve laplacian gibi birçok filtre örnek olarak verilebilir.

Sobel: Sobel kenar algılama algoritması verilen herhangi bir görüntüdeki kenarları elde etmeye yarayan, literatürde en çok kullanılan en temel filtrelerden biridir. Bu filtre kullanılarak görüntü içerisinde isteğe yönelik nesnelerin algılanması, belirlenmesi sağlanabilir. Sobel algoritmasında iki adet konvolüsyon yani evrişim çekirdek şablonu kullanılır. Bunlardan birisi yatayda tarayarak kenarları bulurken diğeri dikeyde tarayarak kenarları bulmaya yarar. Bu şablonlar görüntü içerisinde ışık yoğunluğunun fazla olduğu yerleri belirlememizi sağlar. Doğru bir metodla, 1-D analize dayanarak 2-D bir görüntünün türevi hesaplanabilir. Sobel işlemi uygulanırken bir görüntüde 2-boyutlu özel bir eğim ölçümü gerektirir. Bu işlem taranan görüntüde her bir noktada eğim genliğini bulmak için kullanılır. Sobel kenar belirleyici; biri x-yönündeki eğimi, diğeri y-yönündeki eğimi hesaplamak için 3x3'lük bir konvolüsyon maske çifti kullanır. Konvolüsyon maskesi hesaplamalar için görüntü boyutundan daha küçük seçilir. Maske mantık olarak bir kerede bir pikseli işleyerek görüntüde hareket eder ve görüntüyü tarar böylece x ve y yönünde filtre yapılır.

Görüntünün üst köşesine yerleştirilen maske, merkezine denk gelen pikselin değerini değiştirerek bir adım atlar ve sonraki piksele geçer. Aşağıdaki örnek görüntüde, maskenin merkezi değiştirilecek pikselin etrafına gelecek şekilde yerleştirilir. Satır, sütun değerlerini ifade eden (i, j) değerlerinin değiştirilmesi ile maske hareket ettirilir. 3x3'lük maske ile ilk ve son satır ve ilk ve son sütundaki piksellerin işlenemeyeceği şekilden görülmektedir. Eğer

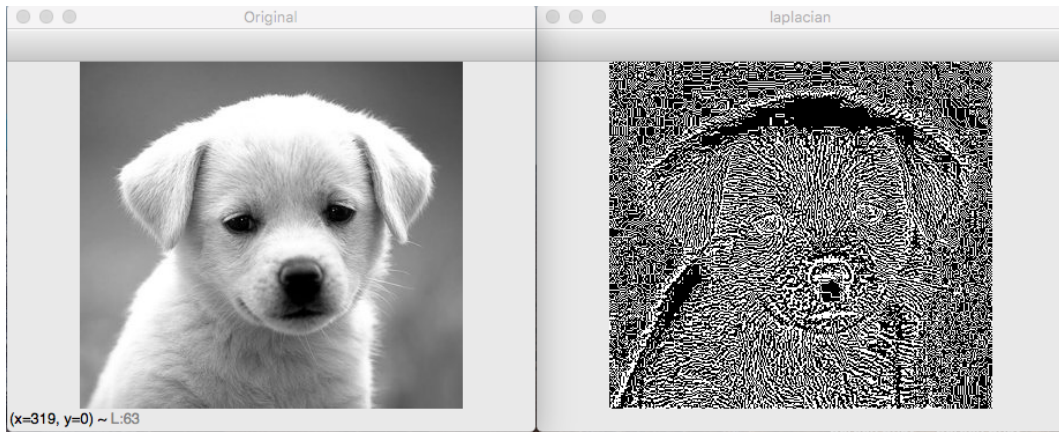
maske ilk satırdaki piksele yerleştirilseydi, tabii olarak bu görüntü sınırlarının dışına taşacaktı. Bu durumu çözmek için görüntüye birer satır ve sütun eklenebilir.

Gx maskesi yatay yöndeki kenarları, Gy maskesi dikey yöndeki kenarları vurgular. Bunların toplanması ile her iki yöndeki kenarlar tespit edilir.



Şekil 2.12: Sobel kenar belirleme filtre çıktısı [21]

Laplacian: Görüntü işlemede sınır belirleme ve netleştirme amacıyla farklı yönlerdeki kenarlara aynı tepkiyi veren Laplace yöntemi kullanılmaktadır. Bu yöntem ikinci türev operatörü olan bir metottur [14].



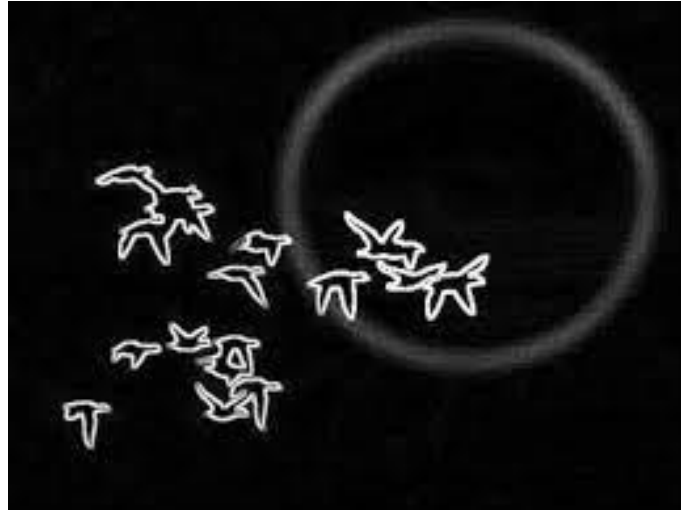
Şekil 2.13: Laplace kenar belirleme filtre çıktısı [22]

Prewitt: Prewitt birinci derece türevlerine eşit olan Gx ve Gy maskelerini kullanarak görüntüyü filtreleyen bir kenar belirleme yöntemidir. Diğer filtrelere göre görüntü üzerinde filtreleme işlemini daha kısa sürede tamamlayan bir filtredir. Ama sobele göre daha gürültülü sonuçlar verebilir [15].



Şekil 2.14: Prewitt kenar belirleme filtre çıktısı [23]

Log: Laplacian of Gauss olarak bilinen bu filtre Laplace operatörünün ikinci türevine göre çalışır ve ayrıca yönden bağımsız bir filtredir. Laplace işlecine dayalı yöntemler gürültüler için hassas olduğu için kenar belirleme işlemi gerçekleşmeden önce filtre ile temizlenmesi iyi sonuçlar alınmasını sağlar. Bu amaçla bu noktada Gauss filtresi kullanılırsa Log (Laplacian of Gauss) kenar belirleme operatörü elde edilmiş olur. Bu yöntem kullanılarak, Gauss filtresi $g(x,y)$ ile görüntü $f(x,y)$ katlanarak yumuşatılmış olur. Böylece gürültülü noktalar ve istenmeyen desenler izole edilmiş olur ama yumuşatma ile kenarlarda yayılma gözlemlenir. Görüntüyü iyileştirmek için gauss filtresi, ikinci türev elde etme bölümünde de kullanılır. Böylece gürültülü noktalar ve desenler izole edilir. Log filtresinin uygulama çıktı görüntüsü şekilde verilmektedir.



Şekil 2.15: Log kenar belirleme filtre çıktısı [24]

Canny: Canny kenar belirleme yöntemi çok adımlı bir olan bir işlemdir. İlk önce görüntü bir Gauss filtre ile yumuşatılır daha sonra yumuşatılmış görüntüye bölgelerini belirlemek üzere iki boyutlu basit bir türev operatörü uygulanır. Böylece eğim yoğunluklu görüntüde kenarlar da belirginleştirilmiş olur. Ardından tepe noktalar takip edilerek tepe olmayan bütün noktalar 0 yapılır. Böylece çıkış görüntüde ince bir çizgi verilmiş olur.



Şekil 2.16: Canny kenar belirleme filtre çıktısı [25]

2.6. GÖRÜNTÜ SIKIŞTIRMA

2.6.1. Görüntü Sıkıştırma Yöntemleri

Sıkıştırılmamış grafikler, ses ve görüntü verileri oldukça büyük depolama kapasitesi ve bilgi gönderme band genişliğine gerek duyar. Bilgi depolama alanında hızlı gelişmeler olmasına rağmen, sayısal haberleşmede gerekli olan depolama kapasitesi ve veri transfer band aralığı gereksinimi var olan teknolojinin kapasitelerini zorlamaktadır. Günümüzde kullanılan etkileşimli sayısal ses, sayısal görüntü ve video (çoklu ortam) tabanlı web uygulamaları, sinyal ve görüntüleri kodlamada daha yoğun ihtiyaçları karşılamak için bu sinyal ve görüntülerin sıkıştırılmasından yararlanmıştır [4,5].

Sayısal görüntü, sıkıştırılmamış ham haliyle oldukça büyük depolama alanlarına ihtiyaç duyar. Tablo 2.1’de çoklu ortam uygulamalarında kullanılan çeşitli görüntü tipleri ve bu görüntüleri depolamak için gerekli saklama alanları miktarları verilmiştir.

Tablo 2.1: Çeşitli görüntüler ve kapladıkları depolama alanları

Çoklu Ortam Verisi	Boyut	Çözünürlük (bit/piksel)	Ham Büyüklüğü
Bir sayfa yazı	11"x8,5"	Değişen çözünürlük	16-32 Kbit
Gri seviyeli görüntü	512x512 piksel	8 bpp	2,1 Mbit
Renkli görüntü	512x512 piksel	24 bpp	6,3 Mbit
Medikal görüntü	2048x1680 piksel	12 bpp	41,3 Mbit
Tam hareketli video	640x480 piksel, 10 sn	24 bpp	2,21 Gbit

Tablo 2.1'deki örneklere bakıldığında ham görüntünün depolanması için ne kadar büyük saklama ihtiyacı olduğu açıkça görülür. Bu yüzden, böyle çoklu ortam verileri depolanmadan veya gönderilmeden önce sıkıştırılır. Sıkıştırılan görüntü gönderildikten veya bir depolama cihazına saklandıktan sonra, tekrar açılıp oynatılır. Sabit görüntüleri sıkıştırma işlemine görüntü sıkıştırma, sıkıştırılan görüntülerin kullanılmak için tekrar açılmasına da yeniden oluşturma denir.

Görüntü sıkıştırma işlemi, görüntü bilgilerinin korunarak, sayısal görüntünün ham büyüklüğünden çok daha az miktarda depolama alanı kullanacak şekilde görüntüyü temsil etmektir. Temel kural görüntüyü oluşturan verilerin gereksiz tekrarlarından kaçınmaktır [5].

Görüntü sıkıştırma işlemi genellikle iki basamaktan oluşur. Birinci basamakta görüntü bir mesaj serisine dönüştürülür. İkinci basamakta bu mesaja uygun kodlar atanır ve sıkıştırma gerçekleştirilir. Birinci nesil görüntü sıkıştırma teknikleri ikinci basamak üzerine yoğunlaşırken, ikinci nesil yöntemler birinci basamakta yoğunlaşıp, ikinci basamağa anlamlı mesajlar yaratma stratejisine odaklanır. Görüntü bölütleme yaklaşımlı görüntü sıkıştırma, kodlama basamağına geçmeden önce görüntüyü anlamlı bölgelere bölerek uygun kodlar atama öncesinde bir ön işlem olup ikinci nesil bir görüntü sıkıştırma yöntemidir [1,15].

Birçok görüntünün bazı ortak özellikleri vardır. Örneğin komşu pikseller birbiriyle yüksek derecede ilgilidir ve bu yüzden fazla miktarda gereksiz tekrar içerir. Sıkıştırma yapabilmek için öncelikle görüntü bilgisini içeren pikseller birbiriyle bağımsız hale getirilmelidir. Sıkıştırma işleminde genellikle görüntünün iki temel özelliği üzerinde durulur.

Bunlar gereksiz bilgi tekrarı (redundancy) ve alakasızlıktır (irrelevancy). İşte bu ikisi ortadan kaldırılmaya çalışılır. Genel olarak sayısal görüntü ve videoda üç çeşit gereksiz tekrar tanımlanır [5]:

- Uzaysal gereksiz tekrar (spatial redundancy): Komşu piksel değerleri arasındaki gereksiz tekrarlardır.
- Spektral gereksiz tekrar (spectral redundancy): Farklı renk planları veya spektral bantlar arasındaki ilişkidir
- Zamansal gereksiz tekrar (temporal redundancy) (video): Görüntüler serisindeki bitişik katmanlar arasındaki ilişkidir

Görüntü sıkıştırma araştırmaları görüntüyü mümkün olduğu kadar uzaysal ve spektral gereksiz tekrardan arındırıp ilgili görüntüyü temsil etmek için gerekli bit sayısını azaltmayı amaç edinir.

Sıkıştırma tekniklerini sınıflandırmada birçok farklı yöntem izlenebilir. Genel olarak asıl görüntü ile sıkıştırma sonrası yeniden oluşturulan görüntü arasındaki ilişkilere göre bakıldığında, sıkıştırma çeşitlerini iki sınıfa ayırmak mümkündür [4]. Bunlar:

- Kayıplı sıkıştırma (lossy compression) ve
- Kayıpsız sıkıştırma (lossless compression).

Kayıpsız sıkıştırma sıkıştırma sonrası yeniden oluşturulan görüntü, asıl görüntü ile piksel-piksel karşılaştırıldığında sayısal olarak fark olmaz. Yani oluşturulan görüntüyle asıl görüntü birbirinin aynısıdır, hiçbir bilgi kaybı yoktur. Kayıplı sıkıştırma ise yeniden oluşturulan görüntü ile asıl görüntü arasında fark vardır, yani bilgi kaybı olmuştur. Bu bilgi kaybı genellikle çok önemli olmayan bilgilerin yok edilmesinden oluşmaktadır. Kayıplı sıkıştırma sayısal olarak görüntüler arasında hata olmasına rağmen, görsel kalite açısından kabul edilebilir oranda bir kayıp oluşur. Böyle bir hata tercih edilmesinin sebebi kayıplı sıkıştırmanın daha yüksek sıkıştırma oranlarıyla görüntü sıkıştırmayı başarabilmesidir.

Literatürde en çok kullanılan kayıpsız sıkıştırma biçimleri Huffman kodlama, koşu-uzunluğu kodlama (run-length coding), aritmetik kodlama, Lepmel-Ziv (LZ) ve LZW kodlamalarıdır [7]. Bu kodlama teknikleri sıkıştırma işlemini verinin kullanım sıklığına göre her bilgiye değişken uzunluklarda kodlar atayarak gerçekleştirirler. Böylece, sadece görüntü içindeki gereksiz tekrarları daha kısa kodlarla göstermiş ve gereksiz tekrarları ortadan kaldırmış olurlar. Dolayısıyla bilgi kaybı oluşmaz. Kayıpsız kodlamanın bir dezavantajı

monokrom manzara görüntülerinde genellikle en fazla (3:1) gibi bir sıkıştırma oranı sağlamasıdır ki bu sıkıştırma oram çok büyük değildir.

Kayıplı kodlamada ise, kayıpsız kodlamanın tersine, sıkıştırma oranının büyüklüğü birinci planda tutulur. Bu kodlamaların blok gösteriminde genellikle bir kuantalayıcı vardır ve kaybın en büyük sebebi kuantalayıcıdan kaynaklanmaktadır çünkü kuantalayıcı bazı değerlere yaklaşım yapmaktadır. Kayıplı sıkıştırma biçimleri öngörücülü (predictive) kodlama ve dönüşüm kodlama (transform coding) olmak üzere ikiye ayrılabilir. Sıkıştırma değerlendirmesi veri gereksizliği üzerine değil de boşluk (space) tabanına göre yapılır. Öngörücülü kodlamada daha önce gönderilen veya güncel olarak kullanılan bilgi gelecekteki değerleri tahmin etmek için kullanılır ve ikisi arasındaki fark değerleri kodlanır. Bu yaklaşım, görüntü veya uzaysal tanım kümesinde yapıldığı için ifade edilmesi kolay olup yerel görüntü özelliklerine uyarlanmaya hazırdır. DP CM (differential pulse code modulation) olarak da bilinen diferansiyel tetik kodu modülasyonu öngörücülü kodlamaya bir örnektir. Diğer taraftan, dönüşüm kodlama ise öncelikle görüntüye bir dönüşüm tekniği uygulayarak uzaysal gösterimine eşit bir başka gösterime dönüştürüp daha sonra dönüştürülmüş değerleri (genellikle dönüşüm katsayıları) kodlar. Ayrık kosinüs dönüşümü DCT, Hızlı Fourier dönüşümü FFT, Walsh-Hadamard dönüşümü, Karhunen-Loeve dönüşümü gibi ortogonal dönüşümler görüntü enerjisini birkaç katsayıyla temsil eder. Bu katsayıların kodlaması büyük oranda sıkıştırma sağlayacağı için daha fazla hesap gereksinimine rağmen, öngörücülü kodlamayla karşılaştırıldığında dönüşüm kodlama çok büyük avantajlar sunmaktadır. Web uygulamalarında sıklıkla kullanılan JPEG görüntü biçimi DCT dönüşümü tabanlı sıkıştırılmış bir görüntü biçimi olup, transform kodlamaya bir örnek olarak verilebilir [4].

2.6.2. Kayıpsız Görüntü Sıkıştırma Yöntemleri

2.6.2.1. RLE

Bütün veri tipleri için kullanılabilecek bir kodlama olmasına rağmen, aynı sembolün ardışık olarak çok defa tekrar etmesi durumunda iyi bir sıkıştırma performansı elde ettiği için, genellikle görüntü sıkıştırmada kullanılır. RLE, bir değerın ardışık olarak çok sayıda tekrar etmesi durumunda, o değeri bir defa kodlayıp, daha sonra tekrar sayısını kodlama mantığına dayanmaktadır [74]. Örneğin 1-bit renk derinliğine sahip, yani sadece siyah ve beyaz renkte olan bir görüntünün bir satırındaki 1000 pikselin renkleri sırayla; önce 350 adet beyaz, ardından 250 adet siyah ve daha sonra yine 400 adet beyaz düzeninde ise, o satır RLE ile (350, 250, 400) şeklinde kodlanabilir. Burada kod çözücünün ilk pikselin ne renk olduğunu

bilmesi gerekmektedir. Ya ilk pikselin rengi kodlamanın başında belirtilir, ya da ilk pikselin her zaman beyaz olduğu kabul edilir. Bu durumda ilk 350 piksel siyah olsaydı önce 0 ardından 350 kodlanacaktı.

2.6.2.2. Huffman Kodlama

Bilgisayar bilimlerinde veri sıkıştırmak için kullanılan bir kodlama yöntemidir. Kayıpsız olarak veriyi sıkıştırıp tekrar açmak için kullanılır. Huffman kodlamasının en büyük avantajlarından birisi kullanılan karakterlerin frekanslarına göre bir kodlama yapması ve bu sayede sık kullanılan karakterlerin daha az, nadir kullanılan karakterlerin ise daha fazla yer kaplamasını sağlamasıdır. Şayet bütün karakterlerin dağılımı eşitse yani aynı oranda tekrarlanıyorsa, bu durumda Huffman kodlaması aslında blok sıkıştırma algoritması (örneğin ASCII kodlama) ile aynı başarıya sahiptir [75]. Ancak bu teorik durumun gerçekleşmesi neredeyse imkânsız olduğu için her zaman daha başarılı sonuçlar verir.

2.6.2.3. Aritmetik Kodlama

Aritmetik kodlamanın temeli, kodlanacak veri parçasını 0 ile 1 arasında gerçek sayı aralığı ile temsil etmeye dayanır [76]. Veri kümesindeki her sembol bu sayı aralığını daraltır. Veri içerisindeki sembolün kullanım sıklığı az olanlar bu aralığı hızlı şekilde daraltırken fazla olanlar aralığı daha az daraltır. Kullanım sıklığı fazla olan semboller aralığı daha az daralttığından onu temsil eden bit sayısı, Huffman algoritmasında da olduğu gibi, azdır. Veride tekrar eden semboller çoksa veri az sayıda bit ile kodlanabilir. Alfabenin küçük olduğu ve karakterlerin tekrar etme sıklığında büyük farklar olduğu durumlarda, Huffman Kodlaması etkinliğini yitirirken, Aritmetik kodlama bu durumlarda daha başarılıdır.

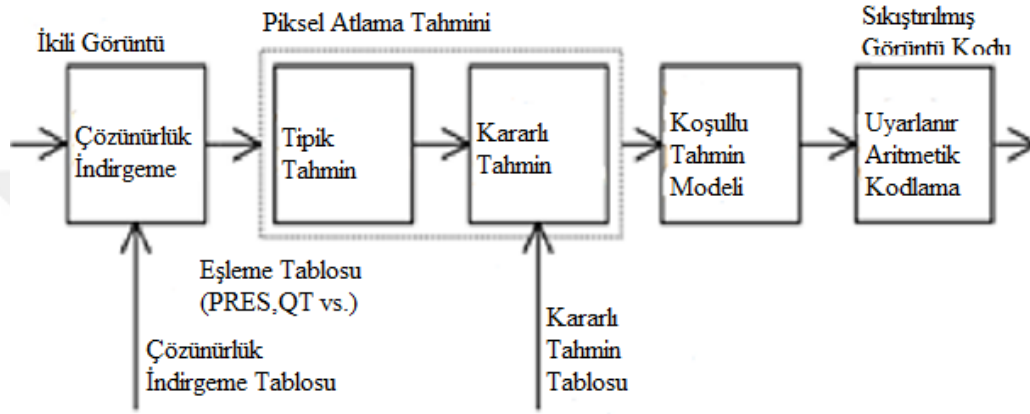
2.6.2.4. JBIG Standardı

JBIG, iki seviyeli (iki renkli) görüntü verilerini sıkıştırmak için kullanılan bir yöntemdir. Kısaltma JBIG, Uluslararası Standartlar Organizasyonu (ISO) içinde yer alan bir standart komitesi olan İki Boyutlu Görüntü Uzmanları Topluluğu ortak grubunun temsilcisidir. Geliştirdikleri sıkıştırma standardı bu komitenin ismini taşımaktadır.

1988'de ISO ve CCITT, ISO / IEC JTC1 / SC29 / WG9 grubuna ve CCITT SGVIII alt grubuna katılarak JBIG'yi kurdu ve iki seviyeli verilerin sıkıştırılmasında standart ve kayıpsız bir yöntem geliştirdi. 1993'te iki seviyeli veri kodlamasının JBIG yöntemini tanımlayan standart tamamlandı. 1993'ten sonra faks iletiminde yaygın olarak kullanılmıştır.

JBIG, aritmetik kodlama tabanlı bir tekniktir.

Aşamalı bir JBIG ikili görüntü sıkıştırma kodlayıcısının bir blok diyagramı Şekil 2.17’de verilmektedir. Algoritma, orijinal görüntüyü, çözünürlük olarak indirgenmiş görüntülere dönüştürerek başlar. Her bir görüntü, en düşük çözünürlüklü görüntü ile başlayarak kodlanır. Anlaşılabacağı üzere, piksellerin tipik ve kararlı tahminleri mümkün olduğunda kullanılır, böylece bu tür pikseller aritmetik kodlama aşamasında atlanabilir. Bu şekilde arzu edilen sıkıştırma hızı ile daha iyi sıkıştırma elde edilir. Boşlanmamış pikseller aritmetik olarak kodlanır.



Şekil 2.17: JBIG sıkıştırma kodlamasının blok diyagramı

İkili görüntülerin hiyerarşik çözünürlüğü azaltma (RR), tek bir görüntüyü kademeli olarak küçük resimlerden oluşan bir piramit haline dönüştürür. JBIG standardına uygun olarak, çözünürlüklerin simetrik olarak iki faktörle azaltıldığını varsayabiliriz. Çözünürlük azaltma piramidinin her katmanı, önceki yüksek çözünürlüklü katmandan ve muhtemelen şu anki katmandan gelen bilgiler kullanılarak oluşturulmuştur. Bir sonraki düşük çözünürlüklü katmandaki her piksel, yüksek çözünürlüklü katmandaki dört pikselin yerini alır. Genellikle, çözünürlük azaltma piramidin her ardışık katmanını oluşturmak için aynı yöntem kullanılır.

JBIG’de önerilen PRES çözünürlüğü azaltma yöntemi QT yönteminden daha karmaşıktır. Amacı, yüksek kaliteli, yani görsel çekici, daha düşük çözünürlüklü görüntüler üretmektir. Çözünürlük azaltılmış piksel değerini belirlemek için PRES yöntemi, aynı katmandan üç piksel ve bir önceki yüksek çözünürlük katmanından dokuz piksel kullanır. Bunun aksine, dörtlü küçültme yöntemi yalnızca dört yüksek çözünürlüklü piksel kullanır.

Piksel tahmini, aynı görüntünün birden çok gösterimi olan hiyerarşik çözünürlük azaltma katmanlarının yüksek düzeyde ilişkili ve bazen de gereksiz bilgi içerdiğini ortaya çıkarmaktadır. Bu tahmin, atlanarak âtil şekilde kodlanmış piksel sayısını azaltarak sıkıştırma hızını arttırmak için kullanılabilir [75-76].

2.7. KAYIPSIZ SIKIŞTIRMADA KULLANILAN GÖRÜNTÜ DOSYA FORMATLARI

2.7.1. GIF

ABD’li bir yazılımcı olan Steve Wilhite tarafından 1987’de icat edilen GIF, hareketli küçük resimler olarak tanımlanmaktadır. GIF kelimesinin açılımı İngilizcede “Graphics Interchange Format”tır ve bu Türkçede grafik değiştirme biçimi anlamına gelmektedir. GIF ile resimler hareketli hale getirilebilmekte, ayrıca videolar da GIF biçimine dönüştürülebilmektedir. Videodan farklı olarak GIF dosya biçimi sesi desteklemez. GIF’ler, sürekli döngü halinde olan ve hiç kimsenin oynat düğmesine basmasını gerektirmeyen bir dizi resim ya da sessiz videodan oluşur. Görüntülerin bu şekildeki tekrarı, GIF’lerin bir şarkı ritmi gibi tanıdık ve akılda kalıcı hale gelmesini sağlar. Lisa Hassell, animasyonların popülerlik kazanmasının sebebini GIF’lerin çabuk algılanmasına bağlamış, GIF’lerdeki dinamik hareketliliğin kavramlarla ve mesajlarla duygulara hitap ettiğini iddia etmiştir. GIF’ler bir bağlamı ve duyguyu hızlı bir şekilde karşı tarafa aktarabilmektedir. Ghaderi ve Afshinfar [77], İngilizce öğrenen orta düzey öğrencilerin, durağan ve hareketli komik resimlerle deyimleri öğrenme ve akılda tutma başarılarını karşılaştırmalı olarak araştırmışlardır. Araştırma sonucunda hareketli komik resimlerin durağan resimlere göre öğrencilerin, deyim öğrenme ve akılda tutma başarıları daha çok arttırdığı görülmüştür. GIF’ler dinamik özellikte olması dolayısıyla fotoğraf ve resimlerle sağlanamayacak ayrıntıyı ve hareketi göstermeye olanak tanımaktadır. Öte yandan GIF’lerin animasyon özelliği paylaşılacak istenen durumu vurgulamaktadır.

Günümüzde iletişim uygulamaları kullanılırken veya sosyal medya platformları üzerinde duygularını daha etkin biçimde ifade etmek istendiği durumlarda GIF kullanımı sıklıkla tercih edilmektedir. İnternetin yaygınlaşması ile GIF’lerin kullanımı giderek artmıştır. Ayrıca çok fazla bant genişliğine ihtiyaç duymaması ve kolay paylaşılabilir olması da GIF’lerin tercih edilme sebeplerindendir. GIF formatının canlandırmayı destekleme özelliğine sahip olması ve sabit boyutlu olması bakımından GIF’ler, internet sitesi tasarımcıları tarafından tercih edilmektedir. İnternet sitelerinde kullanılmaya başlayan GIF materyalleri ile sayfalar içerisindeki kısa video, logo ve banner gibi grafik tasarımların hareketlendirilerek, ziyaretçilerin ilgisi çekilerek İnternet sayfasında daha fazla vakit geçirmeleri sağlanmaktadır. Bu internet sitelerinin yanı sıra Whatsapp, Messenger, Instagram gibi anlık iletişim olanağı sağlayan uygulamalarda da eğlence ve duygu durumunu ifade etmek için hareketli GIF’lerin kullanıldığı görülmektedir [78].

Teknolojinin eğitimle bütünleşmesi, eğitimde teknoloji kullanımının artmasıyla birlikte eğitim alanında da birçok dijital materyal kullanılmaya başlanmıştır. Dil öğretiminin teknoloji ile yapılabildiği her ortamda kullanışlı bir materyale dönüşen GIF'ler, dünyada yabancı dil öğretimi için tasarlanmış uygulama ve sitelerde de yerini almıştır. Bu bağlamda GIF'lerin dikkat çekici ve kolay anlaşılabilir olması, herhangi bir dille sunulmaması açısından birçok alanda bilgi, duygu aktarımı ve iletişim gibi amaçlarla kullanıldığı görülmektedir.

GIF farklı fotoğrafların bir araya getirilmesiyle ya da videoların belirli kesitlerinin kullanılmasıyla oluşturulmaktadır. GIF oluşturmak teknolojik uzmanlık gerektirmez. Dolayısıyla bir GIF yapma programı ya da GIF yapım sitesi aracılığıyla herkes kişisel GIF'lerini oluşturabilmektedir. GIF hazırlamak için kullanılacak ücretli veya ücretsiz birçok site ve uygulamalar bulunmaktadır. GIF oluştururken şu uygulama ve siteler kullanılabilir:

- GIPHY
- GIFS.com
- MakeAGIF
- 5SecondsApp
- GIF Me!
- imgPlay
- Ezgif.com.

Çekilen fotoğraf veya videolar yukarıda sözü edilen site veya benzeri uygulamalara yüklenerek hızlı ve kolay şekilde GIF materyalleri oluşturulmaktadır. Bunun yanı sıra kullanıma hazır GIF'lere de internet arama motorlarından ulaşılabilir.

2.7.2. TIFF

TIFF (Tagged Image File Format), 1986 yılında şu anda Adobe şirketinin bir parçası olan Aldus şirketi tarafından ortaya atılmış ve geliştirilmiştir.

TIFF dosyaları, dijital ortamda .tiff veya .tif dosya uzantıları olarak tutulan dosyalardır. Bu tür dosyalar, orijinal resmin yüksek kaliteli çözünürlüğü, sıkıştırılmaması ve orijinal resim şeffaflık değerlerinin de tutulabileceği biçimde elde edilen dosyalardır. Her pikseli 24 bit değer alabilir. Yüksek kalitede olmaları nedeniyle tutulduğu hafızada diğer dosya biçimlerine oranla daha fazla yer tutarlar.

TIFF dosyaları, genellikle yüksek kalite yazıcı çıktıları, resmin her ince detayının gösterilmesi gerektiği yayımlar, kısaca yüksek renk derinliğine ihtiyaç duyulan durumlarda gösterilmektedirler [79].

2.7.3. PNG

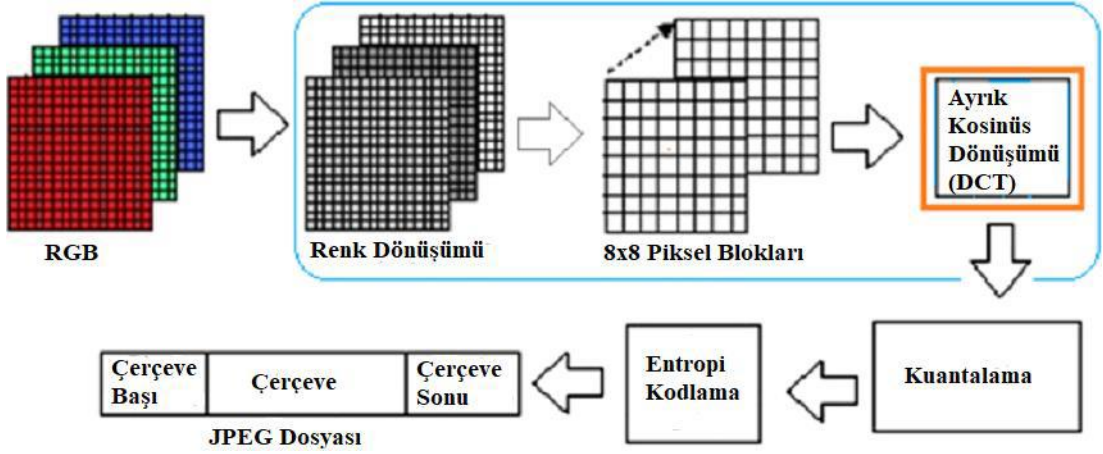
PNG dosya formatı GIF dosya formatının bazı kısıtlarını ortadan kaldırmak ve geliştirmek için 1995 yılında ortaya atılmış dosya formatıdır. GIF formatını oluşturan algoritma aynı zamanda lisans patentine sahip olması nedeniyle, patent durumunun ortadan kaldırılması da yine bu dosya formatının oluşturulma nedenleri arasındadır. Literatürde, Taşınabilir Ağ Grafikleri olarak geçen bu dosya formatı, GIF formatlı dosyaların 256 değişik renge karşı 16 milyona kadar renk barındırabilmektedir. PNG dosyaları sıkıştırılmamış dosyalardır. Her bir pikseli 24 bit RGB ve alfa kanalı ile 32 bit RGBA yani şeffaflık ve opaklığı destekleyecek biçimde geliştirilmiştir. Dijital ortamda .png uzantısı ile gösterilen bu dosyalar, sıkıştırılmamış olmaları ve her bir pikseli 32 bite kadar yer kaplayabilmesi nedeniyle hafızada önemli bir yer tutmaktadırlar.

PNG dosyaları, GIF dosyaları gibi internet tabanlı uygulamalarda tutuldukları gibi bilindiği üzere Mac OS işletim sisteminde ekran görüntüsü alınan resimlerinde varsayılan görüntü biçimidir [80].

2.7.4. Kayıplı Görüntü Sıkıştırma Yöntemleri

2.7.4.1. JPEG

JPEG [81], görüntüleri sıkıştırılmış bir biçimde saklamak için kullanılan bir görüntü sıkıştırma standardıdır. Ortak Fotoğraf Uzmanları Grubu anlamına gelir. JPEG formatı oldukça popülerdir ve dijital kameralar gibi çeşitli cihazlarda kullanılır ve ayrıca internet gibi bant genişliği kısıtlı bir ortamda büyük boyutlu görüntülerin hızlı bir şekilde aktarımı için kullanılan en yaygın standarttır. JPEG formatında kaydedilen görüntü dosyaları genellikle jpg, jpeg veya jpe gibi uzantılara sahiptir. Günümüzde pek çok JPEG görüntü sıkıştırma algoritması mevcuttur ve birçoğu hala geliştirilmeye devam edilmektedir [82]. JPEG görüntü sıkıştırma algoritmasının genel yapısı şekil 2.18'de verilmiş ve bu adımlar aşağıda açıklanmıştır. Kod çözme işlemi ise bu adımların tersten gerçekleştirilmesidir.



Şekil 2.18: JPEG algoritmasının genel yapısı [82]

Renk Dönüşümü:

Görüntüdeki renklerin gösterimi RGB 'den YUV renk uzayına dönüştürülür. Bu dönüşüm işlemi zorunlu değildir. Ancak daha iyi sıkıştırma oranları elde etmek için genel olarak tercih edilir. YUV renk biçimi, renkleri, parlaklık ve renklilik olarak ifade eder. Y parlaklığı, U ve V değerleri de renkliliği ifade eder. Bu dönüşüm bir doğrusal dönüşümdür ve denklem 3.1'de gösterildiği gibi hesaplanır.

$$Y = 0.299R + 0.587G + 0.114B$$

$$U = Cb = 0.564(B-Y) = -0.169R - 0.331G + 0.500B \quad (3.1)$$

$$V = Cr = 0.713(R-Y) = 0.500R - 0.419G - 0.081B$$

Ayrık Kosinüs Dönüşümü (DCT):

Görüntü 8×8 piksel bloklara bölünür ve daha sonra en önemli adımlardan biri olan ayrık kosinüs dönüşümü uygulanır. Ayrık kosinüs dönüşümü uygulandıktan sonra ortaya 64 katsayı çıkar bunlardan ilkinde DC, diğerlerine ise AC adı verilir [83].

Niceleme:

DCT dönüşümü sonrasında ortaya çıkan katsayılar genellikle ondalıklıdır. Bu değerler niceleme yöntemi ile tamsayılara dönüştürülür. Niceleme yönteminde öncelikle her katsayı 1 ile 255 arasında bir tamsayıya bölünür ve daha sonra en yakın tamsayıya yuvarlanır ve görüntüdeki kayıp bu esnada meydana gelir.

Entropi Kodlama:

Son adım olarak entropi kodlama işlemi gerçekleştirilir ve görüntü bu adımda sıkıştırılır. Bu adım kayıpsız bir adımdır ve birçok entropi kodlama algoritması mevcuttur. Bunların en çok bilinenleri Huffman kodlaması, aritmetik kodlama, çalıştırma uzunluğu kodlamasıdır.



Şekil 2.19: PNG Görüntü



Şekil 2.20: JPEG Görüntü

Şekil 2.19’da yer alan PNG dosyasının büyüklüğü 662 KB iken Şekil 2.20’de yer alan dönüştürülmüş JPEG dosyası 63 KB olarak gözlemlenmiştir.

2.7.4.2. JPEG 2000

JPEG2000 standardı, JPEG standardının kısıtlamalarını gidermek ve yüksek kalitede görüntüler elde etmek amacıyla tasarlanmıştır [84]. JPEG2000, ayrık dalgacık dönüşümü (DWT) tabanlı görüntü sıkıştırma standardıdır. 2000 yılında, orijinal ayrık kosinüs dönüşümü tabanlı JPEG standardı [81] yerine geçmek amacıyla Birleşmiş Fotoğraf Uzmanları Grubu komitesi tarafından oluşturuldu. JPEG2000 'in JPEG 'e göre sıkıştırma performansında artış olurken, JPEG2000 'in sunduğu başlıca avantaj, kod akışının önemli esnekliğidir. JPEG2000 ile bir görüntünün sıkıştırılmasından sonra elde edilen kod akışı doğada ölçeklenebilir, birçok yolla kodu çözülebilir [85]. Örneğin, kod akışının herhangi bir noktada kesilmesiyle, görüntünün daha düşük bir çözünürlükte veya sinyal-gürültü oranında bir temsili elde edilebilir. Bununla birlikte, bu esnekliğin bir sonucu olarak, JPEG2000 karmaşık ve hesaplama gerektiren kodlayıcı ve kod çözücülerini gerektirir. Ayrıca JPEG2000’ in JPEG’den üstünlükleri aşağıda maddeler halinde verilmiştir.

- Renkli görüntülerde 48-bit, gri-tonlamalı görüntülerde 16-bit renk uzayına olanak sağlar.
- Kaliteden ödün vermeden daha yüksek sıkıştırma oranı sağlar.
- JPEG’ deki en büyük dezavantajlardan biri, görüntü önce 8×8 ’lik bloklara ayrılıp daha sonra DCT uygulandığı için, özellikle yüksek sıkıştırma oranlarında bloklar arası geçişin keskinleşmesi ve gözle fark edilir hale gelmesidir. Özellikle 1/30 ’dan yüksek oranlarda sıkıştırmış görüntülerde bloklar belli olur. JPEG2000 ’de böyle bir bloklama yapısı olmadığı için bu dezavantaj ortadan kalkmıştır.
- Görüntüdeki önemli parçaların daha yüksek kalitede önemsiz parçaların ise daha düşük kalitede sıkıştırılmasına olanak tanınmasıdır.
- Kayıpsız sıkıştırmaya da izin verir.

JPEG2000 görüntü sıkıştırma algoritması temel olarak aşağıdaki adımlardan meydana gelmektedir.

RGB’den YUV’a dönüşüm veya RCT:

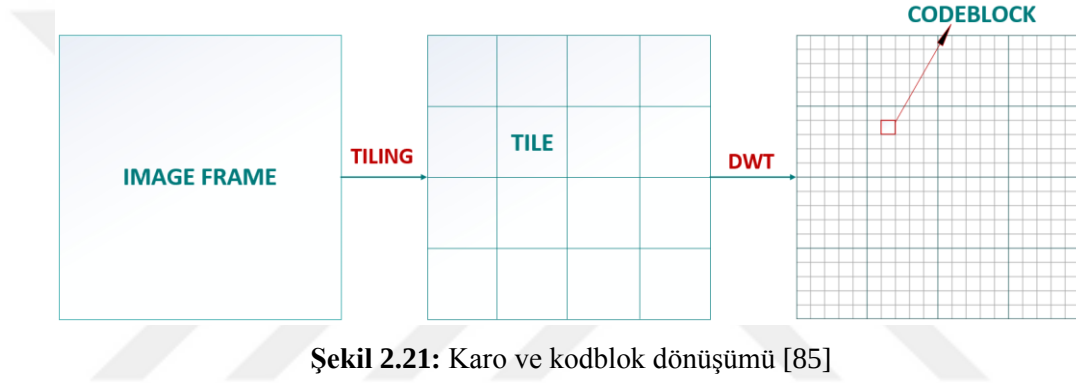
Burada kullanılan YUV yöntemi JPEG yöntemindeki YUV yöntemidir. RGB yöntemi ise kayıpsız sıkıştırmaya olanak tanıyan bir dönüşümdür.

Görüntüyü Parçalama (Tiling):

Bu adımda görüntü bloklara ayrılır. Daha sonra her bir blok ayrı bir şekilde sıkıştırılır. Bu adım eğer istenildiğinde görüntünün tamamını değil sadece belli bölümlerini geri açabildiği için önemlidir.

Ayrık dalgacık dönüşümü (DWT) kullanarak, piksel bilgisini uzamsal etki alanından frekans etki alanına çevirme:

Dalgacıklar veriyi farklı frekans bileşenlerine ayırmak için kullanılan matematiksel fonksiyonlardır. DWT, şekil 2.21’de gösterildiği gibi birçok alt-geçişli (LP, low - pass) ve üst-geçişli (HP, high - pass) sayısal filtre serisi ile gerçekleştirilebilir.



Şekil 2.21: Karo ve kodblok dönüşümü [85]

Bir görüntünün dalgacık ayrıştırmasında, ayrıştırma önce satır satır, daha sonra sütun sütun gerçekleştirilir. Örneğin $N \times M$ boyutlarında bir görüntü için, önce her satır filtrelenir ve filtre çıkışı iki adet $N \times (M/2)$ boyutunda görüntü elde etmek için alt örneklenir. Daha sonra her sütun filtrelenir ve dört adet $(N/2) \times (M/2)$ boyutunda görüntü elde etmek için alt örneklenir.

ROI kodlaması (Region of Interest):

ROI kodlamasında, görüntünün belirli bölgeleri daha yüksek kalitede, diğer yerleri daha düşük kalitede kodlanır. Görüntülerin belli parçalarının daha iyi kodlanmasını istediğimiz uygulamalar için bu adım kullanışlıdır.

DWT dönüşümü sonuç değerlerini Niceleme:

Niceleme adımı DWT’ nin kademeli yapısına uygun olarak tasarlanmıştır. Bu adımda kayıplar meydana gelmektedir. Temel amaç JPEG ’de olduğu gibi yüksek katsayılı olan frekansları sıfıra indirgemektir.

Entropi Kodlama:

Kodlama işlemi algoritmanın son adımıdır. Kodlama işlemi, bir bit düzlemi kodlama tekniğidir. EBCOT algoritmasına dayanır ve her kod bloğunda bağımsız olarak gerçekleştirilir.

Bir görüntünün dalgacık ayrıştırmasında, ayrıştırma önce satır satır, daha sonra sütun sütun gerçekleştirilir. Örneğin $N \times M$ boyutlarında bir görüntü için, önce her satır filtrelendir ve filtre çıkışı iki adet $N \times (M/2)$ boyutunda görüntü elde etmek için alt örneklenir. Daha sonra her sütun filtrelendir ve dört adet $(N/2) \times (M/2)$ boyutunda görüntü elde etmek için alt örneklenir.

Entropi Kodlama:

Kodlama işlemi algoritmanın son adımıdır. Kodlama işlemi, bir bit düzlemi kodlama tekniğidir. EBCOT algoritmasına dayanır ve her kod bloğunda bağımsız olarak gerçekleştirilir.

2.8. GÖRÜNTÜ BÖLÜTLEMEYLE BİRLİKTE KULLANILAN KODLAMA BİÇİMLERİ

2.8.1. Koşu-Uzunluğu Kodlama

Koşu uzunluğu kodlama, aynı elemanın ard arda çok miktarda tekrarlandığı görüntülerde kullanılan bir sıkıştırma biçimidir. Görüntü tek boyutlu ardışık bit akışı biçimine getirildikten sonra kodlanmaya başlanır. Genellikle 2- seviyeli görüntülerde görüntü siyah ve beyaz piksellerden oluşur. Siyah pikseller “0” ile, beyazlarsa “1” ile gösterilir. Koşu, ard arda gelen siyah veya ard arda gelen beyaz pikseller olarak tanımlanır. Birinci pikselden başlanarak, ard arda koşuların uzunluklarının kodlanma işlemi koşu uzunluğu kodlama olarak adlandırılır. Koşu kodlama Şekil 2.22’de özetlenmiştir [7].

1	1	1	0	0
1	1	1	0	0
0	0	1	1	1
0	0	0	1	1
1	1	1	1	1

Koşu kodu: 3234337

Şekil 2.22: Koşu kodlama gösterimi

Şekil 2.22’de gösterilen görüntünün koşu kodlanmasına sol üst köşedeki pikselden başlanır ve satır satır soldan-sağa doğru pikseller karşılaştırılarak ilerlenir. Koşu kodu bilgisinin başına ilave olarak birinci piksel bilgisi eklenebilir. Eklenmemişse bunun “1” veya “0” olduğu farzedilir. Herbir kod sonrası gelen bilgi bir diğer koşuyu temsil edeceğinden

diğer koşuların bilgisini temsil etmeye gerek yoktur. Bu bilgiler ışığı altında aşağıdaki örnek görüntünün koşu kodu şöyle olacaktır: “1”3234337. En baştaki “1” birinci pikselin “1” değeri ile başladığını temsil eder. Hemen onu takip eden bilgi ‘1’in kaç kez arka arkaya geleceğidir. Bir sonraki bilgi diğer bölgenin yani “0”ın kaç kez arka arkaya geldiğini, ondan sonraki diğer bölgenin (‘1’in) kaç kez kullanıldığını belirten uzunluk bilgisidir. Koşu kodunun en yüksek uzunluğu 7 olmuştur. Onluk tabanda 7 sayısı ikilik tabanda 3 bit ile temsil edilebildiği için, uzunluk bilgilerinin herbiri 3-bit ile temsil edilecektir. Orijinal görüntünün ve koşu uzunluğu kodlanan görüntünün ikili tabanda gösterimleri aşağıdaki gibi olacaktır.

Orijinal görüntü: 1110011100001110001111111 : 25 bit

Koşu uzunluğu kodlanmış: 1011010011100011011111 : 22 bit

Şekil 2.28’deki örnek görüntü için koşu uzunluğu kodlamanın sıkıştırma oranı (25/22:1) yani yaklaşık olarak (1,14:1) olmuştur. 1,14’lük bir sıkıştırma oran çok büyük bir oran olmadığı için koşu uzunluğu kodlama tekniği Şekil 2.23’teki görüntü için fazla elverişli değildir. Görüntü boyutları arttığında sıkıştırma oran da görüntüye bağlı olarak artabilir. Koşu uzunluklarının birbirlerine yakın veya büyük miktarlarda arka arkaya uzunluklar olduğu durumlarda koşu uzunluğu kodlama biçimi daha elverişlidir.

1	1	1	1	1
1	0	0	0	0
1	1	1	1	1
0	0	0	0	0
0	1	1	1	1

Koşu Kodu: “1”64564

1110100101110100: 16 bit

Sıkıştırma (1,56:1)

0	0	0	1	1
0	0	0	1	1
1	0	0	1	1
1	0	0	1	1
1	0	0	1	1

Koşu Kodu: “0”323323232

01110111101110111010: 21 bit

Sıkıştırma (1,19:1)

Şekil 2.23: Farklı görüntüler için koşu uzunluğu kodlama

2.8.2. Zincir Kodlama

Görüntüdeki sınırlar uzunluğu ve yönü bilinen bir zincir ile temsil edilebilirler. İki boyutlu bir görüntü düzleminde hareket yönü sekiz durumdan oluşur. Zincirin hareket yönü 8 alternatif yönden birisi olacaktır. Her yön bir numara ile kodlanırsa (3-bit) iki boyutlu bir düzlemde bir başlangıç noktasına geri dönüldüğünde zincir tamamlanacaktır [7].

Zincir kodlama çoğunlukla biçim belirleme işlemlerinde bir nesneyi arka plandan ayırmada bunun kenar haritasını çıkarmada kullanılır. Aynı zamanda, bölge büyütme yaklaşımıyla bölütlenmiş görüntülerde de zincir kodlamadan faydalanılabilir. Zincir kodlama bulanık mantık temellerini kullanarak bölge büyütmeyle bölütlenen görüntülerin sıkıştırılmasında kullanılmıştır. Bölgelerin sadece sınırlarını belirlemek bölgeyi temsil etmek için yeterlidir. Bu sınırlar içerisinde kalan pikseller zaten bölgenin özelliklerini taşıyacaktır. Bu durumda, bölgenin özelliğini temsil eden ortalama gri seviye değeri, zincirin başlangıç konumu ve zincir yön kodları veri dizisi biçiminde oluşturulacak bilgi, kodlanmış veri olacaktır. Sadece bölgenin sınır pikselleri temsil edilecek, bu sınırlar arasında kalan tüm pikseller bölgenin ortalama gri seviye değerine eşit olacaktır. Literatürde çevrit kodlama (contour coding) olarak da anılan kodlama biçimi aslında zincir kodları temeline dayanmaktadır.

2.9. AMAÇ FONKSİYONLU ALGORİTMALARLA GÖRÜNTÜ SIKIŞTIRMA SONUÇLARI

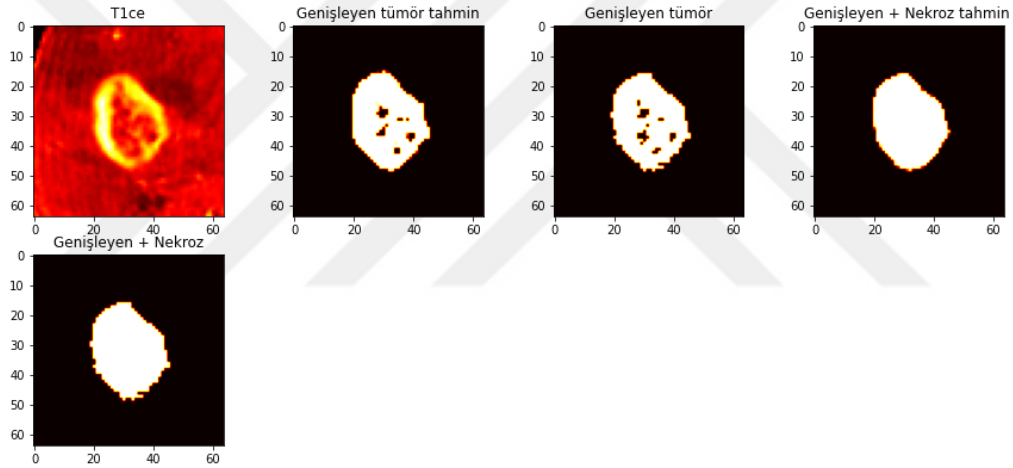
Amaç fonksiyonu tabanlı bölütleme, ham haliyle bir miktar sıkıştırma sunar. Örneğin, $c=4$ bölüte ayrılan görüntü ham haliyle 4 değişken ile temsil edilmektedir ve her bir değişken 2 bitliktir ($2^2=4$). Asıl görüntüde ise her bir değişken gri seviye değeri olup her bir değişken 8 bitle (256 seviye için) temsil edilir. Shannon'un sıkıştırma ilkesine göre aynı miktardaki değişkenlerden biri 8 bit, diğeri 2 bitle temsil edildiğinde gerçekleştirilen sıkıştırma oranı $CR=(8/2)=4$ 'tür. Bu da $c=3$ veya $c=4$ gibi iki bitle temsil edilecek sayıda bölütlenen bir görüntü, aslında orijinal görüntünün dörtte bir sıkıştırılmış haliyle ortaya çıkmış görüntüdür. İki bölüt için yapılan bir sıkıştırma ham haliyle (8:1) bir sıkıştırma oranı sağlar. Özet olarak, görüntü bölütleme işlemi aslında bir görüntü sıkıştırma biçimidir.

En yüksek miktarda sıkıştırma oranı $c=2$ bölüt için gerçekleştirilir. Doğal haliyle (8:1) sıkıştırma sağlayan bölütleme, uygun bir kodlama biçimiyle tamamlandığında daha yüksek oranlarda sıkıştırma sunar. Bu durumda bölütleme işlemi, görüntüyü kodlama öncesi anlamlı

mesaj serisine dönüştürdüğü için, görüntü bölütleme temelli görüntü sıkıştırma işlemi, bir ikinci nesil görüntü sıkıştırma işlemidir.

İki bölüt için “0” ve “1” değerlerinden oluşturulan görüntüde birçok rutin tekrar oluşur. Bu gereksiz tekrarlar ortadan kaldırılırsa bölütleme temelli yaklaşımın sıkıştırma oranı artar. Gereksiz tekrarların sıkıştırmanın en basit yöntemi koşu uzunluğu kodlamada “0” ve “1” ler serisinin aynen bırakılması yerine, seriye karşılık gelen koşu uzunluğu kodlarıyla ifade edilmesi, yüksek oranda kayıpsız bir sıkıştırma gerçekleştirir. Yalnız buradaki kayıpsız sıkıştırma, bölütlenmiş görüntünün kayıpsız kodlanması anlamına gelir. Bölütleme işleminde, görüntünün görsel kalitesini etkileyecek miktarda kayıp olmaktadır.

Amaç fonksiyonlu algoritmalar kullanarak, 5 adet test görüntüsü bölütleme temelli sıkıştırılmıştır. Sonuçlar aşağıdaki şekillerde gösterilmiştir.



Şekil 2.24: Bölütleme temelli sıkıştırılan görüntü

Sıkıştırma uygulanmamış biçimiyle (8:1)'lik bir sıkıştırma oran sağlayan c-2 ile bölütlenen algoritmalar, koşu uzunluğu kodlaması uygulanarak sıkıştırma oranlarını (CR) vermiştir. Görüntünün biçimine göre sıkıştırma oran değişmekle birlikte genellikle (8:1) lik bir orandan (15-20:1) e çıkarır. (32:1) olarak bulunan bu sıkıştırma oranı oldukça yüksektir. 20,3 PSNR sinyal-gürültü oranı ile yapılan sıkıştırma kameraman görüntüsü için çok verimlidir.

Bölüt sayısı artırıldığında sağlanan sıkıştırma oranı düşer. Çünkü, daha fazla bölgeyi daha fazla bitle temsil etmek gerekir. Örneğin iki bölüt için bir bitlik kod yeterli iken 3,4 bölüt için iki bitlik kod, 5,6,7,8 bölüt için üç bitlik kodlar vs. gerekecektir. Bölüt sayısı arttıkça yeniden oluşturulan görüntünün kalitesi de artacaktır ama sıkıştırma oran azalacaktır.

2.10. BÖLGE BÜYÜTME TEMELLİ ALGORİTMALARLA GÖRÜNTÜ SIKIŞTIRMA SONUÇLARI

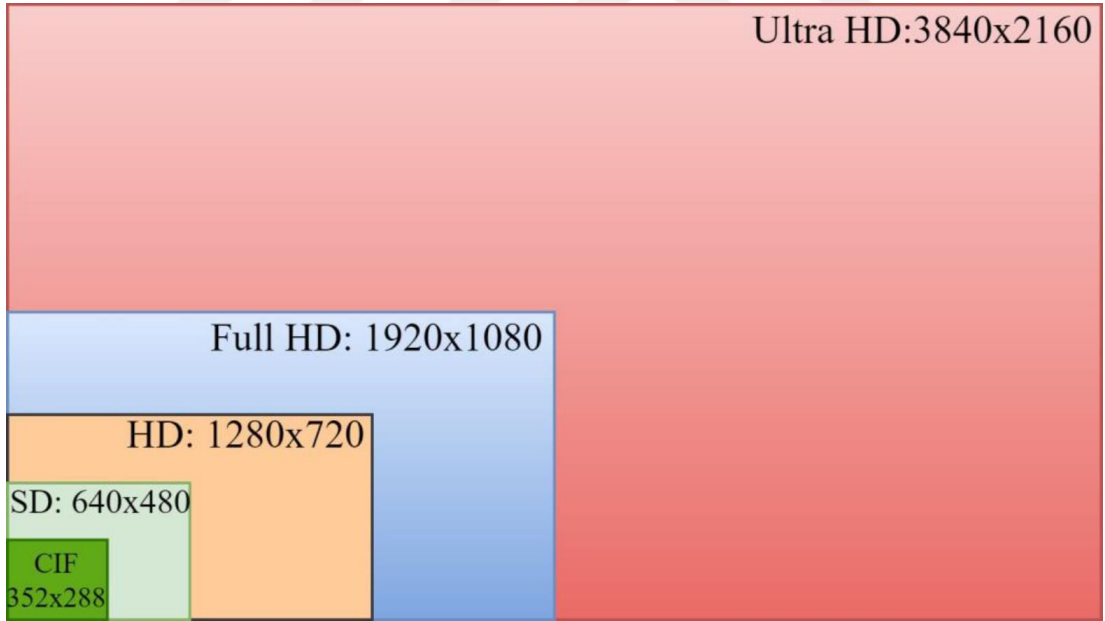
Klasik mantık temeline dayalı bir bölge büyütme algoritması bulanık mantık işleyişine göre uyarlanmıştır. Bunun için bir bulanık çıkarım sistemi tasarlanmış ve bu sistem görüntülere uygulanarak bulanık mantık değerlendirme yapılmıştır. Bölge büyütme yaklaşımı sonucunda görüntü her biri kendi içinde homojen olan bölgelere ayrılır. Bu bölgeler birbirleriyle benzer özellikler taşımamaktadır. Klasik algoritma ile yapılan yaklaşım, sadece gri seviye değerlerini göz önünde bulundurduğu için daha geniş ihtiyaçlara çözüm olamaz. Bulanık mantık temelli algoritma ise 3 giriş değişkeniyle daha hassas değerlendirme yapmaktadır.

Görüntüden bulanık çıkarım sonucu çıkarılan homojen bölgeler, en iyi biçimde temsil edilebilmek amacıyla çevrit kodlama olarak da bilinen bir çeşit zincir kodlama yöntemiyle kodlanabilir. Zincir kodlama biçimi sadece bölgelerin sınır piksellerini yön bilgilerine göre kodlar. Böyle bir kodlama tüm görüntülerde iyi sonuç vermez fakat kendi içinde bölgelere bölünmüş görüntülerde iyi sonuçlar verir. Her yön 3-bit ile temsil edildiği için özellikle büyük ebatlı veya daire biçimli dolgun bölgeler içeren görüntülerde yüksek miktarlarda sıkıştırma sağlar.

2.11. VIDEO KODLAMA

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 [86]. Sinema, televizyon ve video paylaşım uygulamalarının her birinin kullandıkları videoların uzamsal çözünürlükleri farklı olabilmektedir. Şekil 2.25'te ortak ara format (common intermediate format, CIF), standart çözünürlüklü (standard definition, SD) ve yüksek

özünürlüklü (high definition, HD) video çereve örnekleri için piksel boyutları örselleştirilmiştir. Uzamsal özünürlük video kalitesiyle doğrudan ilişkilidir. özünürlük ne kadar yüksek olursa, videodaki piksel sayısı o kadar fazla olur ve videodaki ayrıntıların algılanması o kadar yüksek olmaktadır. Öte yandan yüksek özünürlüklü video için yüksek depolama kapasitesi gerekmektedir. Bu ise teknik olarak başka problemleri ortaya çıkarmaktadır. Örneğın cep telefonuyla 24 bitlik bir RGB videonun Full HD (1920*1080) özünürlük ve 24 fps ile ham olarak kaydedileceğini varsayılın. Bu durumunda 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 düşürölmesi gerekmektedir. Bu işlemlerin ilk adımlarından birisi ham videonun alt-örnekleme işlemidir.

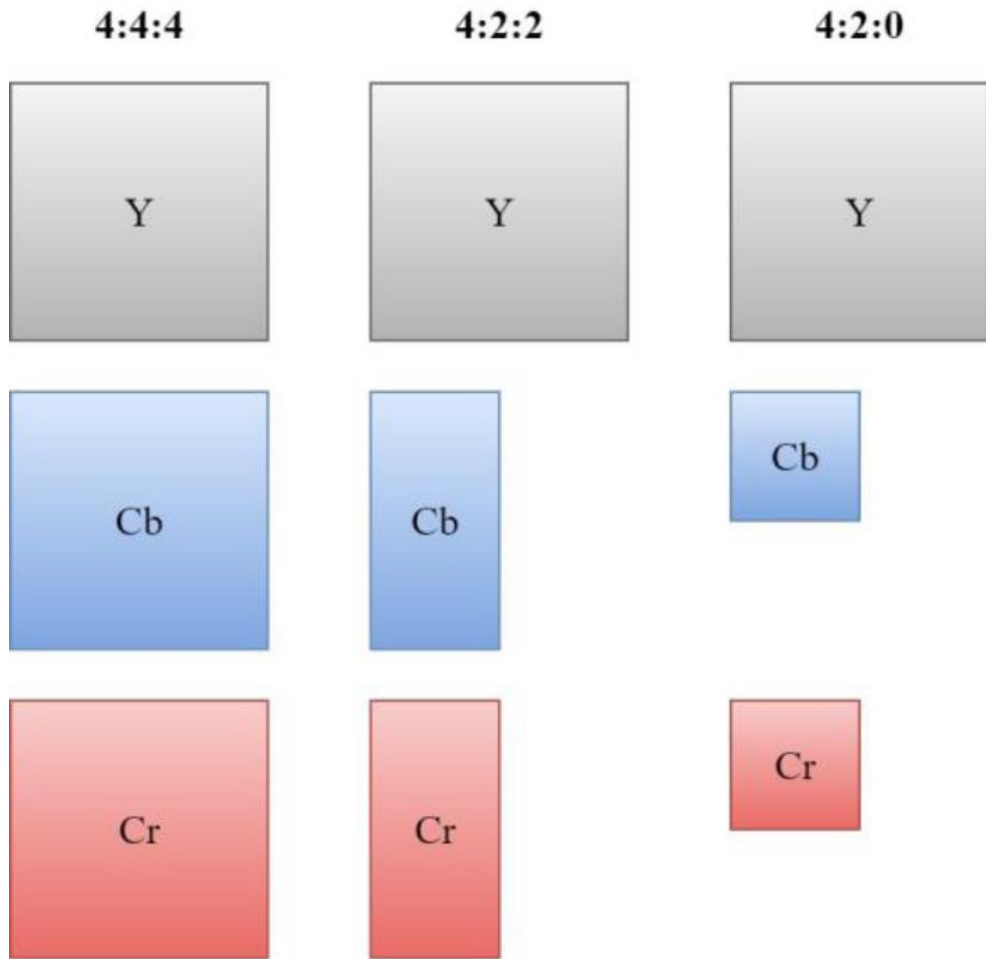


Şekil 2.25: Video formatları için uzamsal özünürlükler

İnsan görsel sisteminin parlaklık değeriğine 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 kullanılabilmektedir [87]. Herhangi bir video çerevesi 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-örnekleme 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 dönüştürölerek gösterilmektedir. Şekil 2.26'da en çok kullanılan kroma alt-örnekleme biçimleri görselleştirilmiştir.

Tüm piksel noktalarında tam çözönlölge 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 yayın iletimi, internet akışı ve tüketici cihazlarında en çok kullanılan formattır.

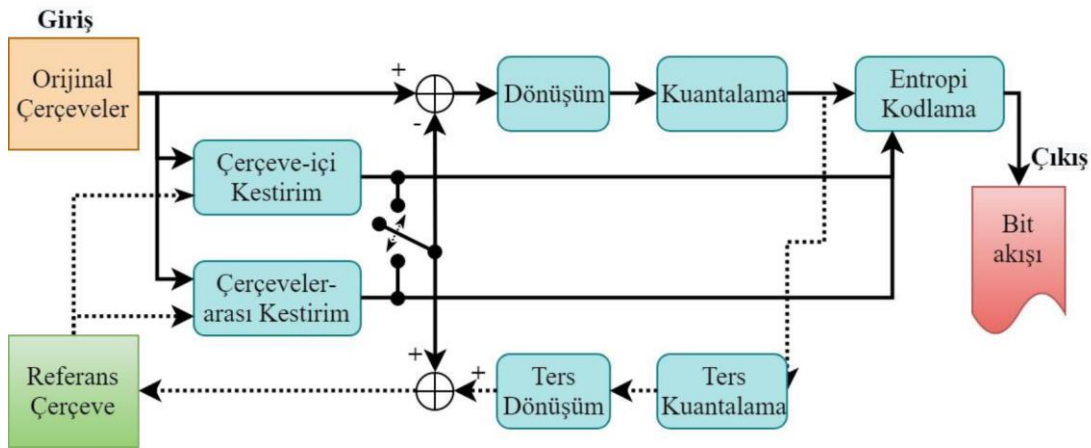


Şekil 2.26: Farklı kroma alt-örnekleme biçimleri

4:2:0 örnekleme hem yatayda hem dikeyde renk farkı bileşenlerini yarı yarıya alt-örnekleyerek algısal renk fazlalığı en üst düzeye azaltmaktadır. Kroma alt-örneklemenin en önemli avantajı, daha kodlama başlamadan bit hızını %50'ye varan miktarda azaltmasıdır [88].

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 alıcıdaki 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 [86]. Şekil 2.27'te video sıkıştırma sürecinin kodlama aşaması verilmektedir.



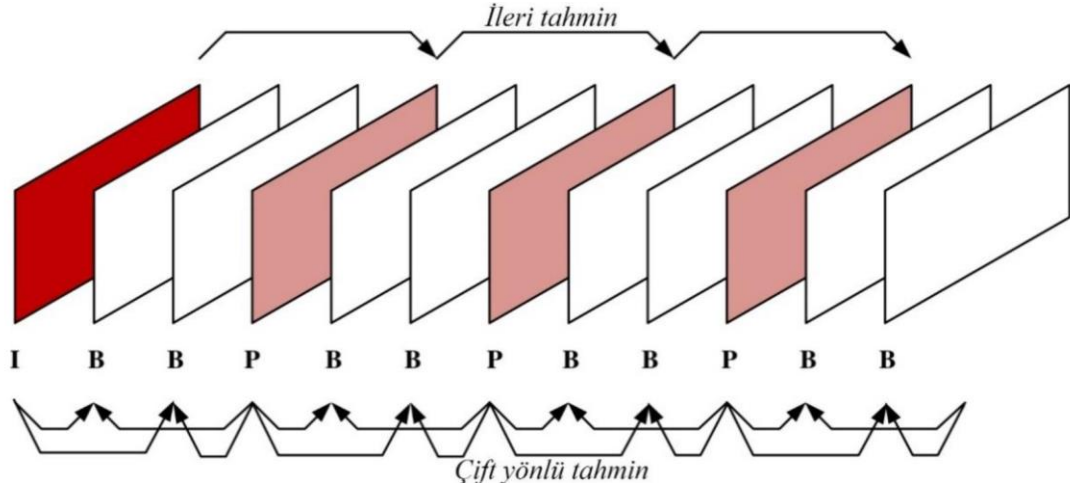
Şekil 2.27: Video kodlama blok yapısı

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-örneklemeye 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 içerisinden mevcut blokla en çok örtüşen blok bulunmaktadır. İki blok arasındaki değişim bir hareket vektörü ile ifade edilmektedir [87].

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 [86]. 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 2.29: Örnek GOP yapısı [88].

Kodlanmış videolarda kod çözücünün hangi çerçeveleri referans olarak alacağı ve çerçevelerin işlenme sırası bir resim grubu (Group of Picture, GOP) ile belirlenmektedir. Şekil 2.29'da 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 [88].

2.11.1. Video CODEC'i

Bir video CODEC'i, bir kaynak görüntüyü veya video dizisini sıkıştırılmış bir biçime kodlar ve kaynak dizinin bir kopyasını veya yakınsamasını üretmek için bunun kodunu çözer. Kodu çözülmüş video sekansı orijinal ile aynıysa, kodlama işlemi kayıpsızdır; kodu çözülmüş sıra orijinalden farklıysa, süreç kayıplıdır [26].

CODEC, orijinal video sekansını, video verilerinin bir yaklaşımını yeniden oluşturmak için kullanılabilecek etkili bir kodlanmış temsil olan bir modelle temsil eder. İdeal olarak model, diziyi mümkün olduğu kadar az bit kullanarak ve mümkün olduğunca yüksek doğrulukla temsil etmelidir. Bu iki hedef, yani sıkıştırma verimliliği ve yüksek kalite genellikle birbiriyle çelişir, yani daha düşük bir sıkıştırılmış bit hızı tipik olarak kod çözücünde düşük görüntü kalitesi üretir [27].

Bir video kodlayıcı üç ana işlevsel birimden oluşur: bir tahmin modeli, bir uzamsal model ve bir entropi kodlayıcı. Tahmin modelinin girdisi, sıkıştırılmamış bir "ham" video dizisidir. Tahmin modeli, komşu video kareleri ve/veya komşu görüntü örnekleri arasındaki benzerliklerden yararlanarak, tipik olarak mevcut video karesinin veya video verisi bloğunun bir tahminini oluşturarak fazlalığı azaltmaya çalışır. H.264/AVC'de tahmin, geçerli çerçevedeki veya

bir veya daha fazla önceki ve/veya gelecekteki çerçevelerdeki verilerden oluşturulur. Komşu görüntü örneklerinden uzamsal ekstrapolasyon, iç tahmin veya kareler arasındaki farkları telafi ederek, ara veya hareket telafili tahmin ile oluşturulur [73]. Tahmin modelinin çıktısı, gerçek mevcut çerçeveden tahminin çıkarılmasıyla oluşturulan bir artık çerçeve ve tahmin içi tipini gösteren veya hareketin nasıl telafi edildiğini açıklayan bir dizi model parametresidir [28].

Artık çerçeve, uzamsal fazlalığı azaltmak için artık çerçevedeki yerel örnekler arasındaki benzerliklerden yararlanan uzamsal modele girdi oluşturur. H.264/AVC'de bu, artık örneklerle bir dönüşüm uygulanarak ve sonuçları nicelendirilerek gerçekleştirilir [72]. Dönüşüm, örnekleri, dönüşüm katsayılarıyla temsil edildikleri başka bir alana dönüştürür. Önemsiz değerleri kaldırmak için katsayılar nicelleştirilir ve kalan çerçevenin daha derli toplu bir temsilini sağlayan az sayıda önemli katsayı bırakılır. Uzamsal modelin çıktısı, bir dizi nicemlenmiş dönüşüm katsayılarıdır [29].

Tahmin modelinin parametreleri, yani tahmin içi mod(lar) veya tahminler arası mod(lar) ve hareket vektörleri ve uzamsal model, yani katsayılar, entropi kodlayıcı tarafından sıkıştırılır. Bu, örneğin kısa ikili kodlarla yaygın olarak oluşan vektörleri ve katsayıları temsil eden verilerdeki istatistiksel fazlalığı ortadan kaldırır. Entropi kodlayıcı, iletilebilen ve/veya depolanabilen sıkıştırılmış bir bit akışı veya dosyası üretir. Sıkıştırılmış bir dizi, kodlanmış tahmin parametrelerinden, kodlanmış artık katsayılardan ve başlık bilgisinden oluşur [30].

Video kod çözücü, sıkıştırılmış bit akışından bir video karesi oluşturur. Katsayıların ve tahmin parametrelerinin kodu bir entropi kod çözücü tarafından çözülür, ardından artık çerçevenin bir versiyonunu yeniden oluşturmak için uzamsal modelin kodu çözülür. Kod çözücü, geçerli çerçevenin bir tahminini oluşturmak için önceden kodu çözülmüş görüntü pikselleriyle birlikte tahmin parametrelerini kullanır ve çerçevenin kendisi, bu tahmine artık çerçeve eklenerek yeniden oluşturulur [31].

2.12. TAHMİN MODELİ

İşlenecek veriler, geçerli çerçeve veya alandaki bir dizi görüntü örneğidir. Tahmin modelinin amacı, verilerin bir tahminini oluşturarak ve bu tahmini mevcut verilerden çıkararak fazlalığı azaltmaktır. Tahmin, önceden kodlanmış çerçevelerden (geçici bir tahmin) veya aynı çerçevedeki önceden kodlanmış görüntü örneklerinden (uzaysal bir tahmin) oluşturulabilir [71]. Bu sürecin çıktısı, bir dizi kalıntı veya fark numunesidir ve tahmin süreci ne kadar doğru olursa, artıktaki o kadar az enerji bulunur. Kalıntı kodlanır ve kodu çözülmüş

kalıntıyı ekleyebilmesi ve mevcut çerçeveyi yeniden oluşturabilmesi için aynı tahmini yeniden oluşturan kod çözücüye gönderilir. Kod çözücünün özdeş bir tahmin oluşturabilmesi için, kodlayıcının tahmini yalnızca kod çözücü için mevcut verileri, yani zaten kodlanmış ve iletilmiş verileri kullanarak oluşturması esastır [32].

2.12.1. Zamansal Tahmin

Öngörülen çerçeve, referans çerçeveler olarak bilinen bir veya daha fazla geçmiş veya gelecekteki çerçeveden oluşturulur. Tahminin doğruluğu genellikle referans çerçeve(ler) ile mevcut çerçeve arasındaki hareket telafi edilerek geliştirilebilir [33].

2.12.2. Mekansal Model: Tahmin İçi

Geçerli görüntü örnekleri bloğu için tahmin, aynı çerçevede önceden kodlanmış örneklerden oluşturulur. Görüntü örneklerinin bloklarının raster tarama sırasında kodlandığını varsayarsak, ki bu her zaman böyle değildir, üst/sol gölgeli bloklar iç tahmin için kullanılabilir. Bu bloklar zaten kodlanmış ve çıktı bit akışına yerleştirilmiştir. Kod çözücü geçerli bloğu işlediğinde, gölgeli üst/sol blokların kodu zaten çözülür ve tahmini yeniden oluşturmak için kullanılabilir [34].

İç tahmin için birçok farklı yaklaşım önerilmiştir. H.264/AVC, bir blok veya makroblok için bir iç tahmin oluşturmak üzere uzamsal ekstrapolasyonu kullanır [69]. Geçerli bloğun üst ve/veya sol taraflarından örneklerin ekstrapolasyonu ile bir veya daha fazla tahmin oluşturulur [70]. Genel olarak, en yakın örneklerin geçerli bloktaki örneklerle büyük olasılıkla yüksek oranda ilişkili olması muhtemeldir ve bu nedenle tahmin bloğunu oluşturmak için yalnızca üst ve/veya sol kenarlar boyunca pikseller kullanılır. Tahmin oluşturulduktan sonra, ara tahmine benzer şekilde bir kalıntı oluşturmak için mevcut bloktan çıkarılır. Kalıntı, tahminin nasıl üretildiğinin bir göstergesi ile dönüştürülür ve kodlanır [35].

2.12.3. İmaj Modeli

Doğal bir video görüntüsü, örnek değerlerden oluşan bir ızgaradan oluşur. Komşu görüntü örnekleri arasındaki yüksek korelasyon nedeniyle, doğal görüntülerin orijinal hallerinde sıkıştırılması genellikle zordur. Uzamsal olarak kaydırılan kopya, orijinal görüntüden herhangi bir yönde uzaklaştıkça, şekilde gösterildiği gibi işlev düşer ve kademeli eğim, yerel bir komşuluk içindeki görüntü örneklerinin yüksek oranda ilişkili olduğunu gösterir [36].

Verimli hareket telafisi veya iç tahmin, artıktaki yerel korelasyonu azaltarak sıkıştırmayı orijinal video karesinden daha kolay hale getirir. Görüntü modelinin işlevi, görüntü veya artık verileri daha fazla ilişkisizleştirmek ve onu bir entropi kodlayıcı kullanılarak verimli bir şekilde sıkıştırılabilecek bir forma dönüştürmektir [37].

2.12.4. Öngörülü Görüntü Kodlaması

Hareket telafisi, bir kodlayıcının önceki veya gelecekteki bir çerçeveye dayalı olarak mevcut çerçevenin bir bölgesinin tahminini oluşturduğu ve bu tahmini bir kalıntı oluşturmak için mevcut bölgeden çıkardığı tahmine dayalı kodlamanın bir örneğidir. Tahmin başarılı olursa, artıktaki enerji orijinal çerçevedekinden daha düşüktür ve artık daha az bit ile temsil edilebilir [38].

2.12.5. Kodlamayı Dönüştürme

Bir görüntü veya video CODEC'deki dönüştürme aşamasının amacı, görüntü veya hareket dengelemeli artık verileri başka bir alana, dönüştürme alanına dönüştürmektir. Dönüşüm seçimi bir dizi kritere bağlıdır [39]:

1. Dönüştürme alanındaki veriler ilişkisizleştirilmeli, yani minimum karşılıklı bağımlılık içinde bileşenlere ayrılmalı ve kompakt, yani dönüştürülen verilerdeki enerjinin çoğu az sayıda değere konsantre edilmelidir [68].

2. Dönüşüm geri alınabilir olmalıdır.

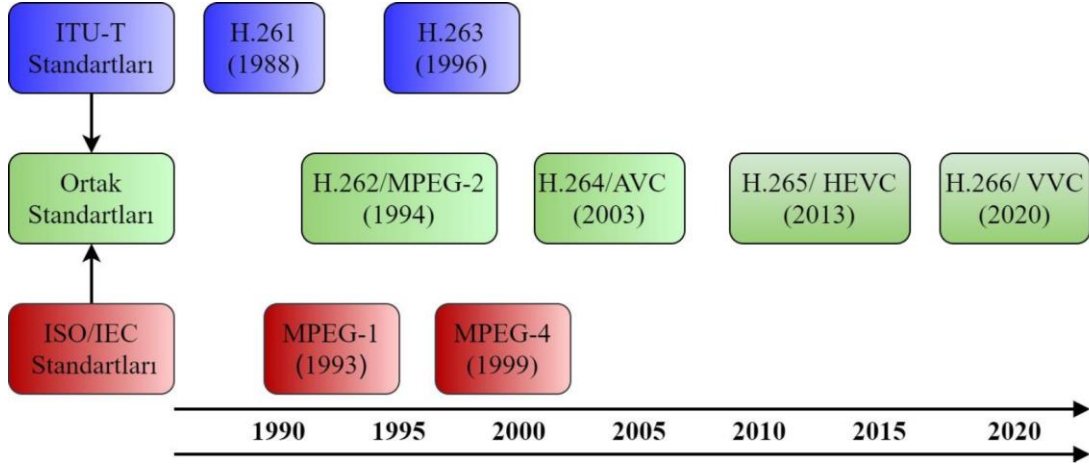
3. Dönüşüm hesaplamalı olarak takip edilebilir olmalıdır, örn. düşük bellek gereksinimi, sınırlı hassasiyetli aritmetik kullanılarak elde edilebilir, düşük sayıda aritmetik işlem vb [40].

Görüntü ve video sıkıştırma için birçok dönüşüm önerilmiştir ve en popüler dönüşümler genellikle blok tabanlı ve görüntü tabanlı olmak üzere iki kategoriye ayrılır. Blok tabanlı dönüşüm örnekleri arasında Karhunen-Loeve Dönüşümü (KLT), Tekil Değer Ayrışımı (SVD) ve her zaman popüler olan Ayrık Kosinüs Dönüşümü (DCT) ve onun yaklaşımları [iv] yer alır. Bunların her biri, $N \times N$ görüntü blokları veya artık numuneler üzerinde çalışır ve bu nedenle görüntü, bir bloğun birimlerinde işlenir. Blok dönüşümleri, düşük bellek gereksinimlerine sahiptir ve blok tabanlı hareket telafi artıklarının sıkıştırılması için çok uygundur, ancak blok kenarlarında bozulmalardan ('blokluluk') muzdarip olma eğilimindedir. Görüntü tabanlı dönüştürmeler, tüm görüntü veya çerçeve üzerinde veya görüntünün 'döşeme' olarak bilinen büyük bir bölümünde çalışır [67]. En popüler görüntü dönüşümü, Ayrık

Dalgacık Dönüşümü, DWT veya sadece 'dalgacık'tır. DWT gibi görüntü dönüşümlerinin durağan görüntü sıkıştırma için blok dönüşümlerinden daha iyi performans gösterdiği gösterilmiştir, ancak tüm görüntü veya döşeme bir birim olarak işlendiğinden ve bloğa iyi "uyması" gerekmediğinden daha yüksek bellek gereksinimlerine sahip olma eğilimindedir tabanlı hareket telafisi. DCT ve DWT'nin her ikisi de MPEG-4 Visual'da bulunur ve H.264'te DCT'ye yaklaşımlar bulunur ve aşağıdaki bölümlerde daha ayrıntılı tartışılır [41, 42, 43].

2.13. VIDEO SIKIŞTIRMA

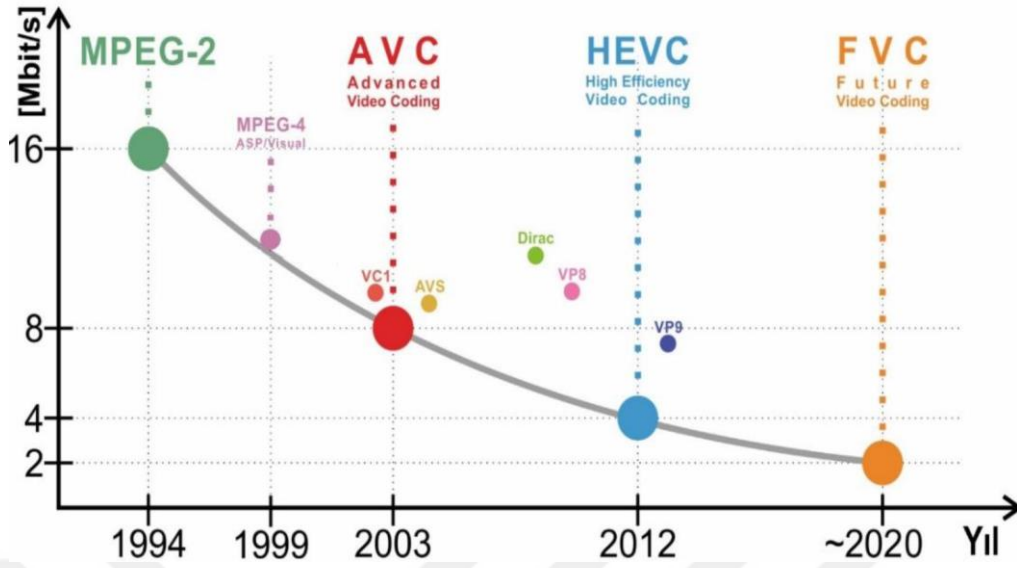
Videoların sıkıştırma yöntemlerinin standartlaşma süreci video teknolojisinin gelişip yaygınlaşması üzerinde büyük etkiye sahiptir. Video kodlama standartları daha düşük bit hızları kullanarak görsel kaliteyi en üst düzeyde tutmayı amaçlamaktadırlar. Başarılı bir video kodlama yönteminin kendisinden 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 [89], 2013 yılında ilan edilen, H.265/HEVC (High Efficiency Video Coding) formatları göze çapmaktadır.



Şekil 2.30: 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 2.30'de 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 [90], BBC yayın organı tarafından geliştirilen Dirac [91], Google öncülüğünde geliştirilen VP8 ve VP9 [92] ve Çin devleti tarafından geliştirilen AVS [93] en çok öne çıkanlardır. VP9 verimlilik bakımından H.264 ile yakın başarımla sunarken H.265/HEVC'ye göre oldukça geride kalmaktadır [94, 95]. Normal videolar için kodlama başarımları HEVC ile kıyaslanabilir düzeyde olan AVS'nin son versiyonu AVS-2 özellikle gözetleme (surveillance) videoları için çok iyi başarımla sahiptir [96, 97].



Şekil 2.31: Full HD bir içerik için video kodlama standartlarının başarımları [98]

Şekil 2.31'de video sıkıştırma standartlarının Full HD bir videoyu sıkıştırma başarımları verilmiştir. Burada görüldüğü gibi ham videoya göre sıkıştırma başarımı oldukça iyi durumdadır. Geliştirilen standartlar sayesinde aynı kalitedeki video için bit oranı değerleri 10 yılda bir yaklaşık %50 aşağı düşerek iyileştirilmektedir [98]. 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.

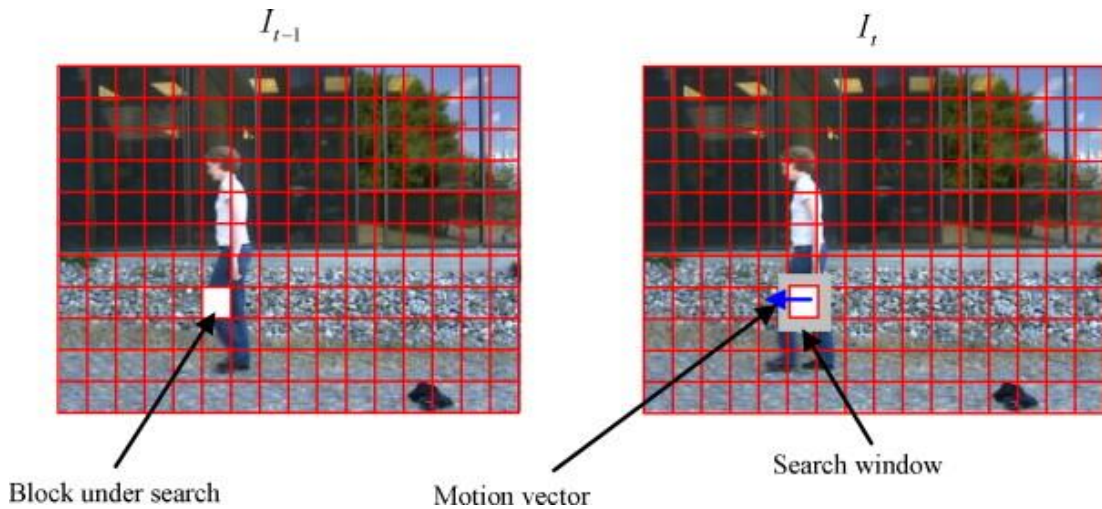
2.14. HAREKETLİ GÖRÜNTÜ SIKIŞTIRMA

Görüntüler sıkıştırılabilir çünkü birbirine yakın pikseller çok ilişkilidir ve uzamsal fazlalıklar içerir. Video sinyalleri, görüntülerin bir zaman serisi olduğundan, ek olarak zamansal boyutta yüksek korelasyon (fazlalık) içerirler. Video sıkıştırma algoritmaları, durağan görüntü sıkıştırma algoritmalarıyla birçok kavramı ve tekniği paylaşır. Bu nedenle, durağan görüntü sıkıştırma çerçevesinin temelleri ile başlıyoruz [44]. Hem görüntü hem de video sıkıştırma algoritmaları, kayıpsız ve kayıplı sıkıştırma yöntemleri olarak sınıflandırılabilir. Kayıpsız sıkıştırmada, orijinal sinyal herhangi bir bozulma olmaksızın tam olarak geri kazanılabilir [45]. Bununla birlikte, insan görsel sisteminin düşük frekanslı içeriğe, yüksek frekanslı içeriğe göre daha duyarlı olduğu bilinmektedir. Bu psiko-görsel fazlalık ile sonuçlanır. Psiko-görsel fazlalıktan yararlanan birçok görüntü/video sıkıştırma yöntemi, yeniden yapılandırılmış sinyalde ihmal edilebilir bozulma ile çok yüksek sıkıştırma oranları elde etmek için bazı yüksek frekanslı bilgileri atmak üzere tasarlanmıştır. Bu sıkıştırma türü,

çoğu gerçek hayattaki uygulamada yaygın olarak kullanılan kayıplı sıkıştırma olarak adlandırılır [46, 47].

İlk olarak, bir xed doğrusal dönüşüm, kaynak görüntüyü bir dönüşüm alanı temsiline eşler. Bir frekans alanı temsili, insanın görsel sistemi göz önüne alındığında, algısal olarak daha önemli olan bilgileri daha az algısal olarak önemli olan bilgilerden ayırmaya yardımcı olur [48, 49]. Dönüşümün kendisi kayıplı değildir ve tersi, orijinal sinyali dönüşüm alanından mükemmel bir şekilde kurtarabilir. Niceleme işlemi, entropi kodlaması için bir nite-alfabe kaynağı oluşturur. Niceleme, düşük frekanslı katsayılar için daha fazla bit ve yüksek frekanslı katsayılar için daha az bit kullanılacak şekilde dönüşüm katsayılarının kesinliğini azaltır, bu da bit hızı düşüşü karşılığında bilgi kaybına neden olur. Daha sonra bir entropi kodlama algoritması, bit hızını daha da azaltmak için nicelenmiş katsayıların kayıpsız sıkıştırmasını gerçekleştirir. Kod çözücü tarafında, kodlama sırasında yapılan işlemlerin tersi kullanılır ve bozuk bir görüntü yeniden oluşturulur [50, 51].

Video sıkıştırma yöntemleri, durağan görüntü sıkıştırmayla birçok temel tekniği paylaşır. Video sıkıştırmadaki en büyük fark, video sıkıştırma algoritmalarının ilişkili video çerçevelerindeki uzamsal ve zamansal fazlalıkları kullanmak için iç ve tahminler arası modlara dayanmasıdır. Çerçeveler arası tahmin yoluyla zamansal artıklığın kullanılması, sabit görüntü sıkıştırma algoritmaları kullanılarak bir videodaki tek tek karelerin sıkıştırılmasına kıyasla çok daha yüksek sıkıştırma oranları elde etmek için ek bilgi sağlar. Hareket tahmini ve hareket dengeleme algoritmaları, video sıkıştırmada zamansal bağıntılardan yararlanmak için anahtar bileşenlerdir [52, 53].



Şekil 2.32: Blok hareket tahmini [55]

Şekil 2.32'de gösterildiği gibi, mevcut çerçeve ile daha önce kodu çözülmüş bir çerçeve (referans çerçevesi) arasındaki hareketi tahmin etmek için blok eşleştirme kullanılır. Geçerli çerçeve üzerinde yakın bir blok eşleşmesi bulunursa, referans ve geçerli çerçeve üzerindeki bloklar arasındaki uzamsal o set (hareket vektörü) kodlanır ve iletilir. Hareket dengeleme, geçerli çerçevedeki piksellerin iç tahminini gerçekleştirmek için iletilen hareket vektörlerini kullanır. Hareket dengeleme adımından sonra kodlayıcı, orijinal çerçeve ile hareket dengelemeli çerçeve arasındaki artık çerçeveyi hesaplar. Kalıntı sıkıştırılır ve hareket vektörleriyle birlikte kod çözücüyeye iletilir [54].

Blok tabanlı hareket tahmininde, genel video sıkıştırma algoritmasını karmaşıklaştıran bazı faktörler vardır. İlk konu, en iyi blok eşleşmesini bulma sorusuna dayanmaktadır [56, 57]. Hareket tahmini için en basit ve en basit yaklaşım, mümkün olan her bloğu değerlendirmek ve minimum kareler toplamını veya mutlak farkların toplamını veren en iyi eşleşmeyi seçmektir [58]. Ancak, yüksek hesaplama yükü nedeniyle kapsamlı arama uygun bir seçenek değildir. Bu sorunu hafifletmek için, hareket tahmin algoritmaları bazı gevşek modlara sahiptir ve sınırlı sayıda aday hareket vektörü üzerinde çalışır [59]. Arama stratejisi nedeniyle, hesaplama karmaşıklığı ile sıkıştırma verimliliği arasında bir değiş tokuş vardır. Çok sayıda aday blok için en iyi eşleşmenin aranması, daha fazla arama işlemi gerçekleştirme pahasına daha küçük kalıntı ve daha iyi oran bozulması performansı sağlar [60, 61].

Bir başka karmaşıklaştırmacı faktör, bir bloğun mevcut ve referans çerçeveler arasındaki yer değiştirmesini en iyi tanımlayan hareket vektörlerinin tamsayı olmayabilmesidir. Video sıkıştırma algoritmaları, alt piksel yer değiştirmelerine izin verecek şekilde zaman içinde geliştirilir. Tamsayı olmayan konumlardaki referans çerçeve piksellerinin tahminini işlemek için enterpolasyon işlemi uygulanır [62].

Video sıkıştırma algoritmaları, sabit görüntü sıkıştırma yöntemleriyle bir kareyi periyodik olarak kodlar. Bu çerçevelere iç kodlu (I) çerçeveler denir ve referans olarak önceden yeniden oluşturulmuş çerçevelere dayanmazlar. Video sıkıştırma algoritmalarında bu tür çerçevelerin kullanılmasının amacı, bit akışında bir hata olması durumunda algoritmanın bir iç çerçeveden yeniden başlamasına izin vermektir [63]. Daha önce yeniden oluşturulmuş çerçeveye referansla tek yönlü referans bilgilerinin kullanıldığı başka bir çerçeve tipi, P çerçevesi olarak adlandırılır. Düşük gecikmeli iletişim gerektiren uygulamalar, I ve P çerçevelerinin kullanımını tercih eder. Düşük gecikmeli video kodlama için basit bir modu desteklerken, tek yönlü sıkıştırmanın bir dezavantajı vardır. Ardışık çerçeveler arasında bir

tıkanma sorunu varsa, önceki çerçeveye referans vermek verimli olmaz. Bu sorunu ele almak için, çift yönlü tahmin edilen B çerçeveleri önerilmiştir [64, 65].

Tipik bir video sıkıştırma algoritması, I kareleriyle birlikte hem P hem de B karelerini içerir. B çerçeve tahmini için bir sonraki referans çerçevesinin kod çözücüsünde bulunması gerektiğinden, çerçevelerin kodlanması ve görüntülenme sıraları aynı değildir [66].

2.15. HAREKETLİ GÖRÜNTÜ SIKIŞTIRMA YÖNTEMLERİ

2.15.1. Kayıpsız Hareketli Görüntü Sıkıştırma Yöntemi

Kayıpsız hareketli görüntü sıkıştırma yöntemleri genellikle gerçek zamanlı video yakalama gibi hızlı kodlama yapmanın gerekli olduğu durumlarda kullanılmaktaydı. Pratikte, kayıpsız bir yöntem ile sıkıştırılmış olan bir hareketli görüntü çok yer kapladığı için ve saklama ünitelerinin kapasiteleri 10 sene öncesine kadar fazla büyük olmadığı için, saklama amacıyla daha sonra kayıplı bir yöntem ile tekrar sıkıştırılmaktaydı. Son yıllarda üretilen yüksek hızlı işlemciler sayesinde, kayıplı bir yöntem ile de gerçek zamanlı video yakalama işlemleri yapılabilmektedir. Bu nedenle günümüzde hareketli görüntü yakalama işlemlerinde de genellikle kayıplı kodlayıcılar tercih edilmektedir. Huffuyv, SheerVideo, CorePNG, MSU Lossless Video Codec, LCL, Lagarith, SKYUV [Smith, 2000], Lossless Motion JPEG gibi birçok kayıpsız sıkıştırma yöntemi vardır.

Huffuyv: Ben Rudiak-Gould tarafından 90'lı yıllarda geliştirilmiş olan Huffuyv, en çok kullanılan kayıpsız hareketli görüntü (video) sıkıştırma algoritmalarından biridir. Sıkıştırma yapmayan YUV'un alternatifi olmuştur. Sıkıştırma ve açma işlemlerinde hızlı bir algoritmadır. Kaynak kodu açık, dağıtımı serbesttir. Huffuyv çerçeve-içi (intra-frame) öngörü kullanarak sıkıştırma yapar. Yani her çerçeveyi bir önceki çerçevenin bağımsız olarak kodlar. Kullandığı öngörü modeli Kayıpsız JPEG ile benzerdir. Öngörülen değer ile orijinal değer arasındaki farkı Huffman kodlaması ile sıkıştırır. Temel olarak 3 öngörü modeline sahiptir:

- Left: Sol taraftaki pikselleri kullanır.
- Gradient: Sol+Üst-ÜstSol kullanır.
- Median: Sol, Üst ve gradient öngörüsünün ortalamasını kullanır.

RGB türünde bir hareketli görüntü daha iyi sıkıştırılabilmesi için R-G, G ve B-G şeklinde kodlanabilir. Her bileşen için (R,G,B veya Y,U,V bileşenleri) ayrı Huffman tablosu kullanılır. Kodlayıcı her bileşen için koleksiyonunda hazır bulunan Huffman tablolarından en uygun olanını seçer. İleriki sürümlerde bu tabloların değişebileceği düşünülerek bu tablolar

istenirse kodlanan hareketli görüntüde saklanabilir. Renk dönüşümü aşamasında renklilik bilgileri alt örnekleme ile azaltılırsa düşük oranda kayıplı sıkıştırma yapılmış olacaktır.

MSU: “MSU (Moscow State University) Graphics and Media Lab.” Tarafından geliştirilmiştir. Her çerçeveyi ayrı bir resim gibi kodlamayıp, çerçeveler arasındaki yüksek oranda benzerliklerden faydalanarak çerçeveler-arası (inter-frame) öngörüsü kullanan bu yöntem, Huffiyuv’a göre daha yavaş çalışmasına rağmen daha iyi sıkıştırma oranı sağlamaktadır. Kayıplı sıkıştırma seçeneği de bulunan MSU, istenirse Huffiyuv gibi, tüm çerçeveleri ayrı bir resim gibi de kodlayabilir.

2.16. KAYIPLI HAREKETLİ GÖRÜNTÜ SIKIŞTIRMA YÖNTEMİ

2.16.1. MPEG-1

MPEG-1 (resmi olarak ISO/IEC 11172 olarak bilinir) [29], ISO/IEC tarafından JPEG ve H.261'e göre geliştirilen ilk genel video sıkıştırma standardıdır, ancak öncelikle dijital video ve sesin CD'lerde depolanması için hedeflenmiştir. Yaklaşık 1,5 Mbps bit hızı [99]. MPEG-1, VCR kalitesinin dijital bir versiyonu olarak düşünülebilir, bu nedenle kaliteden çok sıkıştırma oranına odaklanır. MPEG-1 standardı tarafından sağlanan önemli bir özellik, sıkıştırılmış bit akışında hızlı arama ve ters oynatma için yetenek sağlayan çerçeve tabanlı rastgele video erişimini içermesidir. Standart beş bölümden oluşur: sistemler, video, ses, uygunluk testi ve referans yazılımı. MP3 ses formatı, MPEG-1 standardının en iyi bilinen parçasıdır.

Çoğu video kodlama standardında olduğu gibi, MPEG-1 kodlayıcı üzerinde herhangi bir standardizasyon getirmez ancak yalnızca kodlanmış bir video bit akışının sözdizimini ve bu bit akışının kodunu çözme yöntemini tanımlar. Bu nedenle, bir MPEG-1 kodlayıcı, tasarım için bir esneklik katan, tanımlanan bit akışı sözdizimi uyumlu olduğu sürece, kullanıcıların tercihlerine kadar farklı şekillerde uygulanabilir. Böylece farklı uygulamalar, kalite pahasına farklı karmaşıklıklarla ortaya çıkıyor.

2.16.2. MPEG-2

MPEG-2 Bölüm 2 (resmi olarak ISO/IEC 13818-2 olarak bilinir) veya H.262 [100, 101], dijital depolama ve TV yayıncılığı için ITU-T ve ISO/IEC tarafından ortaklaşa geliştirilmiş bir standarttır. Bu standart, MPEG-1'in daha büyük resim boyutlarıyla ilgili bir uzantı sürümüdür ve video kalitesini artırmak için daha gelişmiş teknikler sağlar. DVD'lerde

ve HDTV'de kullanılan standart sıkıştırma şeması olmuştur; veri hızları, depolama ve iletim için yaklaşık 40 Mbps'ye kadar veya profesyonel uygulamalar için daha da yüksek olabilir [101]. Ayrıca geçmeli video için destek sağlar ve MPEG-1 codec bileşeniyle tam uyumludur.

2.16.3. MPEG-4

MPEG-4 standardı birkaç parçaya bölünmüştür ve temel parçalar MPEG-4 Bölüm 2 (resmi olarak ISO/IEC 14496-2 olarak bilinir) [102, 103] ve MPEG-4 Bölüm 10'dur (MPEG-4 AVC/H .264) [104, 105], burada MPEG-4 Kısım 10 bir sonraki konuda açıklanacaktır. MPEG-4 Visual olarak adlandırılan MPEG-4 Part 2, 1998'in sonlarında tanıtıldı ve ISO/IEC tarafından bir grup ses ve video kodlama formatı için bir standart belirledi. Gözetim uygulamalarında kullanılan klasik MPEG-4 video akış standardıdır. Temel olarak MPEG-1 ve MPEG-2 ile aynı tekniğe dayalıdır ve ayrıca yeni uygulamalara odaklanmıştır. Yeni standart, esnek bant genişliklerini destekler; hem cep telefonları gibi daha düşük bant genişliği tüketen uygulamalar hem de TV yayını gibi daha yüksek bant genişliği tüketen uygulamalar için. Aşamalı ve taramalı video ve farklı renk örnekleme formatları MPEG-4 Visual tarafından desteklenir.

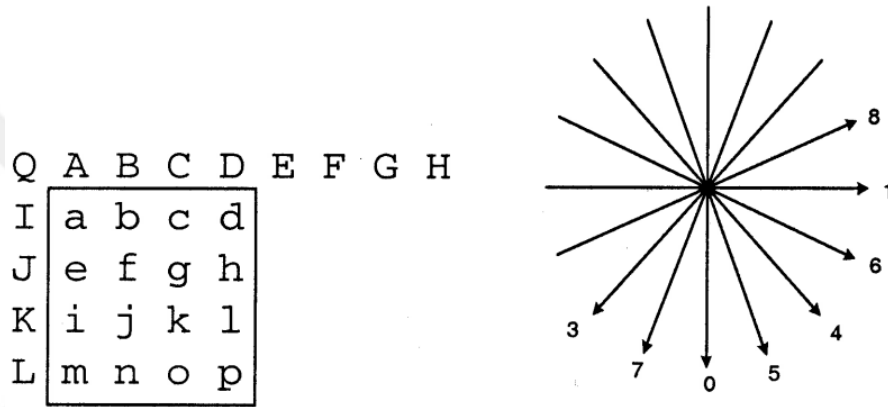
MPEG-4 standardı, kullanıcıların uygulamalara bağlı olarak farklı ayarlar seçmesine izin vermek için profiller ve düzeyler kavramını tanımlar. Gelişmiş Basit Profil (ASP), internet üzerinden multimedya akışı için kullanılan özellikle ilgi çekicidir. Bu profilin DivX, Xvid ve QuickTime 6 gibi birçok codec tarafından da kullanıldığı bilinmektedir.

2.16.4. H-264

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 verisiyonu 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 [106].

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 2.33'te 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.



Şekil 2.33: H.264/AVC Çerçeve içi kestirim modları [106].

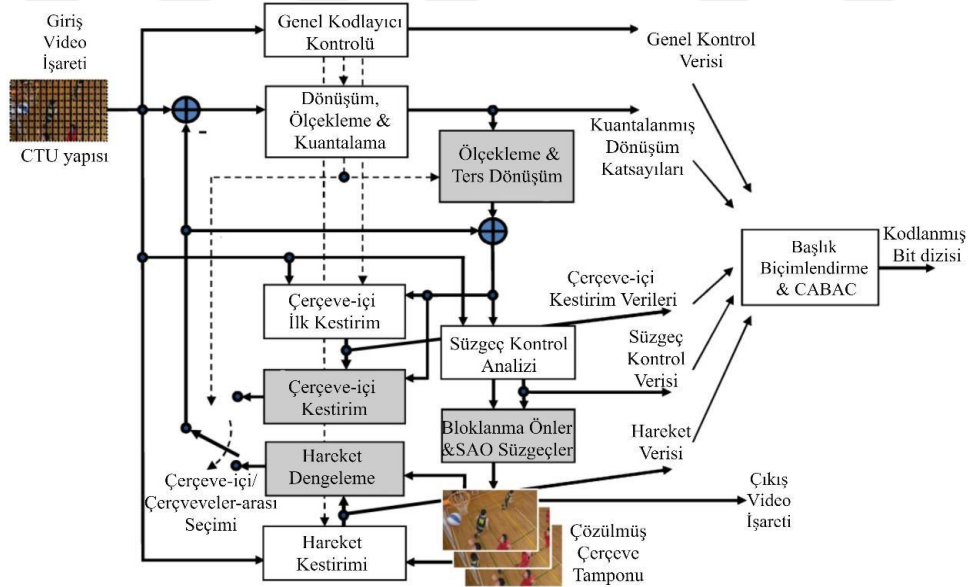
AVC'de sıkıştırma başarımını arttırmak için çerçeve-içi ve çerçeveler arası kestirimde değişken makroblok yapıları kullanılmıştır. Çerçeve-içi kestirimde 16×16 bloklara ek olarak 4×4 blok boyutu da kullanılmıştır. Çerçeveler-arasındaki hareket kestirimi ve hareket dengelemesinde ise mevcut 16×16 makroblok 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 [106].

2.16.5. H-265

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 [107]. 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. 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 2.34: HEVC video kodlayıcı yapısı [107].

Şekil 2.34’te 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 [107].

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’lardan 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 açısall mod, düzlemsel (planar) ve DC) kullanılmaktadır. Bir PU bloğu içerisindeki mod, kendisinden önceki PU blokların modlarına da bağı 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 başarıml ve bellek gereksinimini azaltmak gibi avantajlar sağlanmıştır.

HEVC standardında tasarım özelliklerindeki yenilikler uygulama ve ağı ortamlarındaki çeşitliliğe uyumu arttırmanın ve veri kayıplarına karşı dayanıklılığı arttırmanın yanı sıra daha hızlı işlemlere olanak vermiştir. Bunun için AVC'deki dilimli yapı, çeşitli türlerdeki meta verileri, ağı 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.

3. YÖNTEM

Bu tezde bambaşka bir görüntüden ve videodan rastgele seçilen bir frameden elde edilen evrensel küme ile oluşturulan video görüntüsü daha sonra detaylı bir şekilde anlatılacak olan sympes yöntemi kullanılmıştır. İlk olarak kullanılan matematiksel model aşağıda anlatılmıştır.

3.1. KULLANILAN MATEMATİKSEL MODEL

3.1.1. Sürekli Zaman Bölgesinde Tanım

Bir boyutlu Sympes ile sürekli bir sinyali belirli bir aralığı içerisinde eşitliği (3.1) deki biçiminde yazılabilir,

$$f(t)=Ce(t)s(t); \quad t_1 < t < t_2; \quad T=t_2-t_1 \quad (3.1)$$

denkleminde çıkan $s(t)$, $f(t)$ özgün işaretinin temel istatistiksel özelliğini bulunduran sürekli bir fonksiyondur. Ek olarak T aralığında $f(t)$ yi minimum kareler anlamında temsil edecek biçimde bir olası değere sahip ortonormal fonksiyon kümesinin içerisinde minimum hale getirecek olan fonksiyondur.

$$\delta = \frac{1}{T} \int_{t_1}^{t_2} [f - Cs(t)]^2 dt \quad (3.2)$$

Bu bağlamda $s(t)$ fonksiyonu en küçük kareler anlamında, $f(t)$ işaretinin en büyük enerjisini taşır ve $s(t)$ ve $f(t)$ 'nin enerji veya imza fonksiyonu şeklinde tanımlanır. C reel bir sabit olup

$$C = \int_{t_1}^{t_2} f(t)s(t)dt; \quad t_1 < t < t_2; \quad T=t_2-t_1 \quad (3.3)$$

(3.3) denkleminde bulunabilir. Reel c sabiti Çerçeve kazancı veya Çerçeve Ölçekleme Katsayısı diye isimlendirir. C ve $S(t)$ bulunduğunda $e(t)$ fonksiyonu

$$e(t) = \frac{f(t)}{cs(t)}; \quad t_1 < t < t_2; \quad T=t_2-t_1 \quad (3.4)$$

şeklinde bulunur. (3.4) eşitliğinden anlaşılacağı üzere sürekli sinyalinin enerji fonksiyonu ve ölçekleme katsayısına en küçük kareler anlamında yaklaşmasını sağlamaktadır. Bu sebeple $e(t)$ zarf fonksiyonu olarak adlandırılır.

3.1.2. Ayrık Zaman Bölgesinde Tanım

Ayrık zaman bölgesinde $f(t)$ sürekli sinyali T zaman aralığında;

$$f_i = \{f(t_i); i=1,2,\dots,m\} = \{f_1, f_2, \dots, f_m\} \quad (3.5)$$

şeklinde tanımlanır

$$[f]^T = [f_1, f_2, \dots, f_m] \quad (3.6)$$

olarak tanımlanır. $e_K * s_R$ vektörünü V_i olarak alabiliriz. (6)'dan $[f]$ sütun vektörü m -boyutlu ortogonal vektör uzayı olarak

$$[f] = \sum_{i=1}^m c_i [V_i] \quad (3.7)$$

yazılır. (3.7) deki terimlerin ilk kısmı hariç çıkarılabilir. Bu şekilde

$$[f] = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} \cong c_1 [v_1] = c_1 \begin{bmatrix} v_1^{(1)} \\ v_2^{(1)} \\ \vdots \\ v_m^{(1)} \end{bmatrix} \quad (3.8)$$

yaklaşımı elde edilir. Bulduğumuz $[V_i]$ ortonormal vektörün

$$[R][V_i] = \lambda_i [V_i] \quad (3.9)$$

öz değer probleminden geldiği görülür. Bulduğumuz $[R]$ öz ilintili matris olup $\{[f], [f]^T\}$ 'nin beklenen değeridir, $\lambda_i [V_i]$ vektörünün özdeğeridir.

$$[R] = E[ff^T] \quad (3.10)$$

en yüksek öz değeri $\lambda_{\max}$ olduğu düşünülüp ve bunun öz vektörü $[S] = [V_i]$ düşünüldüğünde (3.9) denklemimiz

$$[R][S] = \lambda_{\max}[S] \quad (3.11)$$

şeklinde ifade edilir. $[f]$ 'in minimum kareler bağlamında (3.12) ifadesinde bulunan yaklaşımı gösterir.

$$[f] = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} \cong C[S] = C \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix} \quad (3.12)$$

burada C reel bir sabit olup

$$C = [f]^T [S] \quad (3.13)$$

denklemleri bulunur. Ayırık zaman bölgesinde işaretimizin $[e]^t = [e_1, e_2, \dots, e_m]$ zarf vektörleri

$$e = \frac{f}{C_s} \quad (3.14)$$

biçimindedir. Burada e'yi bir birim matris ile çarpılması sonucunda

$$[e_d] = \begin{pmatrix} e_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & e_m \end{pmatrix} = \begin{bmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} * [e_1, \dots, e_m] \quad (3.15)$$

1Boyutlu Ayırık Sympes Bölgesinde

$$[f] = C[e_d][S] \quad (3.16)$$

şeklinde tanımlanır.

3.2. SYMPES YÖNTEMİ İLE VIDEO GÖRÜNTÜLERİN KODLANMASI

Sympes yöntemi ile yeniden oluşturulacak videonun her bir frame görüntüsünün Yapı ve Enerjilerini elde etme işlemine kodlama denir. Elde edilen Yapı ve Enerjilerden tekrar görüntü elde etme işlemine çözücü denir.

3.2.1. Sympes Metodu ile Video Görüntülerin Modellenmesi

Bu bölümde modellenecek videonun enerji ve yapı bloklarının oluşumu 3.1 bölümünde anlatılan matematiksel işlemlere göre yazılan MATLAB kodu ile üretilmiştir. Kullanılan videonun her bir karesinden çıkan görüntüler, $M \times N$ boyutunda 0-255 ya da 0-1 aralığında bulunmakta olan gerçek değerlerinin olduğu bir görüntü matrisidir. M ve N değerleri görüntüde olan piksel indislerinin bulunduğu satır ve sütunlara karşılık gelmektedir. İlk olarak videomuz görüntü karelerine ayrılır. Her bir kare $M \times 1$ boyutundaki satır vektörü olduğu düşünülür ve matematiksel model için giriş işareti $VF_j^{(s)} (j = 1, 2, \dots, N_f; s = 1, 2, \dots, S)$ diye adlandırılan birbirine benzemeyen görüntü bloklarına bölünmüştür. Tüm

VF'ler, $L_F \times 1$ uzunluğundaki satır vektörüdür. L_F çerçeve uzunluğu dersek buradan N_F yani çerçeve uzunluğu $N_F = N/L_F$ şeklinde olmaktadır. Videoya ait verinin VF bazındaki elemanları $e_{i,j(s)}$, ($i = 1, 2, \dots, L_F; j = 1, 2, \dots, N_F; s = 1, 2, \dots, S$) diye ifade edilir. O zaman tüm VF çerçeve matrisi $(VF_j^{(s)})$, alt çerçeve vektörleri cinsinden (3.17) gibi gösterilir.

$$VF_j^{(s)} = \begin{bmatrix} e_{(j-1)L_F+1}^{(s)} \\ e_{(j-1)L_F+2}^{(s)} \\ \vdots \\ e_{(j-1)L_F+L_F}^{(s)} \end{bmatrix}, \quad j = 1, 2, \dots, N_F; s = 1, 2, \dots, S \quad (3.17)$$

Her videonun karesinin bloklara ayrılma işlemi sonunda bulunan video ifade matrisi (3.18)'deki gibi ifade edilir.

$$VF^{(s)} = [VF_1^{(s)}, VF_2^{(s)}, \dots, VF_{N_F}^{(s)}] \quad (3.18)$$

Önerdiğimiz yöntemde ölçekleme sabiti, işaretin enerjisi ve bilgi bloklarının çarpımı ile oluşmaktadır. Bu tezde, uzunluğu L_F , herhangi j. VF'nin yaklaşık değeri (3.19)'da gösterilmiştir.

$$VF_j \cong G_j E_{IE} S_{IS}, \quad (j = 1, 2, \dots, N_F) \quad (3.19)$$

G_j reel bir sabit olup, VF_j 'nin kazancıdır. $IE \in \{1, 2, \dots, N_{IE}\}$ daha önceden tanımlı yapı blogunun (CEB) indis sayısı ve $IS \in \{1, 2, \dots, N_{IS}\}$ yine daha önceden tanımlı enerji blogunun (CPB) indis sayısıdır. IS ve IE CPB ve CEB'nin indeks sayılarının miktarıdır. IS , IE , N_{IS} ve N_{IE} değerleri tamsayıdır.

CPB videonun enerji karakteristiğini alan VF'nin inceleme sonucunda bulunana yoğunluk bilgisi sayesinde elde edilmiştir. Bunu matematiksel olarak göstermek istersek (3.20) numaralı eşitlikte gösterilmiştir.

$$|S_{IS}|^T = |S_{IS,1}, S_{IS,2}, \dots, S_{IS,L_F-1}, S_{IS,L_F}| \quad (3.20)$$

$CPB = G_j S_{IS}$ LMS anlamında GF_j 'nin en yüksek enerjisini oluşturmaktadır. Burada G_j bir sabit olup GF_j 'nin yoğunluk seviyesini sağlar.

$$G_j = |CPB|^T S_{IS} \quad (3.21)$$

CEB ise köşegen matris olur ve (3.22) numaralı eşitlik ile gösterilir.

$$E_{IE} = \text{diag}[E_{IE,1} \ E_{IE,2} \ \dots \ E_{IE,L_F-1} \ E_{IE,L_F}] \quad (3.22)$$

E_{IE} , CEB'nin oluşmasında video blok verisinin ayırt edici özelliğini göstermektedir.

Ortonormal vektörler $\{\varphi_{jk}\}$, ile yapılan vektör uzayında her VF düzensiz olabilir. Reel ortonormal vektörlerin transpozunu alırsak $\{\Phi_j^T\}$ ve bunlar dönüşüm matrisinin sütunları olur dersek;

$$\Phi_j^T = [\varphi_{j1} \ \varphi_{j2} \ \dots \ \varphi_{jL_F-1} \ \varphi_{jL_F}] \quad (3.23)$$

şeklinde yazılır. Ve VF_j eşitlik (3.24) gibi yazılır.

$$VF_j = \Phi_j^T \quad (3.24)$$

O zaman $G_j = [g_1 \ g_2 \ \dots \ g_{L_F-1} \ g_{L_F}]$ 'dir. $\Phi_j^T = \Phi_j^{-1}$ eşitliğinden $VF\Phi_j = \Phi_j\Phi_j^{-1}G_j$ eşitliğini çıkartırız. Daha sonra VF_j (3.25) nolu eşitlik ile ortonormal vektörlerin ağırlıkları toplamı şeklinde yazılır.

$$VF_j = \sum_{k=1}^{L_F} g_k \varphi_k, \quad k=1,2,\dots,L_F \quad (3.25)$$

VF 'lerin katsayılarını (3.26) numaralı eşitlikten bulabiliriz.

$$g_k = VF_j \varphi_{jk}^T, \quad k=1,2,\dots,L_F \quad (3.26)$$

$VF_{jt} = \sum_{k=1}^t g_k \varphi_{jk}$, buna göre VF_j 'yi $1 \leq t \leq L_F$ olsun ve $t = L_F$ ise $VF_{jt} = VF_j$ olur ve yaklaşık hatayı (3.27) numaralı eşitlikteki gibi yazarız.

$$\varepsilon_t = VF_j - VF_{jt} = \sum_{k=t+1}^{L_F} g_k \varphi_{jk} \quad (3.27)$$

(3.27)'ye göre hatanın beklenen değeri φ_{jk} 'ya bağlı olarak, LMS anlamında minimum olacak şekilde belirlenir. Buna bağlı olarak φ_{jk} 'lar VF_j 'nin özilintili matrisi ve R_j 'nin özvektörleri hesaplanır. Ortonormalite hesaba katılırsa, LMS hatası ise,

$$\varepsilon_t \varepsilon_t^T = \sum_{k=t+1}^{L_F} g_k^2 \quad (3.28)$$

$$J_t = E[\varepsilon_t \varepsilon_t^T] = \sum_{k=t+1}^{L_F} E[g_k^2] \quad (3.29)$$

şeklinde ifade edilir.

J_t ise $[\varepsilon_t \varepsilon_t^T]$ beklenen değerini belirtirse,

$$E[c_k^2] = E[\varphi_{jk}^T (VF_j^T VF_j) \varphi_{jk}] = \varphi_{jk}^T R_j \varphi_{jk} \quad (3.30)$$

R_j ise VF_j 'nin özilintili matrisi şeklinde tanımlanmıştır. Dönüşümün en doğru biçimde yapılması için, kullanılan t için, J_t 'yi minimuma indirecek ve ortonormallik kısıtlamasına uygun şekilde φ_{jk} 'yı bulunmasıdır. Bu minimuma indirmeyi λ_k (Langrange çarpanı) kullanılarak elde edilir.

$$J_t = \sum_{k=t+1}^{L_{GE}} [\varphi_{jk}^T R_j \varphi_{jk} - \lambda_k (\varphi_{jk}^T \varphi_{jk} - 1)] \quad (3.31)$$

bulunan J_t eşitliğin gradyanı alırsa,

$$\frac{\partial J_t}{\partial \varphi_{jk}} = \frac{\partial}{\partial \varphi_{jk}} \sum_{k=t+1}^{L_F} [\varphi_{jk}^T R_j \varphi_{jk} - \lambda_k (\varphi_{jk}^T \varphi_{jk} - 1)] \quad (3.32)$$

$$2R_j \varphi_{jk} - 2\lambda_k \varphi_{jk} = 0 \quad (3.33)$$

$$R_j \varphi_{jk} = \lambda_k \varphi_{jk} \quad (3.34)$$

R_j özilinti matrisidir aynı zamanda pozitif ve Teoplitz matrisidir.

$$R_j = \begin{bmatrix} r_j(1)r_j(2) & \cdots & r_j(L_F) \\ \vdots & \ddots & \vdots \\ r_j(L_F)r_j(L_F - 1) & \cdots & r_j(1) \end{bmatrix} \quad (3.35)$$

$$r_j(d + 1) = \frac{1}{L_F} \sum_{q=[(j-1)L_F+1]}^{L_F-d} x_q x_{q+d}, \quad d=0,1, \dots, (L_F - 1) \quad (3.36)$$

R_j 'nin özdeğerleride gerçektir ve birbirine eşit değildir. KLD metoduna göre, özilintili matrisi, R_j 'nin özdeğerleri olan $\lambda_{1j}, \lambda_{2j}, \dots, \lambda_{L_F j}$ gittikçe azalan şekilde sıralanmakta ve bu özdeğerlere karşılık gelen özvektörler belirlenmektedir.

$$VF_j^T VF_j = \sum_{k=1}^{L_F} g_{jk}^2 = \sum_{k=1}^{L_F} \lambda_{jk} \quad (3.37)$$

(3.37) numaralı eşitlikte, VF_j 'nin en yüksek enerjisine sahip ilk n tane bileşen alınarak (3.38)'deki gibi gösterilir.

$$VF_j \cong \sum_{k=1}^n g_k \varphi_{jk} \quad (3.38)$$

$n=1$ iken eşitlik (3.38)'un en temel hali (3.39) numaralı şekilde olacaktır. Böylece φ_{jk} enerji vektörü adını alacaktır. Yani enerji vektörünün, LMS anlamında en yüksek enerjiye sahip, VF_j 'nin tüm blokları enerji vektörü olduğu söylenebilir.

$$VF_j \cong g_1 \varphi_{j1} \quad (3.39)$$

Enerji vektörü olarak adlandırılan φ_{j1} , orijinal işarete bakılmasıyla VF 'nin en değerli bilgisini taşımaktadır.

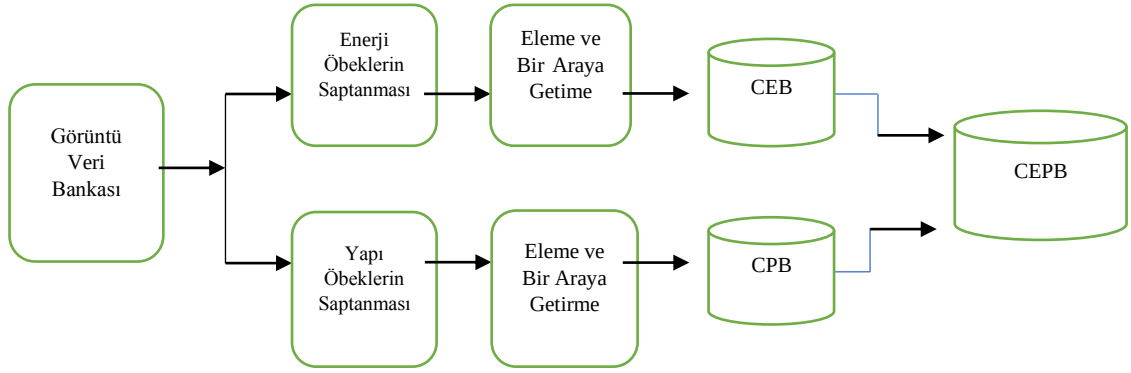
3.2.2. Enerji ve Yapı Bloklarının indirgenmesi

İki vektör arasındaki benzerliği belirlemek için kullanılan en iyi yöntemlerden biri Pearson Korelasyon katsayısıdır. Pearson korelasyon katsayısı;

$$\frac{\frac{\sum_{i=1}^L y_i z_i - [\sum_{i=1}^L y_i \sum_{i=1}^L z_i]}{L}}{\sqrt{\left[\frac{\sum_{i=1}^L y_i^2 - (\sum_{i=1}^L y_i)^2}{L} \right] \left[\frac{\sum_{i=1}^L z_i^2 - (\sum_{i=1}^L z_i)^2}{L} \right]}} \quad (3.40)$$

Şeklinde hesaplanır. Y ve Z iki dizi olsun. $Y = [y_1 y_2 \dots y_L]$ ve $Z = [z_1 z_2 \dots z_L]$ olan vektörlerdir. Pearson korelasyon katsayısı eşitlik (3.42) den de anlaşılacağı üzere her zaman -1 ile +1 değerleri arasında olmaktadır. Eğer katsayı 0 olması halinde sinyallerin birbirine benzemediği görülmektedir. Çalışma kapsamında katsayımızı $0.9 \ll p_{yz} \ll 1$ olursa Y ve dizisi hemen hemen aynı olduğu görülür. Bu şekilde benzer olan enerji ve yapı blokları bu doğrultuda elenecektir. İndirgediğimiz enerji blokları özelleştirilmiş enerji blokları olarak isimlendirilir ve özelleştirilmiş enerji blok bankası adı altında kümelmiştir.

$CEB = \{E_{nie}; n_{ie} = 1, 2, 3, \dots, N_{IE}\}$ şeklinde toplanır. Aynı şekilde enerjisi azaltılmış yapı blokları $CPB = \{P_{nip}; n_{ip} = 1, 2, 3, \dots, N_{IP}\}$ diye toplanır.



Şekil 3.1: CEPB Oluşturma Basamakları Akış Şeması [108]

3.3. ALGORİTMA

3.3.1. Sympes Yöntemi ile Kodlama Algoritması

Girişler:

Kodlanacak video karesinin sütun vektörleri, $VF^{(s)}$

Çerçeve uzunluğu, L_F .

Sayısal Adımlar:

Adım 1: Kodlanacak videonun karelerine bölünmesi.

Adım 2: Her bir video karesinin sütun vektörlerine bölünmesi, $VF^{(s)}$.

Adım 3: Toplam çerçeve sayısının hesaplanması, $N_F = M/L_F$.

Adım 4: Ana çerçeve vektörünün çerçeve vektörlerine bölünmesi, $VF_j^{(s)}$.

Adım 5: Her bir $VF_j^{(s)}$ için $R_j^{(s)}$ özilinti matrisinin hesaplanması.

Adım 6: Her bir $R_j^{(s)}$ için, özdeğerlerinin ($\lambda_{jk}^{(s)}$) ve karşılık gelen özvektörlerinin ($\varphi_{jk}^{(s)}$) hesaplanması.

Adım 7a: Büyükten küçüğe doğru sıralanmış özdeğerlerden en büyük özdeğere $\lambda_{jr}^{(s)}$, $\lambda_{jr}^{(s)} = \max \{\lambda_{j1}^{(s)}, \lambda_{j2}^{(s)}, \dots, \lambda_{jLF}^{(s)}\}$ karşı gelen özvektörün ($S_{jr}^{(s)}$) belirlenmesi ve belirlenen özvektörün çerçeve indisinin $S_{j1}^{(s)}$ olarak değiştirilmesi.

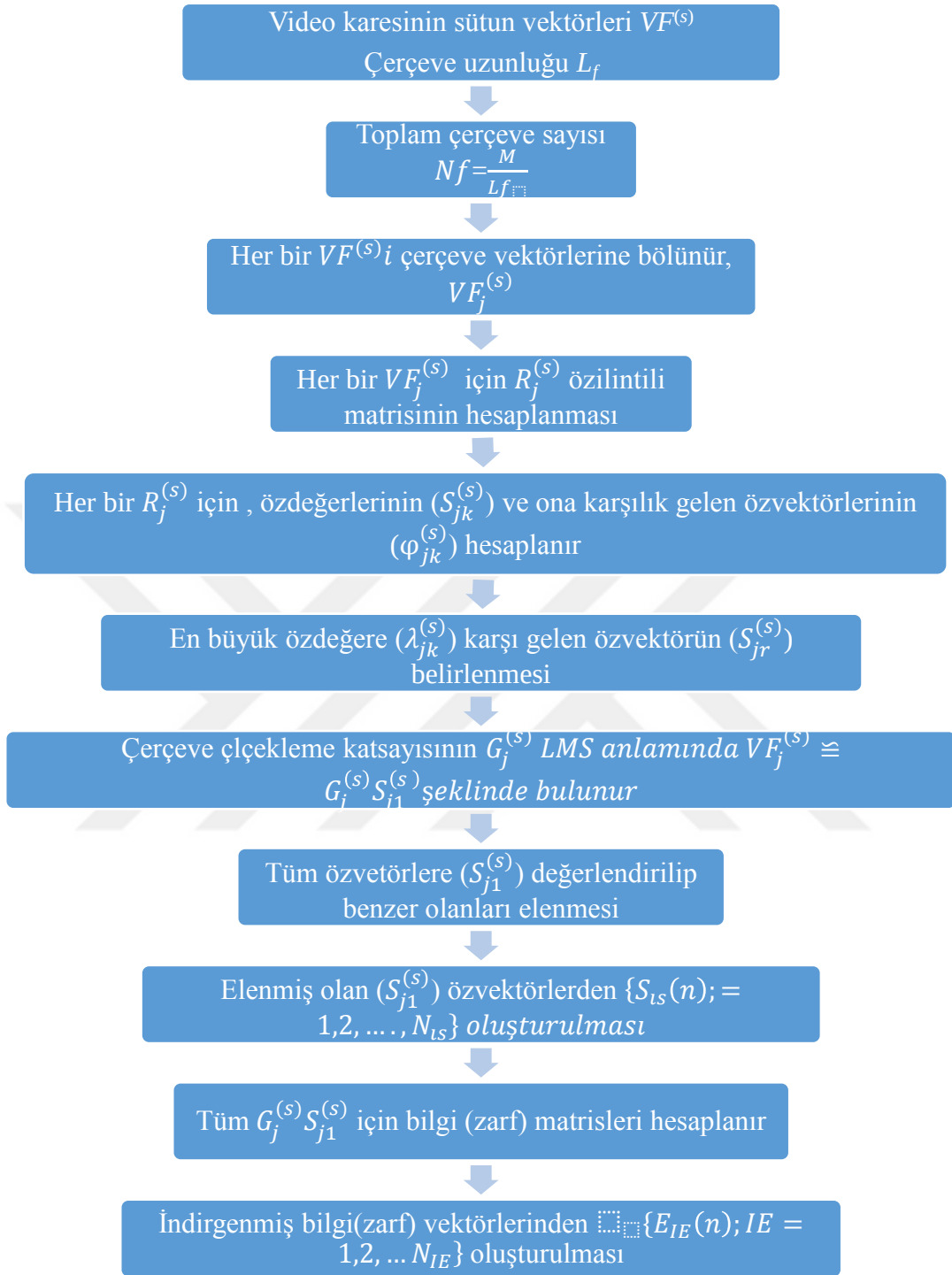
Adım 7b: Çerçeve ölçekleme katsayısının ($G_j^{(s)}$) en küçük kareler anlamında $VF_j^{(s)} \cong G_j^{(s)} S_{j1}^{(s)}$ yaklaşımını elde edecek biçimde bulunması.

Adım 8: Adım 7'nin tüm çerçeveler için ($j = 1, 2, \dots, N_F$) gerçekleşmesi. (Not: Bu adımın sonunda, her bir çerçeve için en yüksek enerjili özvektörler elde edilmiş olmaktadır.)

Adım 9: Adım 8'de elde edilmiş tüm özvektörlerin hızlı ve verimli bir algoritma ile karşılaştırılması ve benzer özvektörlerin elenmesi. (Eleme işlemi için Pearson özilinti sabiti kullanılmıştır.) İndirgenmiş sayıdaki $S_{j1}^{(s)}$ özvektörlerinden önceden tanımlı enerji bloklarının $\{S_{IS}(n); IS = 1, 2, \dots, N_{IS}\}$ oluşturulması. Burada N_{IS} , enerji bloklarındaki özgün enerji vektörlerinin toplam sayısını göstermektedir.

Adım 10: Her bir çerçeve veya Adım 7b'de hesaplanan her bir $G_j^{(s)} S_{j1}^{(s)}$ için köşegenleştirilmiş bilgi (veya zarf) matrisinin köşegen elemanlarının $e_{jr}^{(s)} = \frac{x_{jr}^{(s)}}{G_j^{(s)} S_{j1}^{(s)}}; r = 1, 2, \dots, L_F$ eşitliği yardımıyla hesaplanması.

Adım 11: Adım 9 ile benzer biçimde, elde edilen tüm bilgi vektörlerinin hızlı ve verimli bir algoritma ile karşılaştırılması ve benzer bilgi vektörlerin elenmesi. İndirgenmiş sayıdaki özgün dizilerden bilgi bloklarının $\{E_{IE}(n); IE = 1, 2, \dots, N_{IE}\}$ oluşturulması. Burada N_{IE} , bilgi bloklarındaki özgün bilgi vektörlerinin toplam sayısını göstermektedir. Her bir görüntünün sütun vektörüne ait bilgi ve enerji blokları yukarıdaki adımlar tekrarlanarak üretilir.



Şekil 3.2: SYMPES kodlama kısmı akış diyagramı

3.3.2. SYMPES Kod Çözme Algoritması Aşamaları

Girişler:

Kodlanacak video karesinin sütun vektörleri, $VF^{(s)}$

Çerçeve uzunluğu, L_F .

Kodlama algoritmasının kullanılması ile oluşturulmuş önceden tanımlı enerji (CEB) ve bilgi(yapı) (CPB) blokları.

Sayısal Adımlar:

Adım 1: Modelleneyecek video kareleri görüntü verisinin ($VF(n)$) kodlama algoritmasında olduğu gibi çerçeve vektörlerine bölünmesi.

Adım 2a: Her bir çerçeve için, özgün ve yaklaşılmış çerçeve vektörleri arasındaki uzaklığı veya hatayı ($\delta_{\tilde{IS}} = \|VF_j^{(s)} - G_{\tilde{IS}}^{(s)} S_{\tilde{IS}}^{(s)}\|^2$, $\tilde{IS} = 1, 2, \dots, IS, \dots, N_{IS}$) en aza indiren uygun bir imza vektörünün önceden tanımlı enerji bloklarından çekilmesi. (Bu adım S_{IS} 'nin IS indeksini verir. Bu durumda, $\delta_{IS} = \min \left\{ \|VF_j^{(s)} - G_{\tilde{IS}}^{(s)} S_{\tilde{IS}}^{(s)}\|^2 \right\} = \|VF_j^{(s)} - G_{IS}^{(s)} S_{IS}^{(s)}\|^2$ olur.)

Adım 2b: S_{IS} 'e karşılık gelen IS indis numarasının kaydedilmesi. (Bu durumda, $VF_j^{(s)} \approx G_{IS}^{(s)} S_{IS}^{(s)}$ olur.)

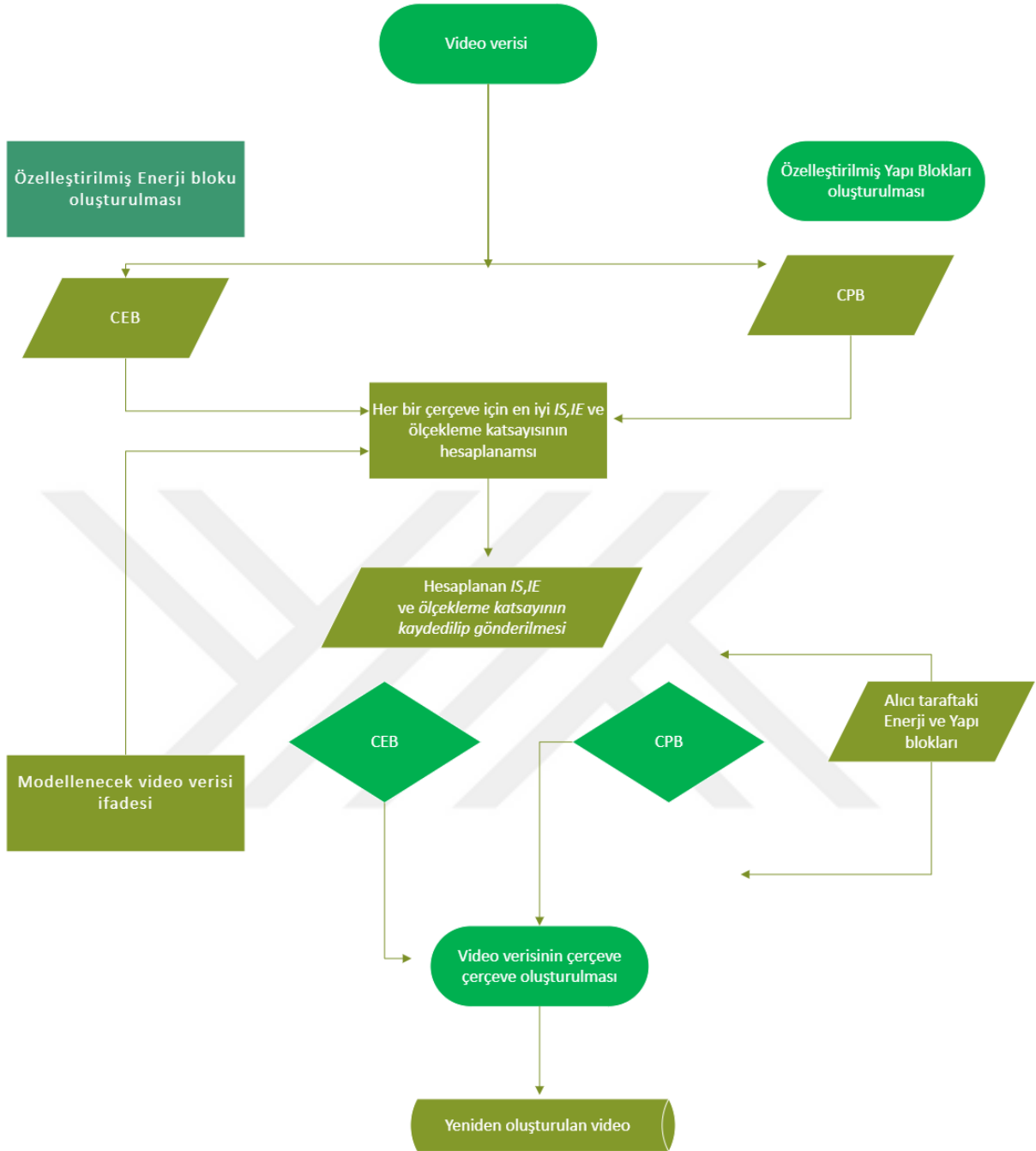
Adım 3a: Hatayı en aza indirecek uygun bir zarf dizisi (veya köşegen bilgi matrisi) E_{IE} 'nin önceden tanımlı bilgi bloklarından çekilmesi. (Bu adım E_{IE} 'nin IE indeksini verir. Bu durumda, $\delta_{IE} = \min \left\{ \|VF_j^{(s)} - G_{IS}^{(s)} E_{IE}^{(s)} S_{IS}^{(s)}\|^2 \right\} = \|VF_j^{(s)} - G_{IS}^{(s)} E_{IE}^{(s)} S_{IS}^{(s)}\|^2$ olur.)

Adım 3b: E_{IE} 'e karşılık gelen IE indis numarasının kaydedilmesi. Bu adımın sonunda, uygun vektörlerin seçilmesiyle en iyi enerji ve bilgi dizileri bulunmuş olmaktadır. Bu durumda, $VF_j^{(s)} \approx G_{IS}^{(s)} E_{IE}^{(s)} S_{IS}^{(s)}$ ifadesi, j. nci VF'nin E_{IE} ve S_{IS} örüntüleri ile en iyi şekilde tanımlanabileceğini göstermektedir.)

Adım 4: Elde edilen sabitlenmiş E_{IE} ve S_{IS} vektörlerini kullanarak, bir bütünsel en küçük (global minimum) hata değeri bulacak biçimde yeni çerçeve katsayılarının ($G_j^{(s)} = (E_{IE} S_{IS})^T VF_j^{(s)} / (E_{IE} S_{IS})^T E_{IE} S_{IS}$) hesaplanması. $G_j^{(s)}$ çerçeve katsayıları $VF_j^{(s)}$ ile $G_{IS}^{(s)} E_{IE}^{(s)} S_{IS}^{(s)}$ vektörleri arasındaki uzaklığı LMS anlamında en aza indirir. Bu durumda, global minimum hata $\delta_{global} = \|VF_j^{(s)} - G_{IS}^{(s)} E_{IE}^{(s)} S_{IS}^{(s)}\|^2$ eşitliği ile ifade edilir.

Adım 5: Yukarıdaki tüm adımlar, modelleneyecek video karelerine ait görüntü verisinin her bir çerçevesinin oluşturulması için sırayla yinelenir.

Adım 6: Görüntü sekansından videonun üretilmesi.



Şekil 3.3: SYMPES Kod Çözme Kısmı Akış Diyagramı

3.3.3. SYMPES Blok Görüntü Yaklaşımı

Blok görüntü yaklaşımı Sütun vektör yaklaşımından farklı olarak görüntü karesi çerçeve boyutuna göre karesel matrislere bölünerek bloklara ayrılır. Blok görüntü yaklaşımı da yine sütun vektörüne dönüştürülür. Sütun vektörü yaklaşımında yapılan işlemler burada tekrarlanır. Sütun vektör yaklaşımı da $L_f = 8$, Blok görüntü yaklaşımında ise $M \times M = 8 \times 8$ 'lik bloklara ayırdık.

3.4. MATERYAL

3.4.1. Evrensel Küme Oluşturulurken Kullanılan Görüntüler

Bu tez kapsamında görüntüler ve oluşturulacak videolar tez yazarının çekmiş olduğu videolar ve fotoğraflardan oluşmuştur. Sympes yöntemi ile yeniden oluşturulacak video için evrensel küme oluştururken iki farklı şekilde görüntü kullanıldı:

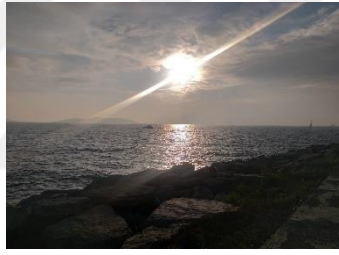
- Videodan içeriğinden bağımsız kendi seçtiğimiz görüntü
- Sympes ile yeniden oluşturulacak videonun herhangi bir frame görüntüsü seçilmiştir.

3.4.2. Evrensel Küme İçin Oluşturulacak Videonun İçeriğinden Bağımsız Görüntüler ve Özellikleri

Tez kapsamında Sympes yöntemi ile oluşturulacak videonun evrensel kümesini oluşturmak için tez yazarının çektiği oluşturulacak videodan bağımsız veri tabanı görüntüleri şekil 3.4, şekil 3.5 ve şekil 3.6’da gösterilmiştir.



Şekil 3.4: Vtg-1



Şekil 3.5: Vtg-2



Şekil 3.6: Vtg-3

Tablo 3.1: Evrensel kümeyi oluşturacak oluşturulacak videodan bağımsız veri tabanı görüntülerinin özellikleri

Veri Tabanı Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Vtg-1	2048px	1536px	96dpi	96dpi	24	JPG	942Kb
Vtg-2	2400px	1800px	96dpi	96dpi	24	JPG	339Kb
Vtg-3	960px	1280px	72dpi	72dpi	24	JPG	387Kb

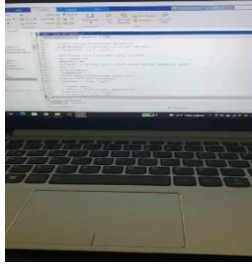
Px: piksel

Dpi: dots per inch

Kb: kilobyte

3.4.3. Evrensel Küme İçin Oluşturulacak Videodan Rasgele Seçilen Görüntüler ve Özellikleri

Tez kapsamında Sympes yöntemi ile yeniden oluşturulacak videonun evrensel kümesini oluşturmak için tez yazarının çektiği oluşturulacak videodan rasgele alınan veri tabanı görüntüleri şekil 3.7, şekil 3.8 ve şekil 3.9'da gösterilmiştir.



Şekil 3.7: Vtg-4



Şekil 3.8: Vtg-5



Şekil 3.9: Vtg-6

Tablo 3.2: Evrensel kümeyi oluşturacak oluşturulacak videodan alınan veri tabanı görüntülerinin özellikleri

Veri Tabanı Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Vtg-4	480px	848px	96dpi	96dpi	24	JPG	35.4Kb
Vtg-5	1280px	720px	96dpi	96dpi	24	JPG	22.4Kb
Vtg-6	1920px	1080px	96dpi	96dpi	24	JPG	167Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

3.4.4. SYMPES Yöntemi ile Yeniden Oluşturulacak Videoların Özellikleri

Tez kapsamında Sympes yöntemi ile yeniden oluşturulacak videoyu oluşturmak için tez yazarının çektiği videoların özellikleri tablo 3.3 verilmiştir.

Tablo 3.3: Sympes Yöntemi ile Yeniden Oluşturulacak Videoların Özellikleri

Video Verisi	Uzunluk	Çerçeve Genişliği	Çerçeve Yüksekliği	Veri Hızı	Toplam Bit Hızı	Resim Kare Hızı	Türü
Vv-1	00:00:01	1920	1080	9207kb/sn	9412kb/sn	30kare/sn	MP4
Vv-2	00:00:02	848	480	2138kb/sn	2197kb/sn	30kare/sn	MP4
Vv-3	00:00:04	1280	720	2819kb/sn	3019kb/sn	30kare/sn	MP4

Kb: kilobyte

Sn: saniye

4. BULGULAR

Tezin bu bölümünde uygulamamızı iki bölümde yaptık. İlk ortaya koyduğumuz yöntemi yani her bir video karesinin sütun vektörlerine bölünmesi ve bu yöntemi karşılaştırmak üzere yine sympes yöntemini kullanarak bu sefer her bir video karesini 8x8'lik bloklara ayırdık ve iki yöntemi karşılaştırdık. İlk 12 uygulamamız her bir video karesini 8'lik sütun vektörlerine ayırma yöntemiyle yaptığımız uygulamalar son 12 uygulamamız ise her bir video karesini bloklarına ayırdığımız uygulamalardır. Test edilen videoların her bir görüntü karesi için ve ayrıca tüm videoya ait psnr (Peak Signal Noise Ratio) ve mse (Mean Squared Error) değerleri açısından incelenmiştir. Bu incelemeler sonucunda orijinal ve yeniden oluşturulmuş video kareleri hakkında bilgi sahibi olundu. Bu bölümde deney seti için farklı özellikteki 6 görüntüden evrensel küme oluşturup ve bu evrensel kümelerden farklı özellikteki videolar oluşturmaktır. Bu bölümde Intel Core İ7-8550U 1.80 GHz teknik özelliklerini içeren bilgisayar ile sonuçlara ulaşılmıştır.

4.1. *Vtg-1* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-1* SONUÇLARI

Vtg-1 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden oluşturulan *Vv-1* isimli tablo 3.3 verilen özelliklere sahip videonun 1,11 ve 21. kareleri ve yeniden oluşturulan 1,11 ve 21.kareleri şekil 4.1, şekil 4.2, şekil 4.3, şekil 4.4, şekil 4.5, şekil 4.6 ve bunların özellikleri tablo 4.1 ve tablo 4.2 verilmiştir. Ayrıca tablo 4.3 ve tablo 4.4'te sonuçların psnr ve mse değerleri verilmiş ve şekil 4.7'de her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



Şekil 4.1: *F-1*



Şekil 4.2: *F-11*



Şekil 4.3: *F-21*



Şekil 4.4: Yof-1



Şekil 4.5: Yof-11



Şekil 4.6: Yof-21

Tablo 4.1: Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulacak Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-1	1920px	1080px	96dpi	96dpi	24	JPG	167Kb
F-11	1920px	1080px	96dpi	96dpi	24	JPG	154Kb
F-21	1920px	1080px	96dpi	96dpi	24	JPG	158Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.2: Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulan Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-1	1920px	1080px	96dpi	96dpi	24	JPG	164Kb
Yof-11	1920px	1080px	96dpi	96dpi	24	JPG	156Kb
Yof-21	1920px	1080px	96dpi	96dpi	24	JPG	160Kb

Px: piksel

Dpi: dots per inch

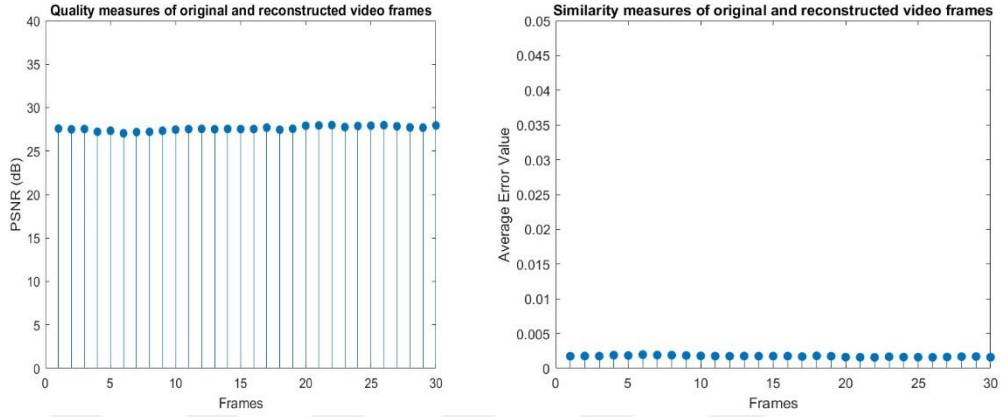
Kb: kilobyte

Tablo 4.3: Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulacak Vv-1 Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
F-1-Yof-1	27.5918	0.0017
F-11-Yof-11	27.5374	0.0018
F-21-Yof-21	27.9528	0.0016

Tablo 4.4: *Vtg-1* Evrensel Kümesi ile Yeniden Üretilen *Vv-1* Video Karelerinin Ortalama Psnr ve Mse Değerleri

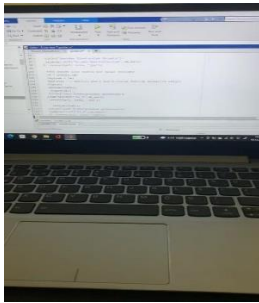
Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-1</i>	27.6058	0.0017



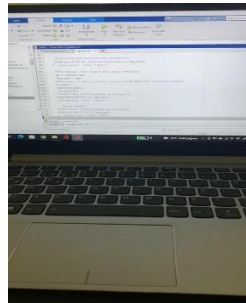
Şekil 4.7: *Vtg-1* Evrensel Kümesi ile Yeniden Üretilen *Vv-1* için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.2. *Vtg-1* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-2* SONUÇLARI

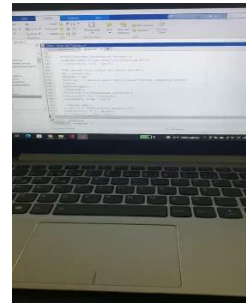
Vtg-1 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden oluşturulan *Vv-2* isimli tablo 3.3 gösterilen özelliklere sahip videonun 33,44 ve 87. kareleri ve yeniden oluşturulan 33,44 ve 87.kareleri şekil 4.8, şekil 4.9, şekil 4.10, şekil 4.11, şekil 4.12, şekil 4.13 ve bunların özellikleri tablo 4.5 ve tablo 4.6 verilmiştir. Ayrıca tablo 4.7 ve tablo 4.8’de sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.7’de her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



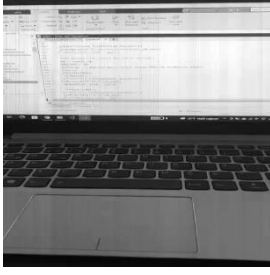
Şekil 4.8: *F-33*



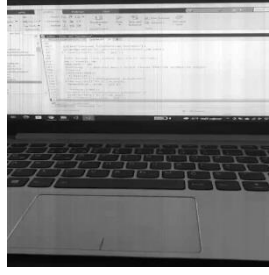
Şekil 4.9: *F-44*



Şekil 4.10: *F-87*



Şekil 4.11: Yof-33



Şekil 4.12: Yof-44



Şekil 4.13: Yof-87

Tablo 4.5: Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulacak Vv-2 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-33	480px	848px	96dpi	96dpi	24	JPG	34.9Kb
F-44	480px	848px	96dpi	96dpi	24	JPG	37.7Kb
F-87	480px	848px	96dpi	96dpi	24	JPG	37.9Kb

Px: piksel
Dpi: dots per inch
Kb: kilobyte

Tablo 4.6: Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulan Vv-2 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-33	480px	848px	96dpi	96dpi	24	JPG	37.8Kb
Yof-44	480px	848px	96dpi	96dpi	24	JPG	39.2Kb
Yof-87	480px	848px	96dpi	96dpi	24	JPG	41.3Kb

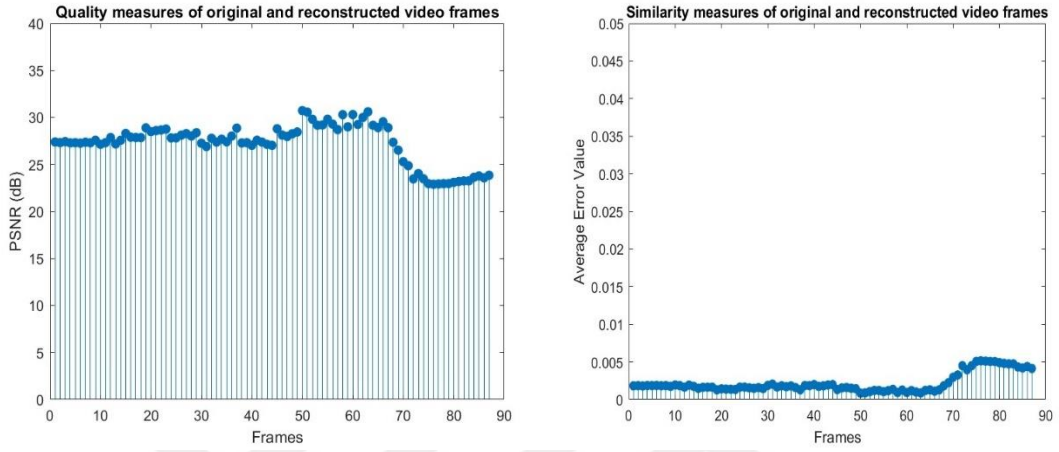
Px: piksel
Dpi: dots per inch
Kb: kilobyte

Tablo 4.7: Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulacak Vv-2 Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
F-33-Yof-33	27.3809	0.0018
F-44-Yof-44	27.0336	0.0020
F-87-Yof-87	23.8448	0.0041

Tablo 4.8: *Vtg-1* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-2</i>	27.2563	0.0022



Şekil 4.14: *Vtg-1* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.3. *Vtg-1* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-3* SONUÇLARI

Vtg-1 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden oluşturulan *Vv-3* isimli tablo 3.3 gösterilen özelliklere sahip videonun 100,110 ve 123. kareleri ve yeniden oluşturulan 100,110 ve 123. kareleri şekil 4.15, şekil 4.16, şekil 4.17, şekil 4.18, şekil 4.19, şekil 4.20 ve bunların özellikleri tablo 4.9 ve tablo 4.10 verilmiştir. Ayrıca tablo 4.11 ve tablo 4.12’de sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.21’de her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



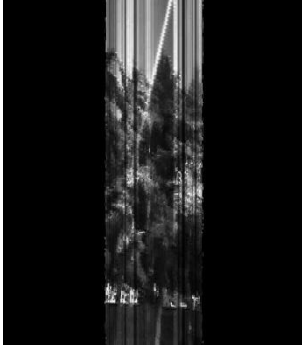
Şekil 4.15: *F-100*



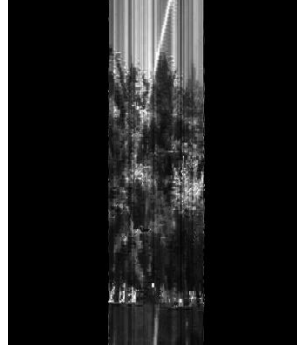
Şekil 4.16: *F-110*



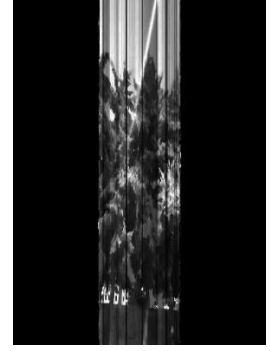
Şekil 4.17: *F-123*



Şekil 4.18: Yof-100



Şekil 4.19: Yof-110



Şekil 4.20: Yof-123

Tablo 4.9: Sympes Yöntemi ile Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulacak Vv-3 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-100	1280px	720px	96dpi	96dpi	24	JPG	26.6Kb
F-110	1280px	720px	96dpi	96dpi	24	JPG	26.3Kb
F-123	1280px	720px	96dpi	96dpi	24	JPG	26.7Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.10: Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulan Vv-3 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-33	1280px	720px	96dpi	96dpi	24	JPG	46.9Kb
Yof-44	1280px	720px	96dpi	96dpi	24	JPG	44.3Kb
Yof-87	1280px	720px	96dpi	96dpi	24	JPG	45.5Kb

Px: piksel

Dpi: dots per inch

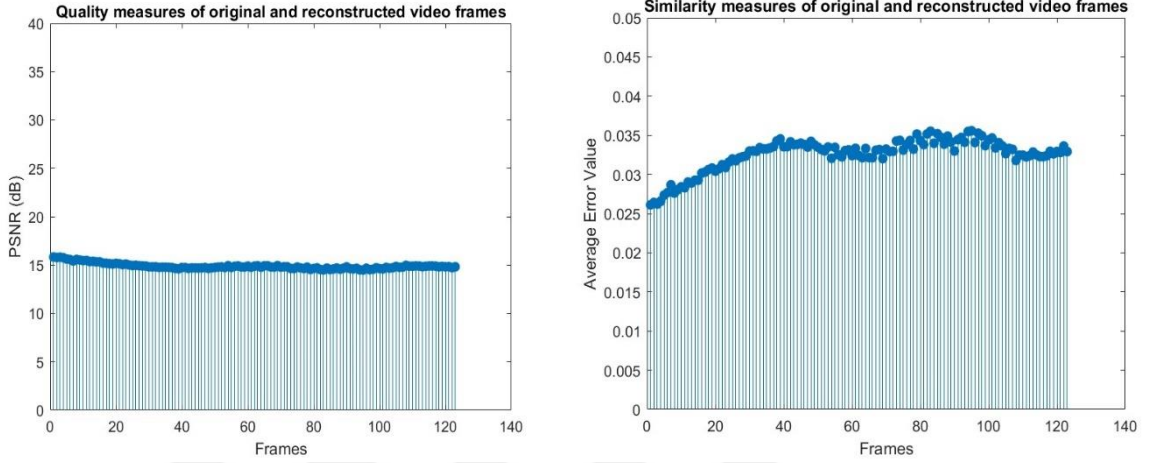
Kb: kilobyte

Tablo 4.11: Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulan Vv-3 Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
F-100-Yof-100	14.6318	0.0344
F-110-Yof-110	14.8873	0.0325
F-123-Yof-123	14.8249	0.0329

Tablo 4.12: *Vtg-1* Evrensel Kümesi ile Yeniden Üretilen *Vv-3* Video Karelerinin Ortalama Psnr ve Mse Değerleri

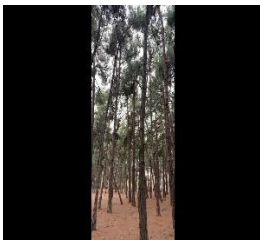
Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-3</i>	14.8845	0.0325



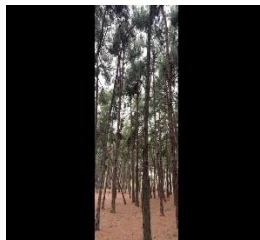
Şekil 4.21: *Vtg-1* Evrensel Kümesi ile Yeniden Üretilen *Vv-3* için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.4. *Vtg-2* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-1* SONUÇLARI

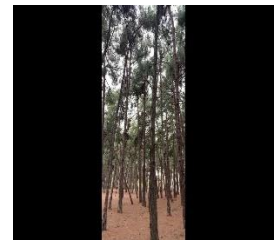
Vtg-2 ile oluşturulan evrensel küme ile Sympes yöntemi ile yeniden üretilen *Vv-1* isimli tablo 3.3 gösterilen özelliklere sahip videonun 4,18 ve 28. kareleri ve yeniden oluşturulan 4,18 ve 28. kareleri şekil 4.22, şekil 4.23, şekil 4.24, şekil 4.25, şekil 4.26, şekil 4.27 ve bunların özellikleri tablo 4.13 ve tablo 4.14’te verilmiştir. Ayrıca tablo 4.15 ve tablo 4.16’da sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.28’de her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



Şekil 4.22: *F-4*



Şekil 4.23: *F-18*



Şekil 4.24: *F-28*



Şekil 4.25: Yof-4



Şekil 4.26: Yof-18



Şekil 4.27: Yof-28

Tablo 4.13: Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulacak Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-4	1920px	1080px	96dpi	96dpi	24	JPG	161Kb
F-18	1920px	1080px	96dpi	96dpi	24	JPG	157Kb
F-28	1920px	1080px	96dpi	96dpi	24	JPG	155Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.14: Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulan Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-4	1920px	1080px	96dpi	96dpi	24	JPG	157Kb
Yof -18	1920px	1080px	96dpi	96dpi	24	JPG	157Kb
Yof -28	1920px	1080px	96dpi	96dpi	24	JPG	156Kb

Px: piksel

Dpi: dots per inch

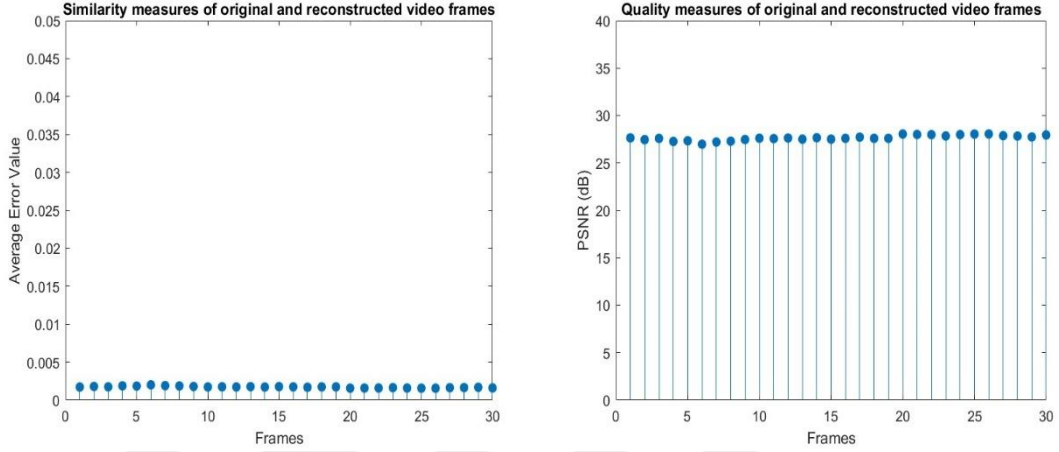
Kb: kilobyte

Tablo 4.15: Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulan Vv-1 Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
F-4-Yof-4	27.2717	0.0019
F-18-Yof-18	27.5906	0.0017
F-28-Yof-28	27.8379	0.0016

Tablo 4.16: *Vtg-2* Evrensel Kümesi ile Yeniden Üretilen *Vv-1* Video Karelerinin Ortalama Psnr ve Mse Değerleri

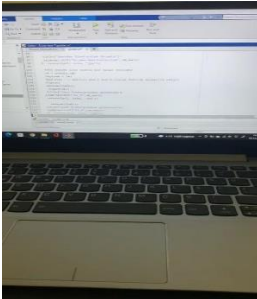
Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-1</i>	27.6337	0.0017



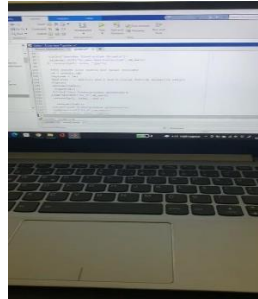
Şekil 4.28: *Vtg-2* Evrensel Kümesi ile Yeniden Üretilen *Vv-1* için her bir Karesinin Mse ve Psnr Değerleri Grafiği

4.5. *Vtg-2* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-2* SONUÇLARI

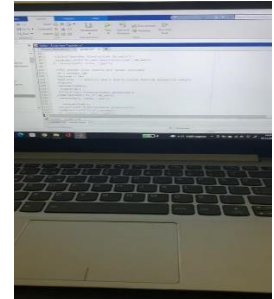
Vtg-2 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen *Vv-2* isimli tablo 3.3 gösterilen özelliklere sahip videonun 5,45 ve 85. kareleri ve yeniden oluşturulan 5,45 ve 85. kareleri şekil 4.29, şekil 4.30, şekil 4.31, şekil 4.32, şekil 4.33, şekil 4.34 ve bunların özellikleri tablo 4.17 ve tablo 4.18’de verilmiştir. Ayrıca tablo 4.19 ve tablo 4.20’de sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.35’te her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



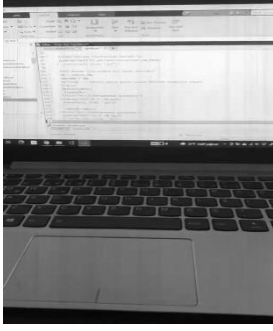
Şekil 4.29: *F-5*



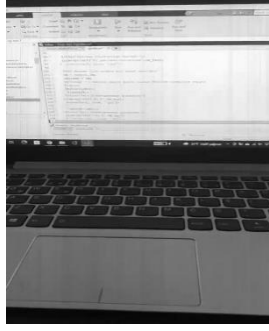
Şekil 4.30: *F-45*



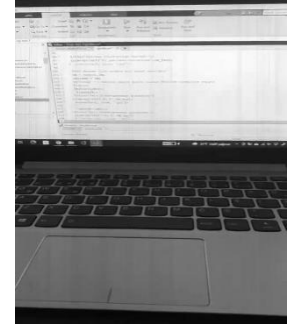
Şekil 4.31: *F-85*



Şekil 4.32: Yof-5



Şekil 4.33: Yof-45



Şekil 4.34: Yof-85

Tablo 4.17: Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulacak Vv-2 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-5	480px	848px	96dpi	96dpi	24	JPG	35.1Kb
F-45	480px	848px	96dpi	96dpi	24	JPG	37.4Kb
F-85	480px	848px	96dpi	96dpi	24	JPG	38.6Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.18: Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulan Vv-2 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-5	480px	848px	96dpi	96dpi	24	JPG	36.6Kb
Yof-45	480px	848px	96dpi	96dpi	24	JPG	39.3Kb
Yof-85	480px	848px	96dpi	96dpi	24	JPG	41.2Kb

Px: piksel

Dpi: dots per inch

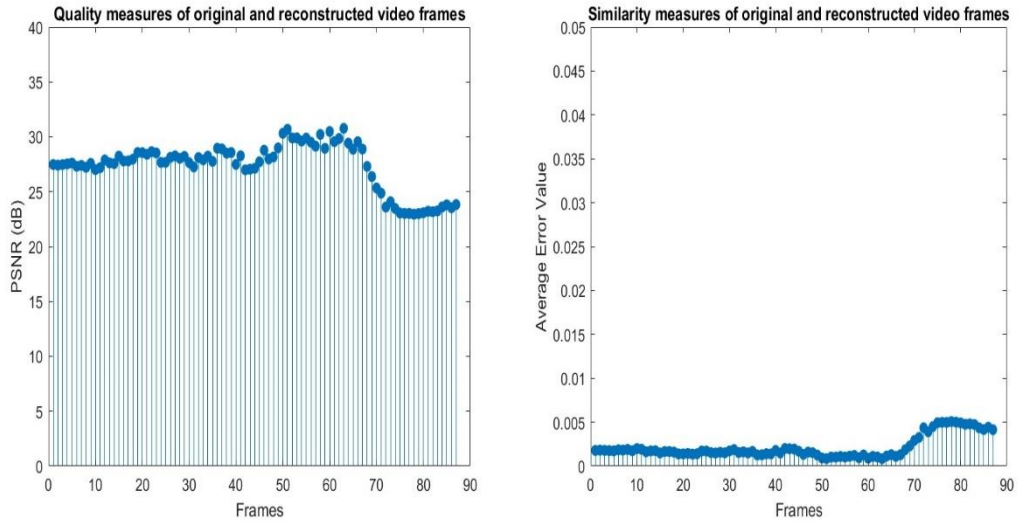
Kb: kilobyte

Tablo 4.19: Sympes Yöntemi ile *Vtg-2* Evrensel Kümesiyle Yeniden oluşturulan *Vv-2* Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
<i>F-5-Yof-5</i>	27.6092	0.0017
<i>F-45-Yof-45</i>	27.7301	0.0017
<i>F-85-Yof-85</i>	23.8077	0.0042

Tablo 4.20: *Vtg-2* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-2</i>	27.3640	0.0021



Şekil 4.35: *Vtg-2* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.6. *Vtg-2* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-3* SONUÇLARI

Vtg-2 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen *Vv-3* isimli tablo 3.3 gösterilen özelliklere sahip videonun 7,97 ve 117. kareleri ve yeniden oluşturulan 7,97 ve 117. kareleri şekil 4.36, şekil 4.37, şekil 4.38, şekil 4.39, şekil 4.40, şekil 4.41 ve bunların özellikleri tablo 4.21 ve tablo 4.22 verilmiştir. Ayrıca tablo 4.23 ve tablo

4.24'te sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.42'te her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



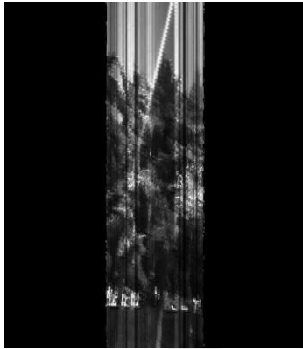
Şekil 4.36: F-7



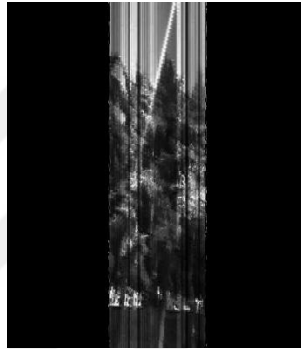
Şekil 4.37: F-97



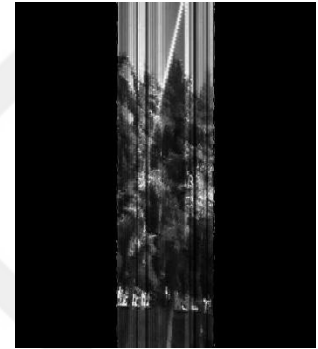
Şekil 4.38: F-117



Şekil 4.39: Yof-7



Şekil 4.40: Yof-97



Şekil 4.41: Yof-117

Tablo 4.21: Sympes Yöntemi ile Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulacak Vv-3 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-7	1280px	720px	96dpi	96dpi	24	JPG	22.5Kb
F-97	1280px	720px	96dpi	96dpi	24	JPG	26.7Kb
F-117	1280px	720px	96dpi	96dpi	24	JPG	26.6Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.22: Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulan Vv-3 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-7	1280px	720px	96dpi	96dpi	24	JPG	46.5Kb
Yof-97	1280px	720px	96dpi	96dpi	24	JPG	48.2Kb
Yof-117	1280px	720px	96dpi	96dpi	24	JPG	48.6Kb

Px: piksel

Dpi: dots per inch

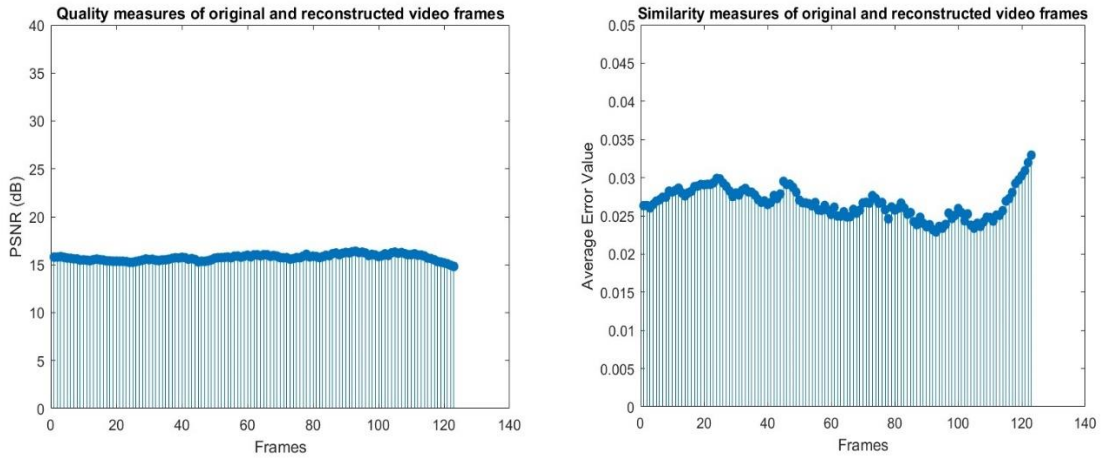
Kb: kilobyte

Tablo 4.23: Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulan Vv-3 Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
F-7-Yof-7	15.6552	0.0274
F-97-Yof-97	15.9588	0.0254
F-117-Yof-117	15.5194	0.0281

Tablo 4.24: Vtg-2 Evrensel Kümesi ile Yeniden Üretilen Vv-3 Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
Vv-3	15.7450	0.0267



Şekil 4.42: Vtg-2 Evrensel Kümesi ile Yeniden Üretilen Vv-3 için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.7. Vtg-3 İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN Vv-1 SONUÇLARI

Vtg-3 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen Vv-1 isimli tablo 3.3 gösterilen özelliklere sahip videonun 2,20 ve 30. kareleri ve yeniden oluşturulan 2,20 ve 30. kareleri şekil 4.43, şekil 4.44, şekil 4.45, şekil 4.46, şekil 4.47, şekil 4.48 ve bunların özellikleri tablo 4.25 ve tablo 4.26’da verilmiştir. Ayrıca tablo 4.27 ve tablo 4.28’de sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.43’te her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



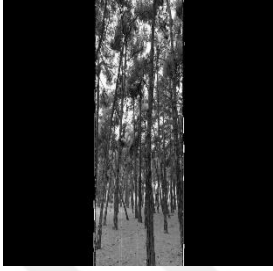
Şekil 4.43: F-2



Şekil 4.44: F-20



Şekil 4.45: F-30



Şekil 4.46: Yof-2



Şekil 4.47: Yof-20



Şekil 4.48: Yof-30

Tablo 4.25: Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulacak Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-2	1920px	1080px	96dpi	96dpi	24	JPG	164Kb
F-20	1920px	1080px	96dpi	96dpi	24	JPG	158Kb
F-30	1920px	1080px	96dpi	96dpi	24	JPG	155Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.26: Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulan Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-2	1920px	1080px	96dpi	96dpi	24	JPG	157Kb
Yof -20	1920px	1080px	96dpi	96dpi	24	JPG	157Kb
Yof -30	1920px	1080px	96dpi	96dpi	24	JPG	156Kb

Px: piksel

Dpi: dots per inch

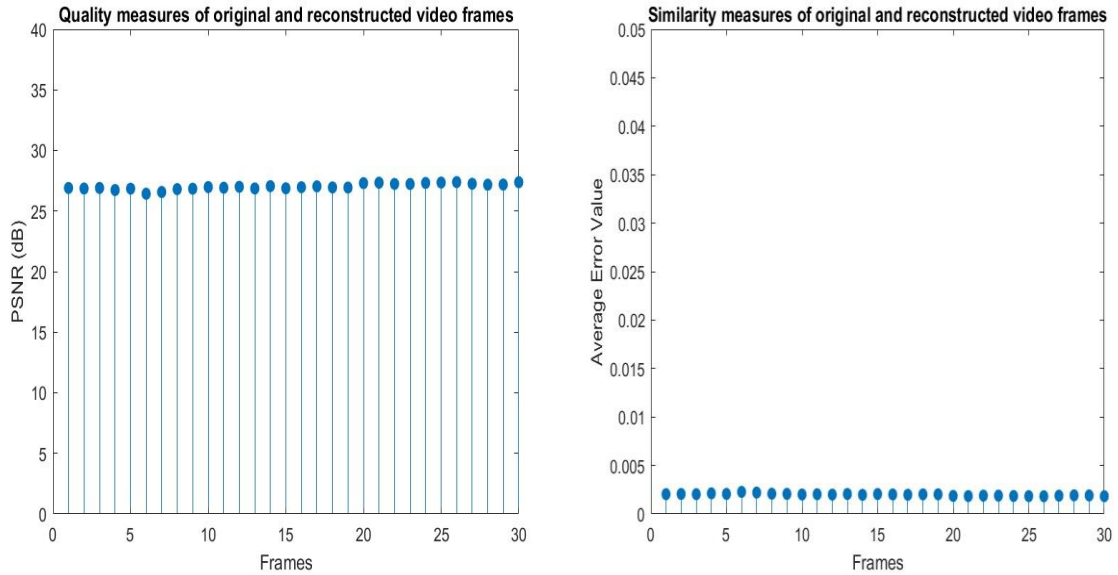
Kb: kilobyte

Tablo 4.27: Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulan Vv-1 Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
F-2-Yof-2	26.8593	0.0021
F-20-Yof-20	27.3072	0.0019
F-30-Yof-30	27.3806	0.0018

Tablo 4.28: Vtg-3 Evrensel Kümesi ile Yeniden Üretilen Vv-1 Video Karelerinin Ortalama Psnr ve Mse Değerleri

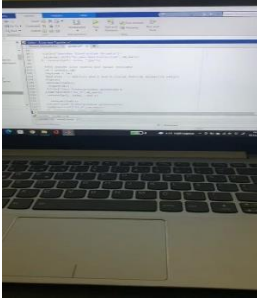
Video Verisi	Ortalama PSNR	Ortalama MSE
Vv-1	27.0025	0.0020



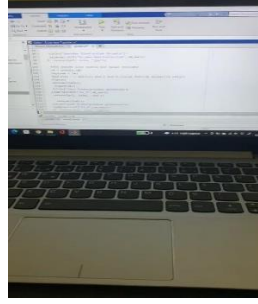
Şekil 4.49: Vtg-3 Evrensel Kümesi ile Yeniden Üretilen Vv-1 için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.8. Vtg-3 İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN Vv-2 SONUÇLARI

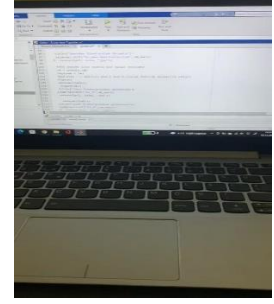
Vtg-3 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen Vv-2 isimli tablo 3.3 gösterilen özelliklere sahip videonun 34,54 ve 74. kareleri ve yeniden oluşturulan 34,54 ve 74. kareleri şekil 4.50, şekil 4.51, şekil 4.52, şekil 4.53, şekil 4.54, şekil 4.55 ve bunların özellikleri tablo 4.29 ve tablo 4.30’da verilmiştir. Ayrıca tablo 4.31 ve tablo 4.32’de sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.56’da her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



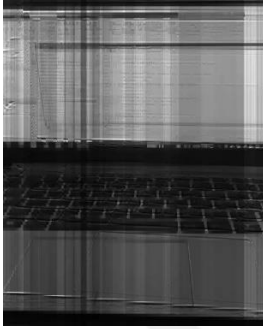
Şekil 4.50: F-34



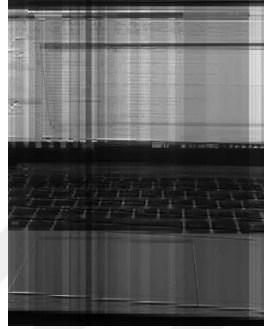
Şekil 4.51: F-54



Şekil 4.52: F-74



Şekil 4.53: Yof-34



Şekil 4.54: Yof-34



Şekil 4.55: Yof-74

Tablo 4.29: Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulacak Vv-2 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-34	480px	848px	96dpi	96dpi	24	JPG	36.1Kb
F-34	480px	848px	96dpi	96dpi	24	JPG	36.1Kb
F-74	480px	848px	96dpi	96dpi	24	JPG	37.8Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.30: Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulan Vv-2 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-34	480px	848px	96dpi	96dpi	24	JPG	36.6Kb
Yof-54	480px	848px	96dpi	96dpi	24	JPG	35.6Kb
Yof-74	480px	848px	96dpi	96dpi	24	JPG	38.5Kb

Px: piksel

Dpi: dots per inch

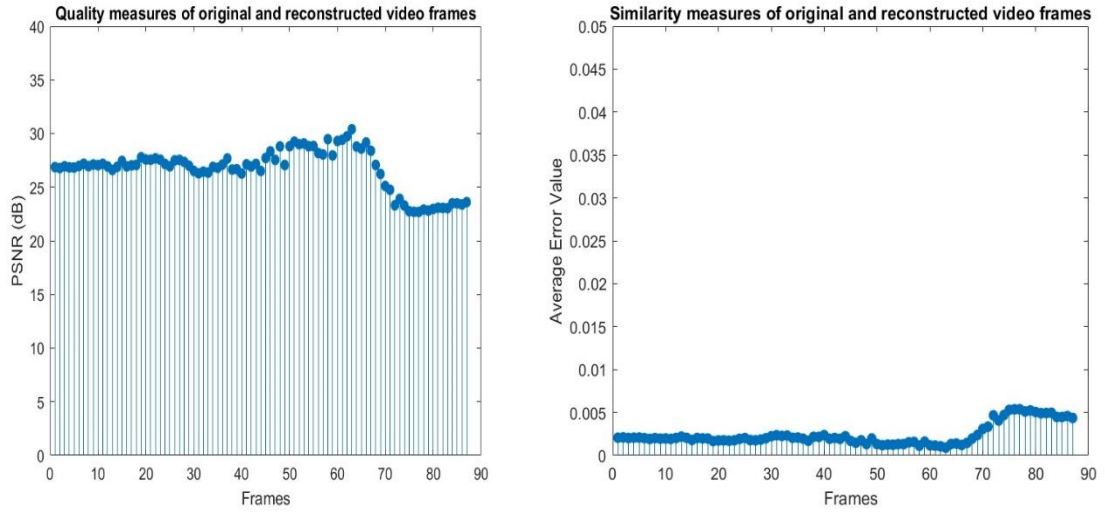
Kb: kilobyte

Tablo 4.31: Sympes Yöntemi ile *Vtg-3* Evrensel Kümesiyle Yeniden oluşturulan *Vv-2* Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
<i>F-34-Yof-34</i>	26.8660	0.0020
<i>F-54-Yof-54</i>	28.8115	0.0013
<i>F-74-Yof-74</i>	23.2768	0.0047

Tablo 4.32: *Vtg-3* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-2</i>	26.7015	0.0024



Şekil 4.56: *Vtg-3* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.9. *Vtg-3* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-3* SONUÇLARI

Vtg-3 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen *Vv-3* isimli tablo 3.3 gösterilen özelliklere sahip videonun 42,102 ve 122. kareleri ve yeniden oluşturulan 42,102 ve 122. kareleri şekil 4.57, şekil 4.58, şekil 4.59, şekil 4.60, şekil 4.61, şekil 4.62 ve bunların özellikleri tablo 4.33 ve tablo 4.34'te verilmiştir. Ayrıca tablo 4.35 ve tablo 4.36'da sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.63'te her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



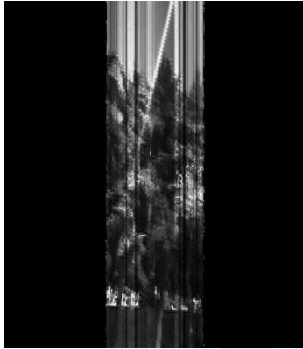
Şekil 4.57: F-42



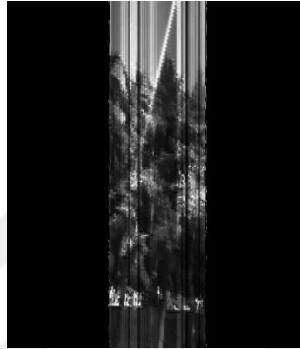
Şekil 4.58: F-102



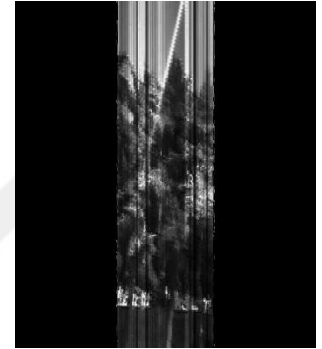
Şekil 4.59: F-122



Şekil 4.60: Yof-42



Şekil 4.61: Yof-42



Şekil 4.62: Yof-42

Tablo 4.33: Sympes Yöntemi ile Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulacak Vv-3 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-42	1280px	720px	96dpi	96dpi	24	JPG	22.5Kb
F-102	1280px	720px	96dpi	96dpi	24	JPG	26.8Kb
F-122	1280px	720px	96dpi	96dpi	24	JPG	26.8Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.34: Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulan Vv-3 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-42	1280px	720px	96dpi	96dpi	24	JPG	46.8Kb
Yof-102	1280px	720px	96dpi	96dpi	24	JPG	47.1Kb
Yof-122	1280px	720px	96dpi	96dpi	24	JPG	46.6Kb

Px: piksel

Dpi: dots per inch

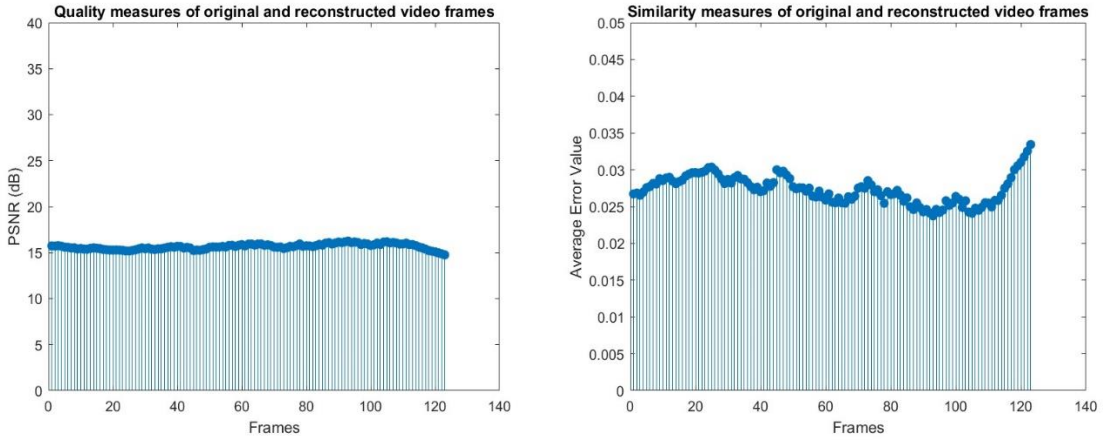
Kb: kilobyte

Tablo 4.35: Sympes Yöntemi ile *Vtg-3* Evrensel Kümesiyle Yeniden oluşturulan *Vv-3* Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
<i>F-42-Yof-42</i>	15.4954	0.0282
<i>F-102-Yof-102</i>	16.0395	0.0249
<i>F-122-Yof-122</i>	14.8760	0.0325

Tablo 4.36: *Vtg-3* Evrensel Kümesi ile Yeniden Üretilen *Vv-3* Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-3</i>	15.6419	0.0273



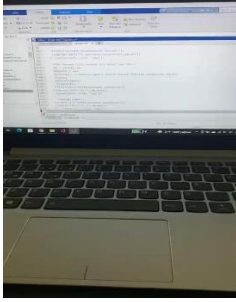
Şekil 4.63: *Vtg-3* Evrensel Kümesi ile Yeniden Üretilen *Vv-3* için her bir Karesinin Psnr ve Mse Değerleri Grafiği

Çalışmada bu zamana kadarki uygulamalar videonun karelerinden bağımsız farklı görüntüler ile oluşturulan evrensel küme ile yapılmıştır. Şimdi sonuçları aşağıda belirtilen uygulamalar ise oluşturulacak videodan rasgele seçilen bir video karesi alınıp ve bu seçilen video karesinin evrensel küme kabul edilmesiyle yapılmıştır.

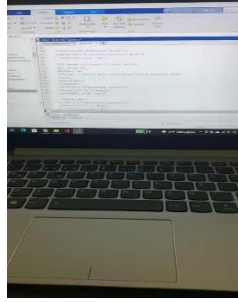
4.10. *Vtg-4* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-2* SONUÇLARI

Vtg-4 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen *Vv-2* isimli tablo 3.3 gösterilen özelliklere sahip videonun 8,17 ve 69. kareleri ve yeniden oluşturulan 8,17 ve 69. kareleri şekil 4.64, şekil 4.65, şekil 4.66, şekil 4.67, şekil 4.68, şekil 4.69 ve bunların özellikleri tablo 4.37 ve tablo 4.38’de verilmiştir. Ayrıca tablo 4.39 ve tablo

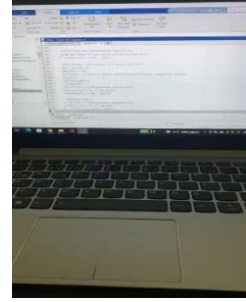
4.40'ta sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.70'te her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



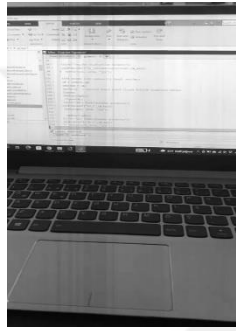
Şekil 4.64: F-8



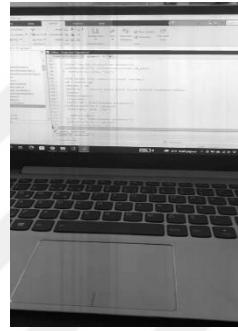
Şekil 4.65: F-17



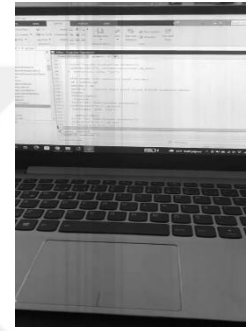
Şekil 4.66: F-69



Şekil 4.67: Yof-8



Şekil 4.68: Yof-17



Şekil 4.69: Yof-69

Tablo 4.37: Sympes Yöntemi ile Vtg-4 Evrensel Kümesiyle Yeniden oluşturulacak Vv-2 Videosundan Alınan Video Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-8	480px	848px	96dpi	96dpi	24	JPG	35.6Kb
F-17	480px	848px	96dpi	96dpi	24	JPG	38.0Kb
F-69	480px	848px	96dpi	96dpi	24	JPG	37.2Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.38: Sympes Yöntemi ile Vtg-4 Evrensel Kümesiyle Yeniden oluşturulan Vv-2 Videosundan Alınan Video Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-8	480px	848px	96dpi	96dpi	24	JPG	35.9Kb
Yof-17	480px	848px	96dpi	96dpi	24	JPG	38.9Kb
Yof-69	480px	848px	96dpi	96dpi	24	JPG	37.5Kb

Px: piksel

Dpi: dots per inch

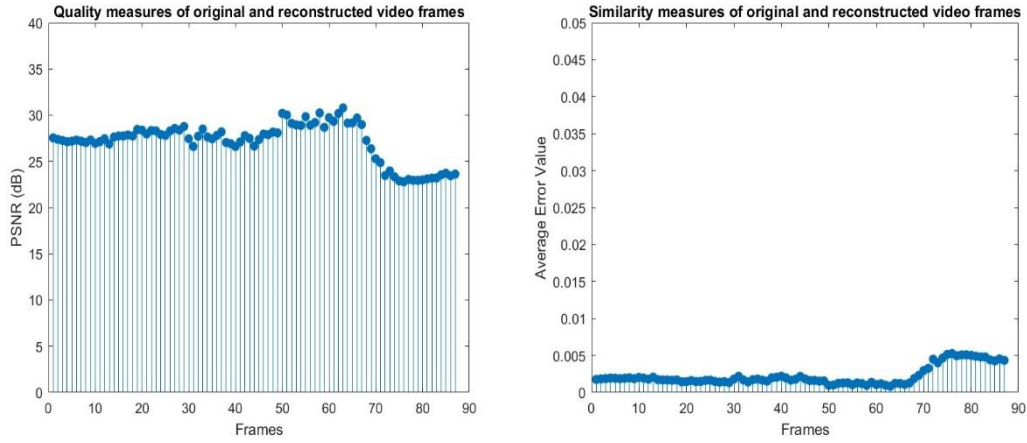
Kb: kilobyte

Tablo 4.39: Sympes Yöntemi ile *Vtg-4* Evrensel Kümesiyle Yeniden oluşturulan *Vv-2* Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
<i>F-8-Yof-8</i>	27.0463	0.0020
<i>F-17-Yof-17</i>	27.8545	0.0016
<i>F-69-Yof-69</i>	26.3679	0.0023

Tablo 4.40: *Vtg-4* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-2</i>	27.1411	0.0022



Şekil 4.70: *Vtg-4* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* için her bir Video Karesinin Psnr ve Mse Değerleri Grafiği

4.11. *Vtg-5* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-3* SONUÇLARI

Vtg-5 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen *Vv-3* isimli tablo 3.3 gösterilen özelliklere sahip videonun 21,41 ve 111. kareleri ve yeniden oluşturulan 21,41 ve 11. kareleri şekil 4.70, şekil 4.71, şekil 4.72, şekil 4.73, şekil 4.74, şekil 4.75 ve bunların özellikleri tablo 4.41 ve tablo 4.42’de verilmiştir. Ayrıca tablo 4.43 ve c’te sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.76’da her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



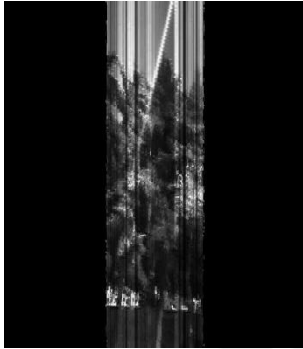
Şekil 4.71: F-21



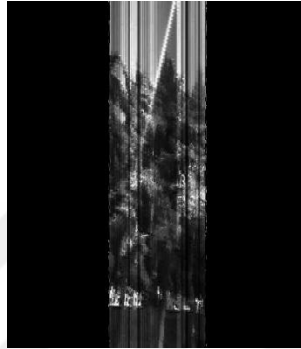
Şekil 4.72: F-41



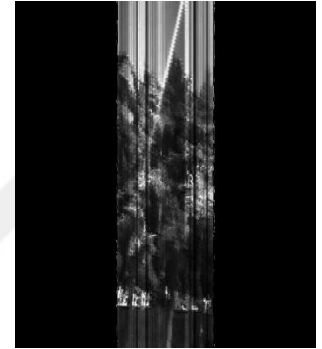
Şekil 4.73: F-111



Şekil 4.74: Yof-21



Şekil 4.75: Yof-41



Şekil 4.76: Yof-111

Tablo 4.41: Sympes Yöntemi ile Vtg-5 Evrensel Kümesiyle Yeniden oluşturulacak Vv-3 Videosundan Alınan Video Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-21	1280px	720px	96dpi	96dpi	24	JPG	22.5Kb
F-41	1280px	720px	96dpi	96dpi	24	JPG	22.1Kb
F-111	1280px	720px	96dpi	96dpi	24	JPG	26.4Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.42: Sympes Yöntemi ile Vtg-5 Evrensel Kümesiyle Yeniden oluşturulan Vv-3 Videosundan Alınan Video Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-21	1280px	720px	96dpi	96dpi	24	JPG	46.0Kb
Yof-41	1280px	720px	96dpi	96dpi	24	JPG	46.3Kb
Yof-111	1280px	720px	96dpi	96dpi	24	JPG	47.2Kb

Px: piksel

Dpi: dots per inch

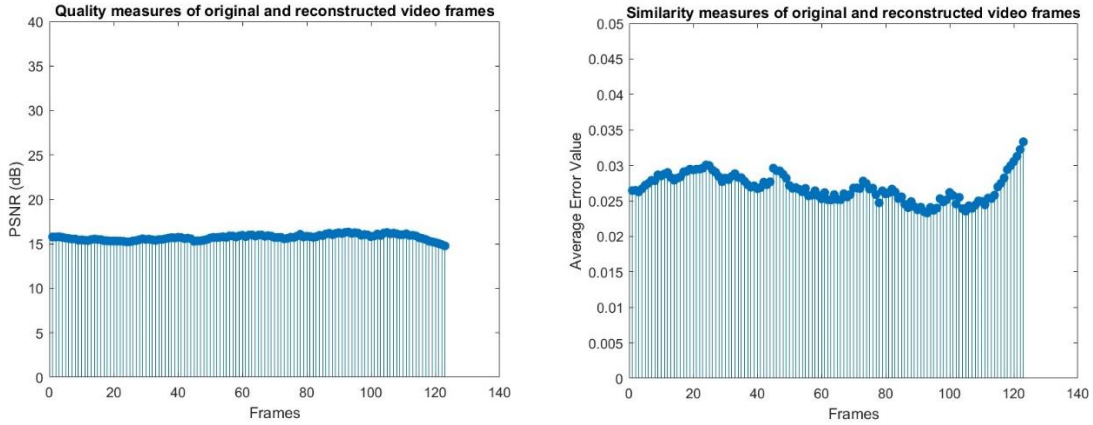
Kb: kilobyte

Tablo 4.43: Sympes Yöntemi ile Vtg-5 Evrensel Kümesiyle Yeniden oluşturulan Vv-3 Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
F-21-Yof-21	15.3073	0.0295
F-41-Yof-41	15.7047	0.0269
F-11-Yof-111	16.1170	0.0245

Tablo 4.44: Vtg-5 Evrensel Kümesi ile Yeniden Üretilen Vv-3 Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
Vv-3	15.7140	0.0269



Şekil 4.77: Vtg-5 Evrensel Kümesi ile Yeniden Üretilen Vv-3 için her bir Video Karesinin Psnr ve Mse Değerleri Grafiği

4.12. Vtg-6 İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN Vv-1 SONUÇLARI

Vtg-6 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen Vv-1 isimli tablo 3.3 gösterilen özelliklere sahip videonun 9,19 ve 29. kareleri ve yeniden oluşturulan 9,19 ve 29. kareleri şekil 4.77, şekil 4.78, şekil 4.79, şekil 4.80, şekil 4.81, şekil 4.82 ve bunların özellikleri tablo 4.45 ve tablo 4.46’da verilmiştir. Ayrıca tablo 4.47 ve tablo 4.48’de sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.83’te her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



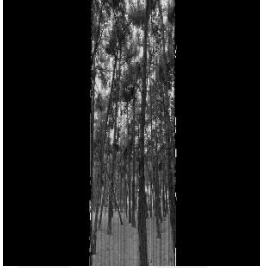
Şekil 4.78: F-9



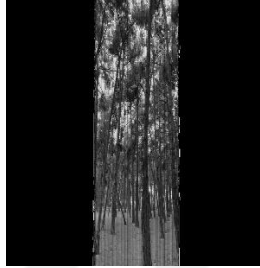
Şekil 4.79: F-19



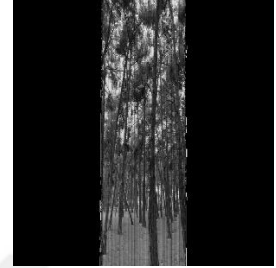
Şekil 4.80: F-29



Şekil 4.81: Yof-9



Şekil 4.82: Yof-19



Şekil 4.83: Yof-29

Tablo 4.45: Sympes Yöntemi ile Vtg-6 Evrensel Kümesiyle Yeniden oluşturulacak Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-9	1920px	1080px	96dpi	96dpi	24	JPG	155Kb
F-19	1920px	1080px	96dpi	96dpi	24	JPG	157Kb
F-29	1920px	1080px	96dpi	96dpi	24	JPG	155Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.46: Sympes Yöntemi ile Vtg-6 Evrensel Kümesiyle Yeniden oluşturulan Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-9	1920px	1080px	96dpi	96dpi	24	JPG	156Kb
Yof-19	1920px	1080px	96dpi	96dpi	24	JPG	158Kb
Yof-29	1920px	1080px	96dpi	96dpi	24	JPG	157Kb

Px: piksel

Dpi: dots per inch

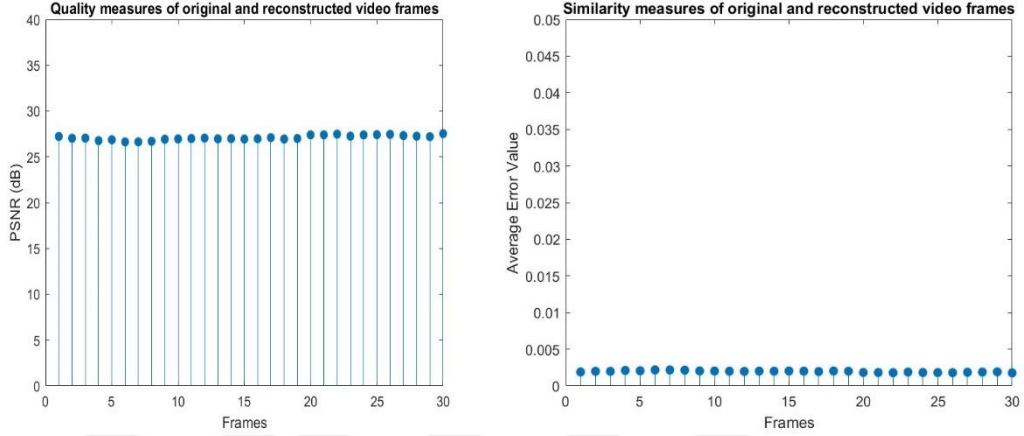
Kb: kilobyte

Tablo 4.47: Sympes Yöntemi ile Vtg-6 Evrensel Kümesiyle Yeniden oluşturulan Vv-1 Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
F-9-Yof-19	26.9192	0.0020
F-19-Yof-19	26.9995	0.0020
F-29-Yof-29	27.2091	0.0019

Tablo 4.48: *Vtg-6* Evrensel Kümesi ile Yeniden Üretilen *Vv-1* Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-1</i>	27.0983	0.0020



Şekil 4.84: *Vtg-6* Evrensel Kümesi ile Yeniden Üretilen *Vv-1* için her bir Video Karesinin Psnr ve Mse Değerleri Grafiği

Bu zamana kadar yaptığımız uygulamalar her bir video karesinin sütun vektörlerine bölünmesi şeklindedir. Bundan sonraki uygulamalar ise yine sympes yöntemini kullanarak bu sefer her bir video karesini 8x8'lik bloklara ayırma şeklindedir. Yine aynı evrensel küme ve aynı videolar üzerinde bu yöntem denenmiştir.

4.13. *Vtg-1* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-1* SONUÇLARI

Vtg-1 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden oluşturulan *Vv-1* isimli tablo 3.3 verilen özelliklere sahip videonun 1,11 ve 21. kareleri ve yeniden oluşturulan 1,11 ve 21.kareleri şekil 4.85, şekil 4.86, şekil 4.87, şekil 4.88, şekil 4.89, şekil 4.90 ve bunların özellikleri tablo 4.49 ve tablo 4.50 verilmiştir. Ayrıca tablo 4.51 ve tablo 4.52'te sonuçların psnr ve mse değerleri verilmiş ve şekil 4.91'de her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



Şekil 4.85: F-1



Şekil 4.86: F-11



Şekil 4.87: F-21



Şekil 4.88: Yof-1



Şekil 4.89: Yof-11



Şekil 4.90: Yof-21

Tablo 4.49: Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulacak Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-1	1920px	1080px	96dpi	96dpi	24	JPG	167Kb
F-11	1920px	1080px	96dpi	96dpi	24	JPG	154Kb
F-21	1920px	1080px	96dpi	96dpi	24	JPG	158Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.50: Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulan Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-1	1920px	1080px	96dpi	96dpi	24	JPG	155Kb
Yof-11	1920px	1080px	96dpi	96dpi	24	JPG	148Kb
Yof-21	1920px	1080px	96dpi	96dpi	24	JPG	153Kb

Px: piksel

Dpi: dots per inch

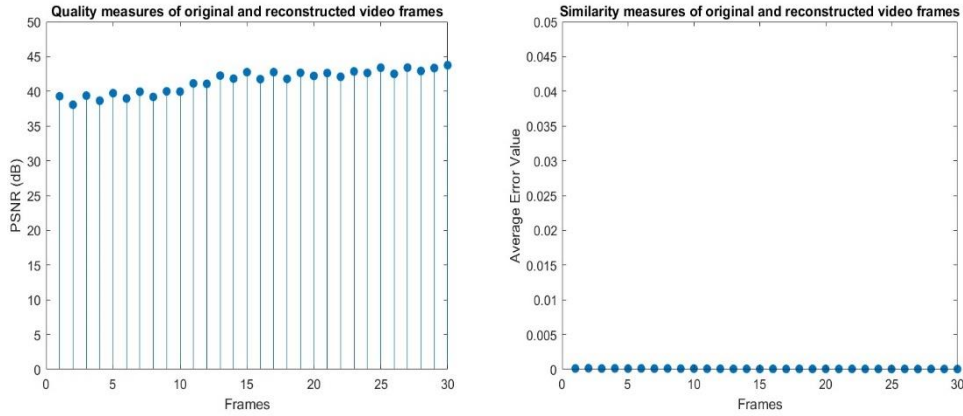
Kb: kilobyte

Tablo 4.51: Sympes Yöntemi ile *Vtg-1* Evrensel Kümesiyle Yeniden oluşturulacak *Vv-1* Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
<i>F-1-Yof-1</i>	39.2823	1.179675893293653e-04
<i>F-11-Yof-11</i>	41.1273	7.713865245323501e-05
<i>F-21-Yof-21</i>	42.6256	5.46316098153132e-05

Tablo 4.52: *Vtg-1* Evrensel Kümesi ile Yeniden Üretilen *Vv-1* Video Karelerinin Ortalama Psnr ve Mse Değerleri

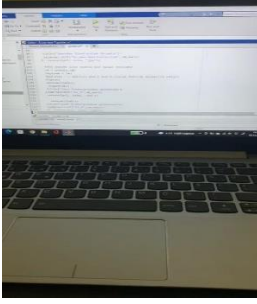
Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-1</i>	41.4282	7.75600255837024e-05



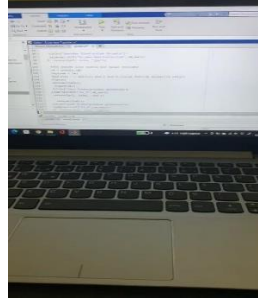
Şekil 4.91: *Vtg-1* Evrensel Kümesi ile Yeniden Üretilen *Vv-1* için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.14. *Vtg-1* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-2* SONUÇLARI

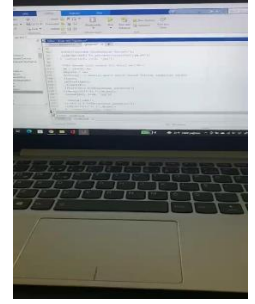
Vtg-1 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden oluşturulan *Vv-2* isimli tablo 3.3 gösterilen özelliklere sahip videonun 33,44 ve 87. kareleri ve yeniden oluşturulan 33,44 ve 87.kareleri şekil 4.92, şekil 4.93, şekil 4.94, şekil 4.95, şekil 4.100, şekil 4.101 ve bunların özellikleri tablo 4.53 ve tablo 4.54’te verilmiştir. Ayrıca tablo 4.55 ve tablo 4.56’da sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.102’de her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



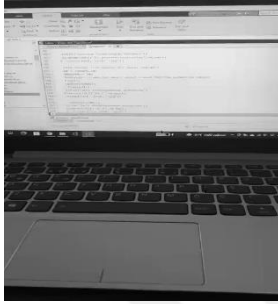
Şekil 4.92: F-33



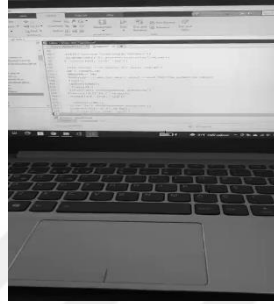
Şekil 4.93: F-44



Şekil 4.94: F-87



Şekil 4.95: Yof-33



Şekil 4.96: Yof-44



Şekil 4.97: Yof-87

Tablo 4.53: Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulacak Vv-2 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-33	480px	848px	96dpi	96dpi	24	JPG	34.9Kb
F-44	480px	848px	96dpi	96dpi	24	JPG	37.7Kb
F-87	480px	848px	96dpi	96dpi	24	JPG	37.9Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.54: Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulan Vv-2 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-33	480px	848px	96dpi	96dpi	24	JPG	34.1Kb
Yof-44	480px	848px	96dpi	96dpi	24	JPG	34.3Kb
Yof-87	480px	848px	96dpi	96dpi	24	JPG	35.4Kb

Px: piksel

Dpi: dots per inch

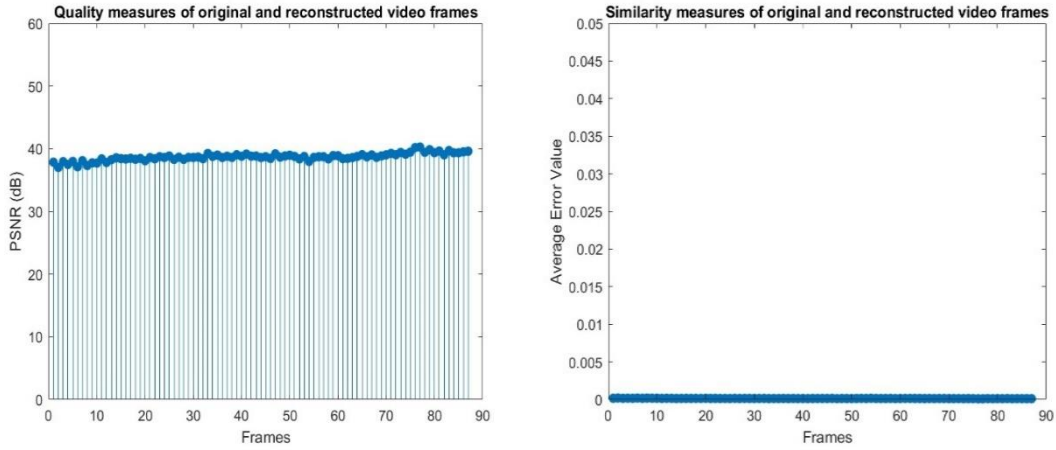
Kb: kilobyte

Tablo 4.55: Sympes Yöntemi ile *Vtg-1* Evrensel Kümesiyle Yeniden oluşturulacak *Vv-2* Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
<i>F-33-Yof-33</i>	39.2394	0.000119139364032702
<i>F-44-Yof-44</i>	38.6017	0.000137983221559681
<i>F-87-Yof-87</i>	39.0267	0.000125122082439662

Tablo 4.56: *Vtg-1* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
Vv-2	38.6992	0.000136207617069781



Şekil 4.98: *Vtg-1* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.15. *Vtg-1* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-3* SONUÇLARI

Vtg-1 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden oluşturulan *Vv-3* isimli tablo 3.3 gösterilen özelliklere sahip videonun 100,110 ve 123. kareleri ve yeniden oluşturulan 100,110 ve 123. kareleri şekil 4.99, şekil 4.100, şekil 4.101, şekil 4.102, şekil 4.103, şekil 4.104 ve bunların özellikleri tablo 4.57 ve tablo 4.58 verilmiştir. Ayrıca tablo 4.59 ve tablo 4.60'ta sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.105'de her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



Şekil 4.99: F-100



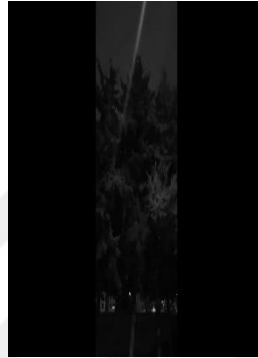
Şekil 4.100: F-110



Şekil 4.101: F-123



Şekil 4.102: Yof-100



Şekil 4.103: Yof-110



Şekil 4.104: Yof-123

Tablo 4.57: Sympes Yöntemi ile Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulacak Vv-3 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-100	1280px	720px	96dpi	96dpi	24	JPG	26.6Kb
F-110	1280px	720px	96dpi	96dpi	24	JPG	26.3Kb
F-123	1280px	720px	96dpi	96dpi	24	JPG	26.7Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.58: Sympes Yöntemi ile Vtg-1 Evrensel Kümesiyle Yeniden oluşturulan Vv-3 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-33	1280px	720px	96dpi	96dpi	24	JPG	24.8Kb
Yof-44	1280px	720px	96dpi	96dpi	24	JPG	24.3Kb
Yof-87	1280px	720px	96dpi	96dpi	24	JPG	25.1Kb

Px: piksel

Dpi: dots per inch

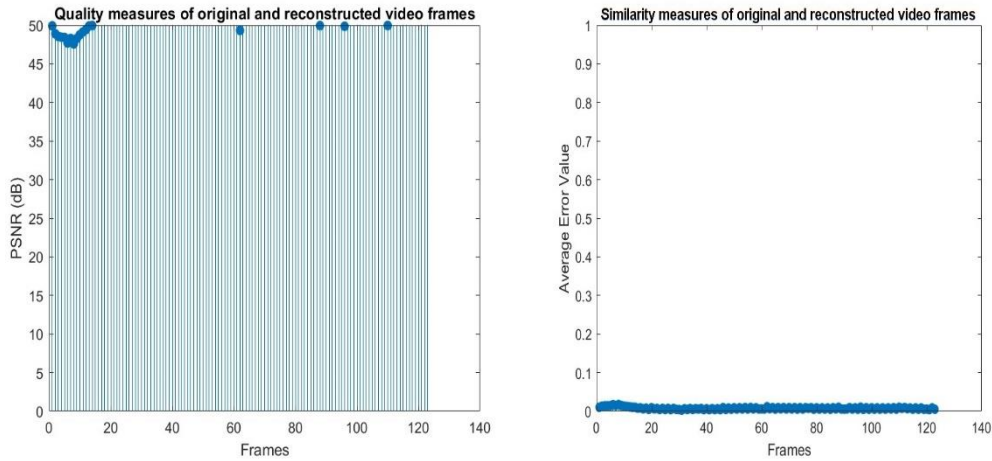
Kb: kilobyte

Tablo 4.59: Sympes Yöntemi ile *Vtg-1* Evrensel Kümesiyle Yeniden oluşturulan *Vv-3* Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
<i>F-100-Yof-100</i>	50.6890	8.53300177331350e-06
<i>F-110-Yof-110</i>	49.9723	1.00640001540397e-05
<i>F-123-Yof-123</i>	53.0365	4.96994529942207e-06

Tablo 4.60: *Vtg-1* Evrensel Kümesi ile Yeniden Üretilen *Vv-3* Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-3</i>	51.5456	7.47235394813299e-06



Şekil 4.105: *Vtg-1* Evrensel Kümesi ile Yeniden Üretilen *Vv-3* için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.16. *Vtg-2* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-1* SONUÇLARI

Vtg-2 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen *Vv-1* isimli tablo 3.3 gösterilen özelliklere sahip videonun 4,18 ve 28. kareleri ve yeniden oluşturulan 4,18 ve 28. kareleri şekil 4.106, şekil 4.107, şekil 4.108, şekil 4.109, şekil 4.110, şekil 4.111 ve bunların özellikleri tablo 4.61 ve tablo 4.62’de verilmiştir. Ayrıca tablo 4.63 ve tablo 4.64’te sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.112’te her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



Şekil 4.106: F-4



Şekil 4.107: F-18



Şekil 4.108: F-28



Şekil 4.109: Yof-4



Şekil 4.110: Yof-18



Şekil 4.111: Yof-28

Tablo 4.61: Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulacak Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-4	1920px	1080px	96dpi	96dpi	24	JPG	161Kb
F-18	1920px	1080px	96dpi	96dpi	24	JPG	157Kb
F-28	1920px	1080px	96dpi	96dpi	24	JPG	155Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.62: Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulan Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-4	1920px	1080px	96dpi	96dpi	24	JPG	150Kb
Yof -18	1920px	1080px	96dpi	96dpi	24	JPG	151Kb
Yof -28	1920px	1080px	96dpi	96dpi	24	JPG	150Kb

Px: piksel

Dpi: dots per inch

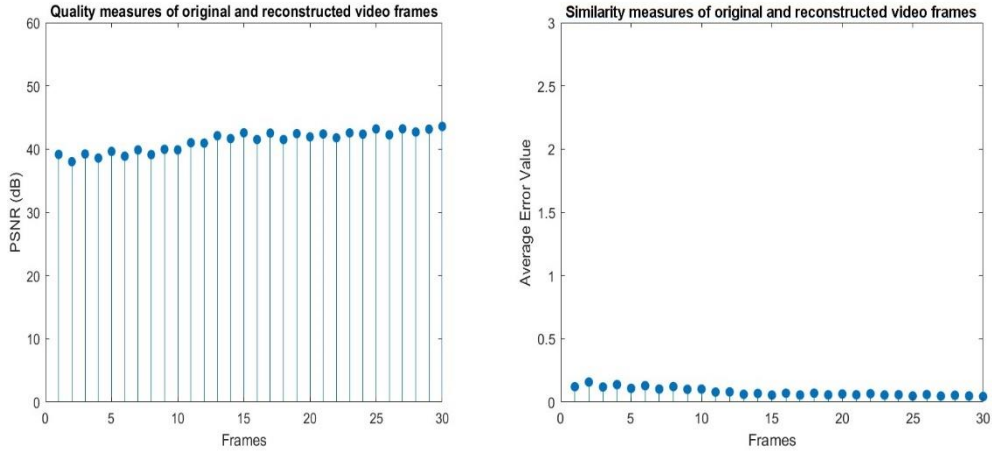
Kb: kilobyte

Tablo 4.63: Sympes Yöntemi ile *Vtg-2* Evrensel Kümesiyle Yeniden oluşturulan *Vv-1* Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
<i>F-4-Yof-4</i>	38.5935	0.000138244290242120
<i>F-18-Yof-18</i>	41.5179	7.05035842102515e-05
<i>F-28-Yof-28</i>	42.7077	5.36078315676307e-05

Tablo 4.64: *Vtg-2* Evrensel Kümesi ile Yeniden Üretilen *Vv-1* Video Karelerinin Ortalama Psnr ve Mse Değerleri

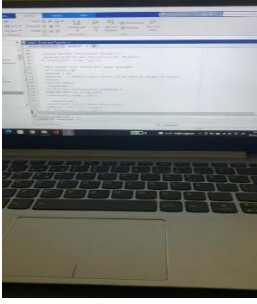
Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-1</i>	41.2660	8.00703878138038e-05



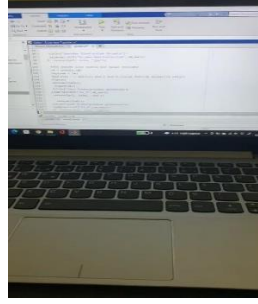
Şekil 4.112: *Vtg-2* Evrensel Kümesi ile Yeniden Üretilen *Vv-1* için her bir Karesinin Mse ve Psnr Değerleri Grafiği

4.17. *Vtg-2* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-2* SONUÇLARI

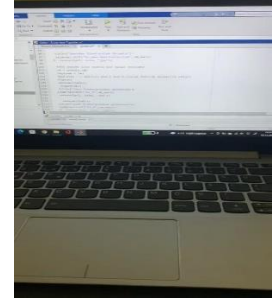
Vtg-2 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen *Vv-2* isimli tablo 3.3 gösterilen özelliklere sahip videonun 5,45 ve 85. kareleri ve yeniden oluşturulan 5,45 ve 85. kareleri şekil 4.113, şekil 4.114, şekil 4.115, şekil 4.116, şekil 4.117, şekil 4.118 ve bunların özellikleri tablo 4.65 ve tablo 4.66’da verilmiştir. Ayrıca tablo 4.67 ve tablo 4.68’de sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.119’de her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



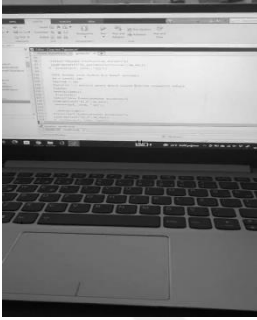
Şekil 4.113: F-5



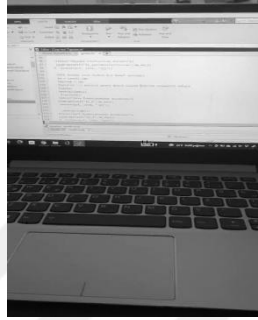
Şekil 4.114: F-45



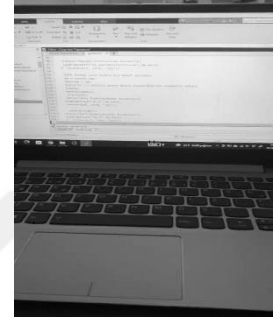
Şekil 4.115: F-85



Şekil 4.116: Yof-5



Şekil 4.117: Yof-45



Şekil 4.118: Yof-85

Tablo 4.65: Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulacak Vv-2 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-5	480px	848px	96dpi	96dpi	24	JPG	35.1Kb
F-45	480px	848px	96dpi	96dpi	24	JPG	37.4Kb
F-85	480px	848px	96dpi	96dpi	24	JPG	38.6Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.66: Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulan Vv-2 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-5	480px	848px	96dpi	96dpi	24	JPG	32.6Kb
Yof-45	480px	848px	96dpi	96dpi	24	JPG	35.0Kb
Yof-85	480px	848px	96dpi	96dpi	24	JPG	36.3Kb

Px: piksel

Dpi: dots per inch

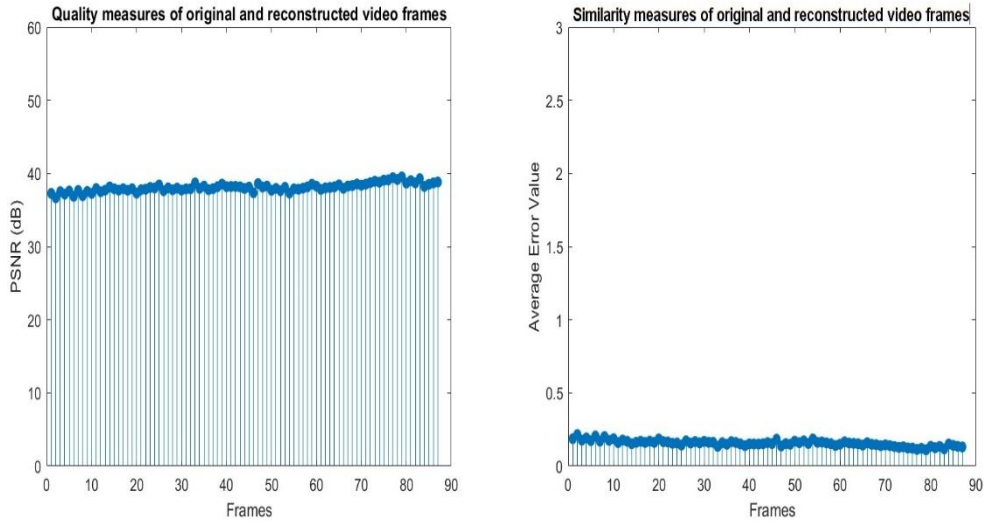
Kb: kilobyte

Tablo 4.67: Sympes Yöntemi ile *Vtg-2* Evrensel Kümesiyle Yeniden oluşturulan *Vv-2* Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
<i>F-5-Yof-5</i>	37.6022	0.000173693626544217
<i>F-45-Yof-45</i>	38.1326	0.000153724530190102
<i>F-85-Yof-85</i>	38.5099	0.000140932515588182

Tablo 4.68: *Vtg-2* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-2</i>	38.1023	0.000156069000532268



Şekil 4.119: *Vtg-2* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.18. *Vtg-2* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-3* SONUÇLARI

Vtg-2 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen *Vv-3* isimli tablo 3.3 gösterilen özelliklere sahip videonun 7,97 ve 117. kareleri ve yeniden oluşturulan 7,97 ve 117. kareleri şekil 4.120, şekil 4.121, şekil 4.122, şekil 4.123, şekil 4.124, şekil 4.125 ve bunların özellikleri tablo 4.69 ve tablo 4.70 verilmiştir. Ayrıca tablo 4.71 ve

tablo 4.72’de sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.126’de her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



Şekil 4.120: F-7



Şekil 4.121: F-97



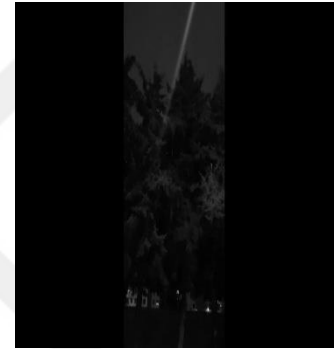
Şekil 4.122: F-117



Şekil 4.123: Yof-7



Şekil 4.124: Yof-97



Şekil 4.125: Yof-117

Tablo 4.69: Sympes Yöntemi ile Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulacak Vv-3 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-7	1280px	720px	96dpi	96dpi	24	JPG	22.5Kb
F-97	1280px	720px	96dpi	96dpi	24	JPG	26.7Kb
F-117	1280px	720px	96dpi	96dpi	24	JPG	26.6Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.70: Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulan Vv-3 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-7	1280px	720px	96dpi	96dpi	24	JPG	20.2Kb
Yof-97	1280px	720px	96dpi	96dpi	24	JPG	25.2Kb
Yof-117	1280px	720px	96dpi	96dpi	24	JPG	25.2Kb

Px: piksel

Dpi: dots per inch

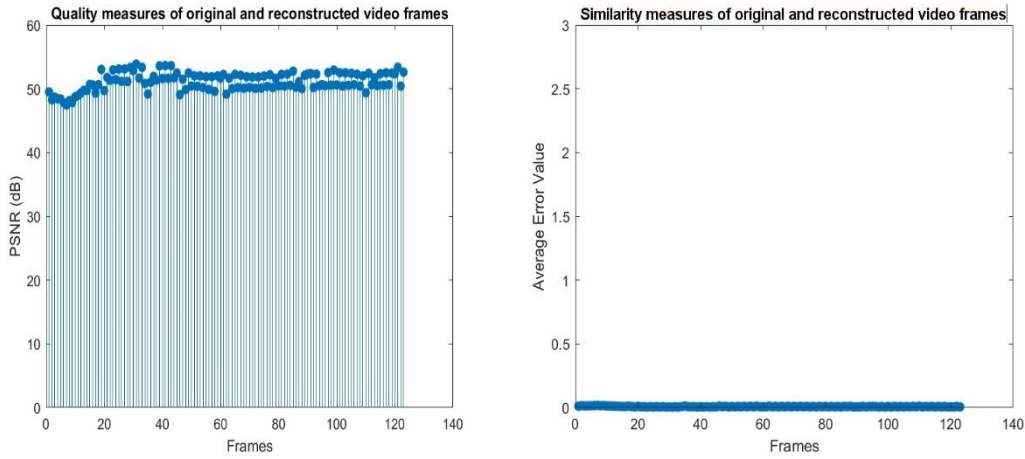
Kb: kilobyte

Tablo 4.71: Sympes Yöntemi ile Vtg-2 Evrensel Kümesiyle Yeniden oluşturulan Vv-3 Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
F-7-Yof-7	47.4578	1.79562942981773e-05
F-97-Yof-97	52.5275	5.58792200608809e-06
F-117-Yof-117	52.5047	5.61733542768831e-06

Tablo 4.72: Vtg-2 Evrensel Kümesi ile Yeniden Üretilen Vv-3 Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
Vv-3	51.1151	8.17569256390004e-06



Şekil 4.126: Vtg-2 Evrensel Kümesi ile Yeniden Üretilen Vv-3 için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.19. Vtg-3 İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN Vv-1 SONUÇLARI

Vtg-3 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen Vv-1 isimli tablo 3.3 gösterilen özelliklere sahip videonun 2,20 ve 30. kareleri ve yeniden oluşturulan 2,20 ve 30. kareleri şekil 4.127, şekil 4.128, şekil 4.129, şekil 4.130, şekil 4.131, şekil 4.133 ve bunların özellikleri tablo 4.73 ve tablo 4.74'te verilmiştir. Ayrıca tablo 4.75 ve tablo 4.76'da sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.134'te her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



Şekil 4.127: F-2



Şekil 4.128: F-20



Şekil 4.129: F-30



Şekil 4.130: Yof-2



Şekil 4.131: Yof-20



Şekil 4.132: Yof-30

Tablo 4.73: Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulacak Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-2	1920px	1080px	96dpi	96dpi	24	JPG	164Kb
F-20	1920px	1080px	96dpi	96dpi	24	JPG	158Kb
F-30	1920px	1080px	96dpi	96dpi	24	JPG	155Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.74: Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulan Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-2	1920px	1080px	96dpi	96dpi	24	JPG	149Kb
Yof-20	1920px	1080px	96dpi	96dpi	24	JPG	150Kb
Yof-30	1920px	1080px	96dpi	96dpi	24	JPG	149Kb

Px: piksel

Dpi: dots per inch

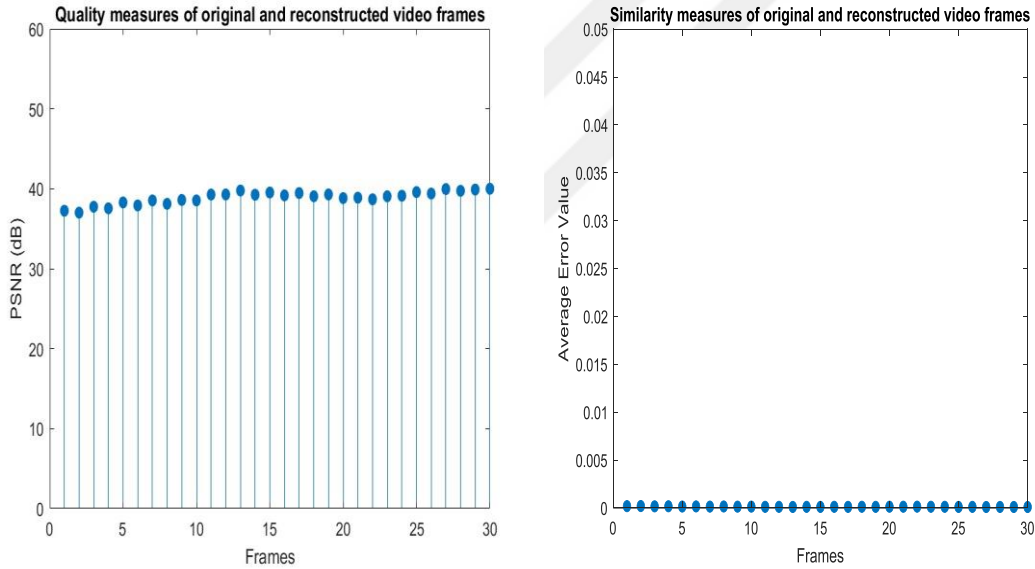
Kb: kilobyte

Tablo 4.75: Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulan Vv-1 Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
F-2-Yof-2	37.0112	0.000199014268016575
F-20-Yof-20	38.8332	0.000130822546883677
F-30-Yof-30	39.9996	0.000100009463359297

Tablo 4.76: Vtg-3 Evrensel Kümesi ile Yeniden Üretilen Vv-1 Video Karelerinin Ortalama Psnr ve Mse Değerleri

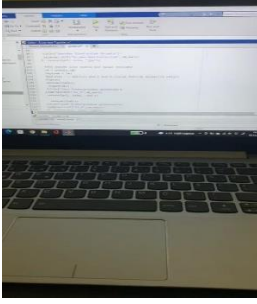
Video Verisi	Ortalama PSNR	Ortalama MSE
Vv-1	38.8965	0.000131178613193643



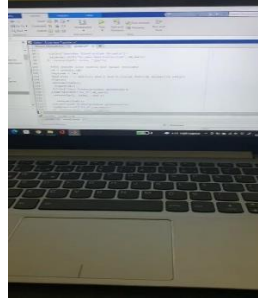
Şekil 4.133: Vtg-3 Evrensel Kümesi ile Yeniden Üretilen Vv-1 için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.20. Vtg-3 İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN Vv-2 SONUÇLARI

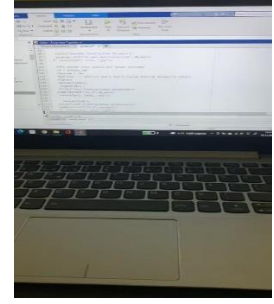
Vtg-3 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen Vv-2 isimli tablo 3.3 gösterilen özelliklere sahip videonun 34,54 ve 74. kareleri ve yeniden oluşturulan 34,54 ve 74. kareleri şekil 4.134, şekil 4.135, şekil 4.136, şekil 4.137, şekil 4.138, şekil 4.139 ve bunların özellikleri tablo 4.77 ve tablo 4.78’te verilmiştir. Ayrıca tablo 4.79 ve tablo 4.80’de sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.140’da her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



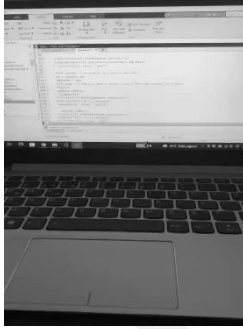
Şekil 4.134: F-34



Şekil 4.135: F-54



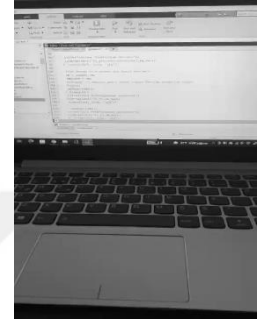
Şekil 4.136: F-74



Şekil 4.137: Yof-34



Şekil 4.138: Yof-34



Şekil 4.139: Yof-74

Tablo 4.77: Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulacak Vv-2 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-34	480px	848px	96dpi	96dpi	24	JPG	36.1Kb
F-34	480px	848px	96dpi	96dpi	24	JPG	36.1Kb
F-74	480px	848px	96dpi	96dpi	24	JPG	37.8Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.78: Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulan Vv-2 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-34	480px	848px	96dpi	96dpi	24	JPG	33.6Kb
Yof-54	480px	848px	96dpi	96dpi	24	JPG	33.3Kb
Yof-74	480px	848px	96dpi	96dpi	24	JPG	34.8Kb

Px: piksel

Dpi: dots per inch

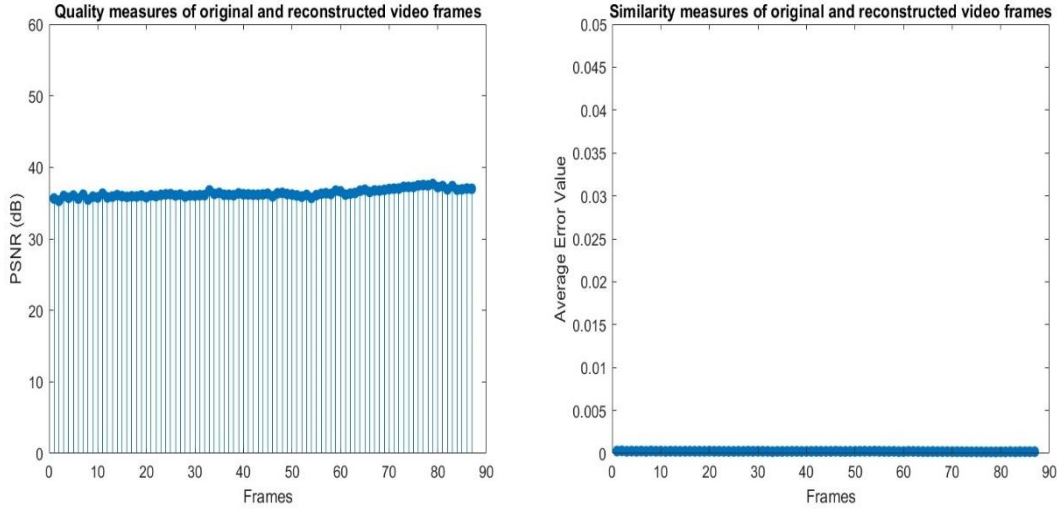
Kb: kilobyte

Tablo 4.79: Sympes Yöntemi ile *Vtg-3* Evrensel Kümesiyle Yeniden oluşturulan *Vv-2* Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
<i>F-34-Yof-34</i>	36.2758	0.000235731926176047
<i>F-54-Yof-54</i>	36.2726	0.000266902536424008
<i>F-74-Yof-74</i>	37.2560	0.000188104118653400

Tablo 4.80: *Vtg-3* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-2</i>	36.3720	0.000232149983788791



Şekil 4.140: *Vtg-3* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* için her bir Karesinin Psnr ve Mse Değerleri Grafiği

4.21. *Vtg-3* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-3* SONUÇLARI

Vtg-3 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen *Vv-3* isimli tablo 3.3 gösterilen özelliklere sahip videonun 42,102 ve 122. kareleri ve yeniden oluşturulan 42,102 ve 122. kareleri şekil 4.141, şekil 4.142, şekil 4.143, şekil 4.144, şekil 4.145, şekil 4.146 ve bunların özellikleri tablo 4.81 ve tablo 4.82’te verilmiştir. Ayrıca tablo 4.83 ve tablo 4.84’te sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.147’de her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



Şekil 4.141: F-42



Şekil 4.142: F-102



Şekil 4.143: F-122



Şekil 4.144: Yof-42



Şekil 4.145: Yof-42



Şekil 4.146: Yof-42

Tablo 4.81: Sympes Yöntemi ile Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulacak Vv-3 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-42	1280px	720px	96dpi	96dpi	24	JPG	22.5Kb
F-102	1280px	720px	96dpi	96dpi	24	JPG	26.8Kb
F-122	1280px	720px	96dpi	96dpi	24	JPG	26.8Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.82: Sympes Yöntemi ile Vtg-3 Evrensel Kümesiyle Yeniden oluşturulan Vv-3 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-42	1280px	720px	96dpi	96dpi	24	JPG	20.6Kb
Yof-102	1280px	720px	96dpi	96dpi	24	JPG	24.8Kb
Yof-122	1280px	720px	96dpi	96dpi	24	JPG	24.8Kb

Px: piksel

Dpi: dots per inch

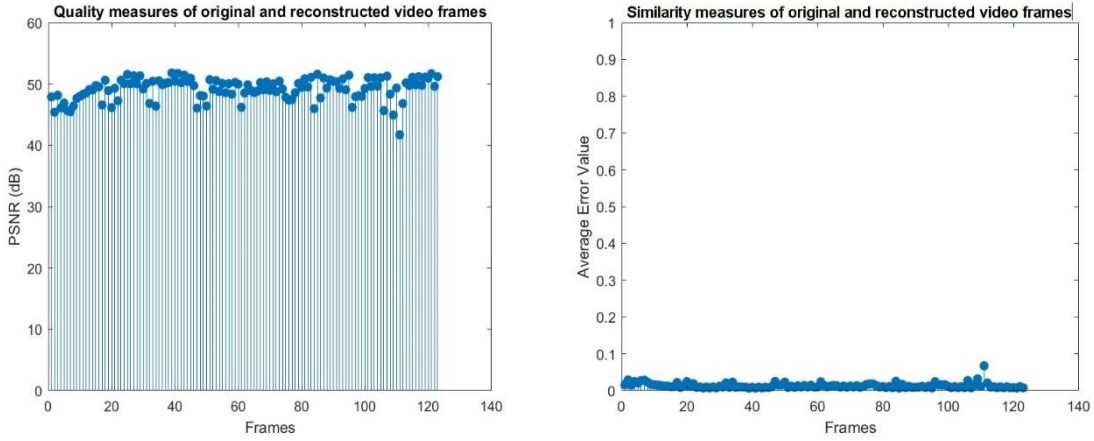
Kb: kilobyte

Tablo 4.83: Sympes Yöntemi ile *Vtg-3* Evrensel Kümesiyle Yeniden oluşturulan *Vv-3* Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Farklı Görüntü Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
<i>F-42-Yof-42</i>	50.2290	9.48630582260519e-06
<i>F-102-Yof-102</i>	49.6064	1.09485721832619e-05
<i>F-122-Yof-122</i>	49.6069	1.09406413412788e-05

Tablo 4.84: *Vtg-3* Evrensel Kümesi ile Yeniden Üretilen *Vv-3* Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-3</i>	49.1484	1.34450530059104e-05



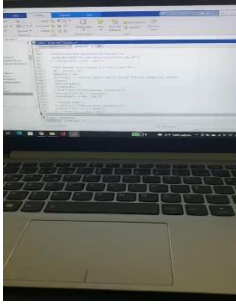
Şekil 4.147: *Vtg-3* Evrensel Kümesi ile Yeniden Üretilen *Vv-3* için her bir Karesinin Psnr ve Mse Değerleri Grafiği

Çalışmanın bu kısmında ise sonuçları aşağıda belirtilen uygulamalar ise oluşturulacak videodan rasgele seçilen bir video karesi alınıp ve bu seçilen video karesinin evrensel küme kabul edilmesiyle yapılmıştır.

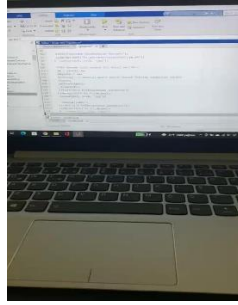
4.22. *Vtg-4* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-2* SONUÇLARI

Vtg-4 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen *Vv-2* isimli tablo 3.3 gösterilen özelliklere sahip videonun 8,17 ve 69. kareleri ve yeniden oluşturulan 8,17 ve 69. kareleri şekil 4.148, şekil 4.149, şekil 4.150, şekil 4.151, şekil 4.152, şekil 4.153 ve bunların özellikleri tablo 4.85 ve tablo 4.86’da verilmiştir. Ayrıca tablo 4.87 ve

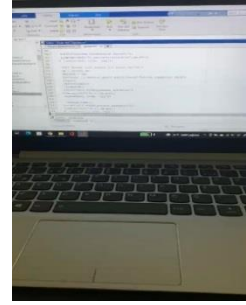
tablo 4.89’te sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.154’te her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



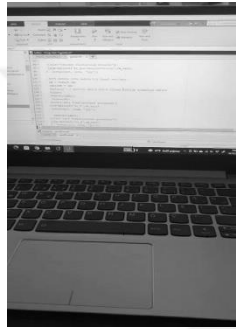
Şekil 4.148: F-8



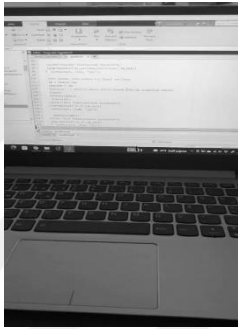
Şekil 4.149: F-17



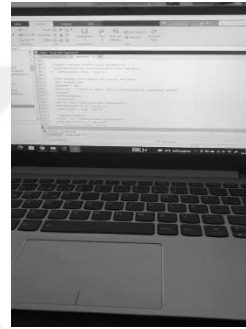
Şekil 4.150: F-69



Şekil 4.151: Yof-8



Şekil 4.152: Yof-17



Şekil 4.153: Yof-69

Tablo 4.85: Sympes Yöntemi ile Vtg-4 Evrensel Kümesiyle Yeniden oluşturulacak Vv-2 Videosundan Alınan Video Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-8	480px	848px	96dpi	96dpi	24	JPG	35.6Kb
F-17	480px	848px	96dpi	96dpi	24	JPG	38.0Kb
F-69	480px	848px	96dpi	96dpi	24	JPG	37.2Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.86: Sympes Yöntemi ile Vtg-4 Evrensel Kümesiyle Yeniden oluşturulan Vv-2 Videosundan Alınan Video Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-8	480px	848px	96dpi	96dpi	24	JPG	32.8Kb
Yof-17	480px	848px	96dpi	96dpi	24	JPG	35.6Kb
Yof-69	480px	848px	96dpi	96dpi	24	JPG	34.9Kb

Px: piksel

Dpi: dots per inch

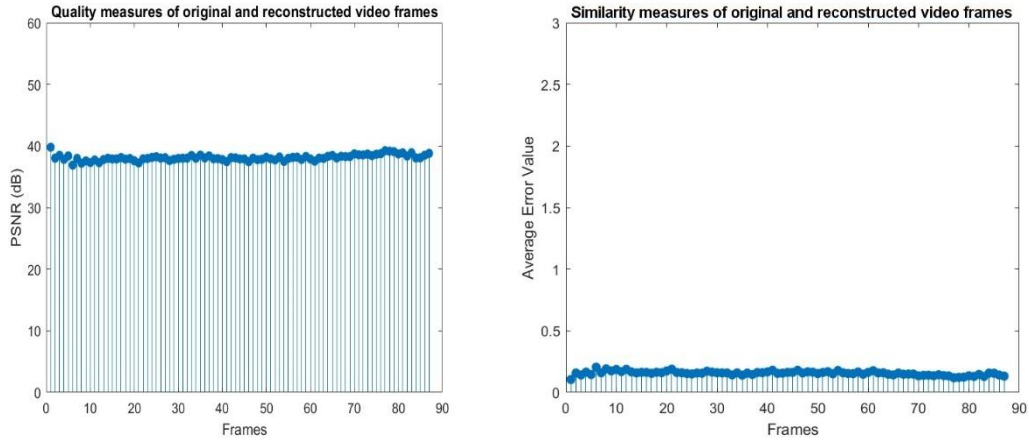
Kb: kilobyte

Tablo 4.87: Sympes Yöntemi ile *Vtg-4* Evrensel Kümesiyle Yeniden oluşturulan *Vv-2* Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
<i>F-8-Yof-8</i>	37.1936	0.000190828726977287
<i>F-17-Yof-17</i>	38.1141	0.000154378856765298
<i>F-69-Yof-69</i>	38.2770	0.000148696992814138

Tablo 4.88: *Vtg-4* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-2</i>	38.1140	0.000155282419037542



Şekil 4.154: *Vtg-4* Evrensel Kümesi ile Yeniden Üretilen *Vv-2* için her bir Video Karesinin Psnr ve Mse Değerleri Grafiği

4.23. *Vtg-5* İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN *Vv-3* SONUÇLARI

Vtg-5 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen *Vv-3* isimli tablo 3.3 gösterilen özelliklere sahip videonun 21,41 ve 111. kareleri ve yeniden oluşturulan 21,41 ve 11. kareleri şekil 4.155, şekil 4.156, şekil 4.157, şekil 4.158, şekil 4.159, şekil 4.160 ve bunların özellikleri tablo 4.89 ve tablo 4.90’da verilmiştir. Ayrıca tablo 4.91 ve tablo 4.92’te sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.161’de her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



Şekil 4.155: F-21



Şekil 4.156: F-41



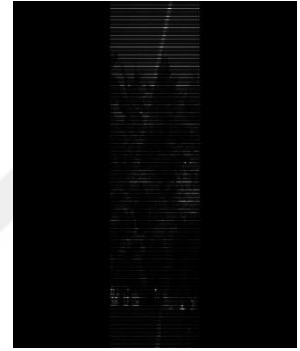
Şekil 4.157: F-111



Şekil 4.158: Yof-21



Şekil 4.159: Yof-41



Şekil 4.160: Yof-111

Tablo 4.89: Sympes Yöntemi ile Vtg-5 Evrensel Kümesiyle Yeniden oluşturulacak Vv-3 Videosundan Alınan Video Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-21	1280px	720px	96dpi	96dpi	24	JPG	22.5Kb
F-41	1280px	720px	96dpi	96dpi	24	JPG	22.1Kb
F-111	1280px	720px	96dpi	96dpi	24	JPG	26.4Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.90: Sympes Yöntemi ile Vtg-5 Evrensel Kümesiyle Yeniden oluşturulan Vv-3 Videosundan Alınan Video Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-21	1280px	720px	96dpi	96dpi	24	JPG	43.4Kb
Yof-41	1280px	720px	96dpi	96dpi	24	JPG	44.2Kb
Yof-111	1280px	720px	96dpi	96dpi	24	JPG	55.4Kb

Px: piksel

Dpi: dots per inch

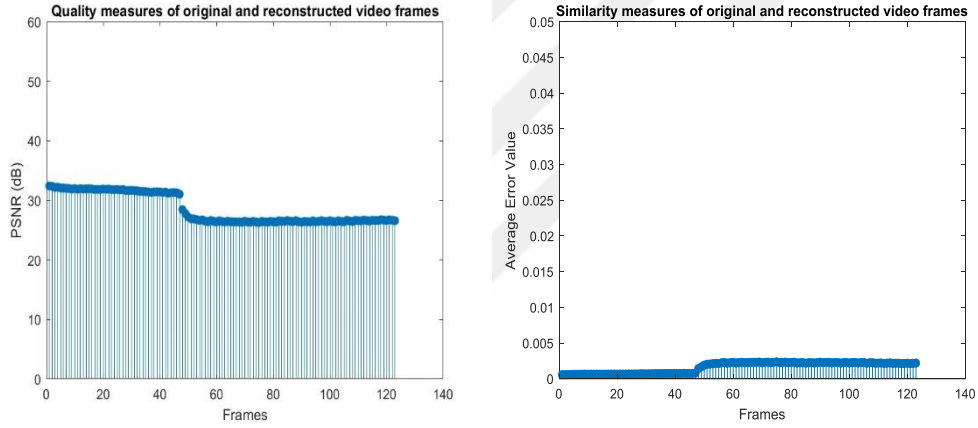
Kb: kilobyte

Tablo 4.91: Sympes Yöntemi ile Vtg-5 Evrensel Kümesiyle Yeniden oluşturulan Vv-3 Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
F-21-Yof-21	31.8511	0.000652960942524787
F-41-Yof-41	31.3202	0.000737877658903325
F-11-Yof-111	26.5625	0.002206754056586940

Tablo 4.92: Vtg-5 Evrensel Kümesi ile Yeniden Üretilen Vv-3 Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
Vv-3	28.5352	0.0016



Şekil 4.161: Vtg-5 Evrensel Kümesi ile Yeniden Üretilen Vv-3 için her bir Video Karesinin Psnr ve Mse Değerleri Grafiği

4.24. Vtg-6 İLE OLUŞTURULAN EVRENSEL KÜME İLE ÜRETİLEN Vv-1 SONUÇLARI

Vtg-6 ile oluşturulan evrensel küme ile Sympes yöntemim ile yeniden üretilen Vv-1 isimli tablo 3.3 gösterilen özelliklere sahip videonun 9,19 ve 29. kareleri ve yeniden oluşturulan 9,19 ve 29. kareleri şekil 4.164, şekil 4.165, şekil 4.166, şekil 4.167, şekil 4.168, şekil 4.169 ve bunların özellikleri tablo 4.93 ve tablo 4.94’te verilmiştir. Ayrıca tablo 4.95 ve tablo 4.96’da sonuçların psnr ve mse değerleri verilmiş ayrıca şekil 4.170’te her bir kare için psnr ve mse değerlerinin grafiği gösterilmiştir.



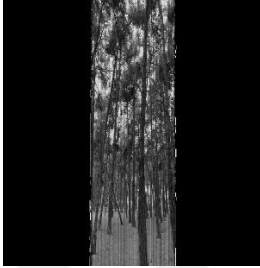
Şekil 4.162: F-9



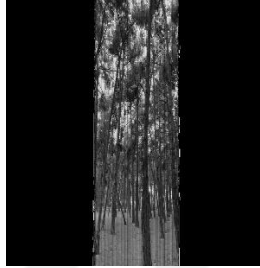
Şekil 4.163: F-19



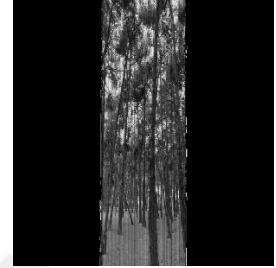
Şekil 4.164: F-29



Şekil 4.165: Yof-9



Şekil 4.166: Yof-19



Şekil 4.167: Yof-29

Tablo 4.93: Sympes Yöntemi ile Vtg-6 Evrensel Kümesiyle Yeniden oluşturulacak Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
F-9	1920px	1080px	96dpi	96dpi	24	JPG	155Kb
F-19	1920px	1080px	96dpi	96dpi	24	JPG	157Kb
F-29	1920px	1080px	96dpi	96dpi	24	JPG	155Kb

Px: piksel

Dpi: dots per inch

Kb: kilobyte

Tablo 4.94: Sympes Yöntemi ile Vtg-6 Evrensel Kümesiyle Yeniden oluşturulan Vv-1 Videosundan Alınan Farklı Görüntü Karelerinin Özellikleri

Video Karesi Görüntüleri	Genişlik	Yükseklik	Yatay Çözünürlük	Dikey Çözünürlük	Bit Derinliği	Resim Türü	Boyut
Yof-9	1920px	1080px	96dpi	96dpi	24	JPG	210Kb
Yof-19	1920px	1080px	96dpi	96dpi	24	JPG	212Kb
Yof-29	1920px	1080px	96dpi	96dpi	24	JPG	212Kb

Px: piksel

Dpi: dots per inch

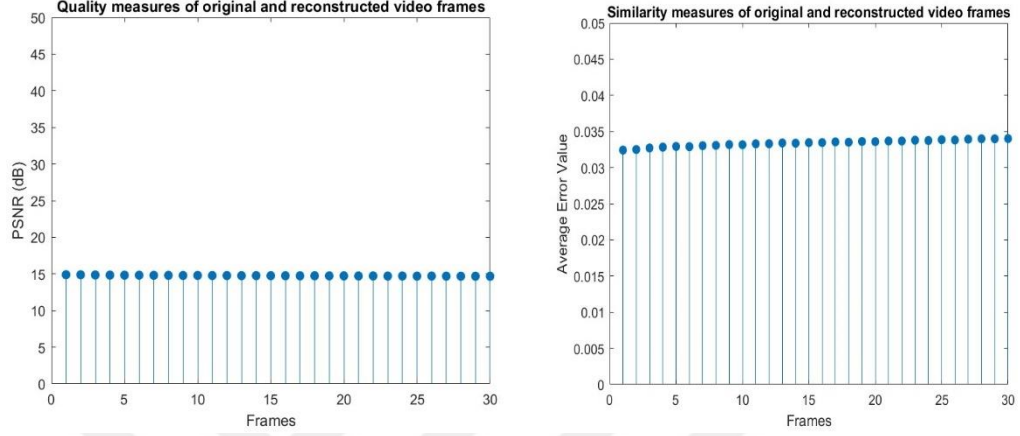
Kb: kilobyte

Tablo 4.95: Sympes Yöntemi ile Vtg-6 Evrensel Kümesiyle Yeniden oluşturulan Vv-1 Videosundan Alınan ve Yeniden oluşturulan Videodan Alınan Karelerinin Psnr ve Mse Değerleri

Video Karesi Görüntüleri	PSNR	MSE
F-9-Yof-19	14.7897	0.0332
F-19-Yof-19	14.7348	0.0336
F-29-Yof-29	14.6870	0.0340

Tablo 4.96: *Vtg-6* Evrensel Kümesi ile Yeniden Üretilen *Vv-1* Video Karelerinin Ortalama Psnr ve Mse Değerleri

Video Verisi	Ortalama PSNR	Ortalama MSE
<i>Vv-1</i>	14.7633	0.0334



Şekil 4.168: *Vtg-6* Evrensel Kümesi ile Yeniden Üretilen *Vv-1* için her bir Video Karesinin Psnr ve Mse Değerleri Grafiği

5. TARTIŞMA

Bu tez kapsamında 6 farklı görüntüden oluşturduğumuz birbirlerinden farklı çözünürlüklere sahip evrensel küme ve yine çözünürlükleri ve süreleri birbirinden farklı 3 farklı video ile çalışılmıştır. Çalışılan bu evrensel kümelerin 3 tanesi video karelerinden farklı ve her biri farklı çözünürlüklere sahip görüntülerden alınmıştır. Çalışılan diğer 3 görüntü ise Sympes yöntemi ile tekrar oluşturulmak istenen videoların herhangi bir video karesinden alınmış olup bu kareler evrensel küme olarak kullanılmıştır.

Vtg-1 Vtg-2 Vtg-3 isimli evrensel kümeleri *Vv-1 Vv-2 Vv-3* videoları için kullanılmıştır. *Vtg-4* evrensel kümesi *Vv-2*, *Vtg-5* evrensel kümesi *Vv-3* ve *Vtg-6* evrensel kümesi *Vv-1* isimli videolar için kullanılmıştır.

Sympes yöntemi ile yeniden oluşturduğumuz videoları, kullandığımız her bir evrensel küme için her iki yöntem uygulanarak psnr, mse değerleri, videoyu yeniden oluşturma süreleri ve blok bazında sıkıştırma yüzdeleri aşağıdaki tablolarda gösterildi.

Tablo 5.1: *Vv-1* video verisi için psnr, mse değerleri

Evrensel küme	<i>Vv-1</i> (PSNR, MSE)	<i>Vv-1</i> (PSNR, MSE)
	<i>Sütun Vektör Yaklaşımı (Lf=8)</i>	<i>Görüntü Blok Yaklaşımı (MxN=8)</i>
<i>Vtg-1</i>	(27.60588, 0.0017)	(41.4282, 7.7560025583702e-05)
<i>Vtg-2</i>	(27.65370, 0.0017)	(41.2660, 8.0070387813808e-05)
<i>Vtg-3</i>	(27.00250, 0.0020)	(38.8965, 0.00013117861319364)
<i>Vtg-6</i>	(27.09830, 0.0020)	(14,7633, 0.0334)

Vv-1 videosunu oluşturmak için kullandığımız 4 farklı evrensel küme ve bunların psnr ve mse değerleri tablo 5.1’de gösterilmiştir.

Tablo 5.2: *Vv-2* video verisinin psnr ve mse değerleri

Evrensel küme	<i>Vv-2</i> (PSNR, mse)	<i>Vv-2</i> (PSNR, mse)
	<i>Sütun Vektör Yaklaşımı (Lf=8)</i>	<i>Görüntü Blok Yaklaşımı (MxN=8)</i>
<i>Vtg-1</i>	(27.2563, 0.0022)	(38.6992, 0.0001362076170697)
<i>Vtg-2</i>	(27.3640, 0.0021)	(38.1023, 0.0001560690005322)
<i>Vtg-3</i>	(26.7015, 0.0024)	(36.3720, 0.0002321499837887)
<i>Vtg-4</i>	(27.1114, 0.0022)	(38.1140, 0.0001552824190375)

Vv-2 videosunu oluşturmak için kullandığımız 4 farklı evrensel küme ve bunların psnr ve mse değerleri tablo 5.2’de gösterilmiştir.

Tablo 5.3: Vv-3 video verisinin psnr ve mse değerleri

Evrensel küme	Vv-3 (PSNR, mse)	Vv-3 (PSNR, mse)
	<i>Sütun Vektör Yaklaşımı ($L_f=8$)</i>	<i>Görüntü Blok Yaklaşımı ($M \times N=8$)</i>
Vtg-1	(14.8845,0.00325)	(51.5456, 7.472353948299e-06)
Vtg-2	(15.7450,0.02667)	(51.1151, 8.175692563904e-06)
Vtg-3	(15.6419,0.02730)	(49.1484, 1.344505300104e-05)
Vtg-5	(15.7140,0.02690)	(28.5352, 0.0016)

Vv-3 videosunu oluşturmak için kullandığımız 4 farklı evrensel küme ve bunların psnr ve mse değerleri tablo 5.3'te gösterilmiştir.

Tablo 5.1, Tablo 5.2 ve Tablo 5.3'e incelediğimizde uyguladığımız her iki yöntemde görüntüleri bloklara ($M \times N=8$) böldüğümüzde oluşturduğumuz videoların psnr ve mse açısından değerlendirildiğinde görüntüleri sütun vektörlerine ($L_f=8$) böldüğümüz yöntemden hem psnr hem de mse değerlerine göre çok iyi sonuç vermiştir. Görüntüler bloklara bölünmüş yöntemde orijinaline daha yakın çıkmıştır

Sympes yöntemi ile farklı evrensel küme kullanarak yeniden oluşturulan videoların oluşması için geçen süreler tablo 5.4, tablo 5.5 ve tablo 5.6'da verilmiştir.

Tablo 5.4: Evrensel kümeler ile Video Verisi Vv-1 oluşma süreleri

Evrensel Küme	Video Verisi Vv-1	Video Verisi Vv-1
	<i>Sütun Vektör Yaklaşımı ($L_f=8$)</i>	<i>Görüntü Blok Yaklaşımı ($M \times N=8$)</i>
Vtg-1	$\cong 16$ dakika	$\cong 110$ dakika
Vtg-2	$\cong 10$ dakika	$\cong 138$ dakika
Vtg-3	$\cong 7,5$ dakika	$\cong 123$ dakika
Vtg-6	$\cong 8$ dakika	$\cong 104$ dakika

Tablo 5.5: Evrensel kümeler ile Video Verisi Vv-2 oluşma süreleri

Evrensel Küme	Video Verisi Vv-2	Video Verisi Vv-2
	<i>Sütun Vektör Yaklaşımı ($L_f=8$)</i>	<i>Görüntü Blok Yaklaşımı ($M \times N=8$)</i>
Vtg-1	$\cong 27$ dakika	$\cong 47$ dakika
Vtg-2	$\cong 11$ dakika	$\cong 61$ dakika
Vtg-3	$\cong 14$ dakika	$\cong 24$ dakika
Vtg-4	$\cong 13$ dakika	$\cong 21$ dakika

Tablo 5.6: Evrensel kümeler ile Video Verisi Vv-3 oluşma süreleri

Evrensel Küme	Video Verisi Vv-3	Video Verisi Vv-3
	<i>Sütun Vektör Yaklaşımı ($L_f=8$)</i>	<i>Görüntü Blok Yaklaşımı ($M \times N=8$)</i>
Vtg-1	$\cong 41$ dakika	$\cong 111$ dakika
Vtg-2	$\cong 20$ dakika	$\cong 172$ dakika
Vtg-3	$\cong 13$ dakika	$\cong 136$ dakika
Vtg-5	$\cong 13$ dakika	$\cong 131$ dakika

Tablo 5.4, Tablo 5.5 ve Tablo 5.6 incelediğimizde uyguladığımız her iki yöntemde yani görüntüleri sütun vektörlerine ($L_f=8$) böldüğümüz yöntem ile görüntüleri bloklara ($M \times N=8$) böldüğümüzde yöntemi videoları oluşturma süreleri açısından değerlendirildiğinde, görüntüleri sütun vektörlerine böldüğümüz yöntemin videoları daha kısa sürede oluşturduğunu gözlemledik. Ayrıca videoların yeniden oluşma süreleri evrensel küme ile ilgili olmadığı ama yeniden oluşturulacak videonun özellikleriyle alakalı olduğu sonucuna ulaşılmıştır.

Tablo 5.7: Sympes Yöntemi ile Videoların Sıkıştırma Yüzdeleri

Video Verisi	<i>Sütun Vektör Yaklaşımı ($L_f=8$)</i>	<i>Görüntü Blok Yaklaşımı ($M \times N=8$)</i>
Vv-1	%31,25	%92,5
Vv-2	%40,62	%93,3
Vv-3	%34,37	%92,96

Tablo 5.7'e göre bakıldığında Görüntü blok yaklaşımının sütun vektör yaklaşımına göre sıkıştırma yüzdesinin daha yüksek çıktığı gözlemlenmiştir.

Sympes yöntemi ile yeniden oluşturduğumuz videoların blok bazında sıkıştırma yüzdeleri aşağıdaki örnekteki gibi hesaplanır.

Görüntü boyutu 1080*1920 ve çerçeve boyutu, $m = 8$ için,

Görüntü 8 bit derinliğe sahip piksel değerlerini barındırmaktadır.

Sütun vektör yaklaşımında $L_f=8$ alırsak Orijinal sinyal çerçevesinin transferinde 8×8 bit = 64 bit alan kullanılır.

SYMPES yöntemi için hesaplama yaparsak, bu örnekte totalde 259.200 tane görüntü bloğu bulunur. Buna göre 18 bit ie indisi için, 8 bit Kazanç sabiti için, is indisi için 18 bit toplamda 44 bit kullanılır ve blok bazında %31,25 sıkıştırma oranı elde edilir.

Blok görüntü yaklaşımda ise $M \times M = 8 \times 8$ alırsak Orijinal sinyal çerçevesinin transferinde $64 \times 8 \text{ bit} = 512 \text{ bit}$ alan kullanılır.

SYMPES yöntemi için hesaplama yaparsak, bu örnekte totalde 32.400 tane görüntü bloğu bulunur. Buna göre 15 bit ie indisi için, 8 bit Kazanç sabiti için, is indisi için 15 bit toplamda 38 bit kullanılır ve blok bazında %92,5 sıkıştırma oranı elde edilir.



6. SONUÇ VE ÖNERİLER

Sympes yöntemi ile daha önce yapılan çalışmalara bakıldığında ses, EKG, EEG, EMG, medikal görüntüler ve gen dizilimi üzerine çalışmalar yapılmış olup, bu çalışmalar sonucunda büyük başarı sağlanmıştır.

Sonuç olarak bu tez kapsamında geliştirdiğimiz yöntem yeniden oluşturulacak olan videonun, oluşturulduğu görüntü kümesinden bağımsız olarak tekrar oluşturabileceği ispat edilmiştir. Yani herhangi bir görüntüden (evrensel küme) çözünürlük fark etmeksizin evrensel kümeden bağımsız bir video oluştuğu ispat edilmiştir. Geliştirilen yöntemde iletim hattından daha az veri gönderilmesi ile band genişliğini düşürdüğü ve görüntüyü iyi bir yüzde ile sıkıştırdığı gözlemlenmiştir. Ayrıca iletim hattında sadece fark indisleri gönderildiği için veri güvenliği ön planda tutulmuştur. Geliştirdiğimiz metotta kod çözüm süresinin kodlama süresinden yüksek olduğu gözlemlenmiştir. Kullandığımız yöntem CPU üzerinde çalıştığı için yeni nesil işlemcilerde bu sürelerin daha kısa süreceği düşünülmektedir. Geliştirilen yöntemin renkli video uygulamalarında da kullanılması düşünülmektedir.

KAYNAKLAR

- [1]. Gonzalez, R.C., Woods, R.E., and Eddins, S.L., 2009, *Digital Image Processing Using MATLAB*, 2nd edn., USA: Gatesmark Publishing.
- [2]. Gonzalez, R.C., Woods, R.E., 2008, *Digital Image Processing*, 3rd edn., Pearson Prentice Hall.
- [3]. Soille, P., 2002, *Morphological Image Analysis: Principles and Applications*, 2nd edn., Germany, Springer.
- [4]. Rosenblatt, F., 1958, "The perceptron: A probabilistic model for information storage and organization in the brain", *Psychological Review*, 65(3), 386-408.
- [5]. Russell, S.J., Norvig, P., 2010, *Artificial Intelligence: A Modern Approach*, Third Edition, Prentice Hall.
- [6]. Caudill, M., 1987, "Neural Network Primer: Part I", *AI Expert*, 2(12), 46-52.
- [7]. van Engelen, J.E., Hoos, H.H. 2020, "A survey on semi-supervised learning", *Mach Learn*, 109, 373–440.
- [8]. *Matlab for Artificial Intelligence*, <https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>, Alındığı Tarih: 11.10.2022.
- [9]. <https://developer.apple.com/library/content/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html>, Alındığı Tarih: 11.10.2022.
- [10]. Değirmenci, A., Çankaya, İ., and Demirci, R. 2016, "Gradyan Anahtarlama Gauss Görüntü Filtresi Ali," *Düzce Üniversitesi Bilim ve Teknol. Derg.*, vol. 4, no. Doğadan Esinlenen Optimizasyon Algoritmaları ve Optimizasyon Algoritmalarının Optimizasyonu, pp. 293–304, 2016.
- [11]. <https://yavuzbugra.wordpress.com/2011/05/01/goruntu-islemeye-filtreleme/>, Alındığı Tarih: 11.10.2022.
- [12]. Ergün, B. and Şahin, C. 2009, "Günümüzde Kullanılan Optik ve Sayısal Küresel Fotogrametri Teknikleri (Digital Spherical Photogrammetry Techniques Recently in Use)," *Optik (Stuttg.)*, pp. 40–50.

- [13]. <https://www.l3harrisgeospatial.com/docs/BackgroundFastFourierTransform.html>, Alındığı Tarih: 11.10.2022.
- [14]. Sena, B. 2018, “*Agu Early Prognosis of Breast Cancer Using Image Processing Using Image Processing and Machine*,” no. December, 2018.
- [15]. Güzide Miray, P. 2015, “Dijital görüntü işleme teknikleri kullanılarak görüntülerden detay çıkarımı,” Yüksek Lisans Tezi, vol. 151, no. Ocak, pp. 10–17, doi: 10.1145/3132847.3132886.
- [16]. <https://abdulsamet-ileri.medium.com/görüntü-filtrelerini-uygulama-ve-kenarları-algilama-21d42f194db4>, Alındığı Tarih: 11.10.2022.
- [17]. https://stringfixer.com/tr/Median_filter, Alındığı Tarih: 11.10.2022.
- [18]. <https://theailearner.com/2019/01/01/log-transformation/>, Alındığı Tarih: 11.10.2022.
- [19]. https://www.researchgate.net/figure/Diffraction-Pattern-Enhanced-Through-Log-Transformation_fig2_270275573, Alındığı Tarih: 11.10.2022.
- [20]. https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_transforms/py_fourier_transform/py_fourier_transform.html, Alındığı Tarih: 11.10.2022.
- [21]. <https://mesutpiskin.com/blog/opencv-ile-kenar-belirleme-algoritmaları.html>, Alındığı Tarih: 11.10.2022.
- [22]. <https://abdulsamet-ileri.medium.com/görüntü-filtrelerini-uygulama-ve-kenarları-algilama-21d42f194db4>, Alındığı Tarih: 11.10.2022.
- [23]. <https://silo.tips/download/grnt-leme-7hafta-kenar-belrlleme-algortmaları>, Alındığı Tarih: 11.10.2022.
- [24]. http://www.ibrahimcayiroglu.com/Dokumanlar/GoruntuIsleme/Goruntu_Isleme_Ders_Notlari-7.Hafta.pdf, Alındığı Tarih: 11.10.2022.
- [25]. <https://medium.com/operations-management-türkiye/canny-kenar-tespiti-edge-detection-ca5fd65a8227>, Alındığı Tarih: 11.10.2022.
- [26]. Yang, W. and Ngan, K., 2004, “MPEG-4 Based Stereoscopic Video Sequences Encoder”, *Proc. ICASSP*, volume 3, pages 741-744, May 2004.

- [27]. Akbulut, O., 2007, “Bir-Bit dönüşümü temelli blok hareket kestirimlerinin H.264/AVC’ye uygulanması”, Yüksek Lisans Tezi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, İzmit.
- [28]. H.264 Video Kodlama Standardı için Verimli ve Düşük Karmaşıklıkta Kodlayıcı Tasarımı, “http://www2.isikun.edu.tr/personel/hfates/H264_research.htm” (Ziyaret Tarihi: 11.10.2022)
- [29]. Lampert, P., De Neve, W., Dhondt, Y., and Van de Walle, R. 2006, “Flexible macroblock ordering in H.264/AVC”, *J. Vis Commun. Image R.*, 17, 358-375.
- [30]. Schwarz, T., Marpe, D. and Wiegand, T., 2005, “Basics Concepts for Supporting Spatial and SNR Scalability in the Scalable H.264/MPEG4-AVC Extension”, *IEEE International Workshop on Systems, Signals and Image Processing (IWSSIP)*, Greece.
- [31]. Dröse, M., Clemens, C. and Sikora, T., 2006, “Extending Single-View Scalable Video Coding to Multi-View Based on H.264/AVC,” *IEEE Int. Conf. on Image Processing (ICIP)*, Atlanta, GA, USA, October 2006.
- [32]. Choi, Y., Cho, S., Lee, J. and Ahn, C., 2002, “Field-based Stereoscopic Video Codec for Multiple Display Methods,” *IEEE Int. Conf. On Image Processing (ICIP)*, volume 2, pages 253-256, September, 2002.
- [33]. Bilen, C., Aksay, A. and Bozdagi Akar, G., 2006, “A Multi-View Codec Based on H.264”, *IEEE International Conference on Image Processing (ICIP)*, Atlanta, USA.
- [34]. Gorkemli, B. and Civanlar, M. R., 2006, “SVC Coded Video Streaming over DCCP”, *8th IEEE International Symposium on Multimedia 2006 (ISM'06)*.
- [35]. Schafer, R., Schwarz, H., Marpe, D., Schierl, T. and Wiegand, T., 2005, “MCTF and Scalability Extension of H.264/AVC and its Application to Video Transmission, Storage and Surveillance,” *Visual Communications and Image Processing (VCIP'05)*, Beijing, China, 12-15 July, 2005.
- [36]. Yang, W., Ngan, K.N. and Cai, J., 2006, “An MPEG-4-Compatible Stereoscopic/Multiview Video Coding Scheme”, *IEEE Transactions on Circuits and Systems for Video Technology*, volume 16, no. 2, pages 286-290, February 2006.

- [37]. Fecker, U. and Kaup, A., 2005, "H.264/AVC-Compatible Coding of Dynamic Light Fields Using Transposed Picture Ordering", *EUSIPCO 2005*, Antalya, Turkey, September 2005.
- [38]. Mueller, K., Merkle, P., Heiko, S., Hinz, T., Smolic, A., Oelbaum, T. and Wiegand, T., 2006, "Multi-View Coding Based on H.264/MPEG4-AVC Using Hierarchical B Pictures," *Proc. Picture Coding Symposium (PCS)*, Beijing, China.
- [39]. Schwarz, H., Hinz, T., Marpe, D. and Wiegand, T., 2004, "MCTF and Scalability Extension of H.264/AVC", *Proc. Picture Coding Symposium (PCS)*, San Francisco, USA.
- [40]. Guo, X. and Huang, Q., 2004, "Multiview Video Coding Based on Global Motion Model", *PCM 2004, LNCS 3333*, pages 665-672.
- [41]. Balasubramaniyam, B., Edirisinghe, E. and Bez, H., 2004, "An Extended H.264 CODEC for Stereoscopic Video Coding", *Proceedings of SPIE*.
- [42]. Flierl, M. and Girod, B., 2003, "Generalized B Pictures and the Draft JVT/H.264 Video Compression Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, volume 13, no. 3, pages 587-597, July 2003.
- [43]. Schwarz, H., Marpe, D. and Wiegand, T., 2004, "SNR-Scalable Extension of H.264/AVC", *IEEE International Conference on Image Processing (ICIP)*, Singapore.
- [44]. Agustsson, E., Minnen, D., Johnston, N., Balle, J., Hwang, S. J., and Toderici, G. 2020, Scale-space flow for end-to-end optimized video compression. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8500{8509.
- [45]. Liu, Z., Yeh, R. A., Tang, X., Liu, Y., and Agarwala, A. 2017, Video frame synthesis using deep voxel flow. *IEEE Int. Conf. on Computer Vision (ICCV)*.
- [46]. Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A., Bishop, R., Rueckert, D., and Wang, Z. 2016, "Real-time single image and video süper resolution using an efficient sub-pixel convolutional neural network".
- [47]. Wallace, G. K. 1992, "The jpeg still picture compression standard". *IEEE Transactions on Consumer Electronics*, 38(1):xviii{xxxiv}.

- [48]. Yang, R., Mentzer, F., Van Gool, L., and Timofte, R. 2020, "Learning for video compression with hierarchical quality and recurrent enhancement". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [49]. Wang, X., Chan, K. C., Yu, K., Dong, C., and Loy, C. C. 2019, "EDVR: Video restoration with enhanced deformable convolutional networks". In *IEEE Conf. on Comp. Vision and Pattern Recognition (CVPR) Workshops*.
- [50]. Yang, R., Mentzer, F., Van Gool, L., and Timofte, R. 2021, "Learning for video compression with recurrent auto-encoder and recurrent probability model". *IEEE Journal of Selected Topics in Signal Processing*, 15(2):388-401.
- [51]. Lu, G., Ouyang, W., Xu, D., Zhang, X., Cai, C., and Gao, Z. 2019, "Dvc: An end-to-end deep video compression framework". *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [52]. Sullivan, G. J., Ohm, J., Han, W., and Wiegand, T. 2012, "Overview of the high efficiency video coding (hevc) standard". *IEEE Trans. On Circuits and Systems for Video Tech.*, 22(12):1649-1668.
- [53]. Yang, R., Sun, X., Xu, M., and Zeng, W. 2019, "Quality-gated convolutional lstm for enhancing compressed video". In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 532-537.
- [54]. Kirmemis, O., Bakar, G., and Murat Tekalp, A. 2018, "Learned compression artifact removal by deep residual networks". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- [55]. <https://www.sciencedirect.com/science/article/abs/pii/S1568494612004401>, Alındığı Tarih: 11.10.2022.
- [56]. Helmrich, C. R., Siekmann, M., Becker, S., Bosse, S., Marpe, D., and Wiegand, T. 2020, "Xpsnr: A low-complexity extension of the perceptually weighted peak signal-to-noise ratio for high-resolution video quality assessment". In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2727-2731.

- [57]. Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. 2018, The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586-595.
- [58]. Keleş, O., Yılmaz, M. A., Tekalp, A. M., Korkmaz, C., and Dogan, Z. 2021, *On the computation of psnr for a set of images or video*.
- [59]. Sullivan, G.J., Ohm, J.R., Han, W.J., and Wiegand, T. 2012, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
- [60]. Yılmaz, M. A. and Tekalp, A. M. 2020, "End-to-end rate-distortion optimization for bi-directional learned video compression". In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1311-1315.
- [61]. High Efficiency Video Coding, ITU-T Rec. H.265 and ISO/IEC 23008-2 (HEVC), ITU-T and ISO/IEC, April 2013.
- [62]. Pescador, F., Chavarrias, M., Garrido, M.J., Juarez, E., and Sanz, C. 2013, "Complexity Analysis of an HEVC Decoder Based on a Digital Signal Processor," *IEEE Trans. On Consumer Electronics*, vol. 59, no. 2, pp. 391-399, May 2013.
- [63]. Algorithm Description of Versatile Video Coding and Test Model 5, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG11, March 2019.
- [64]. Kalali, E., and Hamzaoglu, I. 2018, "Approximate HEVC fractional interpolation filters and their hardware implementations," *IEEE Trans. on Consumer Electronics*, vol. 64, no. 3, pp. 285-291, Aug. 2018.
- [65]. Cetin, M., and Hamzaoglu, I., 2011, "An adaptive true motion estimation algorithm for frame rate conversion of high definition video and its hardware implementations", *IEEE Trans. on Consumer Electronics*, vol. 57, no. 2, pp. 923-931, May 2011.
- [66]. Lainema, J., Bossen, F., Han, W. J., Min, J., and Ugur, K. 2012, "Intra Coding of the HEVC Standard", *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1792-1801, Dec. 2012.

- [67]. Biatek, T., Lorcy, V., Castel, P., and Philippe, P. 2016, "Low-Complexity Adaptive Multiple Transform for Video Coding", *Proc. Data Compression Conference*, April 2016.
- [68]. Vanne, J., Viitanen, M., Hämäläinen, T.D., and Hallapuro, A., 2012, "Comparative Rate-Distortion-Complexity Analysis of HEVC and AVC Video Codecs", *IEEE Trans. On Circuits and Systems for Video Technology*, vol.22, no. 12, pp. 1885-1898, Dec. 2012.
- [69]. Azgin, H., Kalali, E., and Hamzaoglu, I. 2017, "A Computation and Energy Reduction Technique for HEVC Intra Prediction", *IEEE Trans. On Consumer Electronics*, vol. 63, no. 1, pp. 36-43, Feb. 2017.
- [70]. Azgin, H., Mert, A.C., Kalali, E., and Hamzaoglu, I. 2018, "An Efficient FPGA Implementation of HEVC Intra Prediction", *IEEE Int. Conf. on Consumer Electronics*, March 2018.
- [71]. Abeydeera, M., Karunaratne, M., Karunaratne, G., De Silva, K., and Pasqual, A. 2016, "4K Real Time HEVC Decoder on FPGA", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 236-249, Jan. 2016.
- [72]. Amish, F., and Bourennane, E. B. 2016, "Fully Pipelined Real Time Hardware Solution for High Efficiency Video Coding (HEVC) Intra Prediction", *Journal of System Architecture*, vol. 64, pp. 133-147, March 2016.
- [73]. Kalali, E., Adibelli, Y., and Hamzaoglu, I. 2014, "A Low Energy Intra Prediction Hardware for High Efficiency Video Coding", *Journal of Real-Time Image Processing*, Dec. 2014.
- [74]. Capon, J. 1959, A probabilistic model for run-length coding of pictures. *IRE Transactions on Information Theory*, 5(4), 157-163.
- [75]. Huffman, D. A. 1952, A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9), 1098-1101.
- [76]. Witten, I. H., Neal, R. M., & Cleary, J. G. 1987, Arithmetic coding for data compression. *Communications of the ACM*, 30(6), 520-540.

- [77]. Ghaderi, V., and Afshinfar, J. 2014, A Comparative Study of the Effects of Animated versus Static Funny Pictures on Iranian Intermediate EFL Students' Intake and Retention of Idioms, *Procedia-Social and Behavioral Sciences*, 98 (2014) 522 – 531.
- [78]. Balaban, Y. ve Çevik, İ.F. 2020, İnternetin Başlangıcı ile Canlandırmada Yeni Anlatım Biçimleri; GIF ve Doodle Bağlamında Bir Araştırma. *Turkish Studies-Information Technologies and Applied Sciences*, 15 (4), 417-430.
- [79]. Indiana Üniversitesi, “What is the TIFF graphics file format?”, <https://kb.iu.edu/d/afjn>, (erişim tarihi: 10/12/2022)
- [80]. Greg Roelofs, “PNG Site Map”, <http://www.libpng.org/pub/png/#history>, (erişim tarihi: 10/12/2022)
- [81]. Wallace, G. K. 1992, The JPEG still picture compression standard. *IEEE transactions on consumer electronics*, 38(1), xviii-xxxiv.
- [82]. Artuğer, F., & Özkaynak, F. 2018, JPEG Sıkıştırma Algoritmasının Dünü Bugünü ve Geleceği. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 30(3), 161-167.
- [83]. Rao, K. R., & Yip, P. 2014, *Discrete cosine transform: algorithms, advantages, applications*. Academic press.
- [84]. Taubman, D., & Marcellin, M. 2012, JPEG2000 image compression fundamentals, standards and practice: image compression fundamentals, standards and practice (Vol. 642). *Springer Science & Business Media*.
- [85]. Artuğer, F., & Özkaynak, F. 2018, Performance Comparison for Lossy Image Compression Algorithm. *International Conference on Communication and Signal Processing (ICCSP)*.
- [86]. Correa G., Assuncao P., Agostini L., and Cruz L. A. D., 2016, *Complexity-Aware High Efficiency Video Coding*, Springer, 1.st ed., Cham.
- [87]. Akramullah S., 2014, *Digital Video Concepts, Methods, and Metrics: Quality, Compression, Performance, and Power Trade-off Analysis*, 1.st ed., Apress Open, Online.
- [88]. Bull D. R., 2014, *Communicating Pictures: A Course in Image and Video Coding*, 1.st ed., Academic Press, Oxford.

- [89]. Tan, T. K., Weerakkody R., Mrak M., Ramzan N., Baroncini V., Ohm J., and Sullivan G. J., 2016, Video Quality Evaluation Methodology and Verification Testing of HEVC Compression Performance, *IEEE Transactions on Circuits and Systems For Video Technology*, 26(1), 76-90.
- [90]. Sullivan G. J., 2007, DirectX Video Acceleration Specification for Windows Media Video v8, v9 and vA Decoding (Including SMPTE 421M "VC-1"), *Microsoft Corporation*, 1-102.
- [91]. Borer T., and Davies T., 2005, *Dirac Video Compression using Open Technology*, *BBC Research & Development*, 1-9.
- [92]. Grange A., Rivaz P., and Hunt A., 2016, *VP9 Bitstream & Decoding Process Specification- v0.6*, Google Inc, 1-163.
- [93]. Gao W., 2005, 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.
- [94]. Grois D., Marpe D., Mulayoff A., Itzhaky B., and Hadar O., 2013, 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.
- [95]. Rerabek M., and Ebrahimi T., 2014, 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.
- [96]. Chen Y., Zhou Y., and Wen J., 2016, *Efficient Software HEVC to AVS2 Transcoding*, *Information*, 7(3), 53.
- [97]. Ma S., Huang T., Reader C., and Gao W., 2015, AVS2? Making Video Coding Smarter [Standards in a Nutshell], *IEEE Signal Processing Magazine*, 32(2), 172-183.
- [98]. Karwowski D., Grajek T., Klimaszewski K., Stankiewicz O., Stankowski J., and Wegner K., 2016, 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.
- [99]. LeGall, D. 1991, MPEG: A Video Compression Standard for Multimedia Applications. *Communications of the ACM*. 34, 46–58.

- [100]. ITU. International Telecommunications Union. (1994). Generic Coding of Moving Pictures and Associated Audio Information-Part 2: Video. Rec. ITU-T H.262. ISO/IEC. International Organization for Standardization/ International Electrotechnical Commission. (1994). ISO/IEC 13 818-2 (MPEG-2).
- [101]. MPEG-2 Video. 1994. Available at: <http://mpeg.chiariglione.org/standards/mpeg-2/video>. Accessed 25.12.2022.
- [102]. ISO/IEC. International Organization for Standardization/ International Electrotechnical Commission. (1999). ISO/IEC 14 496-2 (MPEG-4 Visual Version 1).
- [103]. MPEG-4 Visual. 1999. Available at: <http://mpeg.chiariglione.org/standards/mpeg-4/video>. Accessed 25.12.2022.
- [104]. ITU. International Telecommunications Union. 2010, “Advanced Video Coding for Generic Audio-Visual Services”. *Rec. ITU-T H.264*.
- [105]. Wiegand, T., Sullivan, G.J., Bjontegaard, G., and Luthra, A. 2003, “Overview of the H.264/AVC Video Coding Standard”. *IEEE Transactions Circuits System Video Technol.* 13, 560–576.
- [106]. Ziv, J., and Lempel, A. 1977, A Universal Algorithm for Sequential Data Compression. *IEEE Transactions on Information Theory.* 23, 337–343.
- [107]. Witten, I., Neal, R., and Cleary. J. 1987, Arithmetic Coding for Data Compression. *Communications of the ACM.* 30, 857–865.
- [108]. Gökbay, Z. İ. 2013, Sympes Yöntemiyle Biyomedikal Görüntülerin Modellenmesi, Doktora Tezi (Tez Danışmanı: Prof. Dr. B. Sıddık Yarman), İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, Biyomedikal Mühendisliği Anabilim Dalı, İstanbul, Türkiye.

İNTİHAL RAPORU İLK SAYFASI